A First Proposal Towards Anticipatory Shipping Implementation In Automotive Manufacturing Through Machine Learning And Optimization

Juan Manuel García Sánchez

UNIVERSITAT RAMON LLULL

DOCTORAL THESIS

Title	A First Proposal Towards Anticipatory Shipping Implementation In Automotive Manufacturing Through Machine Learning And Optimization
Presented by	Juan Manuel García Sánchez
Centre	La Salle Digital Engineering School
Department	Engineering
Directed by	Dr. Xavier Vilasís Cardona

Abstract

A First Proposal Towards Anticipatory Shipping Implementation In Automotive Manufacturing Through Machine Learning And Optimization

Juan Manuel García Sánchez

This research is framed in collaboration with SEAT S.A., a Spanish car manufacturer which is seeking to deliver the expected vehicle by the customers in the shortest timeframe, named Anticipatory Shipping. This problem is not exclusive to a particular company, but it is shared by multiple actors. Additionally, it has also gained the attention of academia producing literature. The review of the existing state-of-the-art conducted us to find a research gap that this thesis attempts to fill in. The investigation focuses on logistics, demand prediction, online data, and manufacturing optimization.

The proposed solution starts from the easiest to the most complex cost of implementation in the current company's operation. Firstly, it commences by redirecting already manufactured stock cars to destinations where they are expected to remain for shorter durations. Several Machine Learning classification algorithms have undergone testing to determine the most suitable one. Results equal or improve the decisions made by the experts of the company. Following this, an exploration into customer behavior was initiated using data obtained from the company's Car Configurator webpage. This online platform enables users to browse the company's entire product lineup and select their preferred vehicle. This thesis demonstrates that data collected from this tool serves as a reliable source of information for discerning users' purchasing intentions. The process involves comparing the obtained outcomes with various demand prediction models, which may or may not incorporate Car Configurator data, filtering the data by eliminating anomalous values and employing heuristic search algorithms such as genetic algorithms. The objective is to pinpoint the subset of online data with the highest predictive capacity. Ultimately, the findings from this final phase are utilized to adjust the attributes of the cars within the manufacturing pipeline. This optimization approach has effectively mitigated the discrepancy between the stock composition and the anticipated demand.

Presently, this research has yielded with presentations at three globally recognized congresses, along with a publication in a top-quartile indexed journal, and additional documentation awaiting release.

Keywords: **Anticipatory Shipping**; **Automobile industry**; **Car Configurator**; **Machine learning**; **Genetic algorithm**; **Forecasting**

Resumen

Una Primera Propuesta Para La Implementación Del Envío Anticipado En La Fabricación De Automóviles A Través De Aprendizaje Automático Y Optimización

Juan Manuel García Sánchez

Esta investigación se enmarca dentro de la colaboración con SEAT S.A., fabricante español de automóviles que busca entregar el vehículo que los clientes desean en el menor plazo posible, lo que se denomina Envío Anticipado. Este problema no es exclusivo de una empresa en particular, sino que es compartido por múltiples actores. Además, también ha ganado la atención del mundo académico produciendo literatura. La revisión del estado del arte existente nos llevó a encontrar un vacío en la investigación que esta tesis intenta cubrir. La investigación se centra en la logística, la predicción de la demanda, los datos en línea y la optimización de la fabricación.

La solución propuesta inicia de más fácil a más complejo coste de aplicación en el funcionamiento actual de la empresa. En primer lugar, comienza por redirigir los coches de stock ya fabricados a destinos en los que se espera que permanezcan menos tiempo. Se han sometido a prueba varios algoritmos de clasificación de aprendizaje automático para determinar el más adecuado. Los resultados igualan o mejoran las decisiones tomadas por los expertos de la empresa. A continuación, se inició una exploración del comportamiento de los clientes utilizando datos obtenidos de la página web del Configurador de Coches de la empresa. Esta plataforma en línea permite a los usuarios navegar por toda la gama de productos de la empresa y seleccionar su vehículo preferido. Esta tesis demuestra que los datos recogidos en esta herramienta constituyen una fuente de información fiable para discernir las intenciones de compra de los usuarios. El proceso consiste en comparar los resultados obtenidos con diversos modelos de predicción de la demanda, que pueden incorporar o no datos del Configurador de Coches, filtrar los datos eliminando los valores anómalos y emplear algoritmos heurísticos de búsqueda como los algoritmos genéticos. El objetivo es identificar el subconjunto de datos en línea con mayor capacidad predictiva. En última instancia, los resultados de esta fase final se utilizan para ajustar los atributos de los coches dentro del proceso de fabricación. Este enfoque de optimización ha mitigado eficazmente la discrepancia entre la composición del stock y la demanda prevista.

En la actualidad, esta investigación ha dado sus frutos con ponencias en tres congresos de prestigio mundial, junto con una publicación en una revista indexada de primer cuartil, y documentación adicional pendiente de publicación.

Palabras Clave: **Envío Anticipado**; **Industria Automovilística**; **Configurador de Coches**; **Aprendizaje Automático**; **Algoritmo Genético**; **Predicción**

Resum

Una Primera Proposta Cap A La Implementació De L'Enviament Anticipat En La Fabricació D'Automòbils A Través D'Aprenentatge Automàtic I Optimització

Juan Manuel García Sánchez

Aquesta investigació s'emmarca en la col·laboració amb SEAT S.A., fabricant d'automòbils espanyol que busca lliurar el vehicle que els clients desitgen en el menor termini possible, el que s'anomena Enviament Anticipat. Aquest problema no és exclusiu d'una empresa concreta, sinó que és compartit per múltiples actors. A més, també ha cridat l'atenció del món acadèmic produint literatura. La revisió de l'estat de l'art existent ens va portar a trobar un buit de recerca que aquesta tesi intenta omplir. La investigació se centra en la logística, la predicció de la demanda, les dades en línia i l'optimització de la fabricació.

La solució proposada parteix del cost d'implantació, des de més fàcil a més complex en l'operació de l'empresa actual. En primer lloc, s'inicia reorientant els cotxes de stock ja fabricats cap a destinacions on s'espera que romanguin durant una durada més curta. Diversos algorismes de classificació d'aprenentatge automàtic s'han sotmès a proves per determinar el més adequat. Els resultats igualen o milloren les decisions preses pels experts de l'empresa. Després d'això, es va iniciar una exploració del comportament dels clients mitjançant les dades obtingudes de la pàgina web de Configurador de Cotxes de l'empresa. Aquesta plataforma en línia permet als usuaris navegar per tota la gamma de productes de l'empresa i seleccionar el seu vehicle preferit. Aquesta tesi demostra que les dades recollides d'aquesta eina serveixen com a font d'informació fiable per a discernir les intencions de compra dels usuaris. El procés consisteix a comparar els resultats obtinguts amb diversos models de predicció de la demanda, que poden incorporar o no dades del Configurador de Cotxes, filtrar les dades eliminant valors anòmals i emprant algorismes de cerca heurístics com els algorismes genètics. L'objectiu és identificar el subconjunt de dades en línia amb la capacitat predictiva més alta. En última instància, les conclusions d'aquesta fase final s'utilitzen per a ajustar els atributs dels cotxes dins de la línia de fabricació. Aquest enfocament d'optimització ha mitigat de manera efectiva la discrepància entre la composició del stock i la demanda prevista.

Actualment, aquesta investigació ha donat resultats amb presentacions en tres congressos reconeguts mundialment, juntament amb una publicació en una revista indexada del quartil superior i documentació addicional pendent de publicació.

Paraules Clau: **Enviament Anticipat**; **Indústria Automobilística**; **Configurador de Cotxes**; **Aprenentatge Automàtic**; **Algorisme Genètic**; **Predicció**

Recognition of Institutions

This work is partially funded by the Department de Recerca i Universitats of the Generalitat de Catalunya under the Industrial Doctorate Grant DI 2019-34. The authors express their acknowledgements to the different institutions involved in this research: Department de Recerca i Universitats of the Generalitat de Catalunya, Industrial Doctorate Program, La Salle - Universitat Ramon Llull, and SEAT S.A.

A María Encarnación y a Juan José

Contents

List of Figures

List of Tables

Acknowledgments

Esta tesis no soy sólo yo, si no que soy yo y mis circunstancias. Y mis circunstancias, en este caso, tienen nombre propio. Por eso, es digno mencionar a todas las personas que me han acompañado en el camino.

Primeramente, a mi director de tesis Xavier Vilasís. Él vio algo en mí (no sólo el gusto compartido por el humor inteligente, también llamado chistes malos). Y esta tesis, cinco años después, así lo certifica. Además, me presentó a las personas que pasarían a ser mis compañeros de penas y alegrías en la épica del doctorado. Los seniors Elisabet Golobardes, Miriam Calvo y Álvaro García, excelentes espejos en los que mirarse; las juniors con ganas de comerse el mundo, Alexia Martorell, Virginia Jiménez, y Jessie Martín; la soluciona problemas Mireia Reniu; y mis compañeros de fechorías Sergi Bernet, Nuria Valls, Carina Trippl y Guillermo Brugarolas. En definitiva, gracias a todas aquellas personas que formaron o formarán parte del grupo antiguamente conocido como DS4DS.

No obstante, los agradecimientos también se extienden al otro vértice del doctorado. SEAT no es sólo una fábrica de coches. Es el punto de encuentro de gente fantástica. Alexandre Lerma, el otro maestro de ceremonias en la creación de esta tesis. Sus conocimientos, trato personal y capacidad resolutiva son admirables. El proyecto de Customer Driven Supply Chain no puede estar en mejores manos, como tampoco lo pueden estar mis excompañeros de departamento PL6, a los que también agradezco. Además, citar a los fantásticos Joan Ortí, Angela Adamo, Alejandra Alonso y, por extensión, a Manuel Román. Estoy seguro que triunfarán en todo lo que se propongan.

Por supuesto, no puedo ignorar los lugares por los que he pasado y las personas que allí he conocido: Madrid (gracias Lucía Lendínez), Francia y, en especial, Brasil. Todo ello me ha traído a Cataluña y a conocer a la persona más maravillosa del mundo, la mejor veterinaria, tieta de Nahia, orgull clotenc, viajera empedernida, compañera de aventuras, Meritxell Vilà. T'estimo. Gracias también a tu familia, Conxita Soler y Jaume Vilà, por aceptar a este extraño en su casa.

Por último, recordar mis raíces murcianas. Tengo la suerte de tener una familia que me apoya y me anima a embarcarme en más aventuras. Todas las enseñanzas de mis abuelos, tíos, primos, etc. me han convertido en el hombre que soy hoy en día. Especialmente, a mis hermanos María Eugenia y José Alberto, un apoyo en el que siempre confiar; y a mis padres, María Encarnación y Juan José, quienes me han brindado un hogar lleno de amor y comprensión. Su ejemplo es mi mayor inspiración.

Chapter 1

Introduction

The automotive industry serves as a driving force in the economies of the countries [1]. Within the European Union, it represents over 6% of total employment and 7% of GDP, whilst in the United States over 7 million jobs are supported by manufacturers, suppliers, and dealers [2, 3]. Particularly, the sector in Spain, the second largest producer in Europe, amounts to 11% of GDP [4]. Nevertheless, the market is currently undergoing a paradigm shift, as the number of car brands has recently surged, driven in part by China's influence [5, 6, 7]. Furthermore, public expectations are notably high, given that, for a considerable portion of the population, purchasing a car represents the second most substantial financial investment after acquiring a house [8]. As a result, companies, including mass-market manufacturers such as Spanish brand SEAT S.A. (referred to hereafter as SEAT), strategically segment and specialize their activities and target audiences to gain a competitive edge. In its endeavor to efficiently meet customer expectations, SEAT strives to deliver the expected vehicle in the shortest possible timeframe. This thesis contributes to this objective by comprehensively understanding the factory's idiosyncrasies, exploring the state-of-the-art, and proposing a novel approach grounded in Machine Learning, data mining, and optimization, yielding promising results.

Nevertheless, the attempts from part of SEAT in fulfilling this goal started before the commencement of the thesis. From 2017 to 2020, former SEAT CEO Luca de Meo promoted the creation of the Fast Lane project within the company. The project consisted of manufacturing and delivering, in a maximum of 21 days, the vehicle to the final customers [9, 10]. The project initially was launched in Austria, and after its success in this market, it was expanded to Germany and Spain. Regrettably, the COVID-19 outbreak had a negative impact on the project. Production was totally interrupted during the lockdown and, eventually, there was a shortage of electronic components, especially semiconductors, which made difficult the return to the regular production rate. In the current days, the Fast Lane service has not been restarted yet.

Consequently, the doctoral thesis researches in the requirements of the main statement and guides the reader in the proposed solution. It is described not in chronological order, but in ascending complexity. In other words, the ease of implementation from the point of view of the company's operative. Firstly, the proposed solution initiates by redirecting already manufactured stock cars to locations where it is estimated they will remain for a shorter duration. Various classification Machine Learning algorithms have been tested to identify the most suitable one. The results achieved either match or surpass the current procedure followed by the company's experts. Subsequently, an investigation into customer behavior commenced using data collected from the company's Car Configurator webpage. This online portal allows users to explore the entire commercial offering of the

company and select the vehicle of their interest. This thesis successfully demonstrates that the data collected by this tool serves as a reliable source of information to discern users' purchasing intent. The procedure entails comparing the obtained results with various demand prediction models that either include or exclude Car Configurator data, purging the data by excluding anomalous values and employing heuristic search algorithms, such as genetic algorithms. The goal is to identify the subset of online data with the highest predictive capacity. Ultimately, the outcomes from this final block are utilized to update the attributes of the cars within the manufacturing line. The optimization mechanism has effectively reduced the disparity between the stock composition and the estimated demand.

The primary objective of the thesis is to enhance operational efficiency through a more qualitative approach rather than a quantitative one, due to the unique nature of the car industry. Specifically, the thesis aims to achieve a deeper understanding of customer behavior and preferences, enabling the sector to make more informed and strategic decisions about stock allocation, demand prediction, and production adjustments. Expected results include the development of a robust methodology for predicting car demand with higher accuracy by leveraging customer interaction data from the Car Configurator.

The thesis has generated the following documentation, including published and pending to-be-released papers:

- **Influence of Car Configurator Webpage Data from Automotive Manufacturers on Car Sales by Means of Correlation and Forecasting**: Published in the indexed journal Forecasting from MDPI in Special Issue Feature Papers of Forecasting 2022. The paper delivers a methodology to prove the influence of Car Configurator webpage data for automotive manufacturers [11].

- **Binary Delivery Time Classification and Vehicle's Reallocation Based on Car Variants. SEAT: A Case Study**: Presented in the International Catalan Congress of Artificial Intelligence (CCIA 2022). This note provides a solution to the vehicle's compound allocation problem employing Machine Learning Classification algorithms. It is performed using the car attributes and the time that vehicles have spent in the compound regions waiting for the customer delivery day [12].

- **Filtering User Behavior Data without Losing Significance on Non Transactional Websites**: Pending to be submitted in journal paper to be defined of the Operational Research and/or E-commerce area of study. This study develops and assesses different filtering rules, based on users' tracking activity within the Car Configurator webpage. Results show that data significance is preserved when compared with the raw clickstream data. It emerged from a collaboration with the Marketing department of the academical institution.

- **Data Mining Car Configurator Clickstream Data to Identify Potential Consumers: A Genetic Algorithm Approach**: Exhibited in the International Conference on Artificial Intelligence and Soft Computing (ICAISC 2023). This paper investigates whether valuable information can be extracted from Car Configurator data. The data mining technique of genetic algorithms is employed to identify the characteristics that maximize the correlation between clickstream data and car sales [13].

- **Analyzing Car Configurator Impact Through Genetic Algorithm from a Regional Perspective**: Shown in CCIA 2023 edition. This study examines whether visits to the Car Configurator website from a specific area in Spain have a similar impact compared to visits from other locations. To analyze this relationship, genetic algorithms are employed [14].

Regarding the data available, they were provided by SEAT and are related to the four car models produced in their headquarters: SEAT Ibiza, SEAT Arona, SEAT Leon 5D, and SEAT Leon Sportstourer. All data are framed within the Spanish market, including: (a) production and delivery planning of vehicles; (b) sales records; and (c) visits done by users from the Car Configurator webpage of the company. The first block gathers all the cars' deliveries done between January 2017 and January 2020, together with their factory background. In other words, the day on which cars have passed through all the milestones in the production flow, i.e., since they are a dealership's request until the customers receive their purchase. This tracking includes a full description of the vehicles in terms of their components, such as car model, trim level, and engine; and attributes, for instance, color, destination, and order type. All this information is contained in over 200,000 rows. Concerning the sales record, the data collection begins in April 2017 and finishes in January 2020, both included. The temporal frame refers to the day customers book their cars in the dealerships, after paying a booking fee. That's why it has been taking the decision to consider this moment as the purchasing date, instead of following the delivery date. Likewise, the location of the sales is given, as well as, the full description of the vehicle, such as car model, trim level, engine, and color. Nevertheless, it does not include a tag to distinguish between the order types, although fleet cars are excluded from the sample. Additionally, for privacy reasons, the company anonymized the vehicles eliminating all identification signals, such as the name of the buyer or the manufacturing serial number. The data spans more than 120,000 rows. Finally, the visits to the Spanish car configurator webpage extend beyond 19M instances, for the same period that the sales record. Each user is individually identified by means of a unique alphanumeric code derived from their internet browser's cookies. This is the only possible tracking approach, as the webpage does not require login with user and password. The clickstream data gives access to know all the components and attributes of the configuration explored by the user, from which geographical location it was scouted, how many times the user has done the same search, and even, in which step of the process the configuration was interrupted. To culminate, whenever it was needed, the above data were aggregated in a weekly format. The motivation lies in the SEAT production calendar, where planification occurs per week.

The document is structured in the following way. In Chapter 2, the state of the art is presented. The available data to perform the research is shown in Chapter 3. Afterward, Chapter 4 describes the proposed solution to fill the research gap. A detailed explanation of each method or technique included in the solution is found in Chapter 5. The results for each step within the proposed solution are placed in Chapter 6, whilst they are discussed in Chapter 7. Finally, Chapter 8 expresses the conclusions of the thesis.

Chapter 2

State Of The Art

This chapter serves as a compass across the existing literature in the area of research. Firstly, SEAT's approach to fulfilling the mission of achieving the fastest delivery of the vehicle that the customer wants is related. This project has been called SEAT Fast Lane. Afterward, the bibliographic exploration is executed in the three pillars on which the Fast Lane is built: (a) efficient operative; (b) reduced variety of products; and (c) demand forecasting. Additionally, the review includes the studies done in the field of optimizing the modification of the cars in the manufacturing line. Finally, the chapter presents the research gaps found in the literature and gives a summary of the suggested solutions.

2.1 SEAT Fast Lane

The trigger that activated the idea of delivering, in the minimum possible time, the cars requested by customers was Amazon Inc. In 2012, the retail and logistics giant unveiled a patent aimed at shortening delivery times. The system consists of anticipating users' purchases and dispatching the products to the closest Amazon facilities to the user, even prior to the completion of the sale. In case the acquisition is not finally executed, the product moves to another location, or it is offered in the area with a discount. This automatic routine is called Anticipatory Shipping [15]. Despite translating this concept to the automobile industry would be delightful, the migration is not straightforward. There are strong differences between both business models. The product category is not comparable, as customers go to Amazon to purchase low-implication products. Fashion and apparel, leisure and entertainment, or home and DIY categories occupy the first positions among the best-selling product categories within Amazon [16, 17]. On the other hand, cars are a more thoughtful buy. Nevertheless, the main discrepancy is the business sector, one of them a retailer and the other ones are manufacturers depending on dealerships to sell their products.

In a manner to follow the philosophy behind Anticipatory Shipping by Amazon, SEAT began internally the Fast Lane project in 2017. The car models and configurations under this signature would be delivered in a maximum of 21 days to the final customers [9, 10]. From the operations point of view, the manufacturing process was adapted and new internal systems were developed. Fast Lane cars gained priority in the waiting list to be manufactured; there were always available Fast Lane slots within the assembly line ready to be booked by the dealerships; the Fast Lane slots could be easily updated during the manufacturing process in non-dependant electrical attributes, such as alloy wheels, color, etc. in opposition to regular manufacturing orders that were immutable; customers were capable of performing a follow-up of their purchase thanks to the online Tracking Tool, which show to them real videos of their vehicles surpassing all the milestones between purchase and delivery; etc. With these assistances, SEAT was able to guarantee the reduced 21-day delivery time, against the 90-day average period when cars are requested directly to the factory. In summary, Fast Lane was supported by three aspects: (a) efficient and dedicated operations to execute all the manufacturing and logistics, reducing the dealerships' stock volume; (b) shortening the number of choseable configurations, a.k.a., variety in the commercial offer to ease the operations and the supply chain; (c) accurate demand prediction to acquire in advanced the customers' interests in order to activate the supply chain. The review of the literature in these three aspects is performed in this chapter.

2.2 Efficient Operative

The endeavor of vehicle manufacturers to ship cars in the most efficient manner is not a recent development, though it has been more extensively explored from an industrial perspective rather than by academics. This issue is well-described in [18]. The authors of the previous note present a taxonomy of the studies done in the field of automobile shipping optimization by level of decision-making, mode of transportation, and type of optimization decisions. Nevertheless, the focus is on transportation and route optimization. From the side of stock management and production optimization, an early example of these concepts can be found in the early 2000s. Reference [19] proposes the concept of the "3-Day Car". It aimed to achieve a fundamental change across the entire automotive supply chain by emphasizing the pivotal role of logistics. The viability of the 3-Day Car in the UK, a market covering three million annual vehicle movements, in terms of key constraints in the logistics and the possible environmental impact of a more responsive logistics is analysed in paper [20]. More of the ideas of this author were published in book [21], in which remarks the concept of "Build to Order" production as the optimal solution and touring across the practices done in Japanese, European, and American car brands. In more recent years, a review of 49 works about the Build to Order is executed on note [22]. It focuses on capacity, order planning, and presenting a framework for structuring planning tasks. These authors stand out for the critical role played by industrial operations in shaping the impact of forecasting and demand prediction within supply chains. Researchers underline that an overemphasis on forecasting methodologies without due consideration for the intricacies of industrial operations can lead to suboptimal outcomes, such as overproduction or stockouts. Fast Lane exemplifies a proactive approach towards updating logistics to better align with the demands of contemporary markets, without ignoring the potential that reliable forecasting can provide.

2.3 Commercial Offer Variety

The second point of improvement consists of having under control the complexity of the commercial offer. It is a delicate affair to find the balance between the quantity of viable options a customer can select from and lean logistics. In opposition to Henry Ford, who claimed that his customers could have any color they wanted, as long as it was black, nowadays the variety extends from thousands to millions of versions considering all the combinations among car models, drive trains, colors, interior packages and optional choices [23]. It might seem beneficial from the point of view of the customers, as it is easier to find a version that fits their needs. Consequently, the demand and profits of the manufacturer would experience an enhancement. Nevertheless, it is not a straightforward effect, as it depends on market-specific factors as well as costs. In instances where consumers lack familiarity with the available options, a preliminary step of learning about their preferences is needed before arriving at a decision. Excessive diversity in such circumstances can introduce complications or lead to consumer frustration, a phenomenon commonly referred to as "choice overload" or the "paradox of choice" [24, 25]. As articulated in the work [26], the marketer's responsibility in these scenarios is to mitigate the perceived complexity of the assortment. Experiments carried on references [27, 28] validate the effects that variety has on customers. In terms of the industry, the work [29] offers a structured framework to identify relevant complexities among product variety, logistics costs, and logistics performance. The latest incurred the most substantial detriment in instances of a misalignment between manufacturing and distribution. Researchers of the work [30], conducted an empirical investigation within the Philadelphia region to assess the efficacy of the automotive distribution system in managing product variety. Their findings revealed that, despite the extensive range of product variety available, most customers opted to purchase from the dealership's existing stock. This consumer behavior was attributed to the fact that ordering a car directly from the factory incurred a significant six-week delay in the delivery process.

In addition to all these previous aspects related to the variety of products, since the origin of Fast Lane, the company was outermost aware of the role of logistics. The current structure is not prepared to manage the shipment of 21 days of a vast selection of products straight out of the factory. These restrictions caused the commercial offer of Fast Lane cars would be shorter, and the orders' logistics flow deviated with respect to common requests. The best component to illustrate this difference is the cable tree of the cars. In other words, the batch of harnesses and wires that control all the features of the vehicle; starting in the engine, gearbox, and so on, but including as well the radio, A/C, heated seat, cameras, etc. After the drive unit, the cable tree is the most significant item within a car. It possesses a high level of customization. The layout and design of the wires are dependent on the attributes of the vehicle. For instance, for two vehicles with the same configuration except that one of them carries cruise control and the other one does not, the cable tree will be different. The high level of personalization entails that cable trees were exclusively requested to the supplier after the approval of the vehicle's order in the manufacturing line. The delivery time the supplier had with the factory is almost 4 weeks. SEAT, like other car manufacturers, is ruled by a "Just in Time" manufacturing policy. The pieces of the assembly are requested if they are required in the manufacturing line. This shipment period contradicts the 21-day delivery commitment of the company. Consequently, the solution to overpass this issue consisted of the creation of a cable tree depot within the SEAT facilities. After this measure, the supply of cable trees was guaranteed for the Fast Lane orders.

Unfortunately, the solution is not perfect. The capacity of the depot was very restricted, as the sales rate of Fast Lane cars was never planned to be the largest out of the total sales of the firm. Reduced store capacity means smaller product variety, thousands rather than millions of feasible combinations. It provokes another problem expected to be solved, i.e., deciding which car variants will be part of the Fast Lane offer. The first approach to resolving the dilemma selected the best-selling car configurations from those with the highest contribution margin. Normally, these vehicles are the most expensive hence, with this policy, it was intended to award the customer who spent the most with the fastest delivery. Nevertheless, the inconvenient aspect of the current solution was the lack of capturing a broader market share. Moreover, relying solely on past performance might hinder the ability to adapt to changing market demands.

2.4 Demand Forecasting

The aforementioned reasons motivated the company to explore a more robust approach. In particular, it was seen as beneficial the option of anticipating the customers' request and triggering the supply chain in advance. Consequently, the third leg of the Fast Lane was initiated, i.e., demand prediction.

2.4.1 SEAT Background

SEAT contacted a subsidiary company belonging to Volkswagen Group seeking assistance. The name of the firm was Data:Lab Munich (DataLab for so on), a division specialized in data science within the automotive industry structured under a start-up frame [31, 32]. Their approach was based on building the Fast Lane offer with the largest coverage, i.e., capable of reaching the largest segment of customers. The focus of the commercial offer was in terms of vehicles' equipment, i.e., cable trees. Afterward, the future sales of each car variant were forecasted from 7 to 20 weeks ahead of the current day. To accomplish this task, it is fed by the history of car variants found in the sales record and the ones done by the users of the SEAT car configurator webpage. The website is an online tool provided by car manufacturers, such as SEAT, to their potential customers so they can browse among all the options available in the company's portfolio without the need to list them all. Additionally, the user can obtain an estimated price and book a date at the dealership. All the information about the activity done by the potential customers is saved. It is possible to obtain which car variant they have configured, from which location, and when they did it. DataLab combined the clickstream data together with the car configurations history sales and, by means of association rules, they proposed a new Fast Lane offer. During this procedure, the weight of each data source was unequivalent, 70-30 proportion in favor of the sales record. Afterwards, weekly sales of each of the car variants were then forecasted between 7 and 20 weeks from the current date. The algorithm behind the prediction was ARIMA. During the development of the Proof-of-Concept phase, occurring in the last quarter of 2019, it was observed that the dealerships were adverse to relying on the output. The DataLab tool suggested car configurations dealers were not familiar with. Regrettably, before starting a second phase in the collaboration between SEAT and DataLab, the COVID outbreak began, and together with the consequent component crisis, especially semi-conductors, Fast Lean delivery ceased.

2.4.2 Car Configurator As A Helpful Information Source

Despite this interruption, the path initiated by our colleagues was worthy to be explored. The digital activity leaves a trace that can be analyzed to get a better comprehension of customers' behavior. With traditional techniques, such as interviewing or surveying, the researcher is imposing to rationalize a mental process that might be irrational. On the contrary, the online path permits to acquisition of this knowledge without perturbing the subjects. Thus, work [33] for example, permits measuring the weight that the World Wide Web has among other information search channels, such as relative and friends, mass media, and retailers. Precisely, it has become one of the main mottos of this thesis. In light of this, a comprehensive bibliographic review is conducted about the utilization of Internet data in forecasting, inventory, and production optimization.

The literature found evidence of a correlation between Internet data and sales, being helpful in demand forecasting for diverse business sectors. Examples are found on e-commerce [34], the entertainment sector [35, 36], the food industry [37], tourism [38], financial activity [39], retail business [40], and the editorial sector [41]. Especial attention deserves the work [42]. They utilized online retailers' clickstream data and historical sales data to develop a sales forecast, exploring the optimal quantity and timing of products. Their findings indicated that anticipatory shipping led to a 5.96% cost reduction for online retailers and an average reduction of 1.69 days of waiting time for customers compared to non-anticipatory shipping. However, the previous examples make reference to low-value purchases, customers are not highly involved, and there are no relevant differences between brands. Products or services with the opposite characteristics are defined as high-implication purchases. One of the economic sectors that fulfill these criteria is the real estate market. This area is not ignorant about the use of Internet data. Several authors in the literature have explored the utility of the Internet as an external source to capture customers' requests. References [43, 44, 45, 46, 47, 48] are proof of this.

Regarding the car market, the bibliographic exploration reveals that the employment of online data has been treated from two points of view: data acquired from social media or data coming from Internet search queries. As an example of social media data, reference [49] focuses on the sentiment analysis of social media and car review online sites, together with average monthly sales, to perform sales prediction before and after the launch of the vehicle. Another case is found in [50], where they performed a comparison of the outputs given by different multivariate regression models and time series models which combines monthly total vehicle sales in the USA together with sentiment scores from Twitter, stock market values, or a mix of both external information. On the other side, an early example from 2009 is found in [51]; they include Google Trends in a logarithmic autoregressive model to predict vehicle sales. Another interesting case is paper [52]; they use a novelty Bass diffusion model that includes customer Internet search behavior to explain product diffusion, gain significant information in about 84% of the samples, and help to predict new product diffusion. Publication [53] develops a backward induction approach to identify keywords that are frequently used by search engine users of the automotive market and, together with economic variables, the authors can predict monthly car sales. Research done in [54] focuses on the German market and performed long-term prediction by adding the information extracted from macroeconomic variables and online search queries. Similarly, reference [55] does an exercise on the car markets of Germany and the UK. They prove that online search data are correlated across products but to different extents. Hence, they develop a model linking search motives to observable search data and sales. Nevertheless, some examples take advantage of both social

media and search queries, such as paper [56]. They compare the outputs of the linear regression model of about a half million posts on social media for eleven car models in the Netherlands against the predictions derived from Google Trends. Paper [57] customizes the typical Bass predictive model of car sales forecasting by adding user-generated Internet information, search traffic, and macroeconomic data to get more accurate predictions. In every previous case, the addition of Internet data outperformed the results of the rest of the models.

2.4.3 Car Configurator Data Filtering Procedure

Despite everything above mentioned, Car Configurator is an online service with downsides. At first sight, the person who accesses the online tool is willing to purchase a car. Nevertheless, this is not totally true. There are two groups of users. In the first place, people who are in the exploration phase before performing the acquisition of the vehicle. They make extensive use of the tool. On the other side, people who are doing window shopping, i.e., browsing through the car variants without the intention of making a purchase. Distinguishing between both profiles is not straightforward. Authors from [58] found inconsistencies in the way the online environment is characterized to profile online consumer behavior and decision-making process. Additionally, the purchase approach has not evolved as much faster as technology. It is still necessary to head to a dealership in order to execute the purchase. They represent a third party with their commercial interests. Their influence is evident when customers arrive at the dealership intending to purchase a particular car configuration seen online but ultimately purchase a different one. This change in choice may be attributed to several factors, such as unavailability or late delivery of the first option or a generous discount offered on the alternative model. Regrettably, dealerships do not maintain records about deviations between the online and the physical world.

Combining this business categorization together with the fact that the car configurator webpage does not require any user commitment in the form of mandatory login, makes that generated data volume boosts, with the consequent inconveniences. Among the challenges Big Data entities should face, are (a) storage; (b) finding and solving data quality issues; (c) cost-effective escalation and appropriate choosing of big data technology; (d) data governance and validation; and (e) data collection processing and integration. Nevertheless, a well-managed Big Data environment can optimize operative costs, diminish time to market, and favor new products [59].

On one side, authors and references that have treated the problem of dealing with large databases are reviewed. Both from the point of view of technical aspects of managing a database and also from noisy data stored in the database. It is aimed at reducing the volume of clickstream data without compromising its significance, with the ultimate goal of enhancing the utilization of this information source in the decision-making process of the company. The way we define significance is by means of a correlation between the visits to the Car Configurator webpage and company sales. On the other side, it is pursued to find the path that users with real purchase intentions explore. In other words, pointing to the car attributes that signal a future transaction. To achieve this objective, literature about data mining techniques is consulted, but special attention to genetic algorithms. They are a type of optimization algorithm inspired by the process of natural selection. It explores the data gathered by the Car Configurator webpage. Similarly, it will seek to identify the characteristics that maximize the correlation between clickstream data and car sales.

Firstly, the list of technical issues includes lack of memory storage, which authors from paper [60] face in recognizing objects in large images database. That's why they decided to focus only on a small subset of the training features, based on the concept that many local features are unreliable or noisy. They perform a features filtering process thanks to an unsupervised preprocessing that recognizes the matching ones. Another challenge is represented in work [61], i.e., the extensive running time of algorithms in large databases. They propose to employ a Hilbert curve-based similarity searching scheme (HCS) in subsamples of the database, which projects each data point to a low-dimensional space. This strategy diminishes the searching time latency by mapping a certain data points cluster rather than the entire database. Another problem consists of database security, meaning querying and encrypting. The article [62] introduces the concept of a Verifiable Database with Incremental Updates (Inc-VDB), which allows a resource-constrained client to securely outsource a large database to a server while ensuring data integrity and the ability to update records. A good summary of how organizations can organize themselves to overcome these obstacles is found in [63, 64] in terms of data modeling and data management, respectively.

And secondly, there is a concern that needs to be highlighted: the presence of noisy data. It perturbs the data reliability and affects the outputs from any process in which the database is involved, among other facts. However, handling noise in Big Data is a challenge as traditional solutions struggle to deal with such large amounts of data. To overcome this challenge, new algorithms are needed to clean up the noise in Big Data and produce high-quality, clean data. Authors in reference [65] introduce two Big Data preprocessing methods, focused on scalability and performance, to remove noisy examples. They are a homogeneous ensemble filter and a heterogeneous ensemble filter. These methods have been found to effectively produce a clean dataset from any Big Data classification problem. Additionally, methodologies and techniques to achieve a filtered database are described in the book [66], especially in chapters fourth and fifth. Additionally, we find in the literature successful use cases of filtered Big Data, as it is related in note [67] for Small to medium-sized businesses (SMEs) from the agri-food sector. Another case is studied by paper [68], which explains how big data deployment transforms enterprise's policies in retail companies.

From the point of view of customer profiling, i.e., distinguishing the real customers out of the total users of the platform, the approach can be thought of as a feature selection exercise where the most relevant features are selected from the dataset to train machine learning algorithms. A comprehensive examination of the current advancements in methods for selecting relevant features can be found in [69]. Genetic algorithms deserve special mention. Starting with an initial population of potential solutions, these algorithms use a process of selection, crossover, and mutation to evolve the population over multiple generations toward an optimal solution. This approach has been shown to be effective for a range of optimization problems where traditional methods are either impractical or inefficient. Works [70, 71] utilized genetic algorithms for feature selection in optimizing support vector machines and classification tasks, respectively. In the context of data mining, book [72] provides a detailed guide to optimizing feature selection using genetic algorithms.

There are efforts, as well, in the context of using correlation as an assessment metric. It is called correlation-based feature-subset selection (CFS). It has proved its validity in cancer research, as it is explained in note [73]. A gene-search algorithm for analyzing genetic expression data was implemented, which combines a genetic algorithm with correlation-based heuristics for data preprocessing. Nevertheless, it is extended to other uses, such as the integration of data sources to build a Data warehouse, as it is related in paper [74]. It proposes a method for selecting an optimal subset of attributes based on correlation analysis, which identifies redundant attributes that do not significantly contribute to the overall characteristics of the data. Reference [75] proposed a correlation-based filter solution using a genetic algorithm for feature selection, which was able to identify relevant features quickly and accurately in high-dimensional datasets. The last case is found in the field of computer vision. Authors from work [76] enhance the accuracy of identifying apple leaf disease and reduce the dimensionality of the feature space. They select the most valuable features through a combination of genetic algorithms and CFS.

2.4.4 Improved Demand Forecasting With Genetic Algorithm

After examining all the examples of the aid that Internet data supplies in different sectors has opened a new research path. Taking into consideration that genetic algorithms are a powerful tool to perform data mining, and that clickstream data upgrades the performance of forecasting algorithms, the goal is to prove that better demand prediction can take place by combining these two techniques. The first instance in the literature to mention is found in note [77]. They perform exhaustive reviews of the applications of genetic algorithms in the forecasting field, focused on commodity prices such as energy, metals, and agricultural products. Their findings reflect the two paths to incorporate genetic algorithms into the forecasting: (a) parameter tuning; and (b) feature selection.

In the first category, authors from work [78] utilize a genetic algorithm to identify the order and estimation of the parameters for the SARIMA algorithm. The playground to check their methodology is climate data, more specifically the average temperature of India from 2000 to 2017. Their results outperform the prediction accuracy and are executed in parallel. However, genetic algorithms have been also used to parameter tuning more complex machine learning techniques, such as neural networks. Good examples in this area are papers [79, 80]. In the first research, the genetic algorithm delineates the weights, the bias, as well as the number of hidden neurons. Their proposal is evaluated against various benchmark methods, including neural networks utilizing back-propagation, Support Vector Regression, and ARIMA across both the most popular time series datasets and real-life data. They produce better forecasts with the genetic-based optimization technique than the rest of the methods. On the other side, the second paper concentrates on a multivariate LSTM neural network, more specifically bidirectional LSTM (BiLSTM). Outcomes proved that the tuned BiLSTM model surpasses other procedures, having an accuracy of 89% in the sales prediction. Special mention deserve the next references, as they are the only ones related to vehicle sales. Indonesian car sales forecasting is assessed on note [81]. The authors integrate a genetic algorithm to optimize the parameters of Holt-Winters exponential smoothing to predict the demand for the most popular car brands. By means of MAPE comparison, the proposed method outperformed the rest. Finally, the Chinese market of electric vehicles is explored in research [82]. The market penetration of electric vehicles (EVs) in the country for the next ten years is predicted with a comprehensive Bass diffusion model fitted with a genetic algorithm. The model prediction results show that EVs are successful innovative diffusion products.

Nevertheless, our research is more similar to the second approach: feature selection. The stock market is a key player in this context. It is a sector alike to our problem as it presents a large number of indicators to measure the variability of stock price. In the same way, we pursue to select the insights from clickstream data with more predictive power. Authors from reference [83] define the genetic algorithm as a ranker of the factors of importance. Afterward, this lineup feeds an LSTM model for stock prediction of the China construction bank dataset and the CSI 300 stock dataset. Their experiments prove the supremacy of the newer model with respect to all the baseline models for time series prediction. A similar strategy is followed in note [84]. They center on four international stock indices and combine genetic algorithms with Random Forest. The first technique chooses a batch of helpful features among the indices. Therefore, the second algorithm unveils hidden relationships between the set and a particular stock's trend. The rehearsal demonstrated that the hybrid predictive model surpasses the performance of the basic forecast by a significant margin.

In another chapter, climate is another rich-data field where the interaction among the features is not the simplest. We found examples of research forecasting the air quality [85] and the risk of flood [86]. The first paper compares the outcomes from the set of features selected by a genetic algorithm against two other filtering techniques. Next, the daily maximum concentration of two air pollutants is predicted. For the second research, the features batch derived from the genetic algorithm feeds a Linear Regression model that executes the forecasting of the level of the Xingu River. In both cases, the genetic algorithm approach delivers the best outcomes. Another sector that benefits from the power and versatility of genetic algorithms is the electricity market. These researchers [87] explored the advantages of genetic-based feature selection in the Australian market using the M5P forecaster.

Nevertheless, the examples dedicated to sales and/or online data scarves. In the retail sector, note [88] introduces a genetic algorithm to expand the feature set and investigate potential feature values. Their validation dataset is the Kaggle Rossman sales data, and the forecasting algorithm is a tree-based model. The findings indicate that the suggested approach can substantially enhance both the precision and robustness of decision tree algorithms. The last example is found in paper [89], for phishing website detection. Traditional methods dedicated to this task are blacklisted based, however, they are rapidly outdated. The study utilizes a genetic algorithm to pinpoint the most influential features and determine the optimal weights for website characteristics. Consequently, the genetic algorithm's selected and weighted website features are employed in training a neural network to achieve the objective. All resulting metrics demonstrate the superior capacity of this approach.

2.5 Reducing Disparity Between Stock And Demand Updating The Production

The misalignment between the arrangement of the stock and the composition of the demand prediction causes inefficiencies. This miscoordination often leads to excess inventory, resulting in increased logistics costs. Moreover, it may imply production delays due to shortages in crucial components. All these causes are reflected in a growing customer dissatisfaction. Firstly, this situation has been intended to be solved after the vehicles are manufactured and before they are headed to the different stock locations of the com-

pany. Nevertheless, the approach chased in this Section puts its attention on those cars that are still in the assembly stage. The production framework that SEAT employs permits the modification of vehicle components non-dependant on the cable tree. In order to decide about which modifications should be done, demand prediction advice in the decision-making process. Forecasting provides crucial insights into the expected popularity of specific vehicle configurations and options. In case a particular set of features is projected to be in high demand, adjustments can be made in real time.

This problem can be faced thanks to heuristic search. Inside this category, it is found simulated annealing [90, 91] or tabu search [92, 93]. Additionally, it is possible to highlight the evolutionary programming, i.e., genetic algorithms. These last components were extensively explored during the thesis research. That's why it is pursued to explore different options to figure out the most optimum production.

One of the existing solution methods with a large history behind is linear programming. It is a mathematical optimization technique used to find the best outcome in a mathematical model with linear relationships. In other words, it finds the optimal solution to a problem where both the objective function and the constraints can be expressed as linear equations or inequalities [94]. In the current framework, the constraints are subjected to linear behavior. On the contrary, the objective function does not follow this conduct as it is ruled by quadratic terms. Another renowned method is called least-squares. It is defined as an optimization problem in which the objective function is a sum of square terms but with no constraints [94]. Consequently, it is not applicable to this problem as the restrictions are well delimited. Therefore, it is necessary to explore more general techniques. That is how convex optimization has been encountered.

A convex optimization problem is one in which the objective and constraint functions are convex. In opposition to linearity, convexity is more general: inequality replaces the more restrictive equality. And the inequality must hold only for certain values [94]. To ensure the viability of the methodology, Disciplined Convex Programming (DCP) is employed. DCP analysis breaks expressions down into subexpressions and applies a general composition theorem from convex analysis to each one of them. The convexity of $f(expr_1, expr_2, ..., expr_n)$ is determined by two factors: the nature of the function f itself and the characteristics of each of the individual expressions $expr_i$. It is considered convex under the following conditions: (a) f is increasing in argument i and $expr_i$ is convex; (b) f is decreasing in argument i and $expr_i$ is concave; (c) $expr_i$ is affine or constant. If none of these three rules apply, the curvature of the expression $f(expr_1, expr_2, ..., expr_n)$ is categorized as unknown [95]. The tool found to assist in the optimization routine is CVXPY. It is a modeling language integrated with Python, designed for addressing convex optimization problems. It simplifies the process by automatically converting the problem into a standard format, invoking a solver, and then extracting and presenting the solution. More details in [96, 97].

2.6 Research Gap

To sum up, it is possible to find works related to the three fundamentals of the Fast Lane. Despite it being nowadays abandoned, its way of procedure is still valid. It is the base of the advances presented in this document. From the point of view of the operative, the current manufacturing is a restriction. It cannot be modified in any way and/or it is out of the scope of this project to propose it. The same for the different pieces, such as the cable tree, that assemble a car. Therefore, the formula revealed in the literature would be

extremely laborious to build. Consequently, this research assimilates these limiting inputs. On one hand, it deepens in the moment when the car is ready to abandon the factory. To do so, the current SEAT's outbound logistics approach is related, together with a proposal to shorten the stock volume. The idea behind this solution consists of dispatching the manufactured vehicles to the location where it is expected to spend a lesser number of days waiting until a customer purchases them. The problem is structured as a binary classification one based on the attributes that compose a car. It is performed thanks to Machine Learning classification algorithms. Results equal or improve the decisions made by the experts of the company.

On the other hand, the manufacturing process cannot be modified, but it is permitted to change some attributes, the non-electrical dependant ones, of the cars under assembly. That's why forecasting is seen as the perfect tool to embrace this opportunity. A new block in the thesis is opened. The bibliography has presented evidence of works dealing with Internet data to perform predictions, not solely in the car market, but in other sectors as well. However, none of these notes exploit or have access to the data gathered by the Car Configurator, the first contact point between the customer and the company. At the beginning, the reliability of the clickstream data is under inspection. Results analyzing the correlation prove that Car Configurator visits subsequently influence in the sales record. Moreover, the performance of forecast algorithms that add this online source is superior to predicting future sales.

Nevertheless, the clickstream dataset deserves to be truly well managed because of its own nature. Quantitative and qualitative approaches are executed. On one side, the data is filtered applying some rules to eliminate outliers. The bibliography is vast in terms of technical considerations and several solutions are exposed, specific literature is modest in the field of dealing with noisy data. Specially, for a problem with the characteristics previously referred to: filtrate big data without losing significance. Each one of these rules defines an outlier differently or it is applied in a distinct dataset feature. The significance preservation is validated thanks to statistical tests. On the other hand, genetic algorithms are used to find the car variants configured in the webpage that identify the users with real purchase intention. There is no evidence of research that addresses user-generated data from non-transactional webpages. The core of the answer explores the possible choices that define each attribute of a car variant and aggregates all of them into a set of rules for filtering the dataset.

In both cases, results show excellent behavior with respect to the baseline. These experiments were carried out under circumstances of low variety, as they only affect changeable elements, such as the color or destination, of the vehicle. These attributes are non-dependant on the cable tree or spare parts, and the number of combinations is not overwhelming, which transforms them into the perfect candidates. To last up, the efficiency of the genetic algorithm is employed to improve the outcomes of the demand prediction done at the beginning of the block. Despite it having been tested and validated in both tasks of parameter tuning and feature selection, for a range of areas of knowledge, the sector of user-generated online data and car sales has not been under analysis. These predictions are utilized to assign a new destination to the vehicles in the manufacturing line. The purpose is to decrease the gap between the composition of expected demand and the vehicles within the stock. In all cases under analysis, it was achieved an improvement in the objective function, diminishing the disparity between what customers want and what the factory manufactures.

Chapter 3

Exploratory Data Analysis

Throughout this chapter, the available data provided by SEAT is intricately detailed. The focus is confined to the Spanish market and centered on the urban SUV SEAT Arona and the utility vehicle SEAT Ibiza, both belonging to segment B and derived from the same platform. Additionally, attention is given to the segment C SEAT Leon (Leon 5D) and the SEAT Leon Sportstourer (Leon ST), the family-oriented variant of the compact model. Essentially, these are the vehicles produced at the company's headquarters, over which the firm exercises complete operational control. The data encompasses: (a) production and delivery planning of vehicles, (b) sales records, and (c) user visits to the Car Configurator webpage of the company.

3.1 Production And Deliveries

The information regarding the operation of the company spans from January 2017 to January 2020, both inclusive. The dataset encompasses over 200,000 rows. Essentially, it captures the date on which cars traverse every milestone in the production process. It is convenient to understand what is the context of the company in terms of the operatives and logistics. Everything it follows makes reference to the SEAT headquarters, placed in Martorell, approximately 30 km northwest of Barcelona. The sole facility where the company exercises complete authority over the logistics and production within the automotive assembly process. It boasts a notable production capacity of approximately 500,000 units annually.

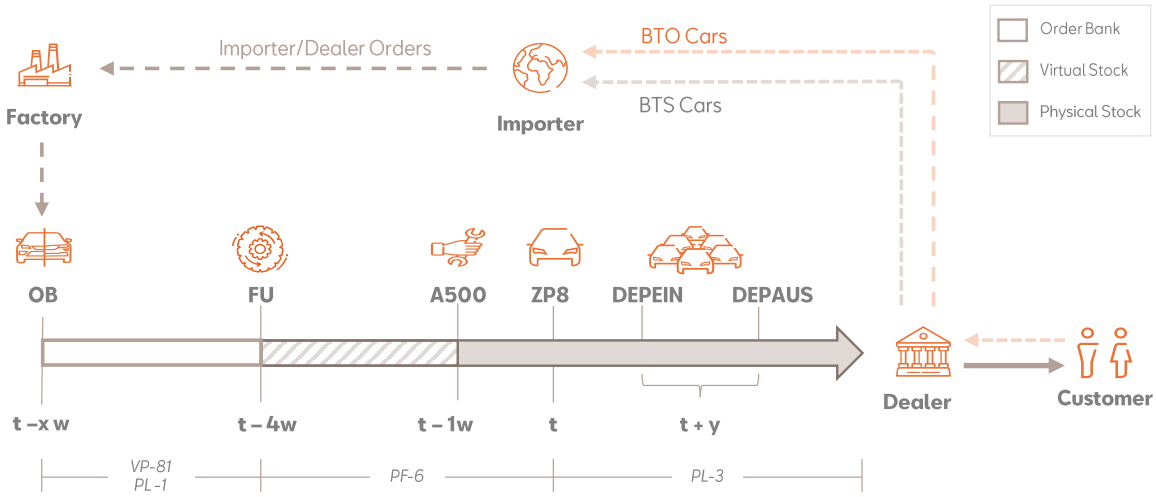


Figure 3.1: Manufacturing flow and supply chain management diagram within SEAT headquarters.

Figure 3.1 illustrates the entire flow of vehicle manufacturing. Prior to the vehicle's delivery to the customer at the dealership, it undergoes various stages involving different stakeholders. The dealership serves as the initiator of this process by sending car orders to the factory, indicating the vehicles they are prepared to sell. At this stage, two distinct request groups can be identified. Firstly, there are orders supported by existing customers, referred to as Build-to-Order (BTO) cars. Conversely, there are requests that the dealership anticipates selling in the future and consequently keeps in its inventory, known as Build-to-Stock (BTS) cars. The data subdivides these order types into four categories: '0' - private customer; '1' - fleet cars; '2' - dealership stock; '3' - importer stock. The first two categories correspond to BTO cars, while the last two categories represent BTS cars. It is worth noting that the role of the importer exists as an intermediary between the dealerships and the factory. The importer consolidates all order requests received from the dealership network and possesses the authority to request cars for its own purposes. These cars form an additional reservoir, supplementing the existing stock, to meet the demands of the dealerships. Ultimately, all these orders reach the factory and are recorded in a repository referred to as the Order Bank.

In order to learn how the order types are manufactured every year, this information is gathered in Table 3.1. For the first two years, the production is balanced between both order types. However, BTO vehicles take the lead in the last year of data. It is necessary to clarify that 2020 data only includes the first month of the year and that 2019 was the best year for the company in terms of sales. That explains the disproportion between groups. Additionally, the trend of the production can be observed along the production weeks in Figure 3.2. This image helps to understand the production cycles of the factory, such as the null manufacturing rate during Christmas holidays and/or summer, but year 2019. Despite the decreasing trend, BTS cars have a significant influence on production.

Table 3.1: Production rate of Build-to-Order (BTO) and Build-to-Stock(BTS) vehicles per year

Order Type		2017	2018	2019	2020
BTO (%)	private customer	29.43	33.65	37.23	50.43
	fleet cars	16.58	15.11	21.94	11.76
BTS (%)	dealership stock	53.99	39.33	33.78	37.8
	importer stock	0	11.91	7.05	0

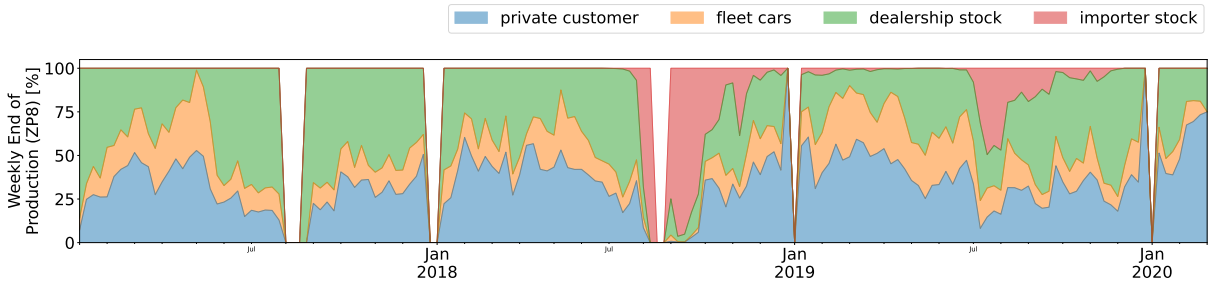


Figure 3.2: Weekly production segmented per Order Type. Data is presented in terms of ZP8 week, i.e., the calendar week in which vehicles abandon the manufacturing line.

Additionally, it is worth giving more details about how it is structured the production of BTS vehicles along the dataset. Figure 3.3 collects the details of BTS weekly production, measured in terms of ZP8 calendar week, per car model. This date represents the week the manufactured car abandons the factory after passing all quality checks. It is seen the period of inactivity of the factory, such as the summer and Christmas holidays. The exception occurs during the summer of 2019. As it was the top seller year of the brand's history, there were enough production orders to keep the factory open. However, the production peak for the four car models does not take place during this year, but it happens in 2018. The maximum manufactured units of SEAT Leon 5D car model were 420 at the 35^{th} calendar week of the year, i.e., the last week of August. For the rest of the car models, the maximum volume happened in the 28^{th} calendar week, during the first half of July. The number of units produced for each car model starts from 246 (SEAT Leon ST) up to 587 (SEAT Ibiza), passing through 368 (SEAT Arona).

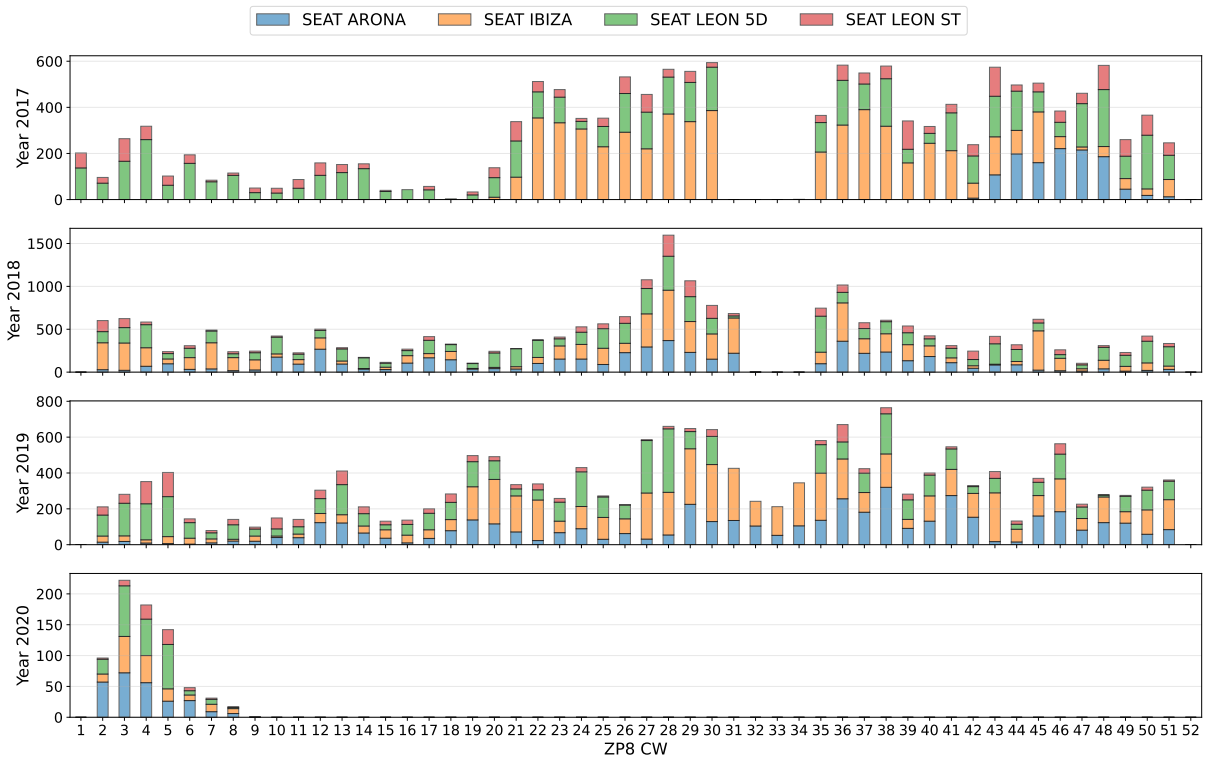


Figure 3.3: Weekly production of Build-to-Stock vehicles segmented per car model and year. Data is presented in terms of ZP8 week, i.e., the calendar week in which vehicles finish the manufacturing line.

The responsibility for managing the Order Bank lies with the commercial department (VP-81) of the brand. In collaboration with the markets and the factory, they determine which batch of orders from the Order Bank will commence production in the following week. This significant milestone is referred to as Order Generation or FU. The selection criteria for orders from the Order Bank vary and can include factors such as prioritization lists provided by dealerships; preference for BTO cars over BTS cars; former Fast Lane vehicles, a first-in, first-out (FIFO) policy; cars' contribution margin; and others. However, the department also takes into account the factory's operational status. In-house Logistics (PL-1) provides information regarding ongoing restrictions, i.e., the availability of components required for car assembly.

Subsequently, the Production department (PF-6) assumes control of orders in the FU status. One of their tasks is to determine the production sequence of vehicles, specifying the production day of the week and shift. They also initiate communication with various suppliers in charge of providing different components except for engines, which are Volkswagen Group's responsibility. Among these components, the cable tree is what defines the TRIM level of the vehicle. TRIM represents the specific features, options, or design elements that distinguish versions of the same model. It ranges from standard to premium. Moreover, cable tree holds critical importance as it has the longest lead time. Consequently, the phase preceding actual manufacturing extends for around three weeks. During this stage, neither the cable tree nor the engine can be modified. However, other elements such as color, alloy wheels, and final destination can still be adjusted, thanks to the improvements carried on during the Fast Lane project. Once all the components are prepared following this preparatory period, the physical production process begins,

and no further changes to the order composition are allowed. This phase commences at a point referred to as A500. It is at this stage that the vehicle undergoes the transformation from a metal sheet in subsequent stations, ultimately becoming a finished product within a single week. Table 3.2 illustrates the manufacturing flow and the changes permitted in each milestone.

Table 3.2: Production process and the modifications of attributes permitted in each milestone. Colored circles represent the attributes that cannot be varied. Empty circles symbolize changeable attributes. W means week

	OB t - xw	FU t - 4w	A500 t - 1w	ZP8 t_0
Platform/Car Model	●	●	●	●
Engine	●	●	●	●
Cable Tree/TRIM Level	○	●	●	●
Non-Electrical (color/alloy wheels)	○	○	●	●
Destination (compound region)	○	○	●	●

The information about the number of available options per car model registered in the data is placed in Table 3.3. The common TRIM levels for the four car models are named, from basic to top: Reference, Style, Xperience, and FR. Additionally, the SEAT Leon family has a supreme version called CUPRA; and SEAT Leon ST, in particular, includes a mid version called Xcellence. The urban SUV is the car model with the largest number of choseable colors due to the car body and roof having distinct coloring. However, it should be noted that not all combinations of attributes are offered within the company's commercial lineup. For instance, certain colors are exclusive to specific TRIM levels, and the same applies to engine or alloy wheel options. As a result, the actual number of combinations observed in the dataset is significantly lower due to these restrictions.

Table 3.3: Number of available elements in each attribute for each car model in SEAT production and deliveries data.

	TRIM	Color	Alloy Wheels	Engines
SEAT Arona	4	48	9	6
SEAT Ibiza	4	12	8	9
SEAT Leon 5D	5	14	16	16
SEAT Leon ST	6	14	17	16

The subsequent steps are overseen by the Outbound Logistics department (PL-3). Regardless of whether a car leaving the factory is BTO or BTS, both types follow the same route. This department is responsible for the car's reception in the stock warehouse, denoted as DEPEIN, as well as its departure from this location, indicated as DEPAUS. In the company jargon, these locations are referred to as redistribution compounds or just compounds. After the manufacturing process, the cars are directed to the nearest compound in close proximity to their final destination. These facilities serve as collection points for cars ordered by dealerships within their respective geographical areas of influence. In the case of Spain, the national territory is divided into six compounds, as depicted in Figure 3.4.

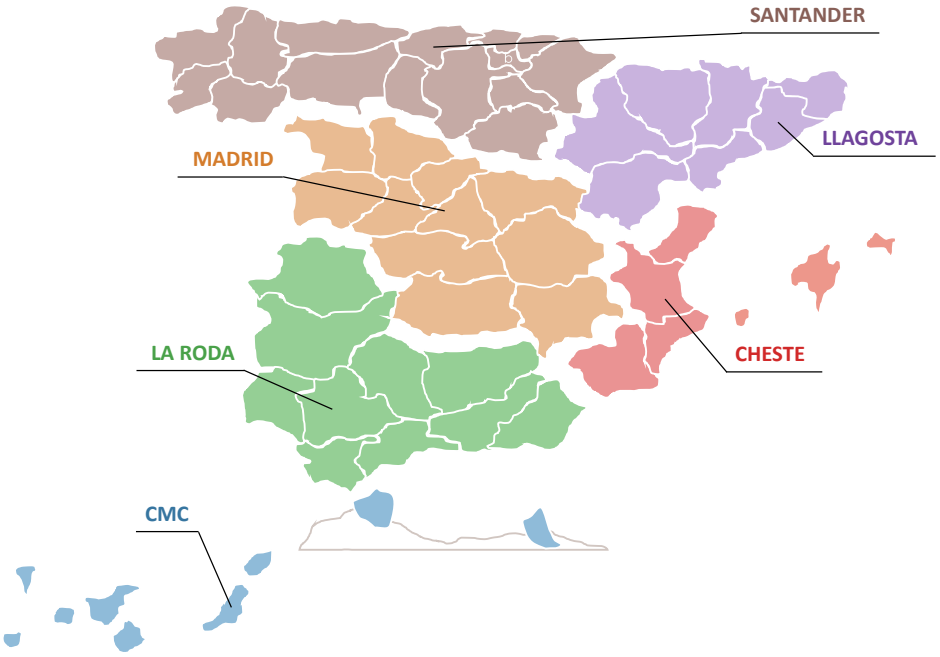


Figure 3.4: Map reflecting the location of the different compounds in which SEAT divides Spanish territory

Another relevant look in this exploratory phase of the BTS vehicles is to understand the behavior per compound region. Table 3.4 collects the weekly average production rate of BTS vehicles per compound region and year. Over the years, the distribution is preserved mainly constant. The compound region with the lowest production rate is CMC, although it has been increasing over the period. On the contrary, the winning compound region varies from MADRID to LLAGOSTA. It is not bizarre as they are the region that accommodates the largest population within the national territory. However, their weight is not very distance from the rest of the compound regions. A reason for this behavior might be found in the production policy. It seems that is more oriented to fill the diverse compound regions according to the dealerships' demands, leaving some room for improvement.

Table 3.4: Weekly average production rate of BTS vehicles per compound region and year

	2017	2018	2019	2020
CMC (%)	4.04 ± 3.56	7.19 ± 5.51	8.33 ± 5.8	11.11 ± 4.05
MADRID (%)	25.91 ± 13.76	20.27 ± 4.51	18.36 ± 4.92	18.41 ± 4.09
LA RODA (%)	20.35 ± 8.4	18.89 ± 7.22	21.16 ± 6.71	16.61 ± 5.31
CHESTE (%)	14.39 ± 5.51	12.91 ± 3.11	12.76 ± 4.38	16.18 ± 9.3
LLAGOSTA (%)	20.96 ± 5.84	22.68 ± 6.06	23.08 ± 6.29	32.01 ± 28.97
SANTANDER (%)	17.92 ± 4.89	18.63 ± 3.8	16.31 ± 3.77	15.4 ± 6.89

The primary distinction between BTO and BTS cars lies in the duration of their stay within the compounds. BTS cars remain in the compounds for an extended period as they await a customer to purchase them. Conversely, BTO cars spend minimal time within the compounds as there is already a client awaiting their delivery. The term to call this milestone is KDUEB, when the customers finally receive their purchased cars. The number of days between ZP8 and KDUEB is named Time in Compound within the data. For instance, 105 car variants have recorded the minimum time, which is a single day. The longest duration within the compound was observed in the case of a dealership's SEAT Leon ST, which remained for 716 days. The main statistics about Time in Compound per compound region are found in Table 3.5. In this block, a car variant is defined as the combination of car model, TRIM level, color, engine, together with order type.

Table 3.5: Main descriptive values for Time in Compound [days] per each compound region individually collected in the production and deliveries dataset.

	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Min. [days]	1	1	1	1	1	1
Mean [days]	62	53	60	52	48	54
Std. Dev. [days]	52	57	63	56	55	55
Q1 [days]	26	14	16	14	14	18
Q2 [days]	46	29	34	27	25	30
Q3 [days]	82	71	82	71	62	69
Max. [days]	716	447	516	490	470	554
No. of Variants	532	1259	1004	1046	1267	1105
No. of Cars	8670	24526	16608	14216	31874	17432

In Table 3.5, it is observed that the CMC compound stands out distinctly from the others due to its significantly higher number of days spent in the compound. Furthermore, it has the lowest number of cars and variants among all the regions. Conversely, the LLAGOSTA compound emerges as the region with the best deliveries. It exhibits the lowest values in the time distribution and the largest figures in terms of the number of variants and cars. However, it is worth noting that the statistical values for all compounds exhibit a similar order of magnitude. This similarity indicates that they follow nearly identical distributions, characterized by a large concentration of cases on the left and a long tail on the right side. This behavior is visually depicted in Figure 3.5, where it becomes apparent that distinguishing one compound from another is practically impossible.

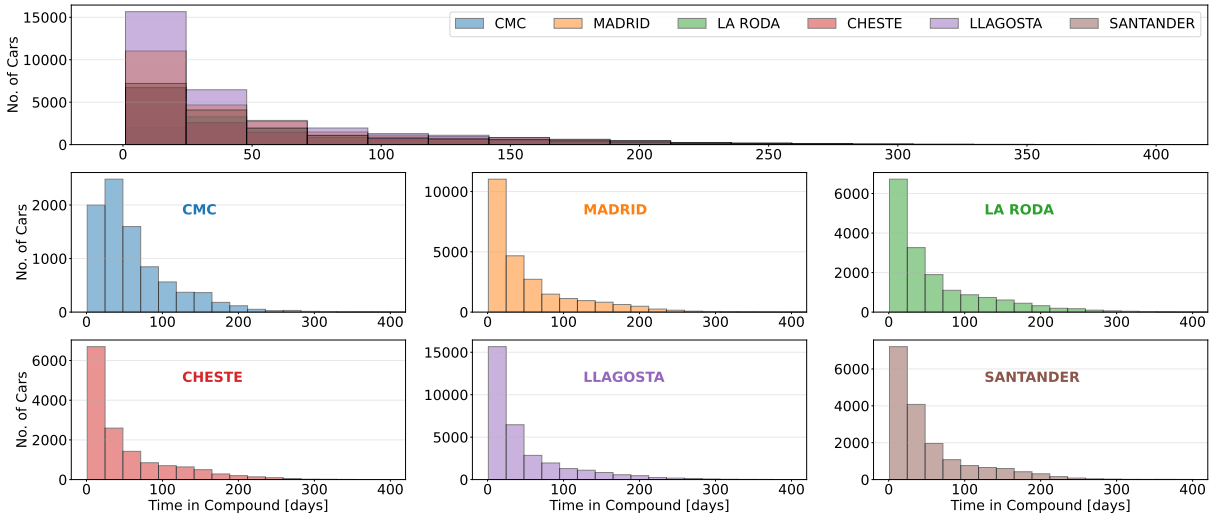


Figure 3.5: Distributions per Compound Region of Time in Compound registered in the production and deliveries dataset.

The analysis of Time in Compound of cars in terms of car model follows the same trend previously observed per compound region. The summary statistics are placed on Table 3.6. On average, SEAT Arona is the model with the fastest delivery time, while SEAT Ibiza occupies the last position. Nevertheless, the most popular car model is the SEAT Leon 5D but the SEAT Leon ST is the least sold. From Figure 3.6, it is learned that the distributions of Time in Compound overlap in the same range for every car model, which is not helpful to differentiate the deliveries in categories.

Table 3.6: Main descriptive values for Time in Compound [days] per Car Model individually collected in the production and deliveries dataset.

	SEAT Arona	SEAT Ibiza	SEAT Leon 5D	SEAT Leon ST
Min. [days]	1	1	1	1
Mean [days]	43	60	55	54
Std. Dev. [days]	46	61	57	62
Q1 [days]	14	17	16	15
Q2 [days]	24	35	31	27
Q3 [days]	56	83	74	68
Max. [days]	470	462	497	716
No. of Variants	547	250	347	379
No. of Cars	26940	26940	33571	18589

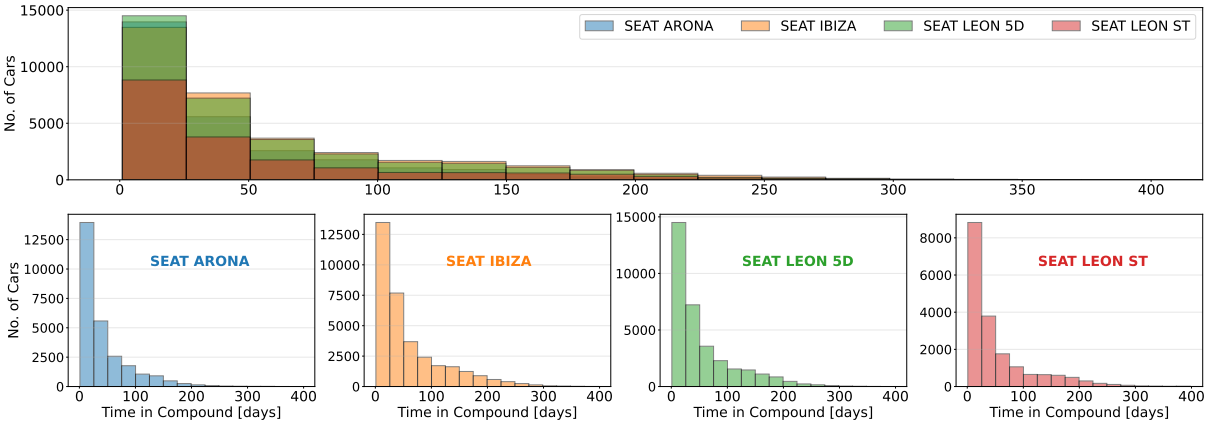


Figure 3.6: Distributions per Car Model of Time in Compound registered in the production and deliveries dataset.

Therefore, it is studied what it is the behavior between BTO cars and BTS cars, identified by the Order Type. Bearing in mind that there are two categories for BTO cars and other two for BTS cars. From Table 3.7, as it might be presumed, BTO cars are faster in the delivery time than the other category, especially those from private customers rather than fleet cars. This information is acquired too from Figure 3.7. However, stock cars represent a significant percentage of the total deliveries, mainly dealerships stock due to importer stock serving as a backup for the former. That's why we consider that optimizing the delivery time within the different compound regions can have such a positive impact, especially when there are epochs in the time range where BTS cars represent a larger percentage in the weekly deliveries than BTO cars, as is shown in Figure 3.8.

Table 3.7: Main descriptive values for Time in Compound per Order Type individually collected in the production and deliveries dataset.

	Build-to-Order		Build-to-Stock	
	Private Customers	Fleet Cars	Dealerships stock	Importer Stock
Min. [days]	1	1	1	2
Mean [days]	25	51	77	60
Std. Dev. [days]	29	49	67	54
Q1 [days]	12	20	26	20
Q2 [days]	16	35	55	41
Q3 [days]	26	65	113	88
Max. [days]	516	716	554	490
No. of Variants	1393	1017	1263	454
No. of Cars	38846	20326	46099	8056

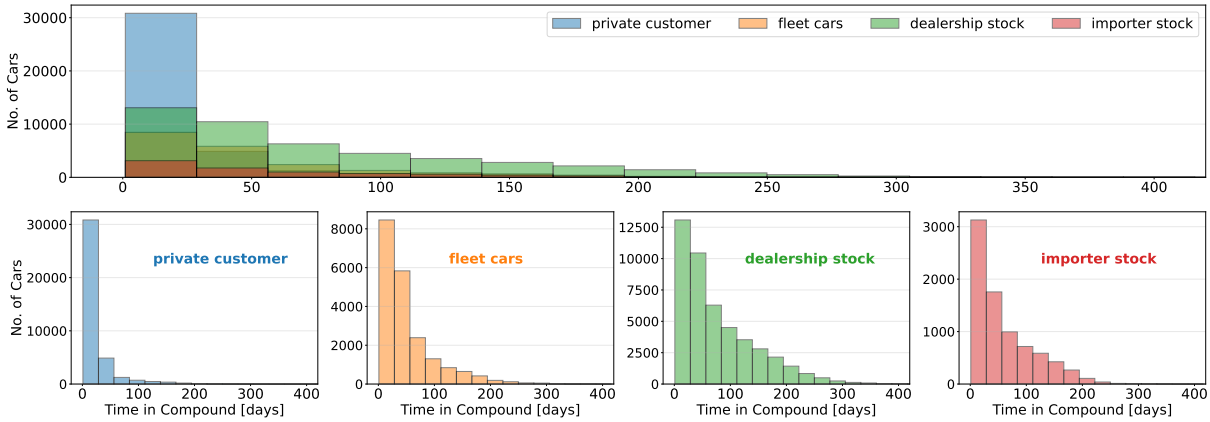


Figure 3.7: Distributions per Order Type of Time in Compound registered in the production and deliveries dataset.

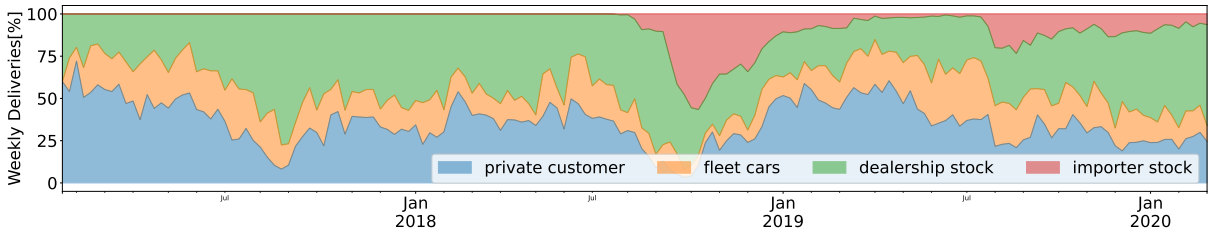


Figure 3.8: Distributions per Order Type of Time in Compound registered in the production and deliveries dataset along the entire timespan.

3.2 Sales Record

The compilation of sales records commenced in April 2017 and concluded in January 2020, encompassing both months. This period translates into 149 weekly instances. The timeframe makes reference to the day when customers reserve their vehicles at the dealerships, subsequent to the payment of a booking fee. Consequently, the decision has been made to designate this moment as the purchase date, rather than using the delivery date. The sales data includes information about the location of the purchase and provides a comprehensive description of the vehicles, such as the car model, trim level, engine specifications, and color, aside from the engine. However, a classification tag distinguishing between different order types is absent. Nevertheless, the firm provided registers exclusively related to regular customers, excluding fleet cars from the sample. Furthermore, to safeguard privacy, the company has taken measures to anonymize the vehicles by removing all identifying markers, including the buyer's name and the manufacturing serial number. The dataset comprises over 120,000 rows of data.

Table 3.8 shows the number of elements found in each attribute within the sales record. Unfortunately, the information about the alloy wheels is not available in the dataset. The gap between these values and the ones shown in Table 3.3, especially in engines, are consequence of the temporal mismatching between both datasets. In other words, the sales record only registers the date of the purchase, not when those vehicles were manufactured. From Table 3.9, SEAT Ibiza exhibited the highest weekly sales figures among all car models, with a maximum of 613 units sold per week. Additionally, it is the most represented car model within the sales (35229 units sold). On the other hand, SEAT Ibiza together with SEAT Arona have the lowest minimum weekly sales, directly affecting the magnitude of the the standard deviation values. The starting epoch has no activity. For the SEAT Arona, it represents the launching of a totally new product. In the other case, SEAT Ibiza suffered a version update, meaning the product was replaced. Sales records from the old version are not available in this study. The overall trend suggests that Leon family models are relatively more balanced. The same descriptive analysis of the main statistics for the weekly sales is done in Table 3.10 at the compound region level. Llagosta first, and afterward MADRID, both are the compound regions with the largest number of sales, average, and rest of metrics. The last position is occupied by CMC (6547 cars sold). Nevertheless, it is not the place with the lowest minimum weekly sales, tied with SANTANDER. This position is occupied by CHESTE. Lastly, Table 3.11 collects the name of the element with the largest and least sales volume in total in the sales record per attribute. The homogeneity in the best-seller TRIM levels is not reflected in the least popular ones. The same occurs in the exterior color of the vehicle. The diversity is large in terms of the engine for both cases. On the contrary, compound regions are a loyal representation of the previous learnings.

Table 3.8: Number of available elements in each attribute for each car model in SEAT sales record.

	TRIM Level	Exterior Color	Engines
SEAT Arona	4	48	8
SEAT Ibiza	4	12	12
SEAT Leon 5D	5	18	31
SEAT Leon ST	6	19	39

Table 3.9: Main descriptive statistics for the weekly car sales per car model in the sales record

	SEAT Arona	SEAT Ibiza	SEAT Leon 5D	SEAT Leon ST
Min	0	0	11	4
Mean	192	236	203	92
Std. Dev.	119	116	81	38
Q1	143	172	147	66
Q2	208	231	189	84
Q3	256	300	248	110
Max	544	613	536	197
Total	28612	35229	30300	13663

Table 3.10: Main descriptive statistics for the weekly car sales per compound region in the sales record

	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Min	1	4	2	0	7	1
Mean	44	145	130	96	182	126
Std. Dev.	22	61	66	46	76	59
Q1	29	108	85	68	132	85
Q2	47	139	122	89	175	117
Q3	59	174	164	121	228	151
Max	97	351	346	260	400	326
Total	6547	21610	19416	14315	27080	18836

Table 3.11: Element with the most and least sales volume per car model in the sales record data

		SEAT Arona	SEAT Ibiza	SEAT Leon 5D	SEAT Leon ST
TRIM Level	Max	Style	Style	FR	FR
	Min	Reference	Xperience	Cupra	Xcellence
Exterior Color	Max	B4B4	B4B4	2Y2Y	2Y2Y
	Min	L5F5	F5F5	S3S3	E4E4
Engine	Max	KX	CV	IX	2X
	Min	GZ	GZ	TY	NU
Compound	Max	LLAGOSTA	LLAGOSTA	LLAGOSTA	MADRID
	Min	CMC	CMC	CMC	CMC

3.3 Car Configurator Data

The visits to the SEAT car configurator Spanish webpage are from the same period as the sales records, from April 2017 to January 2020. The clickstream data gives access to know all the components and attributes of the configuration explored by the users, from which geographical location it was scouted, how many times the user has done the same search, and even, in which step of the process the configuration was interrupted. Nevertheless, the firm exclusively extracted the information of the users who have completed all the steps within the webpage. Despite this restriction, the amount of data collected by the online tool is massive. In our context, plain clickstream data contains close to 19M rows. The webpage has no cost to the user and there is an absence of mandatory login. However, each user is individually identified by means of a unique alphanumeric code derived from their internet browser's cookies. This is the only possible tracking approach.

The cleaning process in which the clickstream data has been involved includes deleting any null values in the relevant variables of the dataset. For example, accessory items of the car have not been registered in the 9% of the samples; car equipment in 2%; and more than 100,000 rows have *null ID*. In the same way, all users placed outside the Spanish national territory are neglected. Therefore, after the cleaning procedure, the information is enclosed in 3,689,418 rows for 1,890,579 visitors. Beware that now each row represents a car variant and a date. A visitor is defined as a user of the online tool who accesses the webpage once a day to create a single car variant. Specifically, if an individual configures the same car variant multiple times within a single day, it is counted as a single visitor. However, if the user explores different car variants on the same day, each configuration is considered a separate visit. This decision is based on the premise that the aim is to assess the level of interest in car variants. It is proposed that growth in car variant visits by multiple users over an extended duration may indicate a triggering effect for future sales.

The data is described, firstly, in Table 3.12. Once again, alloy wheels are a feature not available. In addition, the range of the figures lies in the order of the magnitude of the ones seen in the sales record. With respect to the weekly performance of the visits to the car configurator, Table 3.13 shows the main descriptive stats at the car model level. On the other side, Table 3.14 do it per compound region level. Once again, the zero visits to SEAT Ibiza and SEAT Arona are a consequence of the aforementioned situation. Despite this background, these models are averaging the largest number of visits per week. Nevertheless, SEAT Leon 5D is the most popular car model among the users of the online tool. Regarding the geographical location, the data has been aggregated according to the area of influence of each compound region. Therefore, visits received from the MADRID region overpass the rest of the regions, in which LLAGOSTA occupies the second place and the CMC area the last one. These habits are confirmed in Table 3.15. Other insights are the preference of SEAT Arona visitors over the Xperience trim, being the latter the least popular trim for the rest of the car models. Regarding exterior colors, the most popular ones are shared by vehicles of the same segment, whilst the least visited colors are more diverse. On the contrary, there is consensus on the least preferred engines from part of the users of the car configurator.

Table 3.12: Number of available elements in each attribute for each car model in SEAT car configurator data.

	TRIM Level	Exterior Color	Engines
SEAT Arona	4	47	8
SEAT Ibiza	4	14	12
SEAT Leon 5D	5	15	24
SEAT Leon ST	6	14	33

Table 3.13: Main descriptive statistics for the weekly car configurator visits per car model in the car configurator data

	SEAT Arona	SEAT Ibiza	SEAT Leon 5D	SEAT Leon ST
Min	0	0	699	385
Mean	5096	6749	6803	3360
Std. Dev.	3444	3161	2338	1133
Q1	3508	4524	5158	2361
Q2	5305	6292	6613	3672
Q3	6487	8103	8259	4153
Max	17647	21472	13832	6286
Total	759266	1005556	1013584	500633

Table 3.14: Main descriptive statistics for the weekly car configurator visits per compound region associated with the geographical access point of the user in the car configurator data

	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Min	56	976	386	332	726	403
Mean	462	7161	2900	2381	6258	2846
Std. Dev.	171	2342	1068	878	2127	1023
Q1	335	5462	2117	1659	4519	2121
Q2	436	6920	2695	2358	6156	2803
Q3	560	8937	3679	3068	7713	3572
Max	900	13146	6328	4708	11088	5543
Total	68846	1067007	432072	354721	932410	423983

Table 3.15: Element with the most and least visits to car configurator per car model in the car configurator data

		SEAT Arona	SEAT Ibiza	SEAT Leon 5D	SEAT Leon ST
TRIM Level	Max	Xperience	FR	FR	FR
	Min	Reference	Xperience	Xperience	Reference
Exterior Color	Max	B4B4	B4B4	2Y2Y	2Y2Y
	Min	9P9P	W0W0	L5S7	T4T4
Engine	Max	CV	KX	XX	XX
	Min	GZ	GZ	TZ	TZ
Compound	Max	LLAGOSTA	MADRID	MADRID	MADRID
	Min	CMC	CMC	CMC	CMC

3.4 Comparison Between Sales Record And Car Configurator Data

The previous sections give insights into the sales record and the clickstream data of the Car Configurator webpage individually. However, as the customer journey normally begins with searching for information and finishes when purchasing, it is relevant to comprehend the relationship between both data sources. Figure 3.9 shows that the SEAT Ibiza and the SEAT Leon 5D are the most popular car models in both sales records and online visits. SEAT Arona occupies the third place in the two categories, and SEAT Leon ST closes the ranking. This trend has been constant for all the years, except 2017. It is not atypical, due to it refers to the launching moment of the SEAT Arona. The market behavior was impacted by the introduction of the new model until it was corrected in the subsequent years.

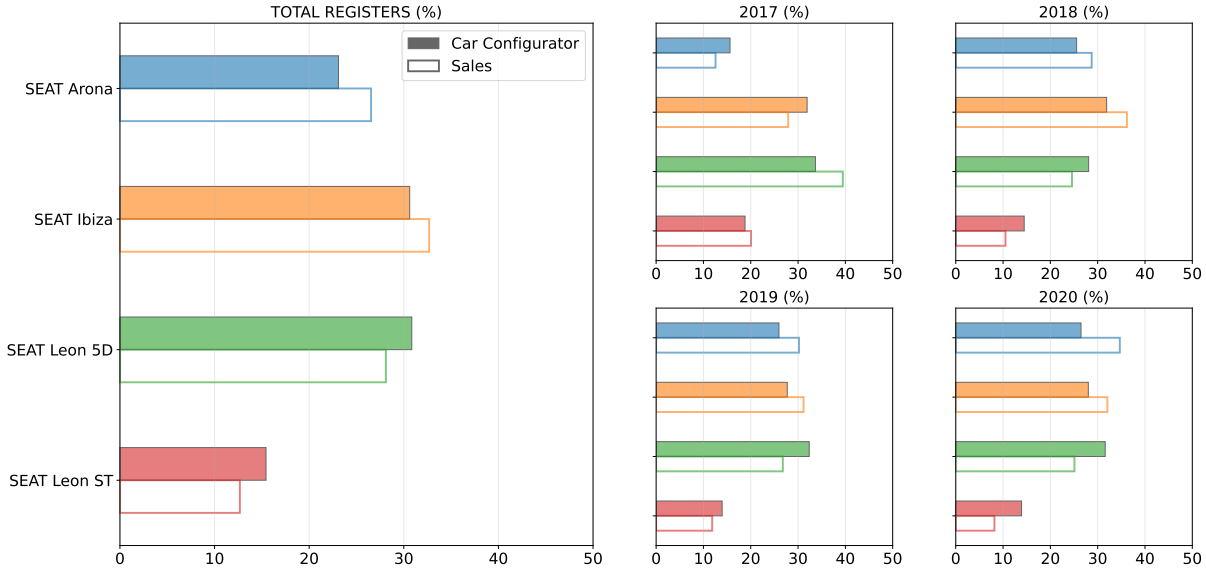


Figure 3.9: Comparison of TMA Level registers within the Car Configurator webpage and Sales record [%]

Regarding the common attributes, the analysis with respect to TRIM Level is visible in Figure 3.10. It might look that Xcellence and Cupra trim are residuals. However, the poor figures are due to they are exclusive equipment level of the Leon family. The contrary occurs for FR trim. Except for SEAT Arona, it is consistently the favorite choice by the users of the Car Configurator webpage. Nevertheless, this is not a mirror of the reality. Sales records do not reflect this behavior. People have a preference to configure the more expensive car variants, but they are not the ones they finally acquire. There is a mismatch between Style and FR trim levels. It is manifest why there is an opportunity to data mining the clickstream information and isolate the users with real purchase intention.

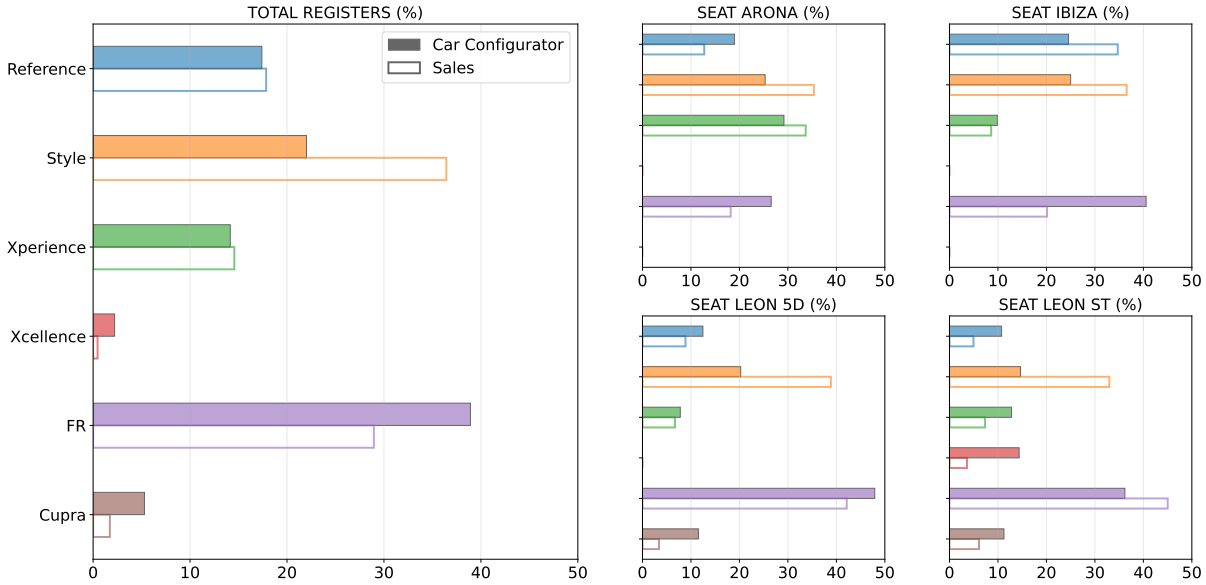


Figure 3.10: Comparison of TRIM Level registers within the Car Configurator webpage and Sales record [%]

The second attribute to discuss is the Exterior Color of the vehicle, shown in Figure 3.11. On this occasion, the information is solely exposed per car model, as there are colors that exclusively belong to one of them. SEAT Arona is the model with the largest number of combinations, as a consequence of the bicolor modality. The roof and car body can have different colors. On one side, car configurator data counts 47 different colors, whilst the sales record collects 48 unique colors. The merge of both datasets derives into 46 common colors, although all of them are shown in the image. Among them, the color named B4B4 is the favorite in the online world and the physical environment. However, this trend is not preserved for the rest of the instances. The second most popular online color, called 9550; is not the second in sales, being E10E. This behavior is repeated for the SEAT Ibiza. The first place is occupied by the same color, B4B4, in both categories, but this overlap does not continue for the next positions. On the contrary, the profile is more stable in the customers of the LEON family. The disproportion between the car configurator and sales record registers is not as much evident. For both car models of the family, color 2Y2Y is the front runner in the two categories. The same for the second and third variants. However, the most noteworthy imbalance is observed in color 9550. It is one of the most visited, but eventually, customers do not acquire it in the same proportion.

The analysis is executed in the engine of the vehicle, as it is exposed in Figure 3.12. The number of available engines in the sales record is equal to or superior to the ones offered in the clickstream data. Each car model has its unique engine selection. The only exception occurs with an engine called MX, which can power all the vehicles. Among all car models, there are engines in which the discrepancy between the sales and online visits is remarkable. Since the engine's choice is attached to the TRIM level, this performance was expected.

Finally, insights into the performance done at the geographical level are gained. Figure 3.13 assesses the behavior at the compound level, whilst Figure 3.14 does it at province granularity. LLAGOSTA and MADRID, the largest Spanish regions, attract the most visits to the webpage. Although the sales record has the core in these locations too, it does not mean that there is no activity in the rest of the country. The well balanced proportion suggests that users normally do not move to other regions seeking a good product, but they prefer to close the deal with the dealership in their area.

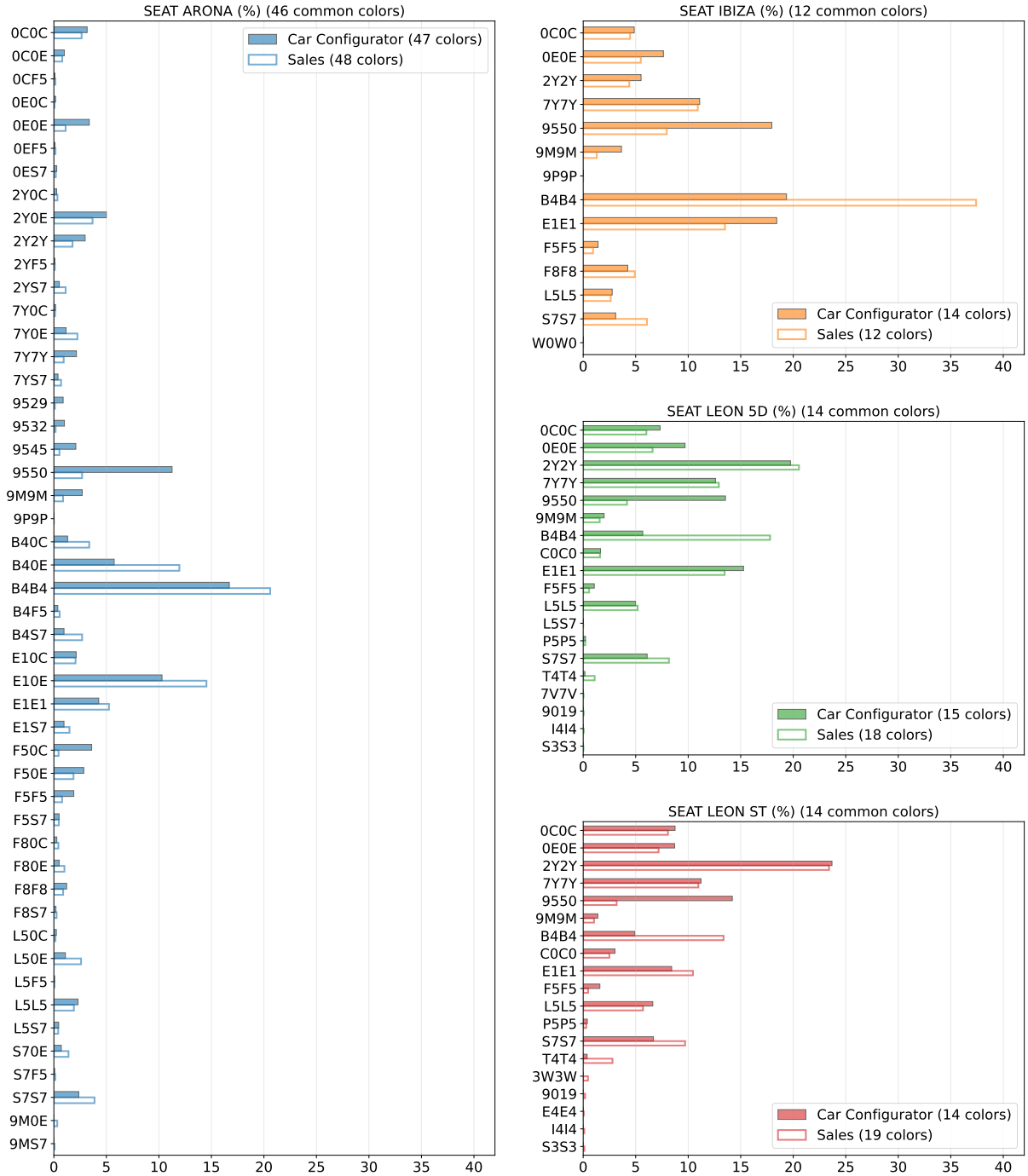


Figure 3.11: Comparison of Exterior Color registers within the Car Configurator webpage and Sales record [%]

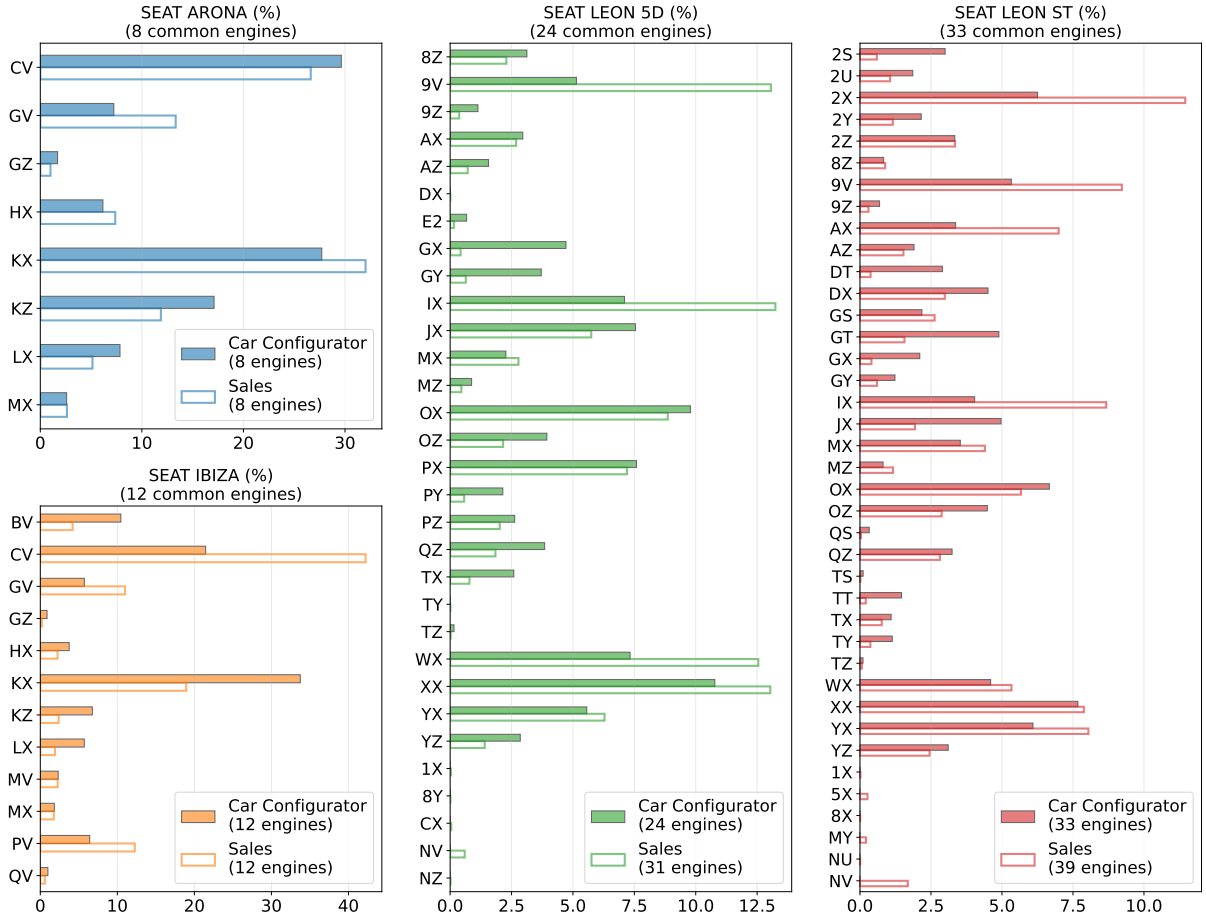


Figure 3.12: Comparison of Engine registers within the Car Configurator webpage and Sales record [%]

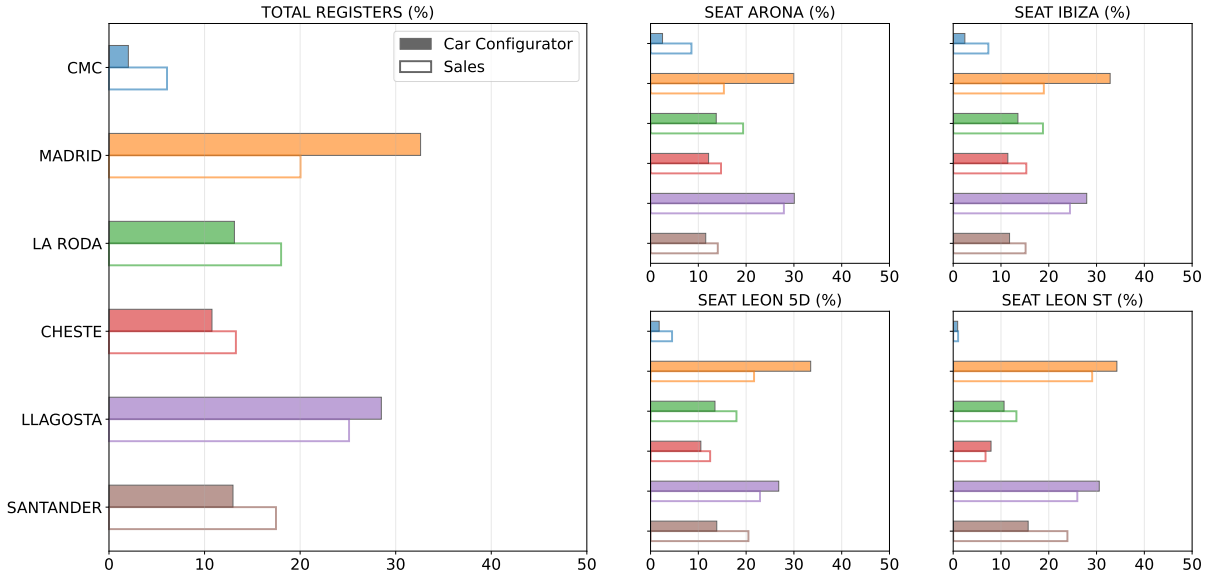


Figure 3.13: Comparison of Compound Location registers within the Car Configurator webpage and Sales record [%]

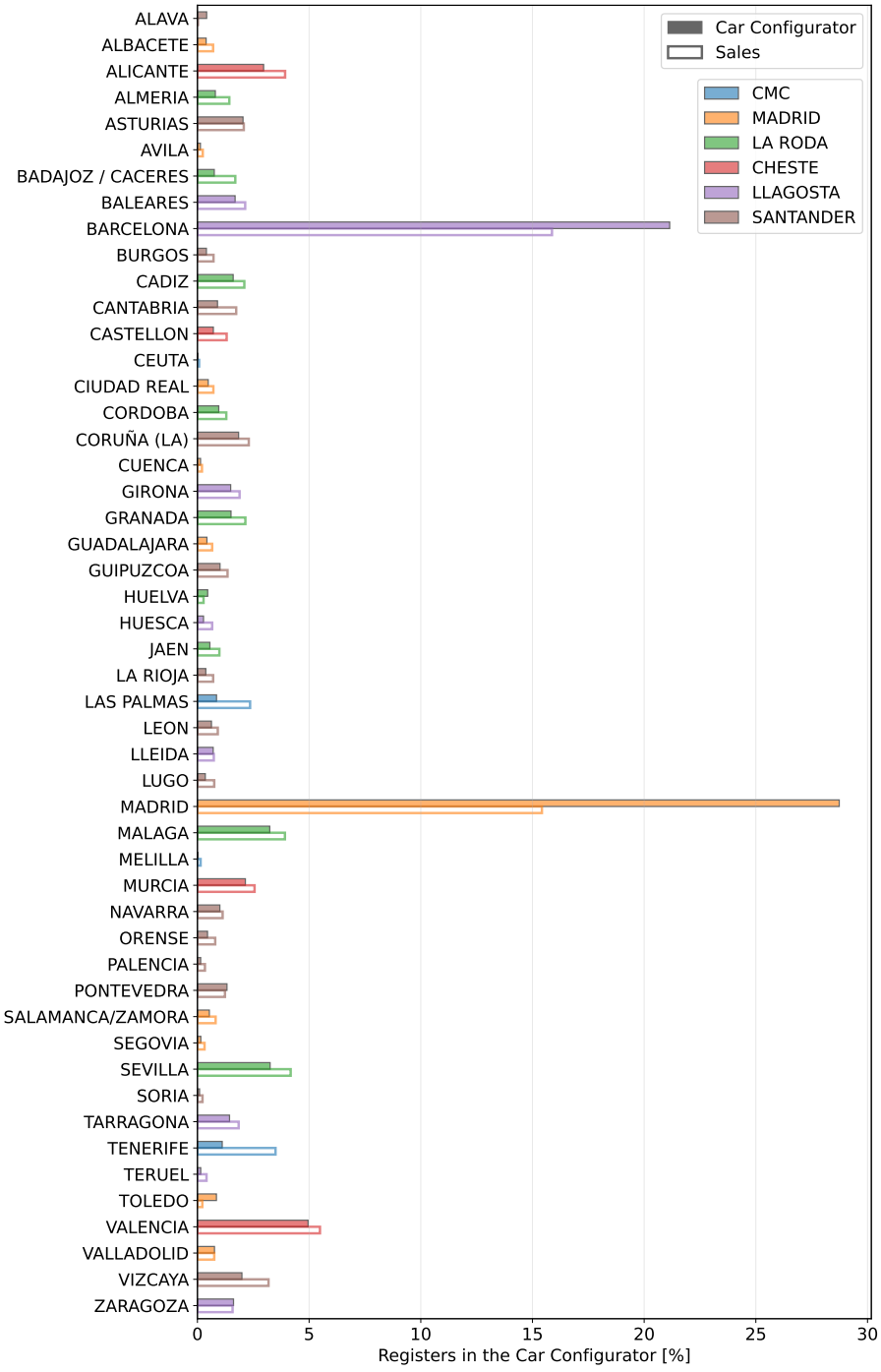


Figure 3.14: Registers of Spanish Provinces within the Car Configurator data and Sales record[%]

Chapter 4

The Proposed Solution

The solution presented in this research is structured following an ascending complexity scale. What this means is the difficulty level the company should face to execute the different options. Consequently, the first step consists of reallocating the already manufactured vehicles to a new destination to improve their purchase likelihood; whilst the proposal evolves up to updating the same attribute meanwhile the cars are in the manufacturing line. In this chapter, the diverse ways of procedure utilized to overcome the existing challenges in the construction of the solution are detailed. Essentially, the discoveries made in the thesis have been released in the form of journal papers or congress proceedings published, or pending to be published.

4.1 Compound Reallocation Of Manufactured Cars

Automotive manufacturers are witnessing a paradigm shift in the traditional vehicle purchase process and are taking decisive action. In recent years, automotive OEMs have been transitioning from the conventional dealership system to the agency model, a trend reported by both newspapers [98, 99, 100] and consulting companies [101, 102, 103]. In the former system, dealers purchased cars from manufacturers based on their requests, subsequently selling them to customers and profiting from the transactions. As a result, the mission of manufacturers can be said to revolve around satisfying the needs of dealerships rather than those of customers.

However, the emerging market trend implies that dealerships will serve as distribution points, diminishing their influence in the vehicle purchase process. Consequently, manufacturers must excel in aligning with customer demand as it directly impacts the financial health of the company. In this context, Machine Learning (ML) can prove to be valuable to automotive OEMs in optimizing vehicle allocation. Given that car brands continue to employ the Built-To-Stock manufacturing strategy, shipping vehicles to regions with the highest likelihood of purchase presents a significant competitive advantage. On one hand, being able to swiftly deliver requested vehicles to customers is an effective way to enhance customer satisfaction. On the other hand, each day a car spends in the compound awaiting a customer incurs logistic costs and may necessitate price discounts to free up space. For instance, note [104] found the correlation between inventory volume and sales in the American automobile market. It is a clear explanation about how dealership system works. The conclusions are that given how vehicles were allocated to dealerships, adding inventory actually lowered sales.

In consequence, it is proposed answering to the problem by conducting a binary classification procedure, assisted by algorithms of machine learning. The features of the problem will be the components defining a car variant, whilst the target is the delivery type a vehicle can be categorized. A car variant is defined as the combination of Car Model, Equipment Level (TRIM), Exterior Color, and Engine. Additionally, the Order Type is a parameter to consider, as it indicates whether the car is BTO or BTS. The first step consists of assigning classes to each car in the dataset according to different time thresholds. These binary classes are Fast Delivery and Normal Delivery, FD and ND hereinafter. Whether a car in the dataset spent fewer days than the threshold, it is tagged as FD. The threshold spans from 1 to 6 weeks waiting in the compound region. We want to explore the performance of the methodology in an extreme case, such as delivering cars in 1 week, although data does not follow this trend. On the other hand, the highest median in Table 3.5 is close to 6 weeks remaining in the compound region. We exclude from the range the numbers of weeks associated with percentile-75, owing to we consider that a car waiting more than 2 months cannot be catalogued as FD. Correspondingly, the weight of the FD class in the production and deliveries dataset varies with the threshold. Table 4.1 reflects the performance of the class per compound region, whilst Table 4.2 does it per car model. LLAGOSTA and CHESTE compounds consistently are the places with the largest rate of cars cataloged as Fast Delivery, whilst CMC is the lowest. Regarding car models, SEAT Arona has the best values, in opposition to SEAT Ibiza, with the least number of Fast Delivery cars.

Table 4.1: Fast Delivery (FD) class percentage per compound region and threshold time, or label, over the total number of vehicles in each compound region

	7 days	14 days	21 days	28 days	35 days	42 days
CMC [%]	1.13	7.47	18.27	29.6	38.93	47.27
MADRID [%]	5.54	25.68	40.94	49.79	56.45	61.32
LA RODA [%]	5.64	21.47	36.12	45.13	51.52	57.12
CHESTE [%]	5.8	26.87	42.64	52.09	58.17	62.94
LLAGOSTA [%]	5.74	26.97	44.29	54.67	61.88	66.95
SANTANDER [%]	3.52	15.83	35.1	48.13	55.92	61.8

Table 4.2: Fast Delivery (FD) class percentage per car model and threshold time, or label, over the total number of vehicles of each car model

	7 days	14 days	21 days	28 days	35 days	42 days
SEAT Arona [%]	6.21	27.31	45.46	56.07	63.09	68.09
SEAT Ibiza [%]	4.27	19.75	33.68	43.55	50.66	56.65
SEAT Leon 5D [%]	4.79	21.29	37.2	47.38	54.76	60.11
SEAT Leon ST [%]	4.95	23.81	41.21	51.53	58.09	63.34

The subsequent stage in the roadmap involves the development of a dependable classification model for each threshold day. This model will be based on the car variant and vehicle's order type. The literature review has identified a collection of Machine Learning algorithms that have already demonstrated efficacy in solving problems within an industrial context [105, 106, 107, 108, 109, 110, 111, 112, 113]. On one hand, there are algorithms rooted in heuristic trees, while on the other hand, there are boosting-based algorithms. Decision Tree (see Section 5.1 for more details) and Random Forest (5.2) appoint the first group, whilst XGBoost (5.3) and CatBoost (5.4) are encountered in

the second family. The listed algorithms will be tuned to obtain the maximum results. For that purpose, both concepts of hyper-parameters tunning and Cross-Validation will be employed (5.5). *BayesSearchCV* unifies these two methods by iteratively exploring the hyper-parameter space based on previous evaluation results and adjusting its search strategy.

The next point is the selection of the most efficient algorithm from the aforementioned list. This task is solved by employing the most appropriate metric. In the case of binary classification problems, there are plenty of suitable options. One of them is accuracy, i.e., the ratio of true results to the total number of cases. It is a preferred choice for well-balanced problems. However, in this framework, it is not always applicable. If it is aimed to obtain certainty regarding the proportion of predicted positives that truly belong to the positive class, precision is the appropriate metric. Conversely, recall measures the proportion of actual positives that are correctly classified. In this specific context, it is desirable to optimize both precision and recall. It is wanted to ensure that cars are correctly classified as "Fast Delivery" (i.e., precision) while capturing as many Fast Delivery cars as possible (i.e., recall). The solution to this requirement lies in the F1 score, which is defined as the harmonic mean of precision and recall. The F1 score is particularly suitable for unbalanced datasets, addressing one of the shortcomings of accuracy. Although there are other metrics such as binary cross-entropy or the area under the ROC curve (AUC ROC), previous studies [114, 115] have demonstrated the effectiveness of the F1 score in binary classification problems within the industrial sector. Additionally, the winning estimator will pass an interpretability assessment. The goal is to comprehend the most helpful features to execute the classification task. Two techniques are used to fulfill this purpose. The first approach is computed according to how frequently and to what extent each feature was used to make decisions. These magnitudes are extracted thanks to the property *feature_importances_* of the algorithm's library. On the other side, it is possible to gain the relevance of each feature per each single observation. The technique behind this concept is called SHAP (SHapley Additive exPlanations) values (see explanation in Section 5.6).

Afterward, the highest-performing classifier is employed to develop the reallocation strategy. The approach involves directing the cars towards compound regions classified as "Fast Delivery." To accomplish this, car variants (including Order Type) pertaining to each compound region are enumerated. Subsequently, if the output of the fitted algorithm corresponds to the "Fast Delivery" category, the car variant conserves its original destination. On the contrary, the destination changes to one of the remaining alternative compounds. In regions where the car variant is classified as "Fast Delivery," the car variant alters its path. However, if none of the alternative destinations are deemed valid options, the car variants maintain their original destination. The pseudo-code representation of this procedure is depicted in Algorithm 1.

Algorithm 1 Reallocation of cars to the most suitable compound region

```
1: procedure REALLOCATION STRATEGY
2:     Inputs: car variant, original compound
3:     compound = original compound
4:     if car variant & compound == Fast Delivery then
5:         compound region = compound
6:     else
7:         for compound in alternative compound do
8:             if car variant & compound == Fast Delivery then
9:                 compound region = compound
10:            else
11:                compound region = original compound
12:    return compound region
```

Finally, the outcomes of the reallocation process are assessed. Specifically, the time distribution per compound region is examined before and after the reallocation process under two approaches: (a) vehicles preserve the same number of days from the original compound; (b) the time in compound is estimated from the existing distribution of days of the vehicles with the same characteristics in the new location. It is assumed that the reality would be between these two scenarios. However, it is important to note that the capacity of the compound regions was not considered during the destination changes. Additionally, no criteria were included to determine the selection between two or more alternative compounds. Nevertheless, the region with the most significant disparity between the compound and demand is recommended as the discrimination criteria.

4.2 Car Configurator Webpage As A Reliable Source

The reallocation strategy is a good starting point to optimize the logistics of the car brand. However, the interest of the potential customers is absent. The hypothesis under study is that this information can be extracted from the data gathered from the SEAT Car Configurator (CC) webpage. The procedure to validate this concept is based on correlation and forecasting.

The customer starts the purchasing path with an exploratory phase, followed by the decision stage, and ends with the acquisition moment within the dealership. There exists a lag between the last and first steps. The objective consists of gaining knowledge about this delay by means of the correlation between the sales and the clickstream data. The moment the correlation is maximum defines the dimension of the period. For that purpose, both information sources are transformed into time series and shifting the sales data over the webpage information. The duration of the shifting lasts for 52 points, i.e., the number of weeks within a full year. On the first trial, the process is implemented at the total weekly sales and users of the CC webpage per car model. Afterwards, the outputs are validated in second granular levels. These ones are defined by joining the color or the compound destination to the car model. The choice of these granularity levels is driven by logistical considerations. These elements are interchangeable without difficulties and independent of spare parts, in opposition to alloy wheels, for instance. We expect to observe the same behavior in CC users but reinforced. Hence, this learning is employed to divide data into time chunks, which will be helpful in the next step.

Within each time chunk, the last month and a half defines the test period. It will serve to predict the sales volume of each car variant, also called the second granular level. Additionally, this division is intended to face all the stages of the product life cycle: introduction, growth, maturity, and decline. Hence, the construction of forecast weekly mix sales will be possible. They are defined as the percentage of sales each car variant has over the weekly sales volume. These mixes are derived from a set of ML algorithms. They are trained with the rest of the data of the corresponding time chunk. The algorithms listed are ARIMA(X) (5.7) and XGBoost (5.3). However, we distinguish between two modalities: univariate and multivariate. The first one only considers past sales data. The latter ones include additionally the information from the automotive brand's webpage. We use these techniques to perform the sales prediction of each car variant.

Inside each time chunk and the car variants belonging to it, some rules were fixed. Firstly, only those colors and compounds with any information from both sales and clickstream data during the test period of each chunk were predicted. Hence, for those algorithms where it is possible to estimate in advance the most precise parameters, such as ARIMA, autocorrelation function and partial autocorrelation function were employed to obtain the moving average (q) and autoregressive parameter (p), respectively. Stationarity, or order of integration (d), of the time series is analyzed by means of the augmented Dickey-Fuller test. In case this procedure is unsuccessful, parameters (p,q) are estimated as first order, by default. For the multivariate version, i.e., ARIMAX, the same tools were applied to perform the forecasting of the exogenous variable. For the case of algorithms of boosting nature, there were no shortcuts, and all parameter combinations within the range of the training set were evaluated. In the case of the univariate version, the algorithm explored up to 3 months of previous sales records to perform the forecast. For the multivariate algorithm, the same methodology took over to predict the exogenous series. Afterwards, the predictions are guided by a rolling strategy. In other words, each point within the test of the time chunk will be predicted individually and using all the original preceding data. It is a scenario to gain accuracy with respect to longer horizons. We select parameter combinations with the lowest mean average error (MAE). The different techniques are assessed under this metric, all car variants and time chunks. It has been decided to employ MAE as an evaluation metric because outliers might be found in the sales record of each variant and this metric is very resistant to these events.

Once the previous step is completed, these outcomes are assessed with respect to the real weekly mix sales. Hence, the results obtained from univariate techniques will be compared to multivariate ones. Traditional metrics such as MAE and root mean squared error (RMSE) were discharged because they are scale-dependent. They are useless to compare different time chunks and car models. One solution arrives in the form of mean average percentage error (MAPE). However, this metric is not able to deal with zero values in any of the series. That is why we propose to compute the correlation between the actual weekly mix sales and the forecast mix. The correlation takes the form of R2 Score. We are inspired by work [116] as a valid framework to compare different forecasting algorithms in the automotive industry. Conclusions will arrive after following a sequential procedure. Firstly, the outputs are averaged over the total length of weeks and time chunks the dataset has. The second step consists of averaging but over each time chunk. Acting in this way, we gain more detail about the performance of each technique. Lastly, the assessment process finishes with the third step. At this level, we count what technique provides the best metric for each week of the test set within each chunk.

4.3 Quantitave Reduction Of Car Configurator Data

The users' quantitative activity is summarized in two metrics: (a) Number of car variants; and (b) Time between connections (TBC). The first item gathers how many car variants have been explored by any users of the online tool. In this scenario, the car variant is defined as the union of car model, trim level, engine, exterior color, and compound region linked to the location of the user. On the other side, the number of days between the first and last connection to the car configurator webpage per each user is stored in TBC. The intention is to track the purchase interest, concentrated on epochs. This behavior can be captured by this metric, instead of any other temporal feature, such as the number of days the user has entered the webpage. The summary of the users's activity is shown in Table 4.3. All users have configured, at least, one car variant, but the average is nearly two car variants. It is remarkable the maximum number of car variants done by a single user is nearly 5000 different. However, this value is affected by the null ID. It is a unique alphanumeric code for all those users who are impossible to track for any reason. From the point of view of time between connections, 75% of the users have accessed only one day to the online tool, but the average is nearly 21 days from the first to the last connection. The statistics prove that they are distributions with very long tail or outliers. For more details see Figure 4.1, where it is plotted without the influence of the null ID.

Table 4.3: Main descriptive statistics of users' activity on the Car Configurator automotive OEM webpage.

	Min	Mean	Std. Dev.	Q1	Q2	Q3	Max
No. of Car variants	1	1.67	4.06	1	1	2	4924
Time between connections [days]	0	20.87	77.84	0	0	0	1031

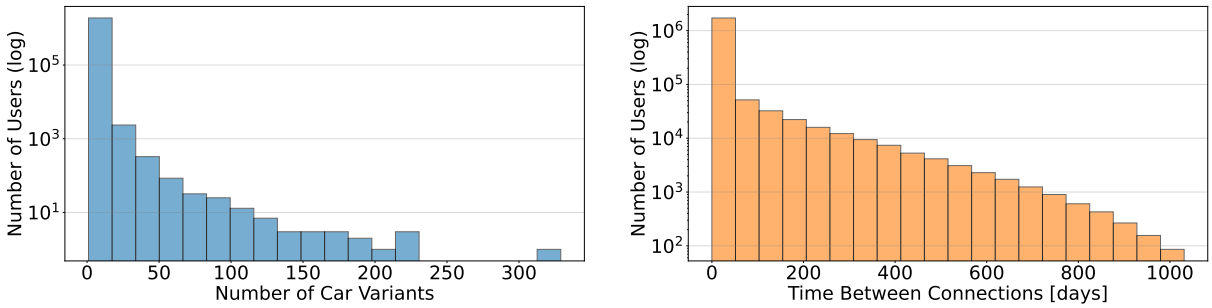


Figure 4.1: Distributions in log-scale of the quantitative activity of the users of the Car Configurator webpage per Number of Car Variants configured (left) and days between first and last connection (right).

The objective is to decrease clickstream data volume without compromising significance, defined by the correlation between Car Configurator webpage visits and company sales. The way of procedure is inspired by the work of [117]. They are capable of matching clickstream and offline purchasing data of a webrooming enterprise that sells doors to other industries. Hence, they introduce a dynamic decision support model that augments the classic inventory planning model. Even so, it is infeasible to perform the same association they did, that's why we follow another strategy in the form of correlation.

However, for the reasons aforementioned, the correlation is not straightforward. Firstly, customers research online the product and, after a while, they head to the dealer shop in order to purchase it. There is a delay between these two moments to give consideration to it. We have established this delay at 8 weeks. This election is not trivial. It enters in the epoch of the largest correlation discovered in the reliability experiment of Car Configurator data. Additionally, the reasons are supported by the manufacturing flow followed by the car manufacturer. Section 3.1 already relates the SEAT production cycle. Although the assembly stage lasts one week, previously it is required a preparation phase where the car's attributes are being defined. This period can span up to 6 weeks. It includes preparation of cars' wire-harness system, and defining sequence in the production line. Therefore, we overlook and expand the range for one week.

Acknowledging this delay period, the computation strategy is as follows. Rather than lagging the full sales record over the entire CC time series and performing the computation, we have proceeded monthly. Figure 4.2 illustrates the method. For each month of CC data within the time range, sales records from 8 weeks in advance, with respect to the first week, are selected to compute the R2 score. The sales range extends as much as the month's number of weeks.

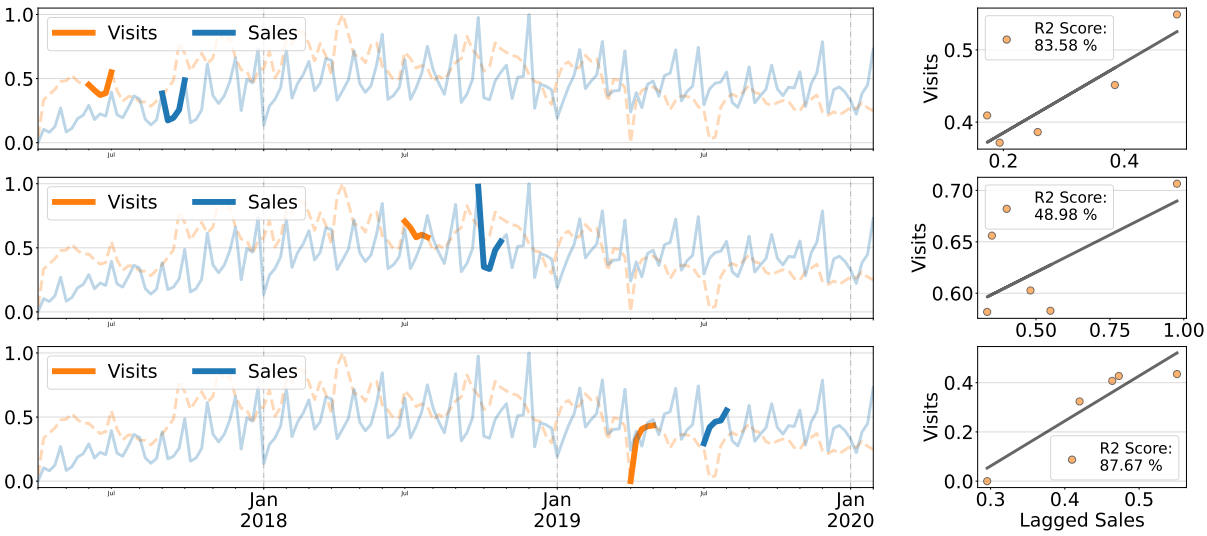


Figure 4.2: Example of R2 Score monthly lagged computation strategy. Soft lines are full weekly time series for Car Configurator data (orange dashed line) and Sales record (blue line). The larger width indicates the period range to compute R2 Score between the inputs. There exists 8 8-week delay between the beginning of both periods.

We propose to apply diverse filters through the entire clickstream data. These filters are based on users' quantity tracking rather than qualitative performance. In other words, how much they have used the tool instead of what they have selected. It has been shown in Table 4.3 that minimum values are not the pain points. For the outliers detection task, we have established two limits. The first is more restricted and it uses the Q3 value of the distribution. Meanwhile, the other one is the popular technique of inter-quartile range (IQR). For more details, see for instance book [118].

The filtering sequence is as follows. The first one consists of eliminating users that have accessed the automotive brand's webpage to configure exclusively one single variant. On the whole, customers with purchase intention normally compare products, but especially prices as it is pointed out in reference [119]. As we do not have this last magnitude, we focus on users with *Number of car variants* larger than one. We call it **rule1**. Secondly, we pay attention to the attribute *TBC*. The purge is motivated by the users' acquisition

"urgency". In other words, if the number of days exceeds by far the normal behavior of the population, it means that purchasing a vehicle is not a high-priority issue for him. The first bifurcation along the filtering process is faced. The Q3-based limit is called **rule2A** while the threshold established by *IQR* is named **rule2B**. Finally, it is necessary to return to the *Number of car variants* each user has consulted. The normal flow a person follows in these online tools is such as: (1) the user selects all attributes he wishes; (2) he realizes the outcome is not affordable, (3) he modifies the product until finding a commitment between price and wished characteristics. Therefore, if the number of variants analyzed during this procedure surpasses the typical practices, it is concluded he is doing window shopping. Once again, the outliers detection thresholds aforementioned are repeated. Consequently, they are applied into the outcome data from rule2A, which derives into **rule3A1** and **rule3A2**; and from rule 2B, whose sons are **rule3B1** and **rule3B2**. A summary of all filtering criteria is placed in Figure 4.3. From the latter, only the most restricted rules are chosen to build weekly CC filtered clickstream time series.

Lastly, it is assessed how the filtering procedure has affected the significance of the dataset. In case it is preserved, it has been found a procedure to reduce the impact of the main concerns of Big Data systems. The manner to measure this indicator is by means of statistical analysis. Significance outcomes from raw clickstream data are used as the reference frame. Therefore, the equivalency of filtered car configurator data outputs with respect to the benchmark is under analysis thanks to statistical Kolmogorov-Smirnov test (see Section 5.9). To indicate how much equivalent both distributions are, the flag is the resulting p-value. The larger the p-value obtained, the larger the certainty of the similarity. More details can be found in the book [118].

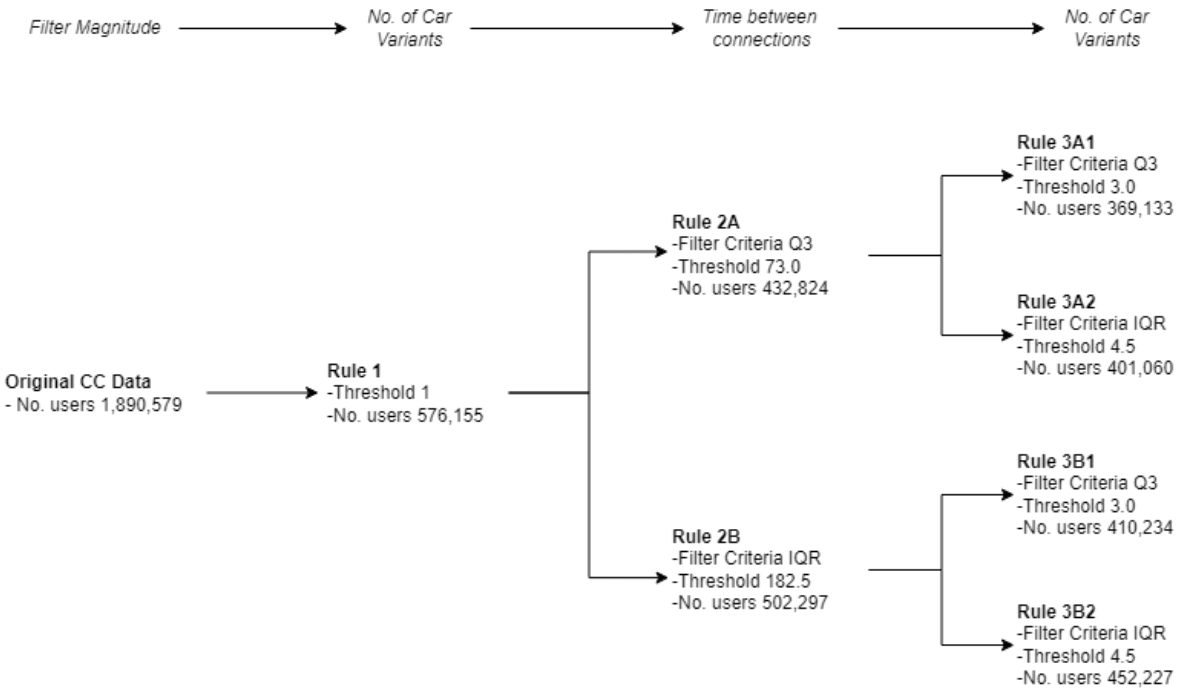


Figure 4.3: Car Configurator clickstream data filtering process.

4.4 Qualitative Filtering Of Car Configurator Data

The Car Configurator (CC) website has proved its validity as a reliable source. Additionally, it is found a path to overcome the obstacles that represent a Big Data environment. On the contrary, it persists the challenge of discriminating the users of the online platform: (a) people with real purchase intention; and (b) users doing window-shopping. To accomplish this effort, genetic algorithms will explore throughout the data gathered by the Car Configurator webpage. Similarly, as it was executed in the previous Section 4.3, it will seek to identify the characteristics that maximize the correlation between clickstream data and car sales.

Customer profiling is performed at two levels. In the first one, the entire filtered clickstream data is correlated with the totality of the sales record. In order to achieve the best results, a genetic algorithm (go to Section 5.8 for learn all the details) takes place under different boundaries. These limits are defined by the set of number of rules, population size, and number of generations. These three parameters establish the search space. Afterward, the genetic algorithm executes five independent trials under identical initial conditions. The goal is to ensure the reliability of our findings. Consequently, outcomes are averaged and compared among them and with the benchmark value. The latter is achieved by applying the fitness function to the plain clickstream data. In the end, the best set of genetic algorithm parameters is chosen. It follows to select the best solution candidate and evaluate it individually. The frequency of the elements chosen by each rule is analyzed and related to the position they place in the clickstream data. Additionally, the fitness of each single rule that composes the best candidate is computed. Therefore, it is understood the weight each individual rule has and it is possible to find which are the Pareto optimum. In other words, what are the subset of filtering rules within the candidate that represent around 80% of the correlation power. Afterwards, associate them with the sales frequency rate.

Thereupon, the second phase of the study begins. It is carried on at the compound region level. The sales record is gathered according to the compound region they belong. For each one of them, the genetic algorithm is executed to optimize the correlation with the clickstream data. The chosen inputs are the winning ones from the previous experiment. Later on, the results are compared with baseline values. Subsequently, a detailed analysis of the filtering rules is performed. This analysis includes not only a count of the number of provinces that are included in the best candidate for each compound but also how they are segmented. Specifically, it is noted which locations belonging to the compound are considered, which ones were discarded, and which ones were added from outside the compound's geographical domain. Finally, a fitness comparison is conducted between the first and last groups. It determines the weight each segment carries in the final solution.

From the point of view of the chromosome, it is structured with the following attributes: day of the week, car model, trim level, engine, exterior color, and geographical location. The dimension of the choromosome is as much big as it is established by the number of rules parameter (see Table 5.1 to learn about the composition of the chromosome). Therefore, the population gathers a set of feasible solutions ranked according to their quality, i.e., the fitness. In order to compute it, it is followed the same strategy illustrated in the previous Figure 4.2 to obtain the significance. The correlation imposing an 8-week lag between the sales and the clickstream data is preserved. Nevertheless, the fitness averages these outcomes to reduce them into one single number, rather than delivering the array. The larger, the better. It is a maximization problem and the theoretical limit value is 100%, i.e., perfect correlation every month. The fitness is computed

between the sales record and the clickstream data filtered according to the set of rules of the chromosome. In case there are no users that fulfill the filtering criteria, the fitness is penalized according to the maximization problem requirements.

Additionally, it has been incorporated another stopping criterion based on antistagnation of fitness. In case the fitness does not vary larger than tolerance for a number of consecutive generations, the mechanism is triggered. This routine can exclusively be repeated for five attempts. After that, the algorithm is interrupted. In this framework, the permitted tolerance is 0.001 and the number of consecutive generations, to activate or deactivate it, is 5% of the population size. It consists of increasing the mutation probability for the next generations until the mutation probability returns to its original value.

4.5 Genetic Algorithm Improves Demand Forecasting

The skeleton of the genetic algorithm utilized in Section 4.4 (and described in Section 5.8) is conserved. The chromosome structure together with the selection, the mutation, the crossover and the elitism mechanisms are the same. However, the fitness function has been adapted to fit the new needs of the problem. Rather than maximizing the correlation between the clickstream data and the sales record, the goal is to decrease the prediction error of the car variant demand. From hereinafter, they are referred to as Variant of Interest (VOI) prediction, as they are computed using exclusively the clickstream data of the car variant under study. On the other side, the predictions derived from the genetic algorithm have been named genetic or chromo, due to they come from the filtering rules found in the chromosome. The chosen metric to rule the fitness function is the mean average error (MAE). It is the magnitude employed to quantify the performance in the initial demand prediction, the ones that confirmed the reliability of Car Configurator data.

Nevertheless, the focus will be on finding the set of optimum parameters that deliver the best figure of merit. In other words, the number of cases of MAE reduction with respect to the VOI benchmark from all the car variants and time chunks analyzed. In case there is more than one candidate, to make a decision on which is the winning subject, it would be explored which one of the aspirants provides more forecasting error reduction. This metric is defined as the average of the difference between the MAE from the VOI forecast and the MAE of the genetic forecast for each time chunk and car variant. This procedure will take place only on the initial demand prediction which is considered to have the largest margin of improvement.

The parameters encompass a range of 50 to 150 rules per chromosome, 20 to 300 chromosomes within the population, and 20 to 200 new generations. It is not attended to scout a search space radically different from the one surveyed in the previous experience with genetic algorithm. The conditions of mutation, crossover, tournament, and elitism probabilities are unaltered.

Finally, the last perspective of the final candidate comes from the assessment of the weekly sales mix, real vs forecast ones, based on R2 Score. This scenario accommodates: (a) averaging the outcomes over the total length of weeks and time chunks; (b) and averaging over each time chunk. In this last attempt, the outcomes derived from the genetic forecast are compared with the ones obtained in the VOI prediction.

4.6 Production Modification Based On Improved Forecasting

Using the innovative approach of applying evolutionary computation to the data gathered by the Car Configurator webpage has two positive consequences. On one side, the reliability of the information is treated from a qualitative point of view. On the other side, more accurate forecasts of customer demand are extracted. These predictions will be used to update the current cars in the production line in terms of non-dependant electrical components.

Following the explanation given in Section 3.1, the update will exclusively impact the Build-to-Stock (BTS) vehicles, and during the time interval occurring between FU and A500. It will be effective for those cars that are one week before A500, named Point of Modification (PM). These modifications search to minimize the gap between the estimated stock composition and the future demand distribution. In other words, the weight each car variant has over the total amount of vehicles. A minimum gap means that the compound region accommodates vehicles in the proportion that customers call. The mathematical expression to be fulfilled is Equation 4.1.

$$F\left(\widetilde{p}_{PM_i}(t)\right)_m : \min\left(\sum_{i=1}^{N}\left(\frac{\widehat{s}_i(t+3)}{\widehat{S}(t+3)} - \frac{\widehat{d}_i(t+3)}{\widehat{D}(t+3)}\right)^2\right)_m \tag{4.1}$$

where $\widetilde{p}_{PM_i}(t)$ makes reference to the production volume of car variant i from car model m that can be updated in week t. N represents the total number of car variants from a given car model. This distinction per car model is caused by it is not possible to transform, at this stage of the manufacturing process, one car model into another type. Capital letters symbolize the added total volume of the car variants, and widehat represents that it is an estimation. Therefore, $\widehat{S}(t+3)$ and $\widehat{D}(t+3)$ mean the total estimated stock and demand volume, respectively, of the third week in advance. Finally, $\widehat{s}(t+3)$ and $\widehat{d}(t+3)$ allude to the volume of car variant i in the estimated stock and demand for the same epoch. The decomposition of the terms of Equation 4.1 is found in the next Equation 4.2, and its equivalent Equation 4.3.

$$\begin{cases} \widehat{s}_i(t+1) = s_i(t_0) + p_i(t_0) - \widehat{d}_i(t_0) \\ \widehat{s}_i(t+2) = \widehat{s}_i(t+1) + p_i(t-1) - \widehat{d}_i(t+1) \\ \widehat{s}_i(t+3) = \widehat{s}_i(t+2) + p_i(t-2) - \widehat{d}_i(t+2) \end{cases} \tag{4.2}$$

$$\begin{cases} \widehat{s}_i(t+1) = s_i(t_0) + p_{ZP8_i} - \widehat{d}_i(t_0) \\ \widehat{s}_i(t+2) = \widehat{s}_i(t+1) + p_{A500_i} - \widehat{d}_i(t+1) \\ \widehat{s}_i(t+3) = \widehat{s}_i(t+2) + \widetilde{p}_{PM_i} - \widehat{d}_i(t+2) \end{cases} \tag{4.3}$$

The previous formulae represent a mass flow equation. In other words, the estimated stock level of a given car variant i for the following week ($\widehat{s}_i(t+j+1)$) is ruled by the current status of the stock $\widehat{s}_i(t+j)$; plus the volume that will enter to the stock ($p_i(t-j)$); minus the outputs, i.e. the demand prediction ($\widehat{d}_i(t+j)$), where $j_\epsilon\,[0,2]$ represent the weeks. The stock of the current week ($s_i(t_0)$) is limited to the number of cars whose production finishes at the week at hand, or before, but they will leave the stock in the actual week or later. Special mention deserves the incoming cars, i.e., the production. Only those cars who are currently in ZP8 are headed to the compound region ($p_i(t_0) == p_{ZP8_i}$). That's why this milestone serves as a reference point and the weeks until arrive to it are counted.

For this reason, the vehicles that are still one week away from reaching are currently in A500 ($p_i(t-1) == p_{A500_i}$). The same logic applies to the cars that are in the Point of Modification, they are two weeks away from ZP8 ($p_i(t-2) == \widetilde{p}_{PM_i}$).

The last step to solve Equation 4.1 is to concrete the circumstances of the problem. In other words, which are the constraints and boundaries. Firstly, the sum of the current volume in PM (P_{PM}) is respected. It is not possible to add or remove cars within the manufacturing. This case is ruled by Equation 4.4. The other limitation claims that all outputs of the optimization problem should be positive. Equation 4.5 relates that the updated values of the production should lay between zero and the total volume in the milestone.

$$\sum_{i=1}^{N} \widetilde{p}_{PM_i} - P_{PM} = 0 \tag{4.4}$$

$$0 \leq \widetilde{p}_{PM_1}, \widetilde{p}_{PM_2}, \cdots, \widetilde{p}_{PM_N} < P_{PM} \tag{4.5}$$

The optimization is only possible in those periods in which information about the expected demand is available. Consequently, it will be computed in the test periods of the time chunks. However, it is introduced the new concept of modification dates. In other words, the subset of four weeks on which the simulated compound region update can take place. It begins with the current week ($s_i(t_0)$) and continues with the three next weeks in advanced, according to the optimization requirements. This process is exemplified in Table 4.4. It is needed to clarify that the optimization process will run isolated on each modification date. This means that the production, stock, etc. of the second set of modification dates is not affected by the new conditions computed in the first modification dates list.

Table 4.4: Exemplification of the modification dates to perform the optimization procedure within a given test period

	Test Period					
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
Modification Dates 1	t_0	t_1	t_2	t_3		
Modification Dates 2		t_0	t_1	t_2	t_3	
Modification Dates 3			t_0	t_1	t_2	t_3

Chapter 5

Methods And Techniques

In this chapter, we delve into a diverse array of methodologies and tools meticulously selected to address the research objectives comprehensively. Machine learning algorithms used for classification tasks and demand forecasting range from the foundational Decision Tree to the classical time series forecasting methods like ARIMA, passing through bagging-based techniques such as Random Forest, or state-of-the-art boosted-based algorithms such as XGBoost or CatBoost. Furthermore, Bayesian Optimization and Genetic Algorithm, and SHAP (SHapley Additive exPlanations) are two blocks introduced as optimization and explicability techniques, respectively. They enrich the analytical toolkit with advanced capabilities. Bayesian Optimization facilitates the exploration of complex optimization spaces, guiding the search for optimal solutions efficiently. Genetic Algorithms facilitate the discovery of optimal solutions in complex optimization scenarios. SHAP, on the other hand, offers insights into model predictions by attributing them to individual features, enhancing interpretability and trustworthiness. Statistical tests like the Kolmogorov-Smirnov test complement these techniques, enabling rigorous hypothesis testing and distributional diagnostics. Each method is meticulously chosen for its relevance, applicability, and effectiveness, collectively forming a robust framework for analyzing and solving the multifaceted challenges addressed in this thesis.

5.1 Decision Trees

Decision Trees [120, 121] represent a fundamental non-parametric supervised learning method extensively applied in both classification and regression tasks. This section focuses on the mathematical formulation and key considerations, with a specific emphasis on the Classification and Regression Trees (CART). Notable alternatives to CART include ID3 and C4.5 algorithms. This method is utilized in Section 4.1.

The foundation of Decision Trees lies in the partitioning of the feature space through simple decision rules derived from the training data. For given training vectors $x_i \in R^n$, where $i = 1, \ldots, l$, and a label vector $y \in R^l$, a decision tree recursively divides the data into subsets at each node. This division is performed to group samples with identical labels or similar target values together. At a specific node m, the data is represented by Q_m with n_m samples. The process of partitioning is directed by a candidate split $\theta = (j, t_m)$, comprising a feature j and a threshold t_m. This split results in two subsets: $Q_m^{left}(\theta)$ and $Q_m^{right}(\theta)$. The evaluation of the quality of a candidate split at node m involves the use of an impurity function or loss function $H()$. The objective is to select parameters $\theta^* = argmin_\theta \; G(Q_m, \theta)$, where $G(Q_m, \theta)$ represents the impurity computation. The

choice of the impurity function depends on the nature of the task, whether it involves classification or regression. The process continues recursively for subsets $Q_m^{left}(\theta^*)$ and $Q_m^{right}(\theta^*)$ until a stopping criterion, such as reaching the maximum allowable depth or a minimum number of samples, $n_m \leq min_{samples}$, is met.

$$G(Q_m, \theta) = \frac{n_m^{left}}{n_m} H\left(Q_m^{left}(\theta)\right) + \frac{n_m^{right}}{n_m} H\left(Q_m^{right}(\theta)\right)$$

For classification outcomes with values $0, 1, \ldots, K-1$, for node m, the impurity measures commonly used are Gini and Log Loss (or Entropy). The Gini index evaluates how often a randomly chosen element in a set would be incorrectly labeled; while Log Loss quantifies uncertainty in classification predictions, indicating how close the prediction probability is to the corresponding actual value.

Gini impurity is

$$H(Q_m) = \sum_k p_{mk}(1 - p_{mk})$$

while Log Loss (Entropy) is set as:

$$H(Q_m) = -\sum_k p_{mk} log(p_{mk})$$

where p_{mk} is the proportion of class k observations in node m.

$$p_{mk} = \frac{1}{n_m} \sum_{y \in Q_m} I(y = k)$$

When predicting continuous values, Decision Trees rely on criteria such as Mean Squared Error (MSE), Half Poisson deviance, and Mean Absolute Error (MAE). These criteria serve as guiding metrics for the decision tree, aiding in the minimization of error or deviation between predicted values and actual values. Additionally, they assist in determining optimal locations for future splits in the decision tree. The Half Poisson deviance criterion is particularly advantageous when the target involves counts or frequencies (count per some unit). It is essential to note that utilizing this criterion requires the condition $y \geq 0$. It is worth highlighting that both Poisson deviance and MAE exhibit slower fitting compared to the MSE criterion.

Mean Squared Error:

$$\overline{y}_m = \frac{1}{n_m} \sum_{y \in Q_m} y$$

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} (y - \overline{y}_m)^2$$

Poisson deviance:

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} \left(y\, log \frac{y}{\overline{y}_m} - y + \overline{y}_m \right)$$

Mean Absolute Error:

$$median(y)_m = \underset{y \in Q_m}{median}(y)$$

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} |y - median(y)_m|$$

To calculate the feature importance fi_j, it is counted the number of node s splits provoked by each feature j and then divide it by the feature importance of all the nodes:

$$ni_m = \frac{n_m}{N}\left[G()_m - \left(\frac{n_m^{left}}{n_m}G()_m^{left}\right) - \left(\frac{n_m^{right}}{n_m}G()_m^{right}\right)\right]$$

$$fi_j = \frac{\sum_{s:node\ s\ splits\ on\ feature\ j} ni_s}{\sum_{k\ \in\ all\ nodes} ni_k}$$

where ni_m is the node importance, N is the total number of samples in the data, n_m number of samples in node m, $G()_m$ is the impurity computation in the node m, and superindexes $left$ and $right$ make reference to descendant nodes from node m.

Decision Trees present numerous benefits, making them a popular choice in various applications. They possess inherent interpretability, demanding minimal data preparation. Their versatility extends to handling both numerical and categorical data, including scenarios with multi-output requirements. Decision trees generate white box models, facilitating straightforward interpretation and validation through statistical tests. Remarkably, despite their simplicity, decision trees demonstrate robust performance. However, certain limitations should be acknowledged. Notably, the current implementation of the algorithm in the widely-used Python library scikit-learn lacks support for categorical variables, recommending alternative options to overcome this restriction. Among the drawbacks, the risk of overfitting arises, potentially resulting in overly complex trees that struggle with generalization. Pruning, i.e., removing unnecessary branches or nodes to diminishing tree complexity; setting a minimum number of samples at a leaf node; or imposing a maximum tree depth are common strategies to mitigate overfitting. Moreover, the NP-completeness of learning an optimal decision tree necessitates reliance on heuristic algorithms like the greedy algorithm, making locally optimal decisions at each node. While these algorithms cannot ensure a globally optimal decision tree, the issue is mitigated by training multiple trees in an ensemble learner. In the same way, ensemble methods are useful to address the sensitivity of decision trees to small data variations. Finally, decision tree predictions are piecewise constant, limiting smoothness and continuity, impacting extrapolation performance. Additionally, they exhibit bias towards the dominant class, necessitating careful consideration of dataset balance.

5.2 Random Forest

Decision trees may experience elevated variance, rendering their outcomes sensitive to the particular training data employed. Mitigating this variance can be achieved by constructing multiple models using various samples from your training data. Ensemble learning, a broad meta-approach to machine learning, aims to enhance predictive performance by combining predictions from numerous models. While there's a myriad of possible ensembles, three predominant methods prevail: bagging, stacking, and boosting. This section will primarily focus on the first method, bagging. Bootstrap Aggregation, abbreviated as Bagging, constitutes an ensemble comprising decision tree models, although it can also be used to combine the predictions of other types of models. It hinges on the concept of a bootstrap sample, which refers to a dataset subset obtained with replacement. Replacement entails that a selected sample from the principal dataset is reinstated, enabling the possibility of its reselection and potential inclusion multiple times in the new sample. Consequently, this process allows for the presence of duplicate examples from the original dataset within the sample [122, 123].

Random forest [124, 125] represents an extension of decision tree bagging, applicable to both classification and regression challenges. This method is utilized in Section 4.1. Diverging from traditional bagging, random forest incorporates a distinctive element involving the selection of a subset of input features at each split point during tree construction. Ordinarily, constructing a decision tree entails evaluating the value for every input variable in the data to determine an optimal split point. However, random forest disrupts this process by diminishing the features to a random subset considered at each split point. This approach compels each decision tree within the ensemble to exhibit greater diversity. For classification tasks, the typical practice is to employ the square root of the total number of features at each split, while for regression problems, it is advisable to use one-third of the features. Nevertheless, identifying the optimal values for these parameters is a task that should be tuned for each problem.

Notably, the trees in the ensemble are unpruned, in opposition to standard decision tree models. This deliberate choice leads to a slight overfitting of the training dataset, fostering greater dissimilarity among individual trees and reducing the correlation in their predictions or prediction errors. The outcome is that predictions and, consequently, prediction errors made by each tree in the ensemble tend to be less correlated. When combined, this often results in improved overall performance. In regression problems, the ensemble's prediction is the average across all trees, while for classification tasks, the prediction corresponds to the majority vote for the class label among the ensemble's trees.

Within each bootstrap training set, approximately one-third of the instances are deliberately excluded, forming what is known as out-of-bag (OOB) samples. In the construction of a random forest, this process is iteratively repeated. As these OOB sets are not utilized in training the model, they serve as a valuable test for assessing the model's performance. The calculation involves the following steps: (1) identifying all decision trees that have not been trained using the OOB instance; (2) determining the majority vote among these models for the OOB instance; (3) comparing this majority vote with the true label of the OOB instance; and (4) compiling the OOB error for all instances in the OOB dataset. Over numerous iterations, the OOB error stabilizes and converges towards the cross-validation error. The OOB method offers the advantage of requiring less computational effort and enables ongoing testing of the model during its training, contributing to an enhancement in the number of variables used per step.

A key advantage of the random forest algorithm lies in its versatility, making it applicable to both regression and classification problems. Its adaptability extends to handling large datasets characterized by high dimensionality, encompassing both numeric and categorical data, as well as accommodating outliers and missing values. Notably, feature scaling is unnecessary, as the algorithm employs a rule-based approach rather than relying on distance calculations. The algorithm can model complex, non-linear relationships between features and the target variable. It effectively mitigates the overfitting issue commonly associated with decision trees and enhances overall accuracy. Moreover, random forest minimizes prediction variance compared to a single decision tree. An insightful feature of this algorithm is its ability to supply information about the importance of each feature, simply averaging the importance derived from each tree in the ensemble, which it is valuable for uncovering underlying patterns. The absence of interdependence between trees facilitates parallelization, accelerating the training time. However, the algorithm does have some drawbacks. It sacrifices the intrinsic interpretability inherent in decision trees. Additionally, random forest can be computationally demanding, especially when dealing with large datasets, necessitating ample memory resources. And, finally, they are sensitive to noisy data which may cause overfitting [126].

5.3 XGBoost

XGBoost, or eXtreme Gradient Boosting [127], stands out as a state-of-the-art machine learning algorithm for regression and classification across diverse domains. Belonging to the family of boosting algorithms, specifically gradient boosting, XGBoost is highly regarded for its proficiency in handling various data types and producing accurate models. Despite the traditionally time-consuming nature of model construction in boosting methods, recent implementations like open-source XGBoost or LightGBM have significantly enhanced computational efficiency, making them widely adopted and leading approaches in the field. This method is utilized in Sections 4.1 and 4.2.

Boosting involves constructing a strong learner by sequentially incorporating weak learners into the ensemble. Traditionally, this entails using a decision stump. Essentially, a decision tree that focuses on a single value of one variable to make a prediction. Boosting can be comprehended by drawing a comparison with bagging. The initial distinction lies in that each decision tree is trained using the same dataset, without any sampling involved. Instead, every instance in the training dataset possesses a weight according to the difficulty the ensemble encounters in predicting that particular example. Another deviation from bagging is that the underlying learning algorithm, e.g., decision tree, needs to take into account the weightings assigned to the training dataset. This implies that ensemble members are constructed with a bias towards making accurate predictions on heavily weighted examples. The third point refers to the construction of the ensemble. Members are added sequentially, until the desired number is reached. Notably, the weightings of the training dataset update based on the capability of the entire ensemble after incorporating the new member. This ensures that the next added member works to rectify errors made by the entire model on the training dataset. The contribution of each model to the final prediction is a weighted sum of the performance exhibited by each individual model. Adaptive Boosting, or AdaBoost for short, stands out as the first successful implementation of the boosting technique [128, 129].

Gradient boosting represents an extension of the AdaBoost set of techniques. Naive gradient boosting operates as a greedy algorithm. It constructs trees by selecting optimal split points based on purity scores or minimizing the loss. The addition of trees occurs one at a time, and the existing trees in the model remain unchanged. A gradient descent procedure is then employed to minimize the loss when incorporating additional trees. The choice of the loss function relies on the nature of the problem at hand. The only limitation is that the loss function must be differentiable. Regularization methods are integrated to penalize specific aspects of the algorithm, enhancing overall performance and mitigating overfitting. Additionally, several enhancements aim to optimize the performance of the gradient boosting approach. These include tree constraints, such as controlling the depth of trees and the quantity of trees within the ensemble; random sampling, involving fitting trees on random subsets of features and samples, labeling the model as stochastic gradient boosting; and weighted updates, such as employing a learning rate or shrinkage to limit the impact of each tree on the ensemble [130, 131].

Now that the model has been introduced, the mathematical background of the algorithm is detailed. It is governed by the objective function, formed by training loss and regularization terms, that should be optimized for the step t.

$$\text{obj}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \omega(f_i)$$

where $\hat{y}_i^{(t)}$ is the prediction value at step t; and $\omega(f_i)$ is the complexity of the tree. The most common measures for the training loss are mean squared error in regression problems, and logarithmic loss for classification tasks.

It can be observed that the essential elements to be learned are the functions f_i, each encompassing the tree structure and leaf scores. An additive strategy is employed, wherein the learned aspects are fixed, and one new tree is added at a time.

$$\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}$$

Therefore, the objective function becomes:

$$\text{obj}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \omega(f_t) + \text{constant}$$

Afterwards, the training loss terms are replaced according to the nature of the problem. Nevertheless, in the general case to simplify the math, the loss function is linearized by means of second-order Taylor expansion:

$$\text{obj}^{(t)} = \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \omega(f_t) + \text{constant}$$

where the gradients g_i and hessian h_i are the first and second derivative of the loss functions, respectively.

$$\begin{aligned}
g_i &= \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \\
h_i &= \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})
\end{aligned}$$

After omitting all the constants, which do not have an impact within the optimization, the specific objective function at step t, only depending on g_i and h_i, becomes

$$\text{obj}^{(t)} = \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \omega(f_t)$$

The training step has been introduced. Nevertheless, the regularization terms need to be defined as well. XGBoost expresses the complexity of the tree as, in which λ is a parameter to encourage prunning, and T is the number of leaves:

$$\omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

The next step in the procedure consists of re-formulating the objective function for the t tree as, where $I_j = \{i|q(x_i) = j\}$ is the set of indices of data points assigned to the j-th leaf:

$$\begin{aligned}\text{obj}^{(t)} &\approx \sum_{i=1}^{n}[g_i w_{q(x_i)} + \frac{1}{2}h_i w_{q(x_i)}^2] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T}[(\sum_{i\in I_j} g_i)w_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2] + \gamma T\end{aligned}$$

in which the definition of tree $f(x)$ has been refined as:

$$f_t(x) = w_{q(x)}, w \in R^T, q : R^d \to \{1, 2, \cdots, T\}.$$

where w is the vector of scores on leaves, and q is a function assigning each data point to the corresponding leaf.

Finally, it arrives at the point of solving for the best vector of scores w_j^* and the best objective reduction obj^* for a tree structure $q(x)$, given that $G_j = \sum_{i\in I_j} g_i$ and $H_j = \sum_{i\in I_j} h_i$:

$$\begin{aligned}w_j^* &= -\frac{G_j}{H_j + \lambda} \\ \text{obj}^* &= -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma T\end{aligned}$$

Keep in consideration that it is not viable to create all possible trees, assess them, and pick the best estimator. Instead, the tree is optimized one level at a time. The gain of the tree is the score on the left leaf, the score on the right leaf, the score on the original lead, and the regularization on the additional leaf:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma$$

A relevant observation emerges: if the gain is less than γ, it is more beneficial not to add the branch. This aligns precisely with the pruning techniques applied in tree-based models.

One advantage of employing gradient boosting lies in the ease of obtaining importance scores for each attribute once the boosted trees are built. For a single decision tree, the importance of each attribute split point is computed based on the improvement it brings to the performance measure, weighted by the number of observations the corresponding node is accountable for. The feature importances are subsequently averaged across all the decision trees within the model. In addition to the previously mentioned positive qualities, such as flexibility and high accuracy, another notable advantage of XGBoost is its speed. This algorithm is specifically designed to perform well, even when dealing with large datasets, and it is optimized for both single- and multi-core processing. Nevertheless, XGBoost is not without its drawbacks. One notable limitation is its memory-intensive nature, particularly evident when dealing with large datasets. This can challenge computers with limited memory, resulting in slower performance. Additionally, XGBoost is often characterized as a black box algorithm, making it difficult to interpret and comprehend the underlying processes that drive its predictions. This lack of transparency makes troubleshooting and fine-tuning more difficult for users seeking to understand and optimize the model.

5.4 CatBoost

CatBoost [132], abbreviated from Categorical Boosting, is a cutting-edge open-source extension that uses gradient boosting on decision trees. It stands out for its ability to effectively handle categorical features, for both classification and regression tasks. This method is utilized in Section 4.1.

Dealing with categorical data in machine learning introduces challenges that require thoughtful consideration to ensure accurate model performance. Diverse techniques, such as One-Hot Encoding or Label Encoding, have appeared to manage the situation. Nevertheless, those methods carry their own limitations and drawbacks. Key concerns include high cardinality, i.e., a large number of unique values within a categorical feature which increases computational complexity; and ordinality assumption, assigning numerical values to categorical variables, creating an artificial ordinal relationship that may mistake the model. Another advanced technique for categorical encoding is Target Encoding. This method involves replacing categories with the weighted mean of the target variable for each category, leveraging information from the target variable. Unfortunately, it implies a high risk of overfitting. As the categorical value is transformed based on the target, an unintentional inclusion of information might occur during the training phase. It is known as data leakage. To eliminate, or diminish, this concern the K-Fold Target Encoding technique is employed. This approach incorporates K-fold cross-validation, dividing the dataset into k folds and performing target encoding k times, with each fold acting as the validation set once [133, 134].

Nonetheless, complete avoidance of data leakage risk remains a challenge. In extreme case scenarios, like employing Leave-One-Out Target Encoding within a single category variable, data leakage appears. CatBoost adopts an alternative categorical data encoding strategy that mitigates leakage concerns. This method introduces an artificial time variable, simulating the sequential handling of each data sample by the algorithm. CatBoost encodes based on all preceding data for the current sample, thereby minimizing the risk of leakage. Because the order of the samples is relevant in the process, the authors called it Ordered Target Encoding. Lastly, the encoding equation is adjusted, substituting the Overall Mean with an initial guess or prior in this approach.

When CatBoost generates a tree, it initiates the process by shuffling the rows of the training dataset. Subsequently, it employs Ordered Target Encoding on discrete variables with more than two options. For binary variables, a transformation to ones and zeros is performed. For regression tasks, continuous values are discretized into equally sized bins. Hence, the model initializes predictions with a prior value and calculates residuals. Then, it constructs a tree with leaves whose outputs are set to zero. Eventually, the score varies according to the average of the residuals in the leaf. This approach ensures that the sample's residual does not influence the leaf output calculation, preventing any form of leakage. To assess the quality of predictions for each split, CatBoost measures the cosine similarity between the leaf outputs and the residuals. The updated predictions result from adding the existing residuals to the current tree's leaf outputs, scaled by a learning rate. This process repeats until reaching the maximum number of trees or failing to achieve a significant performance improvement. Notably, CatBoost builds symmetric trees, meaning they employ the same threshold for every node at the same level. This decision is motivated by the notion that symmetric trees are weaker and, especially, faster learners than the other types of trees.

CatBoost provides a wide range of opportunities to assess feature importance [135]. By default, for non-ranking metrics, features are arranged based on the average change in prediction when the feature value undergoes a change. The greater the variation, the more significant the relevance of the feature. Conversely, for ranking metrics, the relevance is determined by the difference in the model's loss value with and without the feature. The greater the disparity in performance, the more crucial the feature becomes. Since this technique is computationally intensive, CatBoost approximates it. Instead of retraining the model from scratch, it utilizes the original model and virtually eliminates the feature from all the trees in the ensemble. It's important to note that this calculation relies on a dataset, making the derived value dataset-dependent. Consequently, the importance of a feature may vary based on the specific dataset used.

Furthermore, it is possible to analyze the impact of a feature on prediction results for a pair of samples. The process involves calculating the maximum possible change in the difference between predictions when the feature value is altered for both objects. This is particularly beneficial for understanding why a pair of instances might be incorrectly ranked. Another approach is feature interaction, i.e., assessing the dependency between two features in making predictions. For each pair of features, CatBoost examines all the splits in the trees where these features are utilized. If splits of both features exist in the same tree, CatBoost calculates the change in leaf value when these splits have the same value and when they have opposite values. The greater the change, the stronger the interaction between the two features. Finally, CatBoost also allows obtaining SHAP values, but these will be explained in their dedicated section.

5.5 Bayesian Optimization

Machine learning algorithms aim to create models that generalize well to unseen data, necessitating robust evaluation techniques. Cross-validation and Hyperparameters Tuning have emerged as a pivotal method in this context, providing a systematic approach to assess and refine the performance of machine learning models. Together, these techniques form a comprehensive methodology for ensuring the reliability and effectiveness of machine learning models in real-world applications. This method is utilized in Section 4.1.

The K-Fold Cross Validation procedure is a standard method for estimating the performance of a machine learning algorithm or configuration on a dataset. It assesses and mitigates the risk of overfitting by partitioning a dataset into multiple subsets. These subsets, or folds, are used iteratively for training and testing the model. The goal is to obtain a more reliable estimate of the model's performance by evaluating it on different portions of the data. By systematically rotating the folds, cross-validation provides a robust means of assessing how well a model generalizes to unseen instances. Let $D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ represent a dataset with n instances, where x_i is the feature vector and y_i is the corresponding label. In K-Fold Cross Validation, the dataset is partitioned into k non-overlapping folds $D = \bigcup_{i=1}^{k} D_i$, where D_i is the $i - th$ fold. The training and testing procedure is repeated k times, with each fold serving as the test set once. For each iteration i, the model is trained on the union of all folds except D_i, denoted as $D_{train} = \bigcup_{j \neq i} D_j$. The performance metric E_i is then computed by testing the model on D_i. The overall performance E_{CV} of the model is obtained by averaging the performance metrics overall k folds.

Hyperparameters tuning, on the other hand, focuses on optimizing the internal configurations of a model that unlike model parameters, which are learned from the data, are set prior to training. These internal parameters, such as learning rates, regularization strengths, or network architectures, significantly impact a model's performance, so they must be tuned to maximize it. The process of hyperparameter tuning involves systematically searching through a predefined parameter space, evaluating the model's performance for each set of hyperparameters, and selecting the combination that yields the best results. The search space entails the range of values that hyperparameters can take.

Two of the most common methods for executing hyperparameters tuning are Grid Search and Random Search. The former is a straightforward approach that involves defining a set of values for each hyperparameter within a predefined range. Afterwards, for each one from all possible combinations the machine learning model is trained and evaluated. Consequently, it is guaranteed that the optimal solution within the specified search space will be found. Additionally, the simplicity of grid search makes it an attractive choice, especially when the search space is discrete and not overly complex. On the contrary, the primary drawback lies in its computational cost. As the number of hyperparameters and their potential values increases, the search space expands exponentially, leading to an impractical number of combinations to evaluate. Additionally, in scenarios where hyperparameters have a continuous range, grid search may be less effective, as it discretizes the search space, potentially missing optimal configurations between the predefined values.

Regarding Random Search, it takes a probabilistic approach by randomly sampling hyperparameter configurations from the search space. Given a learner $\mathcal{M}$, with parameters x and a loss function $f(x)$, random search tries to find x such that $f(x)$ is maximized, or minimized, by evaluating $f(x)$ for randomly sampled values of x. This stochastic nature allows random search to efficiently navigate high-dimensional and continuous spaces, where an exhaustive grid search would be computationally impractical. Random search typically requires fewer evaluations as it does not explore every possible combination. On the contrary, it does not ensure finding the globally optimal hyperparameter configuration. While it is more likely to explore diverse regions of the search space, it may overlook critical areas that lead to improved model performance. And similar to grid search, random search may face challenges in efficiently exploring discrete hyperparameter spaces.

Hence, Bayesian Optimization (BO) [136] stands out as a sophisticated and effective strategy for hyperparameter tuning. The fundamental concept involves constructing a surrogate model, typically a Gaussian Process that models $f(x)$, to approximate the genuine objective function. This approximation facilitates a more informed selection of subsequent hyperparameter configurations x for evaluation, thanks to the acquisition function. The algorithm can be broadly summarized as follows.

1. Assess $f(x)$ at n initial points.
2. Iterate while $n \leq N$:
   - Update the surrogate model, such as the Gaussian Process, utilizing all available data $\mathcal{D}1 : n$.
   - Calculate the acquisition function $u(x \mid \mathcal{D}1 : n)$ using the current surrogate model.
   - Determine x_{n+1} as the maximizer of the acquisition function, denoted as $x_{n+1} = \text{argmax}_x u(x \mid \mathcal{D}_{1:n})$.
   - Evaluate $y_{n+1} = f(x_{n+1})$.
   - Expand the data set $\mathcal{D}_{1:n+1} = \mathcal{D}_{1:n}, (x_{n+1}, y_{n+1})$ and increment n.
3. Return either the x assessed with the highest $f(x)$ or the point with the highest posterior mean.

A Gaussian Process (GP) [137, 138], understood as the surrogate model, extends the concept of a Gaussian distribution from random variables to a distribution over functions. Initially, it begins with a distribution encompassing all conceivable functions that could potentially have generated the samples, without taking the actual data into account. These functions are referred to as GP priors and represent the uncertainty regarding the true underlying function $f(x)$. Subsequently, the range of functions is refined by incorporating the available samples.

The reality is that it's not necessary to consider every mathematically valid function. Instead, constraints are applied to the prior distribution covering all potential functions. Initially, there is an expectation for our functions to exhibit smoothness, aligning with empirical knowledge about the typical functioning of the world. Points in close proximity within the input space, denoted as $x_1, x_2, ...$, are associated with corresponding $y_1, y_2, ...$ values that are also close to each other. This smoothness is introduced through the covariance matrix, where each element determines the degree of correlation between the (y_i, y_j) variables based on the proximity of values in the input space (x_i, x_j). The distance is measured by the kernel function $k(x, x')$. Figure 5.1 compiles some widely used kernel functions for constructing GPs and the resulting GP priors.

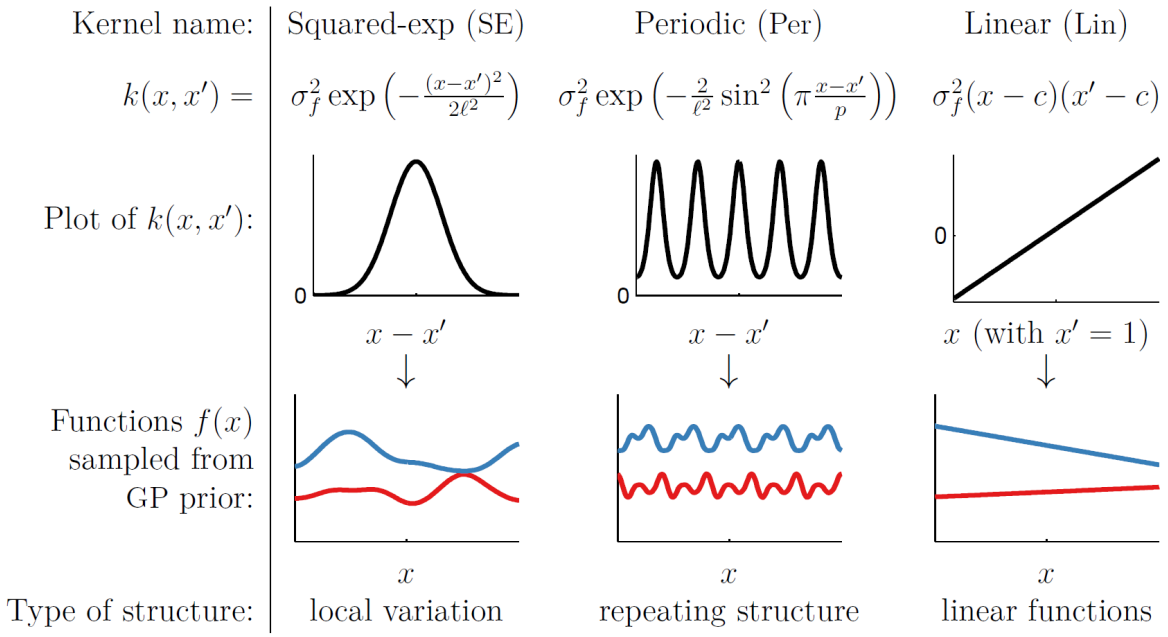


Figure 5.1: Examples of structure expressible by some basic kernels. Source: [139]

Given a kernel $k(x, x')$, the covariance matrix is construct as follows:

$$\Sigma(x, x') = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & k(x_1, x_3) & \dots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & k(x_2, x_3) & \dots & k(x_2, x_N) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & k(x_N, x_3) & \dots & k(x_N, x_N) \end{bmatrix}$$

The covariance matrix $\Sigma(x, x')$ must be positive definite, meaning that the following condition must be met $x^\top \Sigma x > 0, \forall x \neq 0$. Lastly, a mean function $m(x)$ is required to fully characterize the multivariate normal distribution that will simulate the function $f(x)$:

$$f(x) \sim \mathcal{GP}\left(m(x), k(x, x')\right)$$

To find the best point to sample $f(x)$ next from, it will be chosen the point that maximizes an acquisition function (AF) [140]. Instead of maximizing directly $f(x)$, whose analytic form is unknown, another function, that is much easier to do and much less expensive, is maximized. Acquisition functions are constructed so that a high value corresponds to potentially high values of the objective function. By finding the x that maximizes the AF, it is identified the next best guess for $f(x)$ to try. Either because the prediction is high or because the uncertainty is high. This is known as the so-called exploration-exploitation trade-off. There are three often cited acquisition functions: upper confidence bound (UCB), probability of improvement (PI), and expected improvement (EI).

UCB With an upper confidence bound, the exploitation vs. exploration tradeoff is straightforward and tuned via the parameter λ. Concretely, UCB is a weighted sum of the expected performance captured by $\mu(x)$ of the Gaussian Process, and of the uncertainty $\sigma(x)$, captured by the standard deviation of the GP. When λ is small, Bayesian Optimization will favor solutions that are expected to be high-performing, i.e., exploitation. On the contrary, when λ is large, BO rewards the exploration of currently uncharted areas in the search space:

$$a(x;\lambda) = \mu(x) + \lambda\sigma(x)$$

PI Improvement $I(x^\star)$ is defined as an indicator function that measures the positive difference between the function values at the candidate point $x^\star$ and the reference one x.

$$I(x^\star) = \max(f(x^\star) - f(x), 0)$$

In the context of the probability of improvement acquisition function, each candidate $x^\star$ is assigned the probability of $I(x^\star) > 0$. In a Gaussian Process, a Gaussian distribution is associated with each point, and at the specific location $x^\star$, the function value $f(x^\star)$ is sampled from a normal distribution with a mean of $\mu(x^\star)$ and a variance $\sigma^2(x^\star)$.The expression can undergo a reparameterization after assuming that $z \sim \mathcal{N}(0,1)$. Therefore, the improvement function is rewritten as:

$$I(x^\star) = \max(f(x^\star) - f(x), 0) = \max(\mu(x^\star) + \sigma(x^\star)z - f(x), 0) \;\; z \sim \mathcal{N}(0,1)$$

Hence, the probability that $x^\star$ produces an improvement, in other words, the probability of improvement, is captured by $\mathrm{PI}(x^\star) = \Pr(I(x^\star) > 0) \Leftrightarrow \Pr(f(x^\star) > f(x))$. Building upon this, PI can be articulated using the standard normal distribution:

$$\mathrm{PI}(x^\star) = 1 - \Phi(z_0) = \Phi(-z_0) = \Phi\left(\frac{\mu(x^\star) - f(x)}{\sigma(x^\star)}\right)$$

The interpretation of these expressions lies in the cumulative distribution function $\Phi(z_0)$ of the standard normal distribution. The term $1 - \Phi(z_0)$ represents the probability that a randomly sampled point around $x^\star$ yields an improvement. Similarly, $\Phi(-z_0)$ represents the probability that $f(x^\star)$ is better than a randomly chosen point. The development of z_0 considers how much the mean function value at $x_\star$ deviates from $f(x)$ in terms of the standard deviation $\sigma(x^\star)$. This formulation allows for a probabilistic assessment of improvement, taking into account both the mean and variability of the function.

EI Probability of improvement considers only the probability of improving the current best estimate, but it does not factor in the magnitude. On the contrary, expected improvement acquisition function calculates the expected value of $I(x^\star)$, where $\varphi(z)$ is the probability density function of the normal distribution $\mathcal{N}(0,1)$:

$$\mathrm{EI}(x) \equiv \mathbb{E}\left[I(x)\right] = \int_{-\infty}^{\infty} I(x)\varphi(z)\,\mathrm{d}z = \int_{-\infty}^{\infty} \max(f(x^\star) - f(x), 0)\varphi(z)\,\mathrm{d}z$$

In order to calculate this integral, it is necessary to get rid of the max operator. Thus, the expression will be split into two components, one where $f(x^\star) - f(x)$ is positive and other one where it is zero.

$$f(x^\star) = f(x) \Rightarrow \mu + \sigma z = f(x) \Rightarrow z_0 = \frac{f(x) - \mu}{\sigma}$$

$$\mathrm{EI}(x^\star) = \underbrace{\int_{-\infty}^{z_0} I(x)\varphi(z)\,\mathrm{d}z}_{\text{Zero since } I(x)=0} + \int_{z_0}^{\infty} I(x^\star)\varphi(z)\,\mathrm{d}z$$

Therefore, the expected improvement acquisition function evolves such as:

$$\begin{aligned}
\mathrm{EI}(x^\star) &= \int_{z_0}^{\infty} \max(f(x^\star) - f(x), 0)\varphi(z)\,\mathrm{d}z = \int_{z_0}^{\infty} (\mu + \sigma z - f(x))\,\varphi(z)\,\mathrm{d}z \\
&= \int_{z_0}^{\infty} (\mu - f(x))\,\varphi(z)\,\mathrm{d}z + \int_{z_0}^{\infty} \sigma z \frac{1}{\sqrt{2\pi}} e^{-z^2/2}\,\mathrm{d}z \\
&= (\mu - f(x)) \underbrace{\int_{z_0}^{\infty} \varphi(z)\,\mathrm{d}z}_{1-\Phi(z_0)\equiv 1-\mathrm{CDF}(z_0)} + \frac{\sigma}{\sqrt{2\pi}} \int_{z_0}^{\infty} z e^{-z^2/2}\,\mathrm{d}z \\
&= (\mu - f(x))\,(1 - \Phi(z_0)) - \frac{\sigma}{\sqrt{2\pi}} \int_{z_0}^{\infty} \left(e^{-z^2/2}\right)'\,\mathrm{d}z \\
&= (\mu - f(x))\,(1 - \Phi(z_0)) - \frac{\sigma}{\sqrt{2\pi}} \left[e^{-z^2/2}\right]_{z_0}^{\infty} \\
&= (\mu - f(x)) \underbrace{(1 - \Phi(z_0))}_{\Phi(-z_0)} + \sigma\varphi(z_0) \\
&= (\mu - f(x))\,\Phi\left(\frac{\mu - f(x)}{\sigma}\right) + \sigma\varphi\left(\frac{\mu - f(x)}{\sigma}\right)
\end{aligned}$$

The explanation of the above formula is as follows. $EI(x^\star)$ will take high values when $\mu > f(x)$, in other words, the mean value of the Gaussian process is higher at $x^\star$. The equation is also increased when the uncertainty is large, i.e., when $\sigma > 1$. Notably, the expression works for $\sigma(x^\star) > 0$. Otherwise, as the case of observed data where the uncertainty is null ($\sigma(x) = 0$), it holds that $\mathrm{EI}(x) = 0$. Lastly, the trade-off between exploitation vs exploration is ruled by the inclusion of the parameter ξ into the formula for $EI(x^\star)$. The larger the values of ξ, the more explorative the Bayesian Optimization will be. Thus, the full equation is:

$$\mathrm{EI}(x^\star; \xi) = (\mu - f(x) - \xi)\,\Phi\left(\frac{\mu - f(x) - \xi}{\sigma}\right) + \sigma\varphi\left(\frac{\mu - f(x) - \xi}{\sigma}\right)$$

5.6 SHAP

SHAP (SHapley Additive exPlanations) [141] is a technique designed to provide explanations for individual predictions. This method is utilized in Section 4.1. It builds upon Shapley values [142], a concept originating from coalitional game theory, which addresses the fair allocation of rewards among players in a game. In this context, the "game" represents the prediction task for a single instance of the dataset, where the "gain" is defined as the difference between the actual prediction for this instance and the average prediction across all instances. The "players" in this scenario are the features within the instance, working together to contribute towards achieving a certain predicted value.

The Shapley value is determined through a value function, denoted as val, which operates on features within the set S. Each feature's marginal contributions are assessed based on the probability of them making those contributions. Subsequently, the total of all potential coalitions that a feature can engage with to make a marginal contribution is computed. This process yields an expected marginal contribution.

$$\phi_j(val) = \sum_{S \subseteq \{1,\dots,p\} \setminus \{j\}} \underbrace{\frac{|S|!\,(p-|S|-1)!}{p!}}_{Weight} \underbrace{\left(val\left(S \cup \{j\}\right) - val(S)\right)}_{\text{Marginal contribution}}$$

where S is a subset of the features used in the model, $|S|$ is the number of features in the subset and $|S|!$ is the number of ways coalition S can form; x is the vector of feature values of the instance to be explained; $|p|$ is the number of coalitions made out of p features; consequently, $(p-|S|-1)!$ is the number of ways features can join after feature j joins. With respect to the marginal contribution, $val(S)$ is the value of the coalition S excluding feature j, whilst $val\left(S \cup \{j\}\right)$ has the same meaning but including the feature j.

Shapley values offer a fair method for distributing the contribution of each feature. They consider all possible coalitions a feature can engage with, thus capturing both individual feature contributions and interactions between features. However, to support this claim, four properties need to be examined: Efficiency, Symmetry, Null Player, and Additivity. Efficiency ensures that no value of gain is left unaccounted for. Symmetry dictates that two features are interchangeable if they make identical contributions to all coalitions. If a feature yields zero marginal contribution across all coalitions, it is classified as a Null Player and contributes nothing to the total value. Lastly, Additivity stipulates that when combining two predictions, the overall contribution of a feature is the sum of its contributions to the individual predictions. This property assumes that predictions are independent of each other. Unlike other attribution methods, Shapley values are the only ones that satisfy all these properties [143, 144].

SHAP differs from Shapley values for two primary reasons. Firstly, it introduces KernelSHAP and TreeSHAP, which are alternative kernel-based estimation approaches for Shapley values. KernelSHAP is inspired by local surrogate models, while TreeSHAP offers an efficient estimation method for tree-based models. Secondly, SHAP includes various global interpretation methods that are based on aggregations of Shapley values.

KernelSHAP aims to estimate the contributions of each feature value to the prediction for a given instance x. It operates through five key steps:

- Sampling coalitions $z'_k \in 0, 1^M, \quad k \in 1, \ldots, K$, where 1 indicates the presence of a feature in the coalition, and 0 indicates absence.
- Obtaining predictions for each z'_k by first converting them to the original feature space and then applying the model $\hat{f}$: $\hat{f}(h_x(z'_k))$.
- Calculating the weight for each z'_k using the SHAP kernel.
- Fitting a weighted linear model.
- Returning the Shapley values phi_k corresponding to the coefficients obtained from the linear model.

To derive values from coalitions of features that validate data instances, a function $h_x(z') = z$ is required, where $h_x : \{0, 1\}^M \rightarrow \mathbb{R}^p$. This function maps 1's to the corresponding values from the instance x being explained and 0's to values from the instance, which will be replaced by random feature values from the data. The function h_x treats feature X_j and X_{-j} (i.e., the other features) as independent and integrates them over the marginal distribution: $\hat{f}(h_x(z')) = E_{X_{-j}}[\hat{f}(x)]$. This approach disregards the dependence structure between present and absent features, resulting in KernelSHAP suffering from the same problem as all permutation-based interpretation methods. This estimation places excessive weight on improbable instances, leading to unreliable results. However, sampling from the marginal distribution is necessary, despite this limitation.

Concerning TreeSHAP [145], it represents a variation of SHAP tailored for tree-based machine learning models such as decision trees, random forests, and gradient-boosted trees. Unlike the exact KernelSHAP, TreeSHAP achieves polynomial time complexity instead of exponential. Specifically, it reduces computational complexity from $O(TL2^M)$ to $O(TLD^2)$, where T stands for the number of trees, L denotes the maximum number of leaves in any tree, and D represents the maximal depth of any tree. TreeSHAP employs conditional expectation $E_{X_j|X_{-j}}(\hat{f}(x)|x_j)$ to estimate effects rather than marginal expectation. The fundamental concept involves simultaneously propagating all possible subsets S down the tree. However, a challenge with the conditional expectation arises when features lacking influence on the prediction function f receive a TreeSHAP estimate different from zero, violating the Null player property. This scenario can occur when a feature is correlated with another feature that does impact the prediction.

Regarding interpretation methods, there are numerous options available. Some of these will be outlined shortly. SHAP feature importance entails calculating the average of the absolute Shapley values per feature across the dataset. Subsequently, the features are arranged in descending order of importance and visualized accordingly. The summary plot combines feature importance with feature effects, where each point represents a Shapley value for a feature and an instance. The color gradient indicates the feature's value from low to high, with overlapping points slightly jittered along the y-axis to illustrate the distribution of Shapley values per feature. Features are organized based on their importance. However, for a deeper understanding of the relationship between a feature's value and its impact on the prediction, it is necessary to visualize a SHAP dependence plot. This plot depicts each data instance's feature value on the x-axis and the corresponding Shapley value on the y-axis. Additionally, the dependence plot can be enhanced by

highlighting feature interactions, which represent the combined effect of features after accounting for individual effects. Lastly, Shapley values can aid in data clustering by clustering the Shapley values of each instance, grouping instances based on similarity in explanations. The resulting plot comprises multiple force plots, each explaining the prediction of an instance.

5.7 ARIMA(X)

At first glance, time series forecasting seems like a conventional regression problem. However, certain crucial distinctions necessitate attention. Unlike other regression tasks in machine learning, time series possess a temporal order. This temporal indexing must be preserved. Otherwise, the model risks being trained on future information unavailable during prediction, leading to what is known as look-ahead bias. Consequently, the resultant model would likely lack reliability and perform inadequately when making future forecasts. Additionally, time series occasionally lack distinct features. In the absence of additional features, methods must be devised to utilize past values of the time series for forecasting future values. All explanations in this section are derived from the book [146]. This technique is used in Section 4.2.

A time series comprises a sequence of data points arranged chronologically. Additionally, the data typically exhibits uniform time spacing, with consistent intervals separating each data point. Understanding time series can be enhanced by examining their three constituent elements: trend, seasonality, residuals. The trend encompasses the gradual changes observed within a time series. The seasonal component captures recurring patterns that manifest over fixed time intervals, illustrating deviations from the trend. The residuals, representing the unexplained variance not attributable to either the trend or seasonal components, often correspond to random errors, also known as white noise. They embody information that defies modeling or prediction due to its stochastic nature. Indeed, all time series can be decomposed into these three constituent elements. Decomposition refers to the statistical process of partitioning a time series into its constituent components.

In the context of a non-stationary integrated time series, the autoregressive integrated moving average model, abbreviated as ARIMA(p,d,q) [147], can be employed for generating forecasts. This model consists of an autoregressive process AR(p), integration I(d), and a moving average process MA(q). The mathematical representation of the ARIMA(p,d,q) process stipulates that the present value of the differenced series y'_t equals the sum of a constant C, past values of the differenced series $\phi_p y'_{t-p}$, the mean of the differenced series μ, past error terms $\theta_q \epsilon'_{t-q}$, and a current error term ϵ_t.

$$y'_t = C + \phi_1 y'_{t-1} + \cdots \phi_p y'_{t-p} + \theta_1 \epsilon'_{t-1} + \cdots + \theta_q \epsilon'_{t-q} + \epsilon_t$$

The definition of the ARIMA(p,d,q) model introduces new concepts not previously discussed, which will now be elucidated. The order of integration corresponds to the number of differencing operations applied to a series to achieve stationarity. Differencing involves calculating the change between successive time steps, with a first-order differencing denoting a single differencing operation and a second-order differencing indicating two consecutive operations. A stationary time series maintains constant statistical properties over time, including a stable mean, variance, and autocorrelation. Many forecasting models presume stationarity, necessitating its verification. A commonly used test for this purpose is the augmented Dickey-Fuller (ADF) test [148, 149]. This statistical test assesses

the null hypothesis that the time series possesses a unit root, indicating non-stationarity, while the alternative hypothesis suggests the absence of a unit root, confirming stationarity. Consider a simplistic time series where the present value y_t solely depends on its past value y_{t-1}, governed by a coefficient α_1, a constant C, and white noise ϵ_t. This time series attains stationarity only if the root lies within the unit circle, implying its value falls between -1 and 1; otherwise, the series remains non-stationary.

The moving average model represents the present value y_t as a linear combination of the series mean μ, the current error term ϵ_t, and past error terms ϵ_{t-q}. The coefficient θ_q quantifies the extent to which past errors influence the present value.

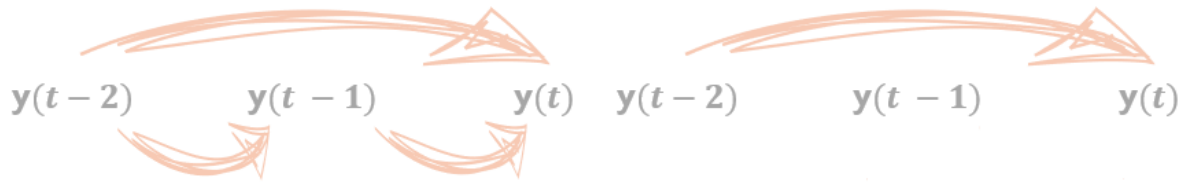
$$y_t = \mu + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \cdots + \theta_q\epsilon_{t-q}$$

Determining the order of the moving average model can be facilitated by examining the autocorrelation function (ACF) [150]. Autocorrelation assesses the linear relationship between lagged values within a time series, with lag representing the number of time steps between two values. Consequently, the ACF illustrates how the correlation between any two values evolves with increasing lag. In the presence of a trend, the ACF plot typically exhibits high coefficients for short lags, which diminish linearly as the lag increases. If the data exhibits seasonality, the ACF plot may also reveal cyclic patterns. In cases where consecutive coefficients exhibit significant changes in behavior, the order of the moving average process corresponds to the latest coefficient before the transition.

However, the ACF plot may also indicate that the time series is governed by an autoregressive process. In such a model, the present value y_t is expressed as a linear combination of a constant C, the current error term ϵ_t (also characterized as white noise), and past values of the series y_{t-p}. The extent to which past values influence the present value is represented by phi_p, signifying the coefficients of the autoregressive model.:

$$y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots \phi_p y_{t-p} + \epsilon_t$$

The partial autocorrelation function (PACF) [151] is utilized for validation and determination of the autoregressive process order. PACF assesses the correlation between lagged values within a time series after eliminating the influence of correlated intermediate lagged values, often referred to as confounding variables. PACF indicates how partial autocorrelation changes with increasing lag, with coefficients becoming insignificant beyond lag p. Figure 5.2 illustrates the disparity between ACF and PACF.



(a) Autocorrelation Function (ACF) (b) Partial Autocorrelation Function (PACF)

Figure 5.2: Summary of the ACF and PACF for a time series

Once the components of the ARIMA(p,d,q) model have been elucidated, the model can be further developed to incorporate additional complexity. If the time series exhibits a seasonal pattern, the SARIMA predictive model is considered. However, particular attention is warranted for examining the influence of an exogenous variable, denoted by X_t, on predictions. In statistics, the term exogenous is used to describe predictors or input variables, whereas endogenous pertains to the target variable. Hence, the ARIMAX model originates from this distinction. It essentially integrates a linear combination of exogenous variables into the ARIMA model:

$$y_t = ARIMA(p, d, q) + \sum_{i=1}^{n} \beta_i X_t^i$$

Incorporating external variables may offer potential benefits, as strong predictors for the target variable could be identified. However, challenges may arise when forecasting multiple future timesteps using the ARIMAX model. This model necessitates forecasting the exogenous variables as well, which can be accomplished using a variant of the ARIMA model. Nonetheless, it is recognized that all forecasts inherently carry some degree of error. Thus, forecasting an exogenous variable alongside the target variable can amplify the prediction error of the target, leading to degradation in predictions as the forecasting horizon extends further into the future. The only recourse to elude this scenario is to limit forecasting to a single timestep ahead and await observation of the exogenous variable before forecasting subsequent timesteps for the target variable. Conversely, if the exogenous variable follows a known function that can be accurately forecasted, there is no detriment in forecasting the exogenous variable and utilizing these forecasts to predict the target variable.

5.8 Genetic Algorithm

The Genetic Algorithm [152, 153] is categorized within the family of stochastic optimization algorithms, particularly those inspired by biological or physical processes. Stochastic denotes the utilization of randomness in the objective function. Optimization, on the other hand, is a mathematical method for determining the maximum or minimum value of a function with multiple independent variables, subject to a set of constraints. Stochastic algorithms are straightforward to conceptualize, flexible, and adaptable to various contexts. They impose no specific requirements on the formulation of the optimization problem. Moreover, they offer an excellent balance between solution quality and computational time. However, they do not guarantee the discovery of the global optimum solution, although they typically excel at finding sufficiently good solutions. The algorithm employs analogs of genetic representation, fitness evaluation, genetic recombination, and mutation. Initially, a population of a fixed size is generated. The main algorithmic loop iterates for a predetermined number of iterations until the optimal value is attained or no further improvement is observed in the best solution after a specified number of iterations. All the steps are illustrated in Figure 5.3. This method is applied in Sections 4.4 and 4.5.

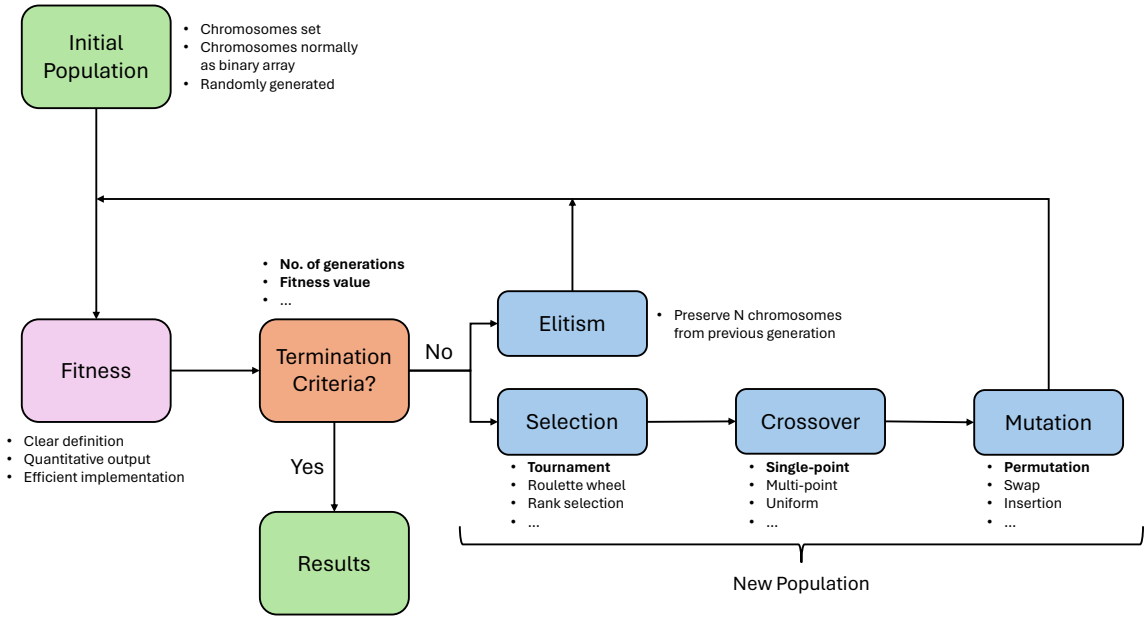


Figure 5.3: Genetic Algorithm Flowchart

The process begins with the generation of the initial population, which consists of a set of chromosomes. This population is randomly generated to ensure a diverse starting point within the solution space, thereby providing a broad search area and avoiding premature convergence to suboptimal solutions. Each chromosome encodes a potential solution to the problem under consideration, representing a point within the solution space. Chromosomes are commonly encoded as binary arrays, although they can be adapted to suit the specific problem. An inadequate representation can lead to a lack of convergence. In the context of this research, the chromosomes are made up of the following attributes: car model, trim level, engine, exterior color, and geographical location as the Spanish province, rather than the compound region, where the users access the online platform from. Additionally, a temporal flag is included in the form of the day of the week (DOW) the connection was done. The skeleton of the chromosome is represented in Table 5.1. Therefore, each chromosome represents a valid solution, i.e., a set of rules for filtering the clickstream data and distinguishing the users with real purchase intention. This strategy is called Pittsburgh [154], in opposition to Michigan [155] which claims that the population is the solution to the problem.

Table 5.1: Example of the structure of the chromosome, composed by the number of rules to find based on the attributes of the search space

	DOW	Car Model	TRIM	Engine	Exterior Color	Location
Rule 1	Monday	SEAT Ibiza	Reference	KX	L5L5	SEVILLA
Rule 2	Thursday	SEAT Leon ST	FR	2X	B4B4	AVILA
Rule 3	Sunday	SEAT Ibiza	FR	PV	9M9M	MURCIA
⋮				⋮		
Rule R	Tuesday	SEAT Arona	Style	GZ	0CF5	SANTANDER

Afterwards, the quality of the solution is quantified by its fitness, which assesses the proximity of a given candidate solution to the optimum solution. The fitness function translates the problem's objective into a form that the algorithm can evaluate. This typically involves computing a numerical score that reflects how well a candidate solution meets the desired criteria. Additionally, it must be carefully crafted to avoid misleading the search process. It should accurately reflect the true quality of solutions, ensuring that top fitness values correspond to better solutions. Nevertheless, efficiency in implementing the fitness function is crucial. Computationally expensive evaluations can significantly slow down the overall performance of the genetic algorithm. Therefore, the function should be designed to provide rapid assessments. In some cases, multi-objective fitness functions are employed to balance multiple criteria simultaneously. These functions aggregate different performance metrics, to provide a comprehensive assessment of solution quality. Along the thesis, two different fitness functions have been developed to match the required criteria. On Section 4.4 is sought to maximize the average correlation between the subset of clickstream data derived from the chromosome solution and the sales record, whilst the fitness function governing the Section 4.5 minimizes the prediction error of a multivariate forecasting. The aforementioned sections provide the details about the construction of the corresponding fitness functions.

New candidates, constituting subsequent populations, are derived from the preceding generation. This process continues until any of the termination criteria are met. After arriving to the maximum number of generations permitted, or achieving the fitness limit value, the genetic algorithm is interrupted. Some candidates from the previous generation are chosen to persist and undergo further evolution. Among various selection strategies, the roulette wheel selection calculates the cumulative probability for each individual and selects the first one whose probability meets or exceeds a randomly generated number. Similarly, rank selection involves sorting all individuals in the population based on their fitness. The selection of parents is determined by the rank of each individual rather than their fitness. This method is particularly useful when individuals in the population have similar fitness levels, which often occurs towards the end of the process. In contrast, truncation selection orders the candidates and selects a portion of the fittest individuals. However, in this context, tournament selection is the preferred strategy. It accommodates negative fitness values, does not necessitate sorting individuals, and operates efficiently on parallel architectures. The old generation is exposed to a tournament among its chromosomes. It generates random indices and picks the corresponding candidates from the old population. The amount of participants is defined by the tournament probability. It signifies the fraction of the population participating in each tournament. In our case, it is fixed at 30%. The genetic algorithm iterates to randomly select the best candidates. The tournament strategy aims to strike a balance between exploring diverse solutions (by allowing weaker candidates to occasionally win) and exploiting strong solutions (by favoring candidates with better fitness values).

The survivor chromosomes, which are called parents, will experience crossover, mutation, both, or none. The first methodology works based on the crossover probability, which is established into 90% in order to promote information exchange and rapid convergence towards optimal solutions. Crossover combines a pair of parents' solutions to create potentially improved children's solutions. To perform crossover, the algorithm randomly chooses a single intersection point on the parents and assembles the parts. An illustration of the methodology is presented in Figure 5.4. This method is called single-point crossover, although multi-point and uniform crossover are valid options.

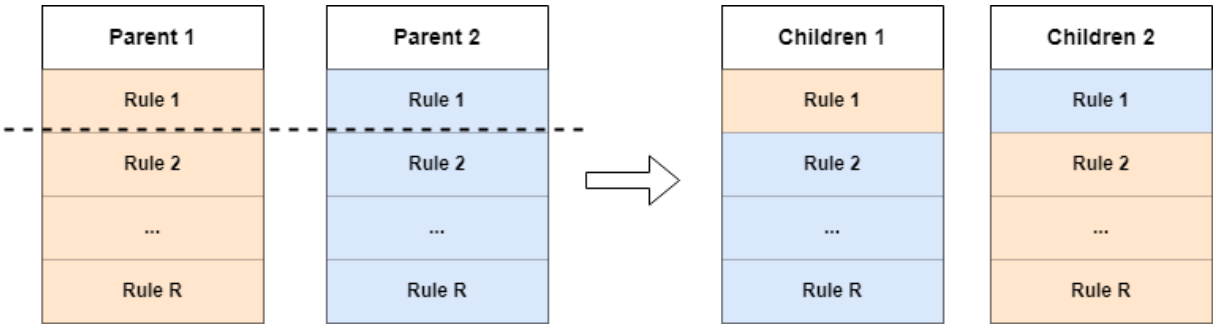


Figure 5.4: Exemplification of single-point crossover between two parents

The next step, the mutation might take place under the circumstance of the mutation probability. In this context, it is set in terms of the population size, such one chromosome, at least, is affected. Mutation introduces small random changes in a chromosome within the population. It serves to maintain diversity and prevent the algorithm from getting stuck in local optima. Whilst crossover recombines existing chromosomes' information, mutation introduces novel variations. However, a low mutation probability is recommended to maintain stability, preserve good solutions, and strike the right balance between exploration and exploitation. The procedure is shown in Figure 5.5. In this scenario, the element's rule is permuted by means of a uniform distribution by other permitted values. The feasibility is relevant for engine and exterior color, such as they are conditioned by the car model and TRIM level.

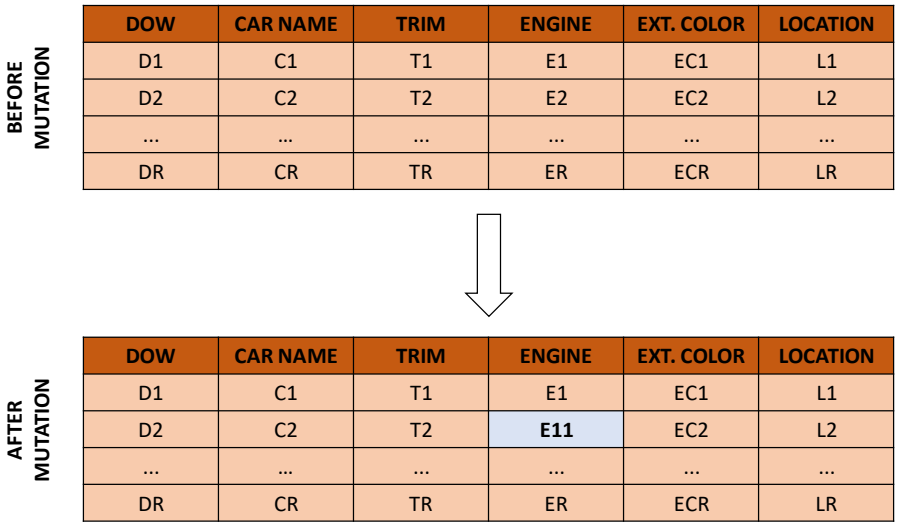


Figure 5.5: Exemplification of mutation of a chromosome of the genetic algorithm

Lastly, elitism is employed to prevent fitness from decreasing in successive generations. It is a strategy that involves preserving a certain number of the best chromosomes from one generation to the next, without subjecting them to any genetic operators. The purpose is to prevent the risk of losing highly fit solutions due to the randomness introduced by genetic operators. In our approach, exclusively the best single solution transfers from the precedent generation to the newer one.

5.9 Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov (KS) test [156, 157, 158] serves to determine whether a sample originates from a population following a specific distribution. Section 4.3 uses this technique. It enables comparison of a sample with a reference probability distribution or between two samples. The KS test relies on the cumulative distribution function, which signifies the probability of the function assuming a value less than or equal to a reference point. This function is a stepwise progression that increments by the inverse of the total number of data points N at each ordered data point's value. Consequently, the KS statistic represents the maximum absolute difference between the two cumulative distributions F_1 and F_2 respectively, as it is shown in Figure 5.6:

$$D = sup_x |F_1(x) - F_2(x)|$$

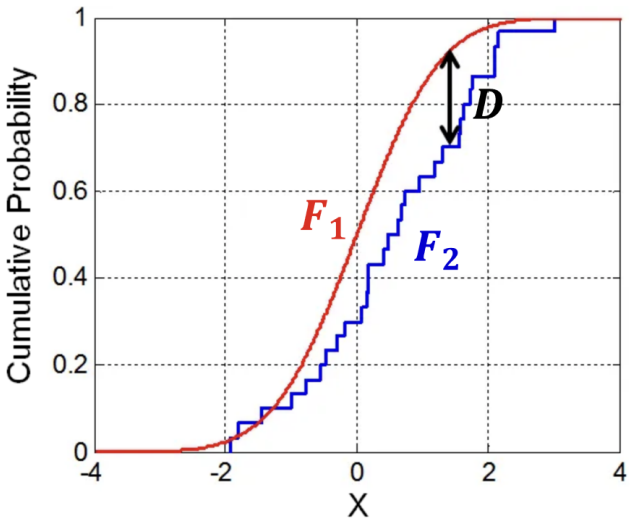


Figure 5.6: Exemplification of Kolmogorov-Smirnov test. Source [158]

The null hypothesis employed in this experiment posits that both samples are drawn from identical distributions. It constitutes a non-parametric and distribution-free test, indicating it does not assume any particular distribution for the data. Furthermore, it is considered an exact test, unlike the chi-square test, which relies on sufficient sample size for valid approximations. Despite these advantages, the KS test is subject to certain limitations. It is applicable solely to continuous distributions and demonstrates greater sensitivity toward the center of the distribution compared to the tails. Lastly, for the KS test to be valid, the distribution must be fully specified. If parameters such as location, scale, and shape are estimated from the data, instead of being known in advance, the standard K-S test results are not valid. In such cases, one usually needs to use simulation methods to determine the correct critical values for the test.

Chapter 6

Results

This chapter presents the results of each one of the topics that assemble the thesis. Each section is introduced by a brief summary of the methodology. Afterwards, results are shown and properly assessed.

6.1 Compound Reallocation Of Manufactured Cars

Four machine learning algorithms will be assessed on this classification task within the production and delivery data: Decision Tree, Random Forest, XGBoost, and CatBoost. The features space includes Car Model, Equipment Level (TRIM), Exterior Color, and Engine, together with Order Type and Compound Region. The target of the problem is the Delivery Type of the vehicle. In other words, whether a car is categorized as Fast Delivery or Normal Delivery, according to the number of days it has spent on the compound region. The threshold to decide between these two classes ranges from 1 to 6 weeks. It is pursued to obtain the best estimator among all the possible configurations of each algorithm. In order to execute this search most efficiently, Bayes Optimization has been employed. More details are found in Section 4.1.

6.1.1 The Best Estimator

The conditions of the Bayesian optimization are limited to 150 number of iterations, so it has room enough for searching for the best solution; 5 stratified folds according to the threshold time under analysis; and F1 Score as the scoring metric. The hyper-parameter space for each algorithm, together with the best combination for each label, are found from Table 6.1 to Table 6.4.

Table 6.1: Hyper-parameter space and best combinations according to threshold days for algorithm Decision Tree

		Decision Tree					
Hyper-param.	Search Space	7 days	14 days	21 days	28 days	35 days	42 days
max_depth	[3,21]	21	14	16	21	19	19
min_samples_split	[2,51]	2	20	21	51	2	2
min_samples_leaf	[1,21]	1	1	21	18	13	21
criterion	[gini,entropy]	gini	gini	entropy	gini	entropy	entropy
max_features	[auto,sqrt log2,None]	None	None	None	None	None	None

Table 6.2: Hyper-parameter space and best combinations according to threshold days for algorithm Random Forest

Random Forest							
Hyper-param.	Search Space	7 days	14 days	21 days	28 days	35 days	42 days
n_estimators	[50, 500]	237	50	440	344	451	500
max_depth	[3,21]	21	21	21	21	21	5
min_samples_split	[2,51]	2	2	16	51	23	46
min_samples_leaf	[1,21]	2	2	16	51	23	46
criterion	[gini,entropy]	gini	gini	entropy	gini	gini	gini
max_features	[auto,sqrt log2,None]	None	None	None	None	None	log2

Table 6.3: Hyper-parameter space and best combinations according to threshold days for algorithm XGBoost

XGBoost							
Hyper-param.	Search Space	7 days	14 days	21 days	28 days	35 days	42 days
n_estimators	[50, 500]	500	500	500	357	500	497
max_depth	[3,21]	19	13	21	15	17	14
learning_rate	[0.001,0.3]	0.300	0.300	0.010	0.207	0.110	0.230
min_child_weight	[1,11]	1	1	1	11	1	1
alpha	[0,10]	0	0	4.342	6.304	10	9.47
lambda	[0,10]	0	10	2.886	0	0	3.596
subsample	[0.8,1]	0.800	0.800	0.870	0.800	0.800	0.982
colsample_bytree	[0.8,1]	1	0.800	0.846	1	1	0.806

Table 6.4: Hyper-parameter space and best combinations according to threshold days for algorithm CatBoost

CatBoost							
Hyper-param.	Search Space	7 days	14 days	21 days	28 days	35 days	42 days
n_estimators	[50, 500]	482	125	349	245	272	262
max_depth	[3,16]	16	16	10	16	16	14
learning_rate	[0.01,0.3]	0.213	0.300	0.120	0.300	0.177	0.178
colsample_bylevel	[0.8,1]	0.828	1	0.827	0.810	0.996	1
reg_lambda	[0.001,100]	2.554	0.001	50.984	85.202	97.431	63.541
subsample	[0.8,1]	0.824	1	0.994	1	0.820	0.800

The goal is to ensure that each of these estimators can not only distinguish between classes but also be reliable in its decision-making. This balance is effectively addressed by the assessment metric, F1 Score. The results, based on this metric, for all the algorithms under analysis, are presented in Table 6.5.

Table 6.5: F1 Score achieved at each threshold days in the training process for each classification algorithm

	7 days	14 days	21 days	28 days	35 days	42 days
Decision Tree	0.078	0.400	0.653	0.709	0.741	0.766
Random Forest	0.056	0.436	0.664	0.716	0.752	0.765
XGBoost	0.151	0.463	0.658	0.724	0.753	**0.781**
CatBoost	0.047	0.423	0.654	0.722	0.751	0.778

Among these algorithms, XGBoost demonstrates the highest and most consistent performance, obtaining the maximum F1 Score of 0.781 at the threshold of 42 days. This indicates that XGBoost is the most effective in accurately predicting the target class. The trend of increasing F1 Scores as the threshold days increase for most algorithms implies that a larger ratio of Fast Delivery cars within the dataset is beneficial for improving predictive performance. It is worth mentioning that while XGBoost achieves the highest F1 Score, the other algorithms also perform competitively and consistently well. All values are in the same order of magnitude as XGBoost, including in this batch Decision Tree. It performs better than Random Forest and/or CatBoost in the first and last threshold, despite its simplicity and limitations.

The interpretability of the results provided by the best estimator has been achieved by comparing two techniques: feature importance and SHAP values. The outcomes from these two procedures are illustrated in Figure 6.1. It is not a surprise that the Private Customers tag is the most relevant attribute by far to identify a car within the category of Fast Delivery. Actually, these are cars Build-to-Order. The time within the compound will be minimal as there is already a customer waiting for the automobile. In the second place of the ranking, with one order of magnitude less than the predecessor, Dealerships Stock appears. Cars belonging to this Order Type are more likely to be categorized as Normal Delivery. With respect to the significance of the compound region, choosing CMC as the final destination has more weight than the other options to be selected as Normal Delivery. It is the location with the longest average and median period of days within the compound (see Table 3.5). Additionally, in the winning threshold days, it is the compound with the lowest percentage of Fast Delivery cars (see Table 4.1). Finally, with respect to the elements that build the car attribute, the most relevant TRIMs are Style and Reference, i.e., the cheapest equipment levels; the most relevant color is the one designated as B4B4; the first car models to appear in the list are SEAT Arona and SEAT Leon 5D. In terms of engines, DS8 and D33 are the earliest options to appear. However, the relevance of these attributes is small paying attention to the SHAP values. To comprehend the distinction between both approaches, it is important to remember that SHAP values assess the influence of a feature on predictions, while features importance estimates the impact of a feature on model fit.

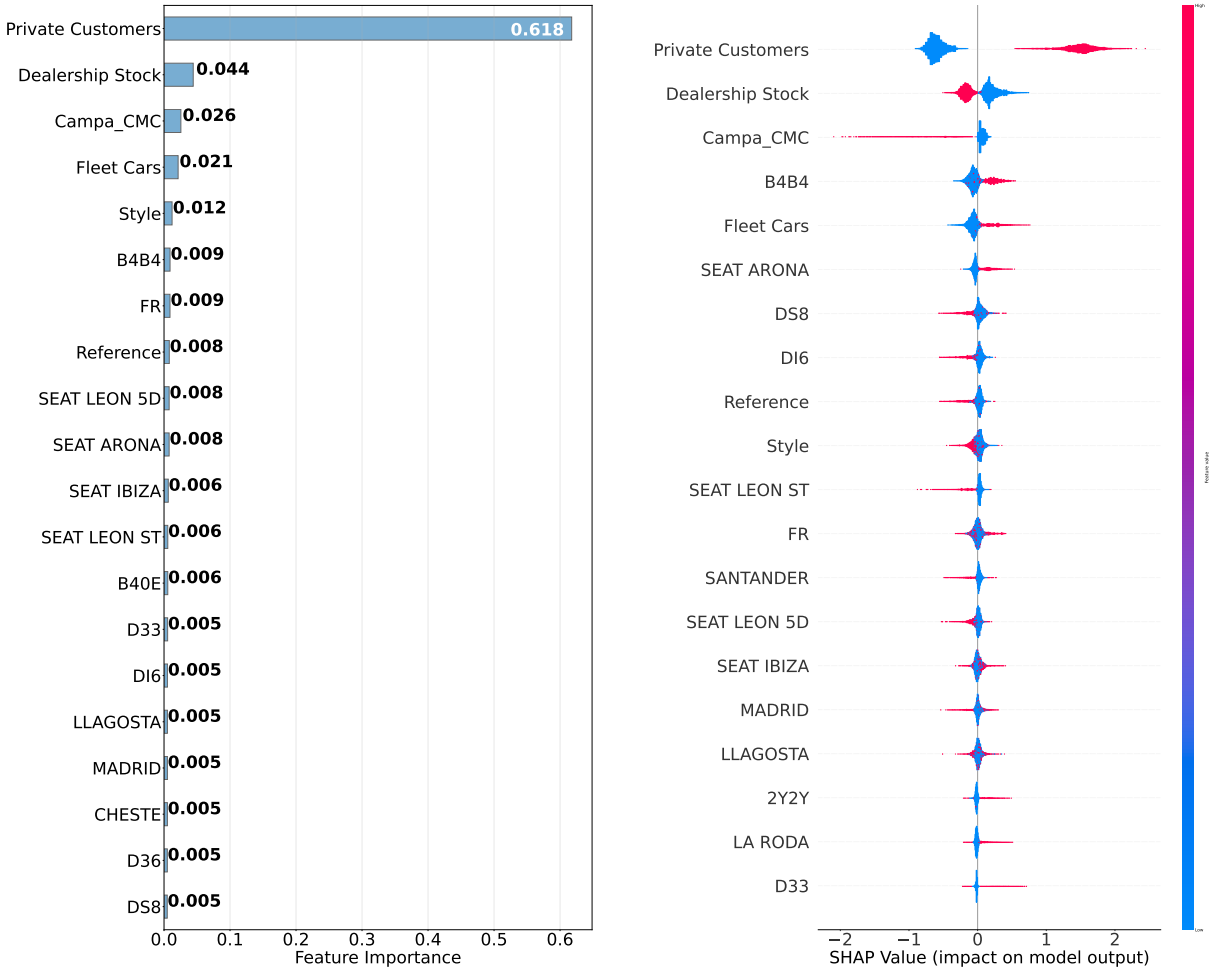


Figure 6.1: Analysis of feature relevance for the best estimator done with feature importances and with SHAP values.

6.1.2 Benefits Of The Reallocation Strategy

It is presented the confusion matrix derived from the best estimator in Figure 6.2. The rows represent the predicted class labels, while the columns represent the actual class labels. True Positives are the dominant category, whilst True Negative occupies the second place. Nevertheless, the estimator has more preference for classifying Fast Delivery cars as Normal Delivery type, rather than the opposite. This trend is favorable. There are more actual Fast Delivery cars occupying the slot of Normal Delivery cars than the other way around. As every day in the compound region has associated logistic costs, it is preferable the misclassification in this direction. They are car variants still attractive to customers.

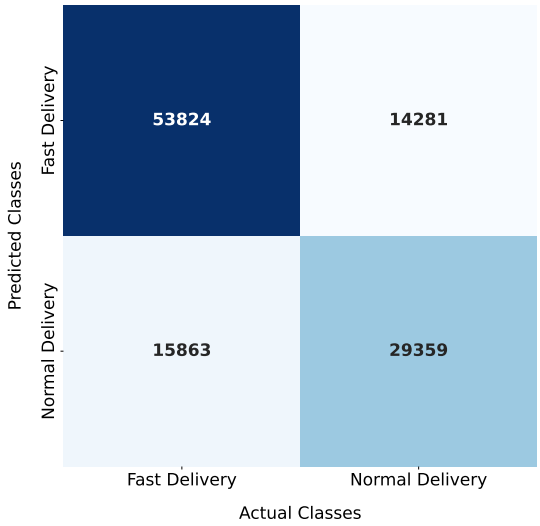


Figure 6.2: Confusion matrix from the best estimator

Based on the discoveries about the interpretability of the model, it is worth exploring the performance of the estimator per Order Type. Figure 6.3 shows the individual confusion matrices for each category. In addition, Table 6.6 collects four evaluation metrics related to each single confusion matrix, and contrasts them with the general outcomes. The best and worst performances, based on the F1 Score criteria, correspond to Private Customer and Dealership Stock, respectively. Especially, the latter category tends to classify cars as Normal Delivery. For the other subgroups, the number of vehicles that are actual Normal Delivery but are classified as Fast Delivery is superior than the inverse scenario.

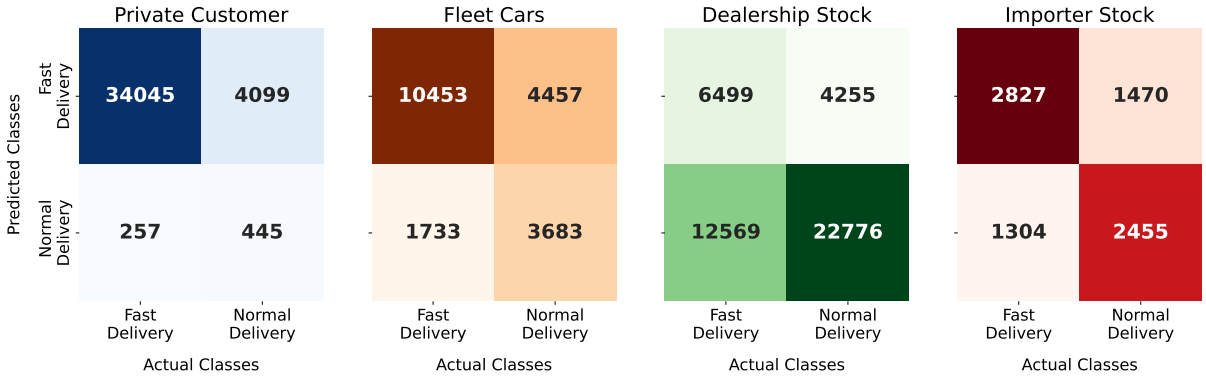


Figure 6.3: Confusion matrices from the best estimator for each one of the classes within the Order Type feature

Table 6.6: Main performance statistics from the confusion matrices derived from the best estimator

	General	Private Customer	Fleet Cars	Dealership Stock	Importer Stock
Accuracy	0.734	0.888	0.695	0.635	0.656
Recall	0.772	0.993	0.858	0.341	0.684
Precision	0.79	0.893	0.701	0.604	0.658
F1 Score	0.781	0.94	0.772	0.436	0.671

Therefore, a point of improvement is detected, as there are cars that based on their configurations are Normal Delivery but they can become Fast Delivery. Nearly 40% of observations can benefit from it. That's why it is pursued to proceed with the reallocation strategy. The best-trained classification system is used to assign an alternative destination in the vehicles that were labeled as Normal Delivery. The attempt is to transform a Normal Delivery car into a Fast Delivery type by changing the compound destination of the vehicle. It was a successful trial for 22301 out of 45222 classified as Normal Delivery cars, having one alternative compound destination, at least, half of the sample. More details are placed in Figure 6.4.

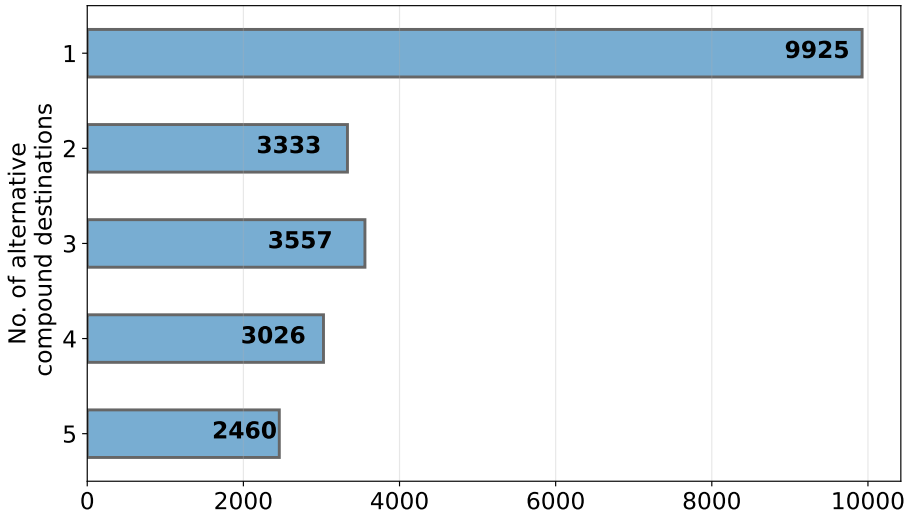


Figure 6.4: Number of labeled Normal Delivery cars updated to Fast Delivery type per number of alternative compound regions available

It follows the analysis at the compound destination level. The comparison between the before and after the reallocation strategy is presented in Table 6.7. As it is expected, the number of cars per compound has suffered modifications and does not respect the total quantity of vehicles within the dataset. We do not discriminate when there is more than one alternative compound destination for the car variant. That's why there are duplicates. Paying attention to CHESTE, the metrics after reallocation do not imply an improvement in the operation of the compound. However, the maximum number of days waiting in the compound has varied positively. On the other side, the median of the cars headed to the CMC compound region has been reduced thanks to the reallocation, despite the average has increased in once of the scenarios. The opposite case occurs in LA RODA, where the average Time in Compound has diminished. Until this point, the reallocation strategy is globally comparable to the decisions taken by the experts of the company. On the contrary, the compounds from MADRID, LLAGOSTA, and SANTANDER experiment an improvement both in average and median in days waiting in the compound.

Table 6.7: Main descriptive values, before and after reallocation strategies, for Time in Compound per each compound region individually. Reallocation A refers to the approach without new Time in Compound computation for the vehicles. For Reallocation B, the Time in Compound for the vehicles with new destination has been estimated from the existing time distribution in that region

	CMC			MADRID		
	Original	Reallocation A	Reallocation B	Original	Reallocation A	Reallocation B
Min	1	1	0	1	1	1
Mean	62	64	46	54	46	36
Std. Dev.	53	57	36	58	50	34
Q1	26	24	24	14	14	14
Q2	46	43	37	29	25	25
Q3	82	88	52	71	59	44
Max	716	716	716	447	554	377
No. of Cars	8670	11153	11153	24526	21355	21355
	LA RODA			CHESTE		
	Original	Reallocation A	Reallocation B	Original	Reallocation A	Reallocation B
Min	1	1	1	1	1	1
Mean	60	56	37	52	56	39
Std. Dev.	64	53	31	57	54	33
Q1	16	17	17	14	16	16
Q2	34	37	31	27	37	32
Q3	82	77	44	71	78	47
Max	516	516	516	490	470	335
No. of Cars	16608	19027	19027	14216	21313	21313
	LLAGOSTA			SANTANDER		
	Original	Reallocation A	Reallocation B	Original	Reallocation A	Reallocation B
Min	1	1	1	1	1	1
Mean	49	42	35	54	48	37
Std. Dev.	55	45	34	56	46	30
Q1	14	14	14	18	18	18
Q2	25	24	24	30	29	28
Q3	62	54	43	69	62	41
Max	470	461	461	554	447	397
No. of Cars	31874	29861	29861	17433	17062	17062

6.2 Car Configurator Webpage As A Reliable Source

This section attempts to verify if data collected from the the car configurator is useful and reliable to capture the interest of potential customers (see Section 4.2). Therefore, the analysis will count on the clickstream data and the sales record of the company. The first step within the study includes measuring the correlation between these two datasets., pursuing to understand the starting of the customer's exploration phase. Afterward, taking advantage of these outcomes, the timeframe will be divided into time chunks, in which the next step of the research will take place. The second phase is oriented to perform demand prediction with and without being assisted by the information gathered by the car configurator. Finally, the assessment of the results, by means of the comparison with respect to the real weekly mix sales, will permit to validate or reject the contribution of the online source.

6.2.1 Correlation Analysis

The time series of the sales record and the clickstream data will be shifted during 52 points. It is attempted to discover the lag between the start of the exploratory phase and the purchase moment. Firstly, it is computed individually per car model. Afterwards, seeking to confirm the outputs, the correlation is performed at the car variant level. In other words, the conjunction of the car model together with color or the compound destination.

Regarding the most aggregated level, the results in Figure 6.5 show that a positive correlation exists for all car models under analysis. It is computed in the form of Pearson correlation coefficient (PCC). Although it does not have the strength it would be expected. None of our four car models reaches a peak close to the unit, being SEAT Arona the one with the largest values. However, it is possible to extract one conclusion. For all car models, the largest correlation is within the first half of the shifting period, as well as the rest of the top five largest points. The unique exception is for SEAT Leon ST, where one of these top five points occurs at the 28th shifted week. Hence, we conclude that purchase likelihood increases within a period of up to 6 months after visiting the Car Configurator webpage.

For the car variant's granular level, results regarding car model and exterior color are displayed in Figure 6.6. They are averaged along the entire lagging period among the set of car variants. At this granular level, the behavior of PCC is similar to the previous one. Correlation is stronger in the first half of the shifting period than in the second half. However, larger values are reached than at the previous granular level, meaning a stronger correlation. The same study is performed at the car model and compound region level. It is illustrated in Figure 6.7. The average value is lower than for the other car variants. The exception is SEAT Arona, in which values are similar. On the contrary, the standard deviation has diminished as well. Therefore, the behavior along the different compound regions is more consistent than among the different colors of each car model. Finally, despite three out of the five largest values of SEAT Leon ST peak occurring in the second half, the pattern is repeated. The correlation exhibits greater strength during the initial half of the shifting period compared to the latter half. To understand more in detail about the lagged correlation of each car variant, see Annex A. Therefore, the time series will be divided into five time chunks of six-month size, where the last month and a half defines the test period.

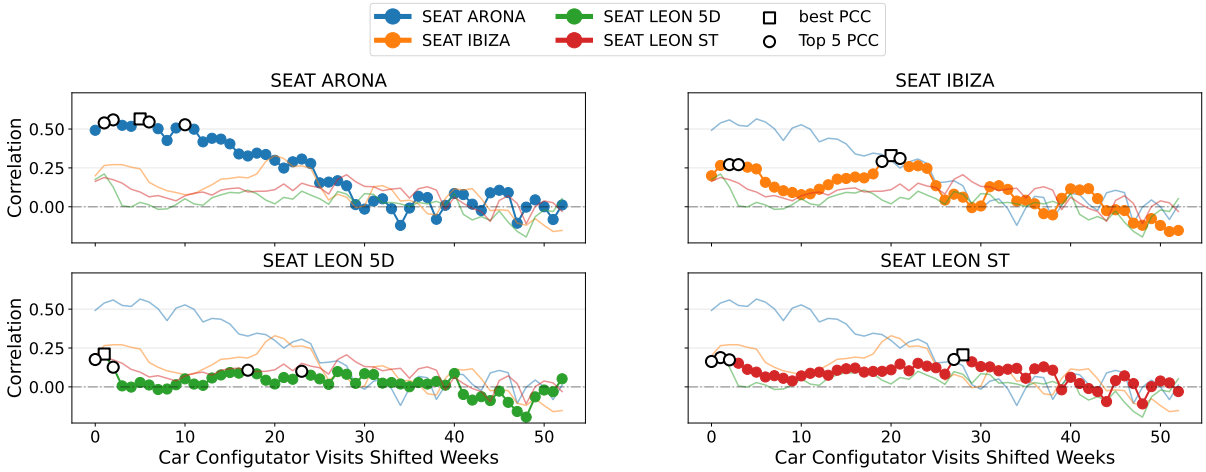


Figure 6.5: Pearson correlation coefficient (PCC) after shifting Car Configurator webpage visits time series over sales time series. A square mark signals the largest positive PCC. Circle marks point the rest of top 5 largest positive PCC.

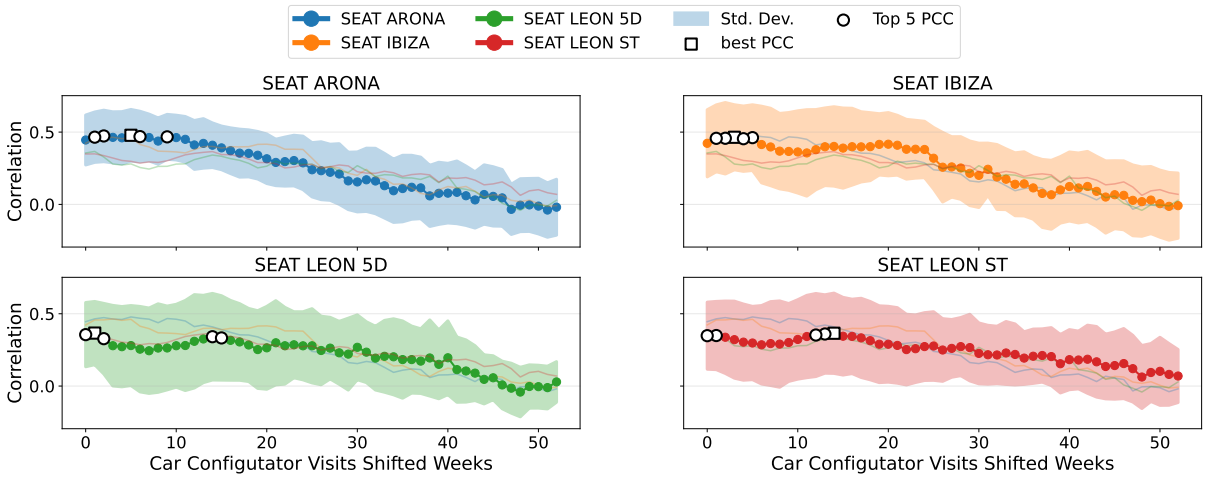


Figure 6.6: Average Pearson correlation coefficient (PCC) at car model and exterior color level. Thicker line represents the average correlation value per lagged week among the car variants. Shadow area symbolizes the standard deviation. A square mark signals the largest positive PCC. Circle marks point to the rest of top 5 largest positive PCC.

6.2.2 Forecasting Performance

For each one of the aforementioned time chunks, the sales volume for all the car variants will be predicted. The forecast will be performed using well-known machine learning algorithms, i.e., ARIMA(X) and XGBoost, both in univariate (only sales record) and multivariate (sales record and clickstream data) modalities.

The time chunk structure is as follows. As the largest correlation points occur within the first part of the lagging interval, the training epoch of these periods will last for six months, i.e., 24 weeks. The next six weeks are used as test data. Predictive algorithms will be evaluated on them. Therefore, there are 5 groups of up to 30 weeks. The test phase of the last time chunk counts with 5, rather than 6 weeks, as there is only information of 149 weeks. Additionally, with the time chunks division, it is intended to face all the stages of the product life cycle: introduction, growth, maturity, and decline. The group of time chunks is illustrated in Figure 6.8.

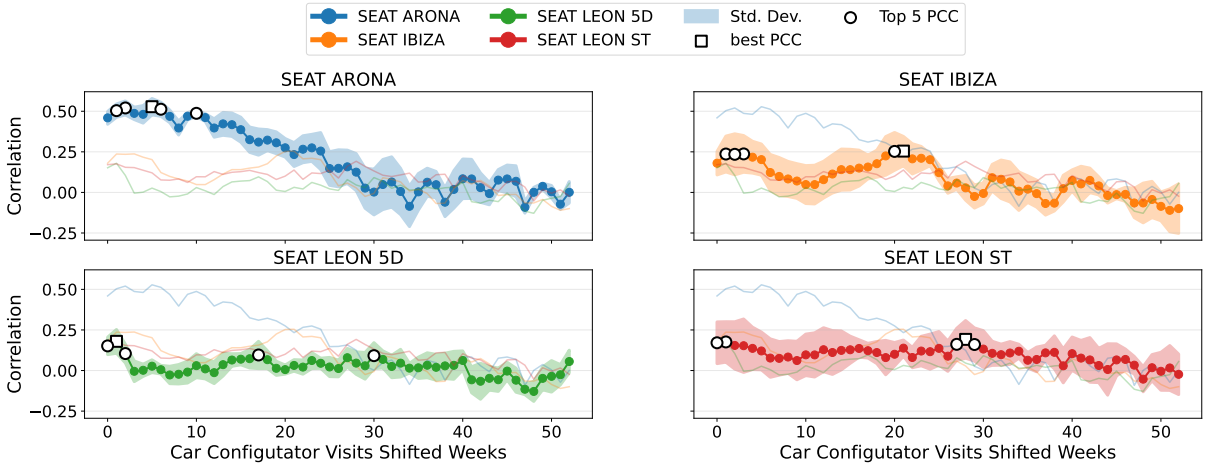


Figure 6.7: Average Pearson correlation coefficient (PCC) at car model and compound region level. Thicker line represents the average correlation value per lagged week among the car variants. Shadow area symbolizes the standard deviation. A square mark signals the largest positive PCC. Circle marks point to the rest of top 5 largest positive PCC.

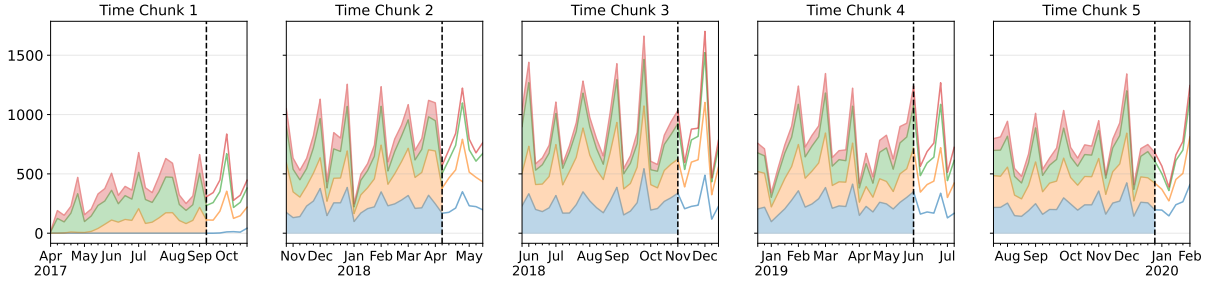


Figure 6.8: Example of the five time chunks the data has been divided. Colored area represents the training epoch. Colored lines symbolize the test phase.

Consequently, the first step consists of assessing the different forecasting techniques in terms of MAE. The decision to utilize MAE is based on its resistance to outliers that may be present in the sales records of each variant. To illustrate the procedure, there is in Figure 6.9 the comparison of the predicted sales record against the actual one in each forecasting algorithm. It reflects the best-seller car variant at the exterior color level. It is referred to as SEAT Ibiza and the color named B4B4 for the third time chunk. The car variant's sales were 1165 units during the test period associated with this time chunk. It lies from the week of 11^{st} November to the week of 16^{th} December 2018. The best technique is XGBoost Multivariate. The MAE of each car variant and forecasting method can be found in Annex B.

The error analysis is extended to encompass the entire dataset. It is executed for both types of car variants. Figure 6.10 collects the results for car model and exterior color, whilst car model and compound region is shown in Figure 6.11. The largest variability is observed for the exterior color car variants, rather than at the compound region level. Nevertheless, the lowest MAE in order of magnitude corresponds to the second time chunk of the SEAT Leon ST. Additionally, the algorithm families follow similar behavior, performing better the gradient boosting type. At these levels, the previous pattern is repeated. Multivariate techniques provide the best outputs. It opens a path to consider Car Configurator webpage data as reliable information.

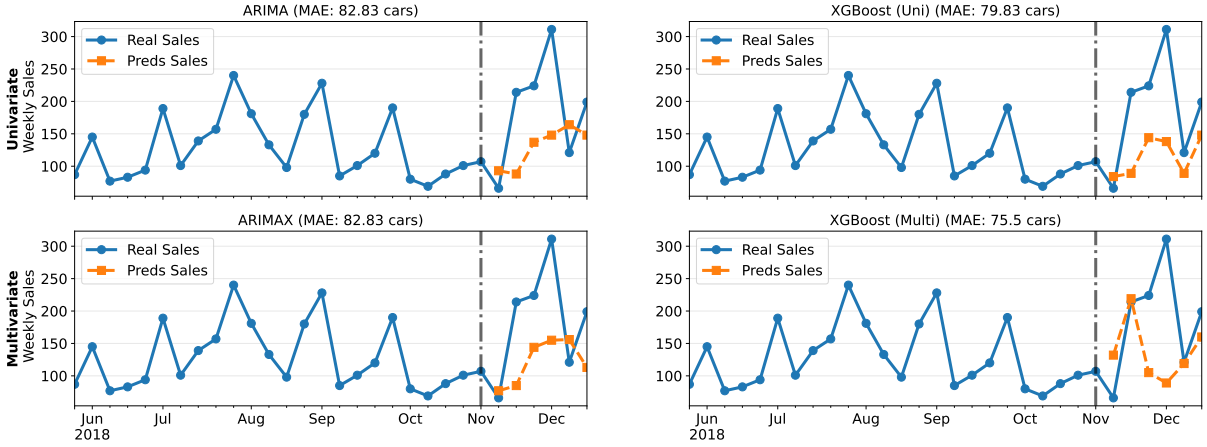


Figure 6.9: Sales predictions obtained for the best seller car variant at third time chunk with the different forecasting techniques.

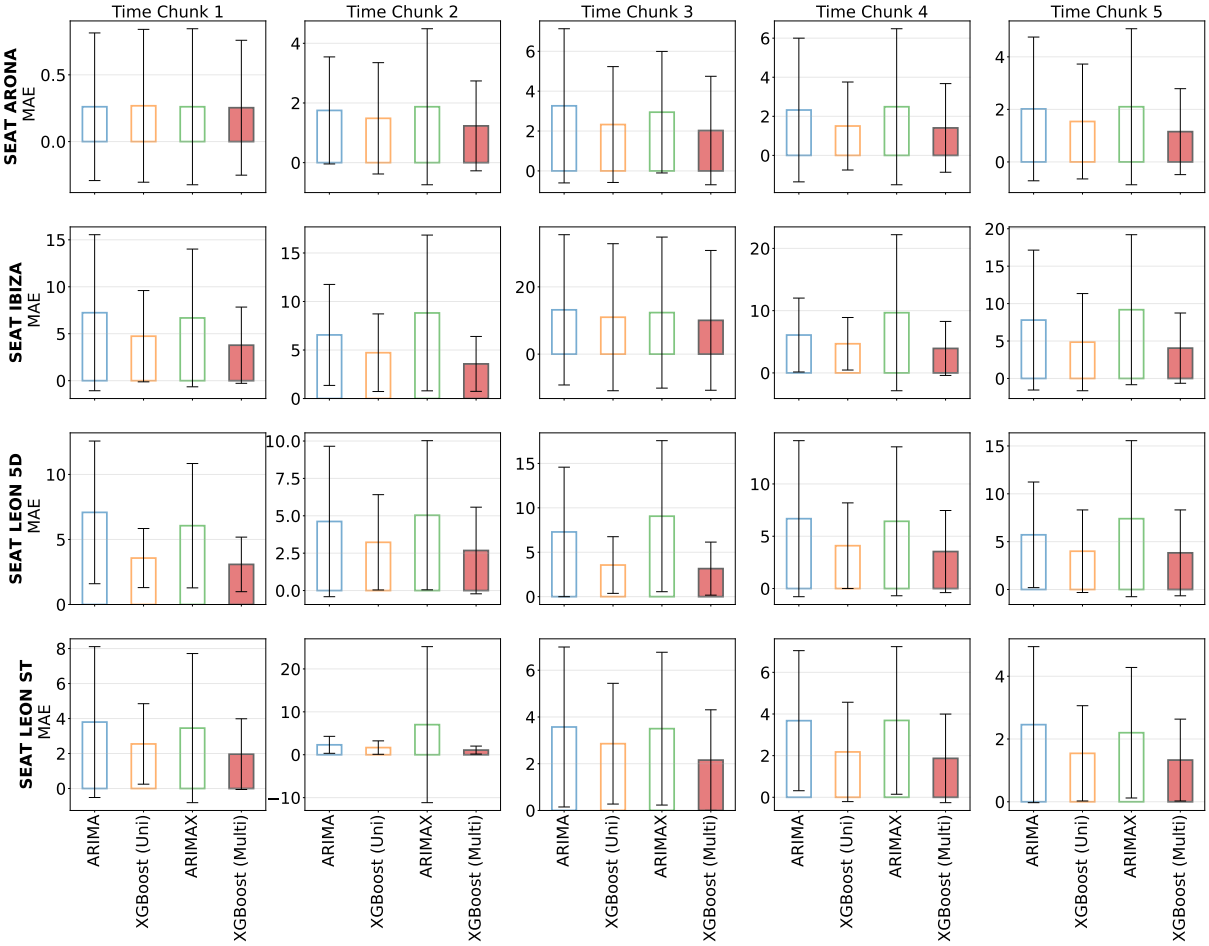


Figure 6.10: Averaged MAE per car variant (car model plus exterior color) and time chunk of each forecasting technique. The colored bar indicates the technique with the best metric. Whiskers represent standard deviation of the metric.

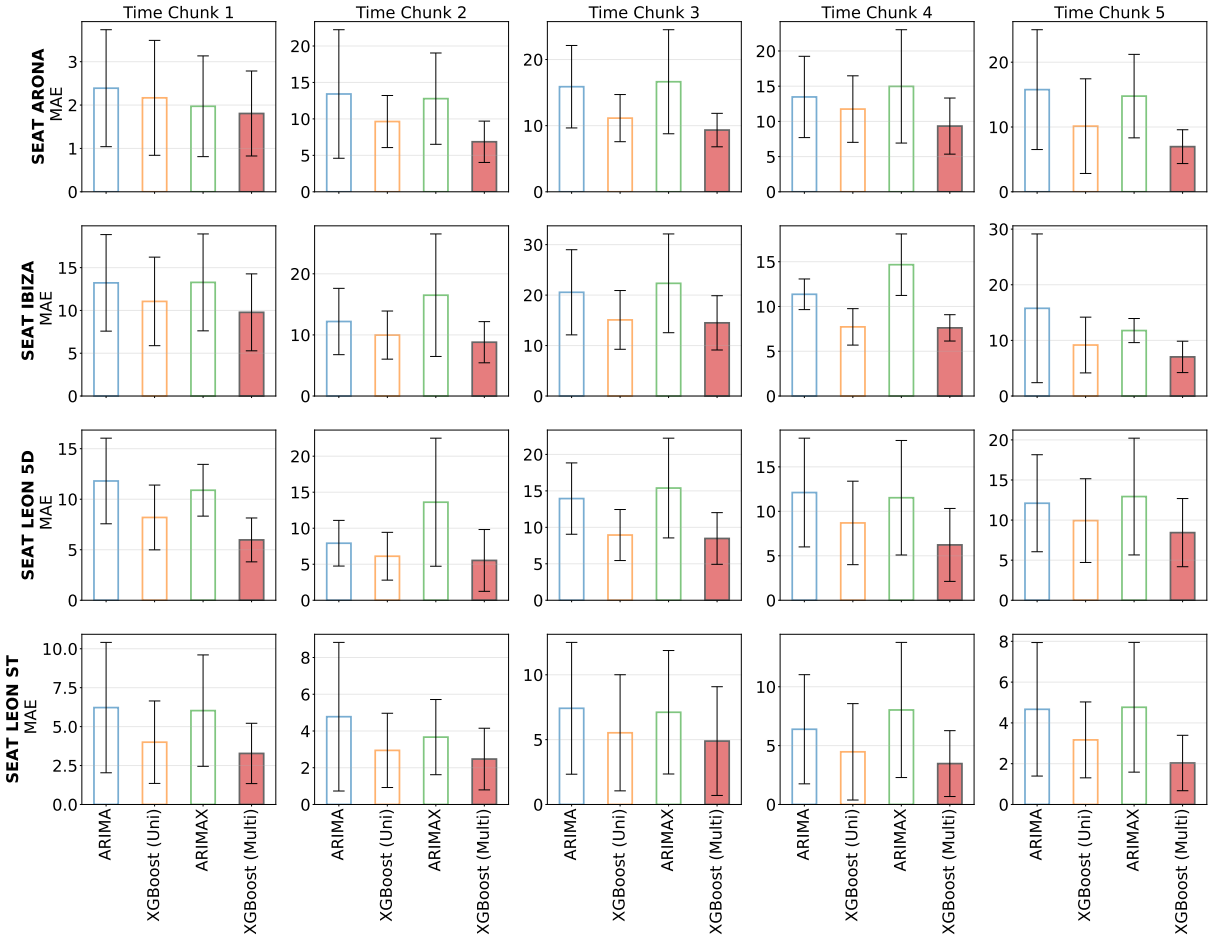


Figure 6.11: Averaged MAE per car variant (car model plus compound region) and time chunk of each forecasting technique. The colored bar indicates the technique with the best metric. Whiskers represent standard deviation of the metric.

6.2.3 Weekly Mix Sales Assessment

After completing the forecasting performance tests, it is proceeded with the evaluation of weekly mix sales. This stage is crucial for validating the accuracy of the previous forecasts. It is found that data from the automotive brand's website consistently proves to be a dependable and trustworthy source. The evaluation process follows the same structured approach as before. Firstly, the metric is exemplified with a car variant. Subsequently, the performance assessment is extended to encompass the entirety of the available data.

The chosen car model is the top seller. In other words, SEAT Ibiza at the third time chunk. Figure 6.12 illustrates the weight each color had on the sales of the first week of the test period. It is observed that one out of every two cars is painted in color B4B4. Therefore, these proportions are compared with the ones derived from each forecasting method. They are presented in the lower grid. The comparison between real and forecast mix is executed thanks to R2 Score. In this scenario, the univariate techniques have a great performance. Nevertheless, the winning method belongs to the multivariate category. ARIMAX outputs present the largest similarity between prediction and reality. On the contrary, despite its good performance in terms of MAE, the outcomes from XGBoost multivariate are the lowest. Nevertheless, this is a representation of a single case, which is why it is important to proceed with the assessment methodology. The R2 Score of each car variant, week, and forecasting method can be found in Annex C.

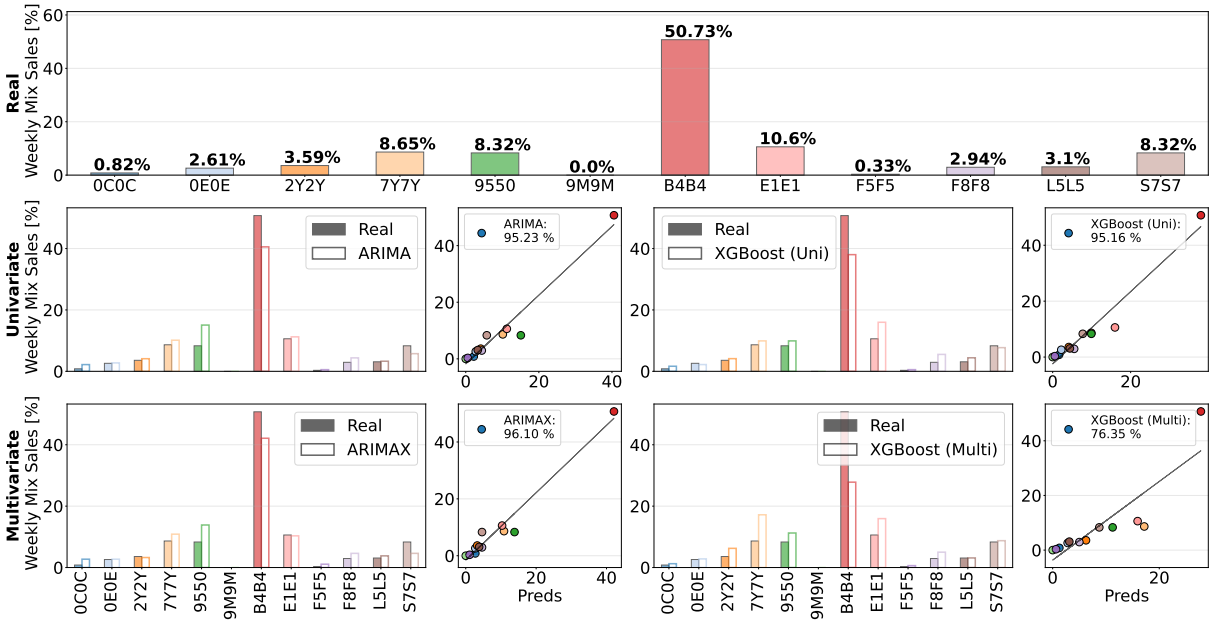


Figure 6.12: Real weekly color mix sales (upper), forecast ones and assessment in the form of R2 Score (lower grid) for SEAT Ibiza in third time chunk.

Therefore, the averaged performance is computed over the total length of weeks and time chunks of the different forecasting techniques. This stage is shown in Table 6.8 regarding exterior color attribute, and in Table 6.9 for compound region level. The leading technique corresponds to the XGBoost in its variant multivariate in both car variants' granular levels. Additionally, the univariate mode of the gradient boosting method consistently delivers the best outputs from its group. The similarity between predictions and reality is larger when it is assessed for the exterior color attribute. Nevertheless, SEAT Arona is the car model with the lowest performance. The launching phase impacts its outcomes.

Table 6.8: Average R2 Score (%) of each forecasting technique for the weekly sales mixes of each car model at exterior color level over the total size of time chunks of the dataset. Bold text signals the largest value

	SEAT Arona	SEAT Ibiza	SEAT Leon 5D	SEAT Leon ST
ARIMA	74.15 ± 32.65	91.07 ± 7.61	80.97 ± 13.26	71.48 ± 18.25
XGBoost (Uni)	75.9 ± 34.13	93.68 ± 4.32	88.76 ± 8.43	82.21 ± 12.82
ARIMAX	67.2 ± 34.62	82.73 ± 18.26	77.22 ± 14.41	71.59 ± 20.01
XGBoost (Multi)	**78.87 ± 31.38**	**94.5 ± 5.93**	**92.42 ± 4.14**	**86.47 ± 12.91**

Table 6.9: Average R2 Score (%) of each forecasting technique for the weekly sales mixes of each car model at compound region level over the total size of time chunks of the dataset. Bold text signals the largest value

	SEAT Arona	SEAT Ibiza	SEAT Leon 5D	SEAT Leon ST
ARIMA	47.72 ± 33.6	60.44 ± 25.44	67.21 ± 23.24	71.94 ± 22.94
XGBoost (Uni)	49.66 ± 31.13	72.94 ± 17.52	69.46 ± 22.0	80.74 ± 17.98
ARIMAX	55.98 ± 34.26	52.19 ± 24.98	57.72 ± 27.66	72.25 ± 22.03
XGBoost (Multi)	**63.74 ± 33.71**	**75.43 ± 19.8**	**77.84 ± 18.5**	**83.86 ± 19.54**

In the second phase, assessment occurs at the time-chunk level. The outcomes of each forecasting technique are averaged over this time level. The intention is to gain more details about the performance. This behavior is displayed in Figure 6.13 and Figure 6.14 for color and compound region granularity levels, respectively. In the first element, XGBoost multivariate clearly dominates the best outcomes per time chunk and car model. There is a single exception in the third time chunk of the SEAT Ibiza. In this frame, XGBoost univariate is ahead of the other algorithms. In the compound destination case, the dominance of XGBoost multivariate is not so wide, but it is for the multivariate algorithms in general. Nevertheless, univariate techniques lead in some time chunks of SEAT Ibiza and SEAT Leon ST.

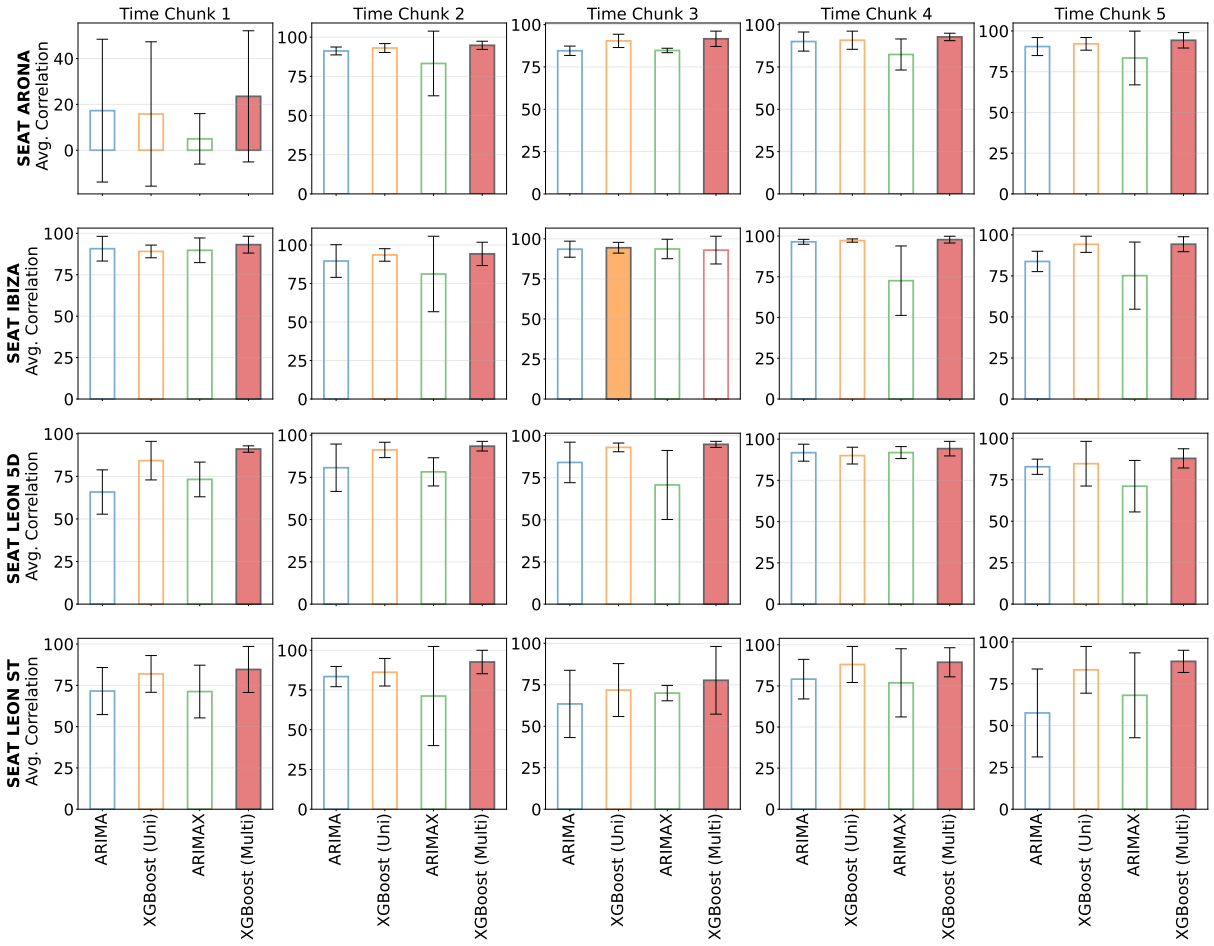


Figure 6.13: Average R2 Score (%) of each forecasting technique for the weekly sales mixes of each car model at exterior color attribute over each chunks of the dataset.Colored bars represent the forecasting algorithm with the largest metric.

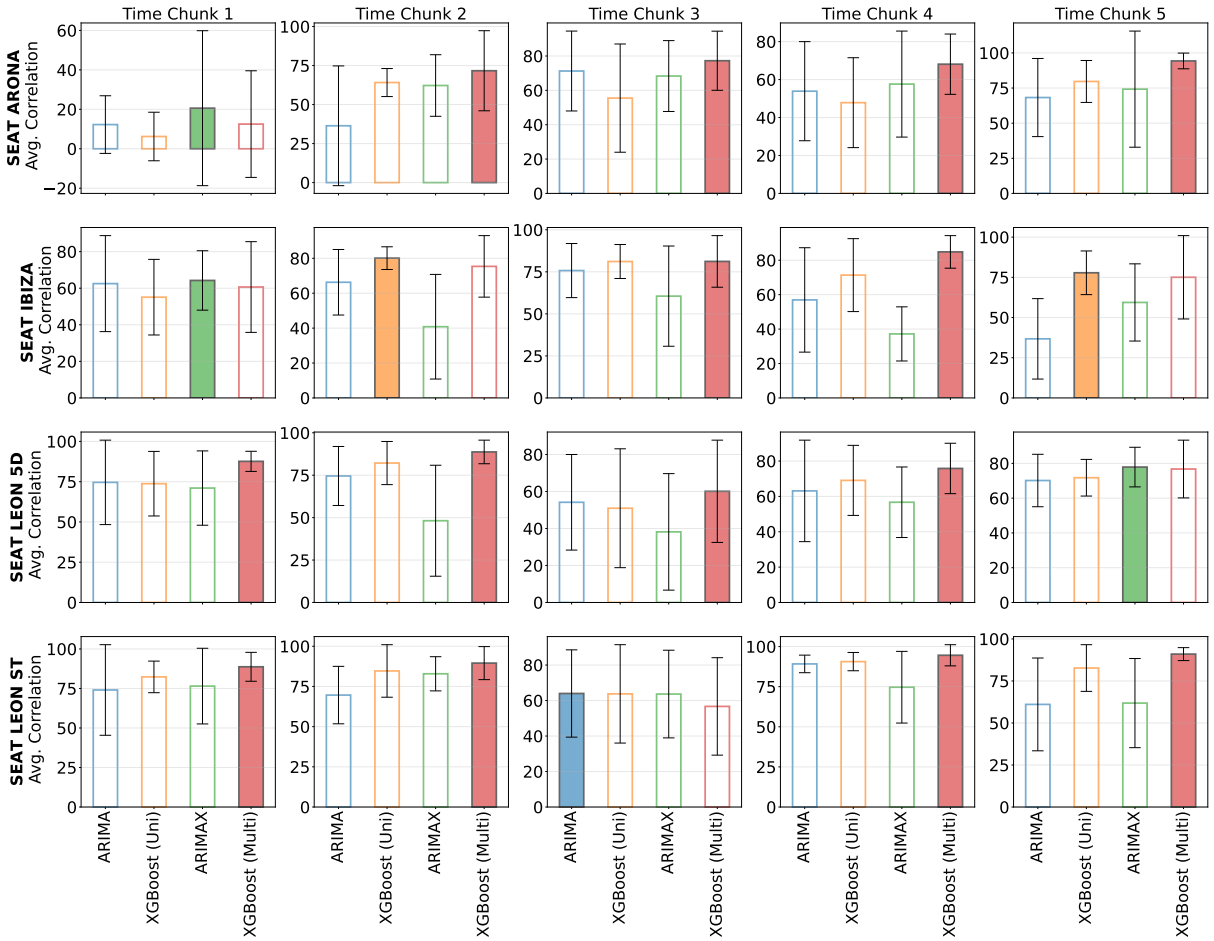


Figure 6.14: Average R2 Score (%) of each forecasting technique for the weekly sales mixes of each car model at compound region level over each chunks of the dataset. Colored bars represent the forecasting algorithm with the largest metric.

The evaluation of weekly color mix sales finishes with the third step. The results for exterior color and compound region car variants levels are summarized in Figures 6.15 and 6.16, respectively. Regarding the color level, there are two scenarios. In the first one, XGBoost multivariate consistently delivers the largest metric along the test period of the time chunk. In the other scenario, there is an even between the gradient boosting algorithms. Nevertheless, the multivariate option is always included. On the contrary, the case at the compound region level presents more variety. All car models have time chunks on which one of the univariate techniques outperforms. However, it is also valid to say that multivariate techniques are more regular in their behavior. It is the leader or it participates in the tie of the algorithms providing the largest metric for most weeks within the test periods.

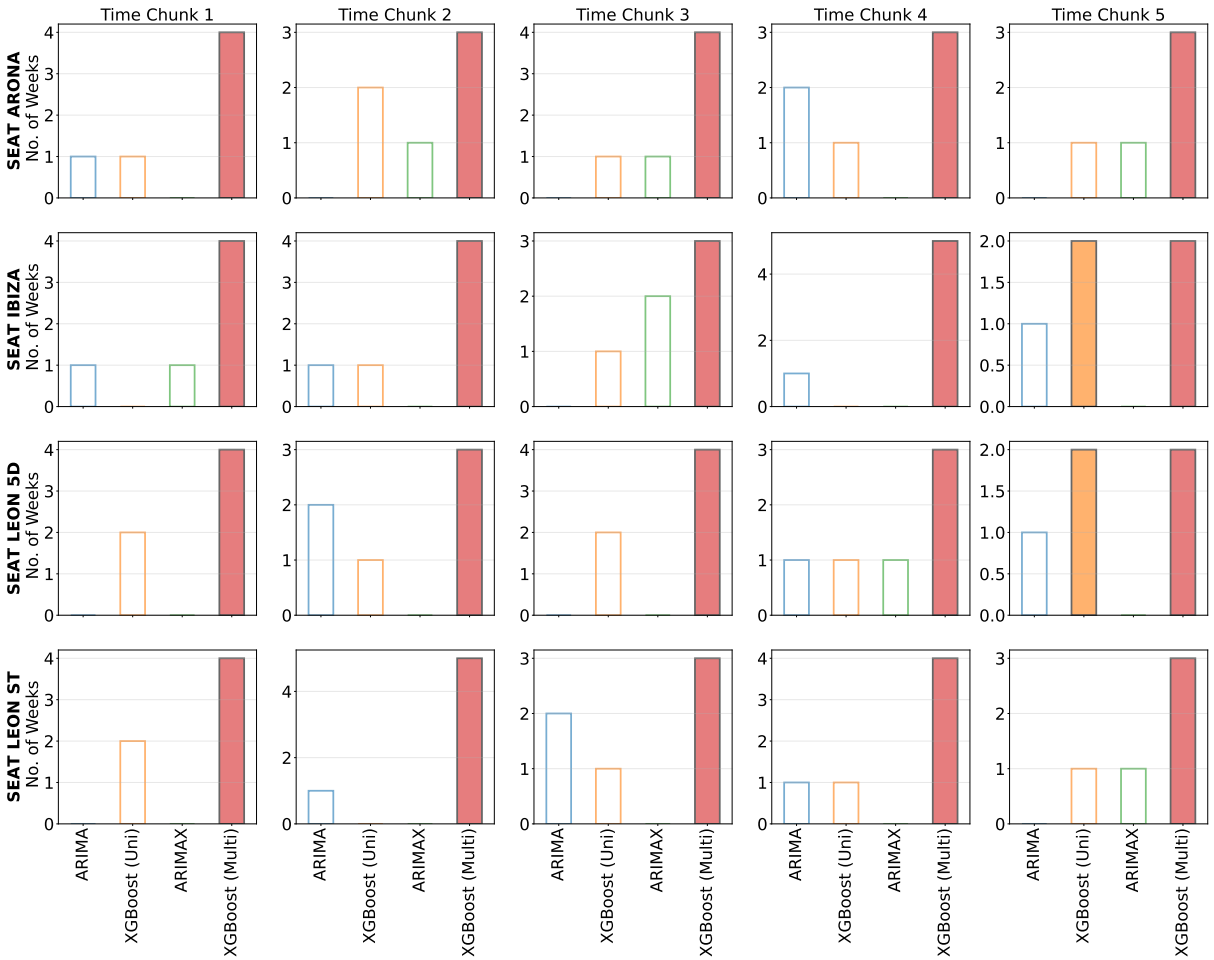


Figure 6.15: Count of what is the forecasting technique that provides the best R2 Score each week of the test period within each time chunk of the dataset for each car model and exterior color attribute. The technique(s) with the largest number of weeks is colored.

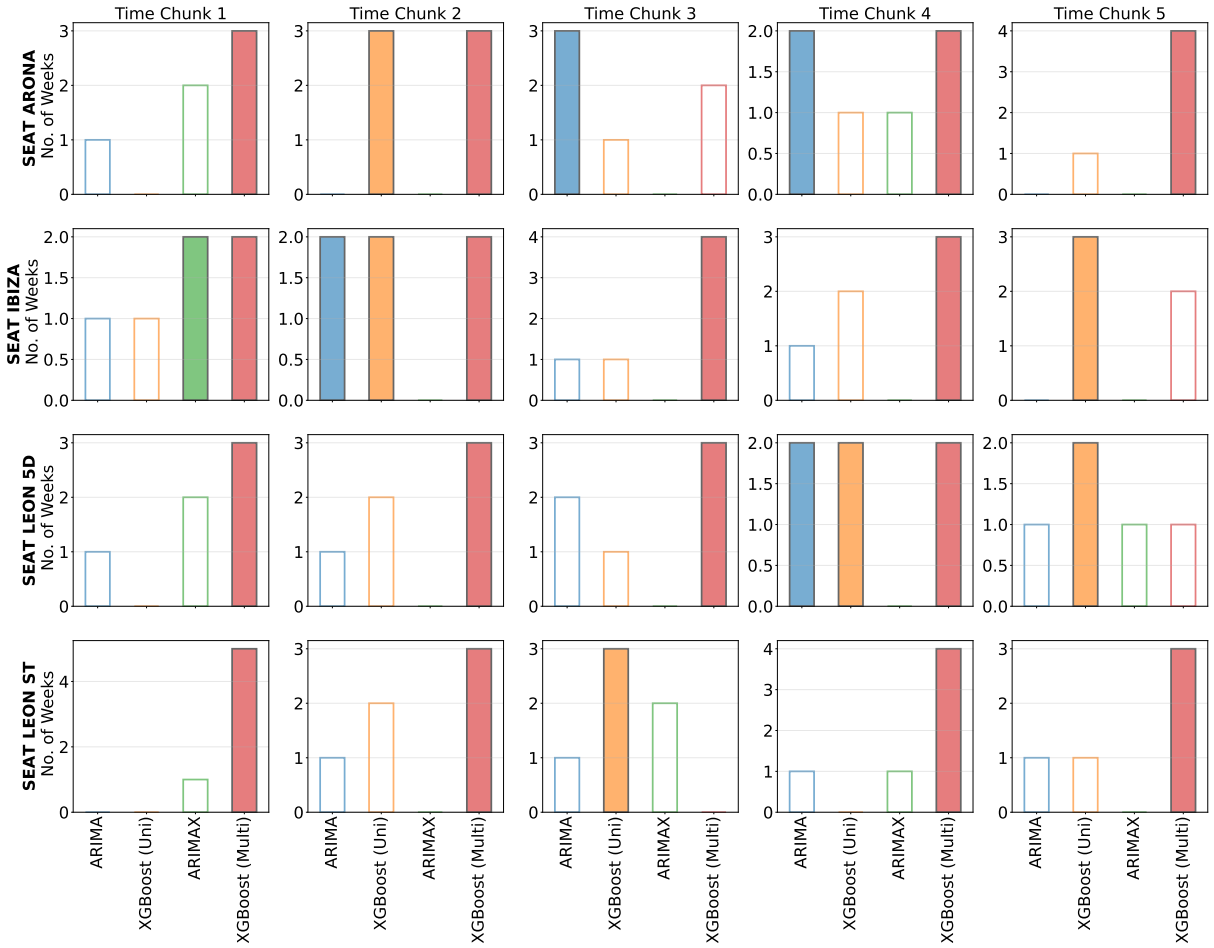


Figure 6.16: Count of what is the forecasting technique that provides the best R2 Score each week of the test period within each time chunk of the dataset for each car model and compound region level. The technique(s) with the largest number of weeks is colored.

6.3 Quantitave Reduction Of Car Configurator Data

The properties of the Car Configurator webpage cause the quantity of data generated by the online service to dramatically boost. Despite the dataset having been processed and cleaned, problems bonded to Big Data environments are still likely, such as noisy and corrupted data. This section presents the results achieved by reducing the volume of clickstream data without compromising significance. The latter is defined as the correlation between Car Configurator webpage visits and company sales. The data disminishing is ruled by a set of filtering rules, based on outliers detection within users' quantitative activity, such as *Number of Car Variants* and *Time Between Connections*. The summary of the filtering sequence can be recovered (see Figure 4.3 and Section 4.3).

6.3.1 Comparison of significance: benchmark vs filtering rules

On one hand, Figure 6.17 presents the R2 Score measuring the correlation between plain Car Configurator data and 8-week lagged sales records. The horizontal axis represents the date on which the correlation calculus took place. Additionally, for the same time period, they are overlapped the outcomes related to each filtered CC data. Nevertheless, the overlapping is not exact in all the periods from the time range. It is distinguished a gap happening in November 2018 for the four cases. Filtered CC data provide larger significance than raw Car Configurator data does. The opposite occurs in February 2019, when raw Car Configurator data gives the largest values.

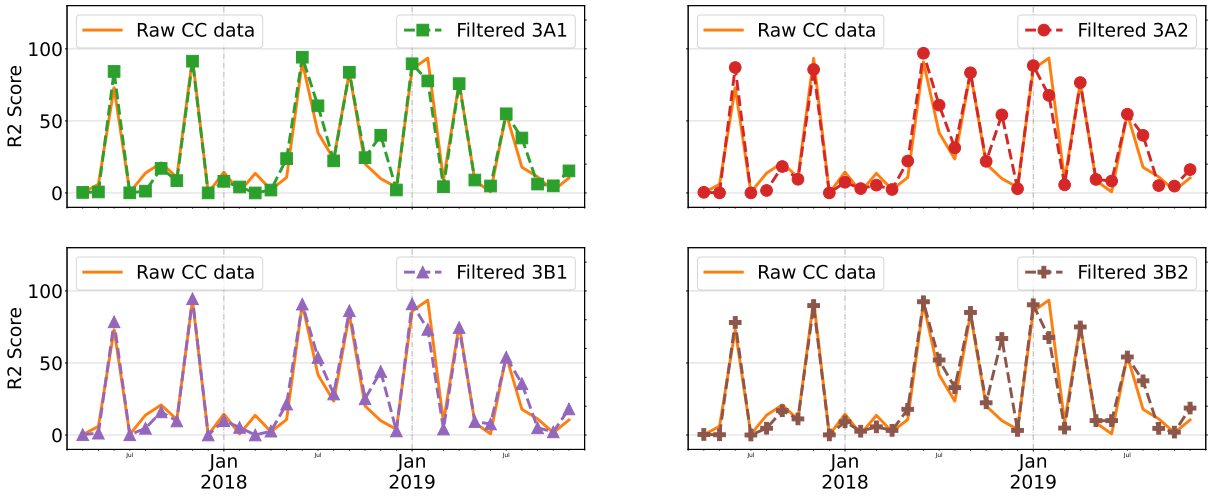


Figure 6.17: Monthly significance value between visits to Car Configurator webpage and lagged sales record. Comparison of the outcomes attained by raw Car Configurator data (orange line) and each one of the filtering rules applied to Car Configurator data (dashed-dotted lines)

Thus, in order to gain precision in the comparison of results, the main statistical values can be found in Table 6.10. It collects the average, the standard deviation as well as the minimum and maximum values from each individual R2 Score. They are computed as supportive variables in an attempt to detect anomalies between series. All of them lay in the range given by the raw clickstream. However, we need to summarize the comparison into a single value. The elected magnitude is the *p-value* from the statistical Kolmogorov-Smirnov test between benchmark and filtering rules' outcomes. The four filtering rules

have proved to be as good as unprocessed Car Configurator data in terms of significance with a lagged sales register. Nevertheless, it is difficult to choose which one is the best. On one side, Filtering Rule 3A2 provides the largest average with not one of the largest standard deviations. On the contrary, the filtering rule most similar to raw CC data is Filtering Rule 3B2, which has the largest *p-value*.

Table 6.10: Main statistical values of the Significance comparison between Raw CC data and Filtered CC data. Column *p-value* collects the outcome of statistical Kolmogorov-Smirnov test between Plain Car Configurator data and filtered Car Configurator data.

	Average	Standard Deviation	Minimum	Maximum	p-value
Plain Car Configurator	28.28	32.94	0.02	93.60	–
Filtering Rule 3A1	29.69	33.73	0.00	94.21	0.838
Filtering Rule 3A2	30.37	33.24	0.03	96.95	0.838
Filtering Rule 3B1	29.72	32.98	0.00	94.59	0.838
Filtering Rule 3B2	30.29	32.90	0.00	92.47	0.968

6.4 Qualitative Filtering Of Car Configurator Data

The challenge of discriminating between users of the Car Configurator webpage is persisting. The chosen approach is using a genetic algorithm that explores all the data generated and identifies the behavior of the users with purchasing intention. To accomplish this task, the genetic algorithm will filter the clickstream data while maximizing the correlation with respect to car sales. Customer profiling is performed at two levels. In the first one, the optimization is performed globally between Car Configurator data and sales record. In the second stage, the optimization takes place at the compound region level. The full methodology is found in Section 4.4.

6.4.1 Results Using General Clickstream Data

For the first part of the analysis, the purpose consists of tuning the parameters of the genetic algorithm. There are three degrees of freedom. The terms related to tournament, crossover, and mutation are constant. The tournament probability is set to 30% the population size; there is 90% chance of crossover between the parents; and mutation probability is limited to 1 over population size. This is a problem oriented into augmenting the fitness, whose theoretical limit value is 100%. This number represents the perfect monthly correlation between clickstream data and sales record.

The experiments carried out are named after the chromosome size, i.e., the number of filtering rules; the population size; and the number of generations beyond the initial, respectively. These parameters range from 50-150 rules within the chromosome; 20-300 chromosomes within the population; and 20-200 new generations. For all these scenarios, five independent trials took place. The outcomes of these attempts are visible in Figure 6.18. It reflects the average fitness achieved among all trials in each generation for every experiment. All lines evolved following the same trend. From the very beginning, all experiments surpass the benchmark value. However, there is an evident gap in the lower parts of the graph. The experiments with inferior population size (30 chromosomes) provide smaller fitness. Additionally, most of the trials were interrupted before arriving at the limit number of generations. The cause is the genetic algorithm stacked in local minima more times than permitted by the anti-stagnation mechanism. The summary of generations computed in each experiment and trial is found in Table 6.11.

These reasons conduct the expansion of the search space. Population size and number of generations boost in the next two scenarios. The best output delivers a maximum average fitness close to 90%. Moreover, both experiments were performed without interruptions. A last attempt to achieve the theoretical limit value was carried on. All free parameters increased their values. In this experiment, despite all trials being stopped by the anti-stagnation mechanism, the fitness delivery improved. However, the difference with respect to the previous experiment is not magnificent. Therefore, it is not possible to decide which experiment is clearly dominant.

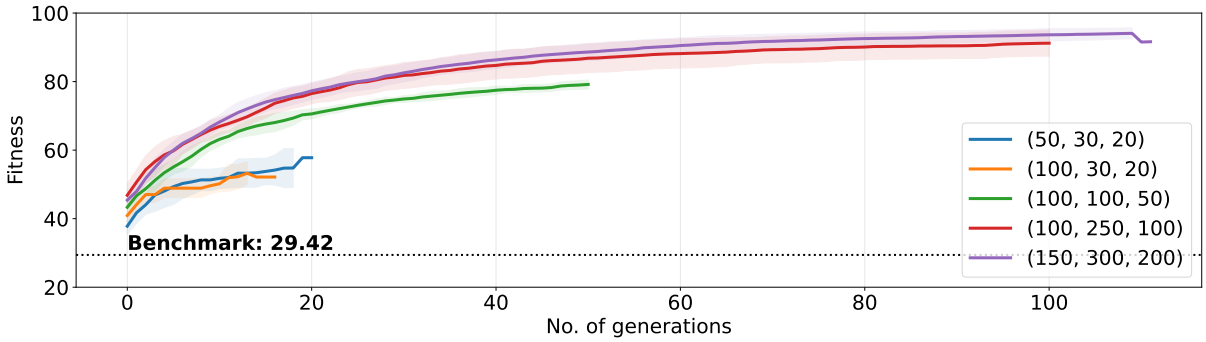


Figure 6.18: Average fitness per generation along all trials for each experiment. Thicker lines represent the average value, whilst the shadow area symbolizes the standard deviation. Experiments are named after number of rules within the chromosome, the population size and the number of generations to explore, respectively.

Table 6.11: Number of generations computed for each trial within every experiment of the genetic algorithm.

No. Rules	Pop. Size	No. Gens	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
50	30	20	18	18	20	15	16
100	30	20	13	12	16	13	13
100	100	50	50	50	50	50	50
100	250	100	100	100	100	100	100
150	300	200	109	109	109	109	111

Other reasons that elude us from making a decision are presented in Table 6.12. It collects the largest fitness values for every trial and experiment. Both candidates deliver the largest maximum values, on average. They surpass about 40 and 15 points in the first two and third cases, respectively. The peaks of the two experiments have the same order of magnitude. The difference is negligible. Hence, it is intended to discover whether simpler scenarios could provide the same performance. That's why a detailed analysis of the best candidate from the two inputs is necessary.

Table 6.12: Maximum fitness values achieved for each trial and experiment of the genetic algorithm. Bold text signals the largest value.

No. Rules	Pop. Size	No. Gens	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
50	30	20	48.22	**58.32**	57.79	53.63	53.83	54.36 ± 4.06
100	30	20	50.12	49.83	52.15	56.01	**56.20**	52.86 ± 3.09
100	100	50	79.51	79.85	76.81	**79.94**	79.77	79.17 ± 1.33
100	250	100	85.34	90.27	**95.45**	91.72	93.28	91.21 ± 3.80
150	300	200	94.39	94.22	**95.98**	94.28	91.63	94.10 ± 1.56

The two candidates under discussion are renamed exp1, the case with 100 rules; and exp2, the scenario with 150 rules. All of these rules are exclusive. There is no single match between the chromosomes of the two candidates. Therefore, for each one of the experiments, the frequency of the elements that compose the rules is computed. Additionally, they are compared against the plain Car Configurator data. The first study takes place in Table 6.13. The row Size GA signifies how many items from all the available options within each feature were explored by the Genetic Algorithm. In both experiments, only the day of the week and the information related to car model and trim were totally scouted. Additionally, the number of chosen items grows together with the size of the chromosomes. A larger number of items were selected in exp2 than in exp1. However, the feature Exterior Color has the most substantial gap. Nearly 50% of available items were chosen. The next two rows, Item_{GA} and Freq_{GA}, represent the item with the largest appearance frequency and this value within the filtering rules. However, the remarkable insight is that there are only two cases where the most popular visited item in the Car Configurator was the most chosen attribute by the genetic algorithm. For exp1, it occurs in the TRIM Level. For exp2, SEAT Leon 5D is the favorite item in both aspects. For the rest of the features, the leading attributes are in long distance of occupying the first places in popularity within the plain clickstream data.

Table 6.13: Frequency of the best items of each chromosome's features in each experiment of the genetic algorithm (GA) and what position that item occupies in Car Configurator (CC) webpage

		DOW	Car Model	TRIM Level	Engine	Ext. Color	Location
exp1	Size_{GA}	7/7	4/4	6/6	34/48	23/51	40/50
	Item_{GA}	Friday	SEAT Leon ST	FR	GV	7Y7Y	ALAVA
	Freq_{GA}	17%	32%	27%	10%	11%	6%
	Pos_{CC}	7^{th}	4^{th}	1^{st}	7^{th}	5^{th}	37^{th}
exp2	Size_{GA}	7/7	4/4	6/6	39/48	26/51	44/50
	Item_{GA}	Thursday	SEAT Leon 5D	Xperience	MX	2Y2Y	ALICANTE
	Freq_{GA}	20%	35%	22%	7%	10%	6%
	Pos_{CC}	5^{th}	1^{st}	4^{th}	15^{th}	4^{th}	6^{th}

The second analysis is shown in Table 6.14. In this case, the most popular item among the users of the Car Configurator is searched in the solutions of the genetic algorithm. It reflects the most visited item per feature individually, not the top car variant configured in the online tool. Besides for TRIM Level and Car Model in exp1 and exp2, respectively, the rest of the information is new. The top visited car model SEAT Leon 5D ranks in second position within the filtering rules of the chromosome from the first experiment.

However, this case is an exception. The rest of the attributes do not occupy these leader rankings. The most dramatic case belongs to the location of exp1. The Spanish province causing the largest number of visits to the Car Configurator aligns in the 23^{rd} position out of 40 explored locations. The other items stands between 4^{th} and 9^{th} place.

Table 6.14: Most popular item per feature within Car Configurator (CC) webpage and what position that item occupies in the best solution of each experiment of the genetic algorithm (GA)

		DOW	Car Model	TRIM Level	Engine	Ext. Color	Location
	Item_{CC}	Tuesday	SEAT Leon 5D	FR	KX	9550	MADRID
exp1	Freq_{GA}	14%	30%	27%	4%	7%	2%
	Pos_{GA}	5^{th}	2^{nd}	1^{st}	9^{th}	6^{th}	23^{th}
exp2	Freq_{GA}	15%	35%	21%	5%	9%	3%
	Pos_{GA}	4^{th}	1^{st}	4^{th}	4^{th}	5^{th}	9^{th}

The findings suggest that the most popular elements within the Car Configurator webpage, while often capturing a significant share of attention, are not necessarily the best indicators of potential consumers among online tool users. This underscores the need for a more nuanced approach to customer profiling analysis. The following trail is built on the fitness of every single rule from the best two candidates. The motivation lies in understanding the weight each individual rule has. To accomplish this purpose, the correlation is computed between the sales record and the reduced clickstream data. This small sample derives from using exclusively the criteria of the rule under analysis, not the entire chromosome. In this manner, from the 100 rules forming the solution of the exp1, 76 of them perform positively to the fitness. The contribution of the 24 remaining rules is null. In the case of exp2, 104 out of 150 rules deliver some individual fitness. The ratio of active rules is similar in both experiments, about 70%. The distinction between both groups of rules is illustrated in Figure 6.19. The rules with zero individual fitness are at the right of the graph and white colored. Additionally, there is green shadow area named Pareto Rules. These regions signal the rules that provide 80% of the value of the accumulated rules fitness. For the first experiment, there are 24 of these Pareto Rules. In the second case, the percentage of Pareto Rules is a little bit higher and increases up to 41 out of 150 rules.

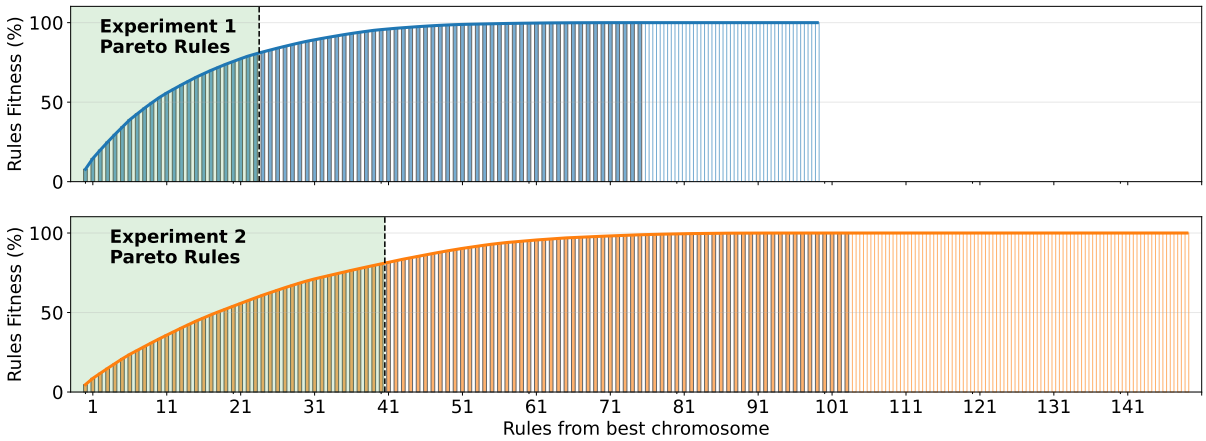


Figure 6.19: Accumulated fitness achieved by each rule within the chromosome of the experiment

It is worth to repeat the appearance frequency comparison exercise. It is performed between the Pareto Rules and the sales record. It is pursued to understand whether it is preserved the same pattern discovered in the clickstream data. The best indicators are not always the most popular elements. Table 6.15 gathers the most chosen item per attribute of the Pareto rules and places it within the ranking of sales record. There is only a match between top preferences. The most common engine from exp2 is the best-seller engine, too. Equal results are found regarding Car Model and TRIM Level, besides the frequency within the Pareto rules is different. The ranking from the rest of the attributes is in the next tiers. On the contrary, Table 6.16 shows the opposite question. Where the best seller items are placed within the Pareto Rules. Clarify that it is presented as the most common item per attribute, not the top-seller car variant. In this case, high-demand items are placed in upper positions than the most visited elements within the Car Configurator webpage. The most distance case occurs in the Location attribute. The village of BARCELONA ranks in the 7^{th} position among the Pareto Rules of the second experiment. This behavior indicates the reliability of the genetic algorithm. It is able to find the filtering rules helping in the correlation with the sales record.

Table 6.15: Frequency of the best items of the Pareto Rules (PR) in each experiment and what position that item occupies in sales record

		DOW	Car Model	TRIM Level	Engine	Ext. Color	Location
exp1	Item_{PR}	Monday	SEAT Leon 5D	Reference	JX	7Y7Y	CASTELLON
	Freq_{PR}	25%	46%	25%	17%	25%	12%
	Pos_{SALES}	5^{th}	2^{nd}	3^{rd}	16^{th}	4^{th}	23^{th}
exp2	Item_{PR}	Tuesday	SEAT Leon 5D	Reference	CV	0E0E	ALICANTE
	Freq_{PR}	27%	39%	29%	12%	15%	12%
	Pos_{SALES}	4^{th}	2^{nd}	3^{rd}	1^{st}	8^{th}	5^{th}

Table 6.16: Most popular item per feature within Sales record and what position that item occupies in the Pareto Rules (PR) of each experiment of the genetic algorithm (GA)

		DOW	Car Model	TRIM Level	Engine	Ext. Color	Location
	Item_{SALES}	Friday	SEAT Ibiza	Style	CV	B4B4	BARCELONA
exp1	Freq_{PR}	12%	29%	21%	12%	12%	12%
	Pos_{PR}	4^{th}	2^{nd}	2^{nd}	2^{nd}	3^{rd}	2^{nd}
exp2	Freq_{PR}	10%	17%	27%	12%	12%	5%
	Pos_{PR}	6^{th}	4^{th}	2^{nd}	1^{st}	2^{nd}	7^{th}

The previous results validate the aforementioned tendency. Frequently favored components, from both sales records and clickstream data, they may not necessarily serve as the most accurate indicators of potential consumers. Nevertheless, it is mandatory to make a decision about what are the best input parameters. The experiment with the largest number of rules, population size, and number of generations is declared as the successful candidate. The behavior shown in the detailed study of the two candidates is equivalent. That's why, the decision is based on objective magnitude. It delivers the largest fitness value. The impact of these parameters at the compound region level is under assessment in the next section.

6.4.2 Results From Compound Region Approach

In this part of the research, the sales data has been divided at the compound region level. Hence, the genetic algorithm seeks to optimize the correlation between the clickstream data and the local sales. Nevertheless, the benchmark value is computed in the next way. Only visits registers and sales coming from the same compound region are employed. That's how they are obtained the results from Table 6.17. The input parameters of the genetic algorithm are the previously chosen. All compound regions exhibit a considerable enhancement in terms of correlation when compared to the baseline values. Among the considered compounds, MADRID displays the most significant improvement, while LLAGOSTA obtains the highest correlation value. On average, the fitness is augmented by 66.25 ± 1.46.

Table 6.17: Maximum fitness value obtained by the genetic algorithm (GA) in each compound and the benchmark value.

	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Benchmark	31.08	29.46	30.49	32.08	35.35	29.91
Fitness_{GA}	96.92	97.22	98.32	98.00	99.25	96.15

The next steps consist of a detailed analysis of the filtering rules. This time, done for each individual compound region. Rather than focusing on the frequency aspect of the items within the candidates, we explore the location attribute. In other words, the locations included in the chromosomes are divided into two groups. Depending on whether the locations belong to the compound region under analysis or they do not.

Therefore, the outputs are placed on Table 6.18. Firstly, it is searched how many locations from each group compose the structure of the candidates. In all compound regions, the Spanish provinces that built it are included. Nevertheless, they are not the majority among the rest of the items. The group of the outsider locations has an appearance frequency that ranges between 70% and 93% within the filtering rules of the candidates. Despite this, the vast presence of outsider locations does not necessarily mean they are more relevant. Therefore, it is interesting to understand whether groups are comparable regarding fitness. To accomplish this task, the individual fitness of each individual rule is computed in the same way was performed in the past. This is the path to achieve the average fitness value provided by each group. The external locations of CMC, MADRID, and SANTANDER provide more fitness than the internal ones, since the average fitness is larger. The contrary occurs in CHESTE and LLAGOSTA. Finally, the difference between the two groups in LA RODA is not significant.

Table 6.18: Outputs of the analysis performed to the solution of each compound region. Suffix In stands for locations belonging to the compound under analysis. Suffix Out is the opposite.

	CMC	MADRID	LA RODA
No. Locations$_{In}$	4	10	9
No. Locations$_{Out}$	45	39	38
Frequency$_{In}$	8.0%	14.0%	19.33%
Frequency$_{Out}$	92.0%	86.0%	80.67%
Average Fitness$_{In}$	0.34 $\pm$ 0.83	1.7 $\pm$ 3.29	2.01 $\pm$ 3.06
Average Fitness$_{Out}$	2.26 $\pm$ 3.15	2.48 $\pm$ 4.15	2.0 $\pm$ 3.45
	CHESTE	LLAGOSTA	SANTANDER
No. Locations$_{In}$	4	8	15
No. Locations$_{Out}$	45	40	34
Frequency$_{In}$	7.33%	20.0%	29.33%
Frequency$_{Out}$	92.67%	80.0%	70.67%
Average Fitness$_{In}$	2.24 $\pm$ 2.17	4.61 $\pm$ 8.53	1.89 $\pm$ 2.9
Average Fitness$_{Out}$	2.16 $\pm$ 4.19	3.0 $\pm$ 4.61	3.53 $\pm$ 5.72

6.5 Genetic Algorithm Improves Demand Forecasting

The power and versatility of the genetic algorithm will be employed to decrease the prediction error of the car variant demand, rather than maximizing correlation. Different executions under different boundary conditions will be carried out. The goal consists of finding the scenario within the search space that provides the largest improvement with respect to the benchmark of all the car variants and time chunks forecasted. In other words, predictions shown in Section 6.2 for compound region level, as it is the granularity with the largest upgrading room. Lastly, the outcomes will be assessed based on weekly mix sales. More details are placed in Section 4.5.

6.5.1 Forecast Comparison

The first part of the analysis includes presenting the number of cases where the genetic algorithm was superior. In other words, the forecast error is smaller than the one already shown in Annex B, i.e., the predictions derived from using the sales record and the clickstream data exclusively from the car variant under study. The forecast errors of the genetic prediction are placed in Annex D. Therefore, the comparison is extended to the full amount of cases, totaling 120 instances. The summary of the analysis is placed on Table 6.19.

Table 6.19: Number of cases per each trial and experiment on which genetic algorithm improved the benchmark results. Bold text signals the largest value.

No. Rules	Pop. Size	No. Gens	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
50	30	20	67	66	71	72	71	69.4 ± 2.7
100	30	20	80	77	74	74	75	76.0 ± 2.5
100	100	50	99	98	99	99	100	99.0 ± 0.7
100	250	100	109	110	107	110	107	108.6 ± 1.5
150	300	200	112	**113**	111	111	112	111.8 ± 0.8

The trend that data evoke shows the rising behavior of the figure of merit attached to the complexity of the experiment. The largest the latest, the best figure of merit. On the other side, the numbers per trial for each experiment remain basically constant. In this scenario, the fifth experiment delivers the largest figure of merit. Additionally, the best outputs correspond to the second trial out of the five trials executed. That's why it is named as the winning candidate. In case there is more than one prospect, to make a decision on which is the winning subject, it would be explored which one of the candidates provides more forecasting error reduction. This metric is defined as the average of the difference between the MAE from the VOI forecast and the MAE of the genetic forecast for each time chunk and car variant. For instance, the average reduction achieved by the winning candidate is 3.89 ± 0.96 cars.

The specifics of the winning candidate are illustrated in Figure 6.20. It reflects the MAE obtained within each car variant and time chunk for the two forecasting approaches. There are only seven cases in which the genetic algorithm is inferior to the benchmark values: two are from the SEAT Arona, both LA RODA in the first and third time chunk; two other cases belonging to the SEAT Ibiza, in the fifth time chunk in the compound regions of LA RODA and SANTANDER; two of them applied to SEAT Leon 5D, linked to SANTANDER and MADRID in the first and third time chunk, respectively; and the last one corresponds to SEAT Leon ST, occurring in the fourth time chunk in CMC compound region. The best and worst differences owe to SEAT Ibiza and LA RODA as compound region. The largest positive deviation (12.7 cars) takes place in the third time chunk, while it is in the fifth time chunk when the negative deviation (-17.6 cars) is maximum.

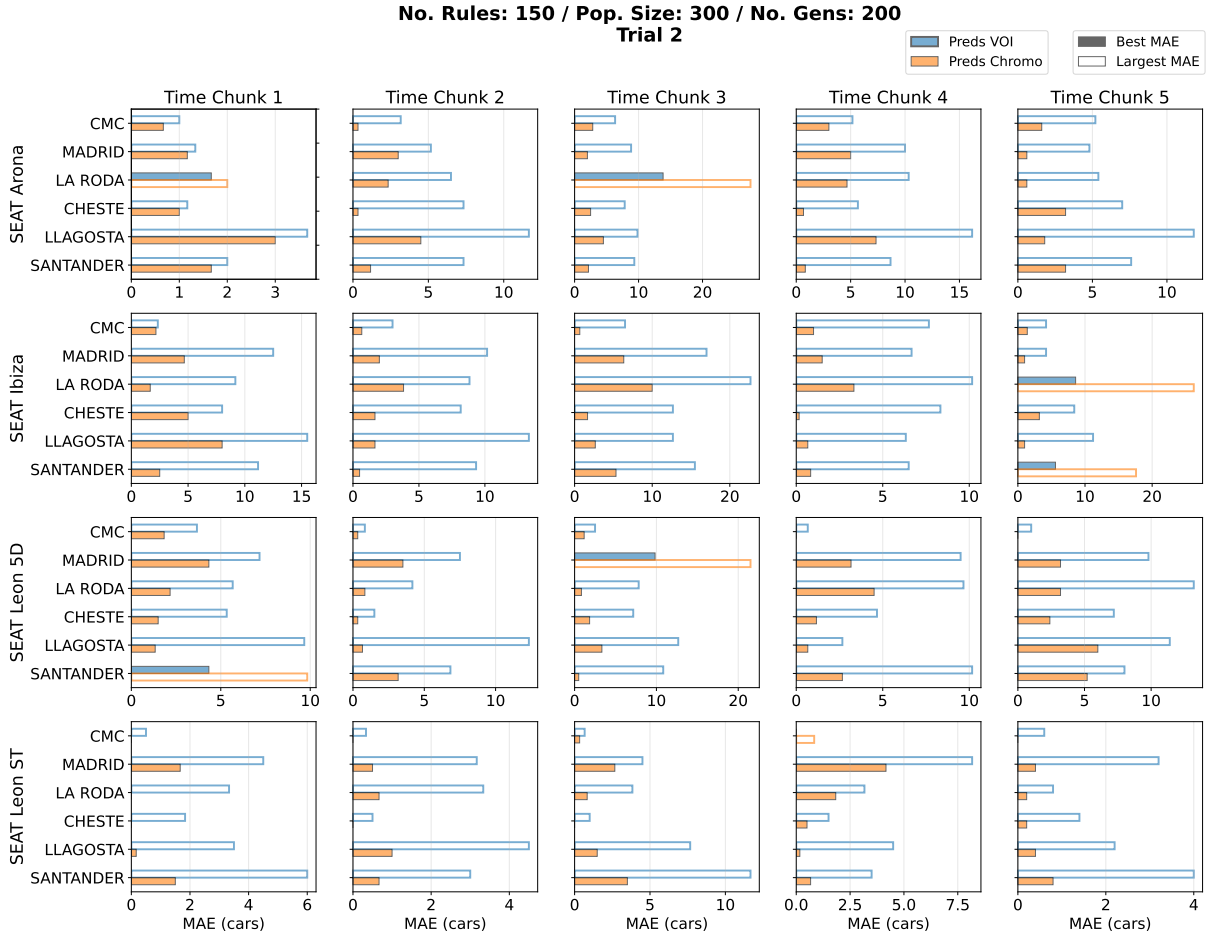


Figure 6.20: Comparison of car variant MAE per time chunk of the best experiment and trial between predictions using the data sole from the car variant under analysis (Preds VOI) and new predictions derived from the genetic algorithm (Preds Chromo). Full colored bar represents the car variant forecast with the lowest error.

Another perspective from the final candidate regards the assessment of the weekly mix sales. The performance of each single instance is placed in Annex D. The average performance across weeks and time chunks is summarized in Table 6.20. All car models show a significant improvement in metrics, ranging from 5 to 15 points. Cases where the genetic algorithm doesn't surpass old predictions are not penalized in this comparison. Lastly, the assessment takes place at the time-chunk level. Table 6.21 gathers the averaged outcomes. The numbers achieved by the genetic algorithm are higher than the ones computed by the original prediction. The exceptions occur at the same items found in Figure 6.20, but SEAT Arona, SEAT Leon 5D, and SEAT Leon ST during first, third, and forth time chunks, respectively. The new forecast error derived from the genetic algorithm is not enough to disturb the upgrade from the rest of the car variants.

Table 6.20: Comparison of the average R2 Score (%) for the predictions used as benchmark (VOI) and the ones obtained by means of the genetic algorithm (Chromo). Average is computed for the weekly sales mixes of each car model at compound region granular level over the total size of time chunks of the dataset

	SEAT Arona	SEAT Ibiza	SEAT Leon 5D	SEAT Leon ST
R2 Score (VOI)	63.74 ± 33.71	75.43 ± 19.8	77.84 ± 18.5	83.86 ± 19.54
R2 Score (Chromo)	69.33 ± 37.23	90.01 ± 18.83	87.94 ± 13.48	96.08 ± 10.08

Table 6.21: Comparison of the average weekly sales mixes R2 Score (%) for the predictions used as benchmark (VOI) and the ones obtained by means of the genetic algorithm (Chromo). Average is computed per time chunk and car model. († *highlights cases in which the VOI outputs are larger than Chromo outputs.* ‡ *represents cases in which there is a car variant where MAE was not improved by genetic algorithm, but average weekly sales mixes R2 Score is not affected.*)

	SEAT Arona		SEAT Ibiza	
	R2 Score (VOI)	R2 Score (Chromo)	R2 Score (VOI)	R2 Score (Chromo)
Time Chunk 1	12.5 ± 27.03	13.71 ± 24.33‡	60.65 ± 24.8	93.38 ± 6.69
Time Chunk 2	71.65 ± 25.65	94.7 ± 8.38	75.38 ± 17.63	98.88 ± 1.56
Time Chunk 3	77.25 ± 17.21†	59.17 ± 32.57	81.18 ± 15.35	96.72 ± 2.76
Time Chunk 4	68.11 ± 15.89	85.33 ± 16.2	84.83 ± 9.46	96.94 ± 6.87
Time Chunk 5	94.28 ± 5.6	98.61 ± 2.03	75.03 ± 25.91†	58.96 ± 29.58
	SEAT Leon 5D		SEAT Leon ST	
	R2 Score (VOI)	R2 Score (Chromo)	R2 Score (VOI)	R2 Score (Chromo)
Time Chunk 1	87.62 ± 6.25†	84.52 ± 15.69	88.71 ± 9.15	95.99 ± 6.27
Time Chunk 2	88.71 ± 6.95	95.11 ± 9.14	89.52 ± 10.27	98.85 ± 1.62
Time Chunk 3	60.11 ± 27.65	72.64 ± 13.8‡	56.69 ± 27.46	87.99 ± 20.34
Time Chunk 4	75.81 ± 14.25	95.1 ± 6.25	94.56 ± 6.6	98.67 ± 2.5‡
Time Chunk 5	76.75 ± 16.61	93.23 ± 3.79	90.99 ± 3.85	99.45 ± 0.37

6.6 Production Modification Based On Improved Forecasting

More precise forecast permits to anticipate the future needs and requirements of the customers. Consequently, the genetic predictions can be utilized to adapt the current production of the company to the expected demand. This section addresses efforts to align stock composition with customer demand by optimizing vehicle destinations through convex optimization, particularly for Build-to-Stock cars in production data. Section 4.6 possesses all the methodology.

6.6.1 Diminishing the gap between compound and demand

Figure 6.21 shows the *points of imbalance* between the composition of estimated stock and future demand before and after the optimization. This magnitude is the resulting number after computing Equation 4.1 with the original and the updated production. The smaller the number, the more aligned the demand and the stock. The outcomes are presented per the starting week of the modification dates. Outcomes are scaled so they can be easily comparable among them, but preserving the gap between cases. It is worth explaining that optimized outputs have an annotation. The primitive updated production values of the solver are float. However, the reality only permits integer units. That's why after the rounding of the values, it was necessary to correct, in case it was needed, the total production volume. In all cases, the compound region with the largest quantity of vehicles assigned is the subject of the correction, by means of addition or subtracting vehicles.

In all cases of Figure 6.21, there has been an improvement. It might seem that this fact does not apply to the entire first time chunk of SEAT Arona and also to the first week of the last time chunk for the SEAT Leon ST. In these events, the differences between before and after the updating process are null. The reason that explains this behavior is that they are in which there is no production for these car models and time ranges. The largest improvements, in real magnitude, correspond to SEAT Ibiza in the last week of the third time chunk, from 36.5 to 25.5 points, whilst the shortest upgrading belongs to SEAT Leon ST car model in the last week of the second time chunk, only $6.5 \cdot 10^{-4}$ points of difference from the starting to last situation. Actually, the average improvement for all cases after removing the outliers, i.e., those situations with zero upgrade or values far from the general trend, is about $3.25 \pm 4.47 \cdot 10^{-2}$ points, for instance, the first week of the fourth time chunk from SEAT Arona. In fact, to give a deepest insight, Table 6.22 presents the original and optimum production for the three cases aforementioned. The original production of car variants with the maximum upgrade was widely distributed along all compound regions possible. Afterward, the number of candidate destinations is reduced to three, being CHESTE the main one. It follows the average case, where it is discovered that not all compound regions will receive vehicles. The optimization process suggests that all production is headed to LLAGOSTA as single destination. The last case, the one delivering the lowest improvement, both original and optimum production are concentrated into an unique destination. However, this is happening because there is one single vehicle to be produced. The optimization output proposes to accommodate this car in SANTANDER, rather than MADRID. It is learned that the updating policy tends to rearrange the production or to concentrate it into a unique destination.

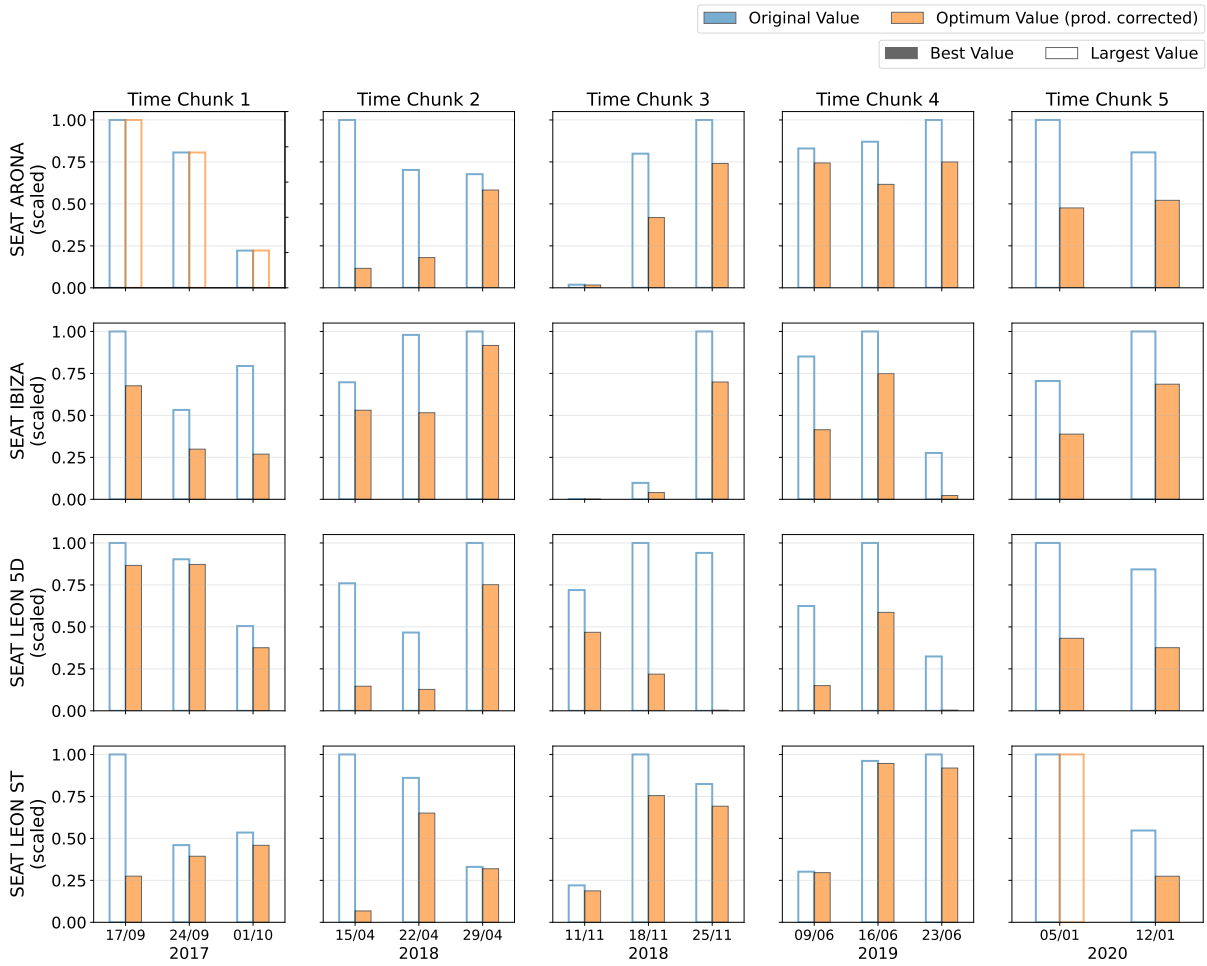


Figure 6.21: Comparison of *points of imbalance* before and after the updating process of the destination of cars in production. Colored bars represent the case with the best output.

Table 6.22: Comparison of the production before (Original) and after (Optimum) the optimization process for three out of the most representative cases: the one with the largest improvement, the one with average improvement, and the one with the least improvement.

	Max Improv.		Avg Improv.		Min Improv.	
	Original	Optimum	Original	Optimum	Original	Optimum
CMC	1	0	0	0	0	0
MADRID	4	11	4	0	1	0
LA RODA	8	5	2	0	0	0
CHESTE	9	26	0	0	0	0
LLAGOSTA	15	0	7	14	0	0
SANTANDER	5	0	1	0	0	1

Chapter 7

Discussion

The path that guided the development of the presented solution attempts to proceed from the least to the most impactful repercussions in the company's operations. Although this thesis has taken place under the constraints of a particular automotive manufacturer, it can be extrapolated to other environments. The limitations of SEAT are not unique since all manufacturers depend on a production planning constrained to certain restrictions; follow some kind of forecast; fabrication is a mix of Build-to-Order and Build-to-Stock; external agents, such as dealerships, may affect the company outcomes, etc. They are seeking to deliver their products in the minimum time with the largest acceptance ratio.

The first step focuses on the cars already manufactured. Based on the attributes that construct a vehicle, a binary classifier is built to distinguish whether the car will be a Fast Delivery or a Normal Delivery within the destination it is headed. These attributes include the Car Model, the Equipment Level (TRIM), and the Engine, together with the Exterior Color of the vehicle and its destination. Moreover, the label that points out a car as Build-to-Order (BTO) or Build-to-Stock (BTS) is added to the feature space. It is discovered that two out of the values within this last parameter lead in the features relevance analysis of the best classifier (see Figure 6.1). Remembering Table 3.7, Private Customers is the Order Type with the lowest mean and median days waiting in the compounds and the second category most populated by number of cars. On the other side, Dealerships Stock is the feature with the largest mean and median in relationship with the number of days cars spend within the compounds. It is difficult for dealers to always order the most suitable batch of vehicles from the factory. When these situations occur they do not suffer any heavy penalties for future orders. Additionally, it is the tag with the largest number of observations, not only from the BTS family but in general. For these reasons, it is coherent that these labels are critical to catalog the instances.

The best classifier has been chosen among a broad number of candidates. Four different machine learning algorithms have been trained and optimized to provide the best outcomes. The list is: Decision Tree, Random Forest, XGBoost, and CatBoost. It ranges from the simplest to the last state-of-the-art technique, but all of them with successful use cases in industrial frameworks. Other approaches, such as neural networks, are well-known for modeling complex relationships and patterns in data. In spite of this, they were disregarded for two reasons. On one side, since they are fed by larger data volume than the one available in the experiment to be trained effectively. On the other side, the explicability of neural networks' solutions is challenging, which poses a significant drawback in scenarios where understanding the decision-making process is crucial. Consequently, more interpretable models were preferred to ensure both performance and clarity in understanding how predictions are made.

The hyperparameters optimization has been accomplished by means of Bayesian Optimization, guiding the estimator in the right path to upgrade the performance and rather than the brute force strategy followed by GridSearch or the blind and aleatory approach of RandomSearch. Additionally, the exploration has been carried out under the conditions of cross validation. The attempt was to diminish the trend of some of the algorithms under analysis to overfit. The optimization has evolved maximizing the F1 Score, instead of other common assessment metrics such as accuracy, recall, or precision. The first one has been rejected because of the conditions of the problems, the dataset is seriously unbalanced under some threshold days. Recall and precision are excellent choices, consequently, it is attempted to optimize both. The needs of the car industry demand that vehicles are correctly classified as Fast Delivery, whilst the most Fast Delivery cars are captured. Consequently, F1 Score is employed as the assessment metric to guide the optimization process since it balances precision and recall.

These thresholds are the maximum number of days a vehicle can stay in the compound region until it is delivered. Above these limits, the car is considered a Normal Delivery. It ranges from the extreme case of shipping in one week up to spending six weeks in the compound region. This decision is motivated since the largest median in the number of days cars stay in the compound regions is close to this value (see Table 3.5). Consequently, the poorest results are delivered in the first threshold, but they improve together with the thresholds since the classes become balanced. It implies that a larger ratio of Fast Delivery cars within the dataset is positive for the algorithms' performance. Henceforth, among all the candidates, XGBoost is the most solid in its outcomes, but it is worth mentioning Decision Tree results in spite of being known to be less effective in capturing complex relationships within the data.

After selecting the best estimator, it is presented the confusion matrix derived from it (see Figure 6.2). It is learned that the trend of the classifier consists of classifying Fast Delivery cars as Normal Delivery ones. This tendency is provoked mainly by cars requested by dealerships. The segmentation of the confusion matrix per Order Type is motivated by the feature relevance analysis, as two out of the four classes within this feature occupy the top positions. However, this misclassification is not critical since Fast Delivery vehicles are car variants attractive to the customers. Hence, future clients would purchase them easily, affecting into the number of days these vehicles would spend within the compound regions.

Hold in awareness that each day spent incurs logistical costs, particularly since a vehicle might be occupying the slot of a car that is more suitable to the client's preferences. Hence, the company is forced to offer a discount on the price of the stocked vehicle with the aim of augmenting its purchasing likelihood. Thus, a reallocation of the cars to the location where they are classified as Fast Delivery gives more opportunities than obstacles. Customers would have their desired car in a shorter time, whilst the company's economics is enhanced. Unfortunately, this last aspect is not quantified since managers of the compound regions did not share their costs. As a result, it has been left out of the scope of the research. Nevertheless, the impact of pursuing a reallocation of vehicles is analyzed. All those vehicles that were classified as Normal Delivery within their original destination will pass again through the classifier, but listing all the alternative destinations. This strategy offers a new perspective to the bibliography on establishing an efficient operation for the automotive industry. This research is situated in a context where it is not feasible to modify the manufacturing process or any of the vehicle assembly components in opposition to the existing literature.

Nearly half out of all Normal Delivery cars found, at least, one alternative destination (see Figure 6.4). When the new distribution of days within the compound is compared against the original, the reallocation procedure matches or enhances the decisions taken by the company's experts (see Table 6.7). The median has decreased in all the locations but two, LA RODA and CHESTE. Especially relevant is the last location, in which the decay is about 10 days for the approach without new Number of Days computations. Regarding the second scenario, it presents concerns due to its random nature. Ideally, this approach would involve multiple repetitions of the experiment. However, it is believed that the actual behavior of the reallocation would fall between the two presented concepts: either the car would spend the same amount of time at the new destination, or the new location would influence the duration.

Although, the estimator has a preference for misclassifying Fast Delivery vehicles. Hence, customers from these regions might find in their dealerships these kind of vehicles easily. Moreover, locations such as LLAGOSTA and MADRID compensate it since they aggregate the largest delivery volume. The impact on the company's operations and on customer satisfaction will be greater. Despite the favorable outcomes, the reallocation strategy lacks a criterion to select among the alternative regions where the car variants are cataloged as Fast Delivery. As a result, there are duplicities in the instances of the dataset that might affect the final outcomes of this procedure. It is proposed to prioritize the destination with the largest disparity between the demand and stock. Therefore, it is recommended to have a reliable estimation of the expected customer demand and the following stock composition. This need encourages the following steps of the proposed solution. Lastly, once a decision about the final destination of the vehicles has been made, the transport planning and logistics of the reallocation output is another area of special interest to manage. Regarding this step, customers' preferences are not deeply taken into consideration. The best performance classifier has inferred them from the sample although the estimator is strongly influenced by the Order Type feature. Therefore, it is proposed in this research to explore the data collected by the SEAT car configurator webpage. It can be used to capture future customers' preferences. The outcomes achieved prove this hypothesis. The correlation analysis between clickstream data and sales records suggests the duration of the purchase period. Furthermore, the addition of visits to the online tool has enhanced the performance of the diverse demand prediction techniques carried out. Supplementary, quantitative and qualitative approaches have been successfully put into practice to diminishing the noise and meaningless data contained in the entire set.

Exploring the correlation analysis between clickstream data and sales records, none of the car models under study is close to the maximum value. Nevertheless, when the correlation values are ranked the top five of them lay in the first half of the lagging period. Exclusively positive correlations are listed since negative values are not strongly significant either. It is irrational considering that having no visits to the car configurator webpage will boost the sales of the company. Because of these top five correlation values, it is concluded that users consult the online tool from 1 to 6 months before the purchase date. Results are consistent at different granular levels. The same experiment has been performed at the car model level and car model plus exterior color granularity, offering the same outputs. These findings align with the discoveries of other authors. Paper [36] was able to find a correlation in the entertainment industry in terms of weeks. For the financial sector, the correlation with online data is found at the day level, as supported in [39]. These timeframes are considered normal for these products. However, in the car purchase process, the period expands considerably, as it is common in high-implication products.

The previous findings are used to divide the temporal range into five subsets, called time chunks. In each one of these, the prediction of the sales record for each car variant will be executed by means of diverse forecasting techniques. Nevertheless, there is a concern about the time chunk division. Possible seasonality or trend of the time series, benefit for the prediction, could be interrupted. However, the exploratory data analysis carried out did not show robust evidence of consistent and helpful time series decomposition for each one of the car variants. Additionally, more complex prediction methods, such as neural networks, were not considered due to their needs of vast amounts of data or black-box behavior. Unfortunately, it is not the case in these circunstances. In this context, traditional techniques can deliver a good performance. This strategy describes the second phase of the planning to confirm the reliability of the clickstream data. A supplementary effect of the temporal division is that the forecasting will be carried out in the stages of the life cycle of a product.

The efficiency of the prediction enhances those forecasting algorithms that are supported by the clickstream data gathered from the car variant under analysis, i.e., multivariate, in opposition to those ones that only rely on the past sales record of the instance, i.e., univariate. Actually, outcomes could be better since the hiperparameters of the tree-based algorithm were not optimized. However, the abscence of fine-tunning is a deliberate choice. The model maintains a balance between simplicity and performance, ensuring that the predictions remain generalizable and not overly tailored to the training data. For the two elected car variant granularities, car model plus exterior color or compound region, the multivariate technique has no rivals (see Figure 6.10 and Figure 6.11, respectively). The granularity levels are chosen due to logistics reasons. They are elements non-dependable of spare parts and easily interchangeable.

Afterward, weekly mix sales are calculated. The chosen metric to evaluate the functioning of this new schema is the correlation between the real and the forecast ones. The motivation behind this decision is that production volume is a parameter depending on other stakeholders. It is beyond the capabilities of this thesis to modify it. Consequently, what is proposed in this research is being accurate with the proportion each car variant should have in the weekly production, rather than the quantity. For this reason, the R2 Score computed between both weekly mix sales represents the behavior of the experiments between zero, or null correlation, and one hundred, i.e., perfect match. The assessment procedure includes averaging the results over the totality of cases, but as well per time chunk. In the first case, the addition of clickstream data benefits the performance of the prediction for both types of car variants (see Table 6.8 and Table 6.9). However, in the subsequent steps, the outcomes derived from the car model plus exterior color variant are more solid than the ones achieved in the case of the car model plus compound region. In the latter, XGBoost multivariate is not as prevailing as much as in the exterior color case. Nevertheless, one of the multivariate forecasting algorithms is in the competition of leading the outputs in the vast majority of experiments. One possible explanation for these events is that car configurator data correlates better at exterior color granularity than at compound region level(see Figure 6.6 and Figure 6.7). It seems that users are more determined about the painting of the vehicle, rather than the location from where they will purchase the vehicle, although exploratory data analysis suggests evidence of the contrary behavior (see Figure 3.11 and Figure 3.13). The plausible explanation is the existence of noisy and meaningless data that worsens the functioning of the forecasting algorithms.

The next point in the proposed solution deals with the problem of diminishing the quantity of worthless data within the information gathered by the online tool. The previous outcomes were achieved with a dataset processed and cleaned. Nevertheless, the car configurator webpage is a service with some implicit difficulties chained. Recording the activity within the webpage exclusively relies on the user's internet browser cookies, since any kind of personal information is not requested. Consequently, performing rigorous tracking is challenging. A user can erase his browser cookies, access from another device, two people can connect from the same gadget, among many other casuistics. All these events cause the amount of data to skyrocket. Additionally, nowadays there does not exist the possibility of finishing the purchase within the webpage. The user must head to the dealership, which can influence him to finally acquire a vehicle different from the one chosen on the online platform. Therefore, it is an arduous task to discriminate between users with real purchase intentions from those doing window shopping. On one hand, reducing the amount of data to a smaller fraction simplifies all the issues linked to Big Data environments, listed as: storage, data quality, data governance, escalation, etc. On the other hand, a genetic algorithm is run to find the patterns of users with genuine purchase intention.

The reduction of data volume is based on the users' quantitative activity, instead of what are the features he has selected. The quantitative activity is summarized in the number of days that have passed between the first and last registered connection, known as Time between connections; and the number of unique car variants he has configured in the webpage. The strategy followed consists of consecutively eliminating the outliers detected within each feature. Outlier detection is based on two well-known techniques. In the end, the amount of car configurator users to analyze has passed from nearly 1,9M to less than 500k (see Figure 4.2). Despite the filtering rules might look arbitrary; they are supported, on the one side, by the user's acquisition necessity; and on the other side, by the typical customer journey. They have proved themselves as effective rules to preserve the significance contained in the raw clickstream data. In all cases, the statistical test delivered a large p-value proving the equivalency between the raw data and the filtered ones (see Table 6.10). The significance is computed based on lagged correlation. The delay is added to simulate the period between browsing in the car configurator and heading to the dealer shop. As a result, the election of the 8-week delay matches with the previous findings of a six-month period and it is time enough to manufacture a vehicle. The goal of the entire procedure is not to discover the exact duration of this period or any other parameters of the experiment but to verify the preservation of the significance of the dataset. Therefore, there is still room for improvement of the results.

The discrimination between users with and without purchase intention is a task suitable for a genetic algorithm. The problem is framed as one that maximizes the average significance between the sales record and the clickstream data that respect the rules contained in the chromosome of the genetic algorithm. In other words, the subset of car variants that a user can select on the webpage together with the day they do it. The chromosome with the largest fitness draws the online path of users with real purchase intention. This methodology has been evaluated at both general and compound region levels. In the general case, it has been improved from a 29.42 fitness to an average of 91.21 ± 3.8 (see Figure 6.18). At the compound region level, the average improvement is 66.25 ± 1.46 in fitness (see Table 6.17).

In the first approach, the purpose consists of tuning the parameters of the genetic algorithms. Some aspects of the algorithm, such as tournament, crossover, or mutation probabilities are fixed. The attempt is to create a stable framework to ease the comparison of the different candidates. They were generated fitting the number of rules within the chromosome, the number of chromosomes that assemble a population, and, finally, the maximum number of generations the algorithm can iterate. The total number of conducted experiments has reached six. For each of these experiments, five independent trials under identical conditions were performed. It is an attempt to obtain robust results that overcome the randomness inherent in this type of solution. Since two feasible candidates from distinct experiments are delivering almost exact outcomes, the two of them are analyzed, focusing on the frequency of the constituent rules. Although all trials from the bigger experiment delivered analogous results, it is intended to comprehend if the similar solutions could be obtained from different size experiments. The appearance rate of the elements listed in the chromosomes is compared against the popularity index within clickstream data and extended to the sales record, as well. Regarding the latter, it is performed employing single Pareto rules, i.e., the rules that individually collect nearly 80% of the total chromosome fitness. The findings suggest that the most popular elements from these sources are not necessarily the best indicators of potential consumers among online tool users.

Afterward, the procedure is individualized per compound region, isolating the online data and the sales record belonging to each one of these locations. In this scenario, the genetic algorithm has been executed with the best parameters achieved in the previous step, rather than exploring again the search space. The maximization objective is preserved. Nevertheless, the assessment is oriented on how the Spanish provinces are segmented. To be more precise, listing which locations are inside and outside the area of influence of each compound region and performing a fitness comparison. The outcomes reveal an underrepresentation of locations that define the compound's influence area. Unfortunately, we were uncapable to find in the data an explanation to those cases in which external locations provide more fitness than internal ones. The reasons might lay in external information that we do not have access to. However, the distribution of individual fitness scores for both groups was found to be equivalent.

To conclude, these data mining experiments have accomplished their objectives. Hence, it is suggested to apply this methodology in the businesses' decision-making process of the automotive sector, although it can be extended to webrooming economical sectors of the same characteristics, in case there are any. The procedure exposed attenuates the counterparts of managing large databases because they escalate together with data size. In the same way, a genetic algorithm is a versatile tool that can be adapted to various contexts. The experiments that have taken place confirm the correlation between clickstream data and sales records. Nevertheless, there is still room for improvement in these outcomes. Future researchers should pursue clickstream data without the current limitations.

Since users can be straightforwardly identified, the genetic algorithm can be modified to obtain a more precise forecast. Consequently, the next part of the proposed solution manages this situation. Because the assessment of the old forecast for car model plus compound region from Section 6.2 is less homogeneous than the other car variant ones, the former will be used as a benchmark in this new scenario. The genetic algorithm, hence, is trained to reduce the forecasting error of the sales predictions. The machine learning algorithm employed is XGBoost multivariate. In this case, it will be fed by the clickstream data that respect the rules within the best chromosome, rather than the visits to the car configurator belonging exclusively to the car variant under analysis.

Considering that the context has shifted, and the purpose is no longer the maximization of the correlation, the free parameters need to be tuned again. However, it has been decided to explore the same search space as in the previous experiments. It is a framework in which behavior and trends are well understood. Therefore, the manner of electing the winning candidate is anchored on counting the number of cases in which genetic predictions outperformed the benchmark. It is a course of action inspired by the literature. The bibliography contrasts the prediction outcomes against the reference values after executing several experiments. Therefore, this logic will be followed listing all the cases in which the genetic algorithm delivered a better result than the benchmark. As was expected from the search space, this figure of merit grows together with the complexity of the experiments executed (see Table 6.19). The best outcome enhances the original results in 113 samples out of the 120 cases. The detailed picture of the solution (see Figure 6.20) explains that the seven cases of difference belong to all the car models and time chunks.

It would be possible to pursue the total efficiency increasing the complexity of the experiments. Nevertheless, the time requested and computational power required to make it a not worthy endeavor. Especially, since the assessment procedure evolves in another direction. Weekly mix sales are rebuilt employing the genetic predictions. The evaluation follows a two-step approach. In the first one, it is learned that the general average R2 Score of each single car model surpasses the previously obtained (see Table 6.20). But when the correlation metric is averaged per time chunk, the results are equally promising. In three out of the seven cases which worse performing, the situation has shifted. The forecast error generated by the genetic algorithm is insufficient to disrupt the upgrade for the remaining car variants.

The merge of two powerful tools such as a genetic algorithm and forecast technique has been proven as a good strategy. At the same instant, it validates the reliability of the car configurator data; proposes a qualitative manner to overpass the concerns related to this data source; and promotes more accurate demand forecasting. This combination of techniques will permit the private partner of the research to react more efficiently to the challenges of its business. The company looks very positively at the investigation done in this part of the thesis and they are already incorporating it into their systems. Additionally, a new methodology is added to the literature, to our knowledge.

Despite this, there is a counterpart in the use of genetic algorithms both applied in the qualitative filtering and the demand prediction. On one side, correlation might be mistaken for causality. Statistical tests such as Granger causality were carried out, but not included in the research since they do not provide an explanation from the physical world. On the other hand, overfitting is a real threat. To mitigate these risks, the fitness function of the genetic algorithm has always been isolated from the benchmark value it was pretended to improve. In this way, the executions were not influenced by the outside. Bearing in mind all these warnings, we are aware of all the problems that they could cause. For instance, misguided decisions or wasted resources. Therefore, our recommendation to the private partner of the research is to create a testing environment in which to apply A/B tests or other strategies to validate or refuse these concepts.

The last step in the proposed solution introduces a new utility to the newer demand predictions. The more accurate forecasts are used in this simulation to update the vehicles in the manufacturing line. Specifically, a new compound region is assigned to each one of the car models. The motivation behind this simulation refers to the first solution developed in the thesis. It attempted to find the most suitable destination for the cars after they were manufactured. Nevertheless, on this occasion, the object of study is the vehicles in the assembly line. Hence, to accomplish this task, it is required to understand

the production flow of the factory, as well as estimate the stock there will be in the different destinations. Accordingly, the equations governing the functioning of the stock are presented (see Equation 4.1). In consequence, the problem can be structured as an optimization one. It is searched to minimize the imbalance between the configuration of the future demand against the composition of the expected stock, being the latter the one dependable of the optimization parameters. In other words, the production volume lies in the point of modification. Additionally, the problem is bounded in such a way the production of each single-car variant cannot be negative and the initial volume should be respected. It is not possible to add or remove any vehicle from the manufacturing line.

Applying all these steps, a simulation has been carried out. It has taken place in the test period from each one of the time chunks. It was seeked the subset of four weeks required by the optimization. Nevertheless, the simulation was launched independently for each one of the modification dates. In all cases under analysis, there has been an improvement of the objective function after applying the optimization of the production. The unique exceptions are a consequence of the null production for the car model at that epoch. Moreover, it has been understood that the upgrading routine makes a rearrangement of the current production or concentrates it into a single destination.

On the contrary, this simulation has its limitations. It is not a faithful representation of reality. The manufacturing policy is ruled by commercial interests, but also by private ones. Currently, it is difficult to imagine a work-frame in which dealerships from one region are willing to transfer their orders to the dealerships from a different place. Especially, if the first is not rewarded in any manner, besides it might represent an adequate business approach from the point of view of economics. Cars are placed in the most likely location to be sold, rather than occupying valuable space and offering a discount so the spot is free. That's why the company is migrating to an agency model, in which the firm will have the ownership of the vehicle they manufacture and dealerships will play the role of advisors and sales point. Under this scenario, this optimization protocol is welcomed and extended to be used for more restrictive attributes and longer timeframes. For instance, the wheels, which are a component of the cars that depend on a supplier. This actor has its own logistics and the flexibility required to execute the optimization procedure deserves its own research.

Chapter 8

Conclusions

From the early stages of this research, the focus has been on understanding what are the needs and problems of the automobile industry. One of the main concerns for any company is learning how to fulfill the expectations of their clients. That's why the document begins with a chapter dedicated to the customers. It has been revealed that the automobile industry is a market very well segmented, but with a common bond. The company that delivers the best customer experience holds a competitive edge over its rivals. In this manner, this research is framed together with a well-known car manufacturer, i.e., SEAT S.A. The brand is looking to improve its operation. Therefore, the document continues with the idiosyncrasy of the manufacturer, such as their most popular products, the lay-up, or the production flow within the main factory.

From the latter part, it is especially relevant the section about the company's car distribution system, based on compound regions or warehouse stock locations. The first contribution of the investigation consisted of proposing a reallocation of Build to Stock vehicles based on where they are projected to spend the minimal time. It is sought to enhance the purchasing likelihood whilst mitigating the logistic costs. Afterward the data exploration, the problem has been settled as a binary classification task, on which the car attributes (car model, trim level, exterior color, and engine, together with order type) are the features. On the other side, the labels consisted of, according to a time threshold, tagging the vehicle as a Fast Delivery or Normal Delivery within the compound region of destination. Four popular Machine Learning algorithms (Decision Tree, Random Forest, XGBoost, and CatBoost) have been under analysis to accomplish this task, all of them trained in cross-validation and hyperparameters fine-tuned. Lastly, it has been set the assessment metric, in the form of f1-score. The decision is motivated by capturing as many positive classes as possible, but ensuring the classification is correctly performed. Therefore, the experiments can take place, resulting in XGBoost with a time threshold of 42 days as the winning candidate, delivering a result of 0.781 in the f1-score. The features relevance analysis shows that order type, together with one of the compound regions available, are the most relevant terms to base the decision of where vehicles should be headed. Lastly, the classification algorithm is employed to perform the reallocation strategy. The outcomes of these last steps prove to be equal to or better than the ones delivered by the experts, in terms of the median average time spent in each one of the compound regions. These findings were presented in CCIA 2022 [12].

Despite the promising results, the previous block does not include into consideration the information from the potential customers. A new perspective begins. Firstly, it is hypothesized whether these insights can be extracted from the information gathered by the Car Configurator webpage. However, this data source carries with it some concerns, related to noisy and irrelevant information, that are managed next. In the first place, the approach has been quantitative, whilst a qualitative view is given in the upcoming part.

As was aforementioned, the first hypothesis to be validated consists of measuring the reliability of the Car Configurator webpage to capture, in advance, customers' demand. The initial part of the investigation was in charge of gauging the correlation between the clickstream data and the sales record at different granular levels. On one side, at the car model level. On the other hand, the granularity has expanded to both car model and color or compound region. Results are consistent at different granular levels. Users browse the online tool within the timeframe of a semester before the purchase date. Secondly, the forecasting capacity of the online data source is under evaluation. For the second level of granularity, and at different test periods built from splitting the data into time chunks, a comparison has been made. Two pairs of forecasting algorithms were trained and assessed. The first group, called univariate, are algorithms based exclusively on past sales records. The second set includes the information of the clickstream date, that's why they are called multivariate. Along all the steps of the assessment process, which includes forecast error and weekly mix sales, the multivariate algorithms deliver better performance, despite the outcomes being more robust in terms of exterior color granularity rather than compound region. These discoveries were published in Forecasting journal [11].

This path was initially explored by the colleagues at DataLab. Their outcomes would have served as a valuable benchmark if the project had continued beyond the initial testing phase. However, comparing the results would have presented certain challenges. DataLab's focus was specifically tied to the vehicles' cable tree and its coverage among customers. This scope significantly diverges from the objectives of this thesis, which concentrates on a different aspect of the problem. Therefore, despite the potential benefits of benchmarking against DataLab's findings, the distinct goals and methodologies of each initiative necessitate separate evaluation criteria and success metrics.

Afterward accomplishing the first objective, it was learned that dealing with the data source is not straightforward. The dataset is massive and it has values that only add noise to the field. The reason behind this is that the webpage is non-transactional. It is not possible to execute a purchase online. Moreover, two groups of users can be distinguished: (a) people who are in the early stages of the acquisition process of a new vehicle, i.e., they do have real purchase intention; (b) people who are doing window shopping, i.e., exploring the company's product catalog without the intention of buy. The dilemma lies on the online service does not save information from their users, in the form of a mandatory login, that permits them to link the online activity with the requests in the dealerships. Therefore, it is not possible to differentiate these two clusters. This concern motivates the next two chapters of the thesis. On the first try, the clickstream data has been filtered according to some rules, based on identifying and removing the outliers. These rules are based on how users have interacted with the tool, i.e., the number of car variants configured and days between the first and last connection; rather than what they have configured exactly. It is expected that the reduced datasets, created by means of these filtering rules, have the same significance as the raw one. This magnitude is defined as the weekly lagged correlation between the clickstream data and the sales record. The outcomes have proved that significance is preserved after removing the outliers element.

The second attempt is more oriented toward understanding what car variants trigger the purchase intention of the users. In other words, the approach is executed from a qualitative point of view. As the search space is truly wide and extensive, it has been relied on the power of genetic algorithms to fulfill this task. They are a type of optimization algorithm inspired by the survival of the fittest. From an initial population of feasible solutions, it evolves by means of different mechanisms (selection, crossover, mutation) until finding the optimal solution. Therefore, this idea has been employed to identify

the car variants that originate the clickstream dataset with the largest averaged lagged weekly correlation with respect to sales records. This approach has been assessed at both general and compound region levels. In order to guarantee the reliability of our findings, five independent trials under identical conditions take place. Thereupon, outcomes are evaluated against the benchmark value. The latter is achieved thanks to applying the fitness function to the plain clickstream data. In the overall scenario, there has been an enhancement from a fitness score of 29.42 to an average of 91.21 ± 3.8. On the specific compound region scale, the average improvement in fitness is 66.25 ± 1.46.

After deconstructing and evaluating the winning candidate, the results indicate that the most popular items from the Car Configurator webpage and sales record may not necessarily serve as the most reliable indicators of potential consumers among users of online tool. Conversely, when investigation occurs at the compound region level, the locations belonging to the compound regions are underrepresented compared to those that are outside of it. Nevertheless, the examination of individual fitness scores for both groups yielded comparable results, as confirmed by a Kolmogorov-Smirnov statistical test, with just one exception. Despite an extensive analysis of the data, the underlying reasons for this behavior remain elusive with the information available. We recommend exploring demographic data, including social and economic factors, as an external data source that might provide insights into this matter. This research produced two publications [13, 14].

Finally, all the discoveries from the previous parts are assembled in this last section. The flexibility of the genetic algorithm is merged with the potency of forecasting. The objective is to demonstrate that superior results in demand forecasting can be attained. Later on, these outputs will be utilized to update the cars under production in the manufacturing line.

Therefore, the adaptation of the genetic algorithm needs to be carried out. In this scenario, the fitness function decreases the prediction error of the car variant under analysis, rather than maximizing the correlation. The forecasting technique running under the genetic algorithm has been the XGBoost, in the multivariate version. The reference frame has been the predictions done at the car model and compound region granular level, as their performance is the least robust. The formula of executing five independent trials per each one of the experiments is preserved, as well as the search space. In consequence, the attempt pursues to find the experiment with the largest figure of merit. In other words, the number of instances where forecast error was reduced compared to the initial benchmark. It has been observed that the figure of merit's magnitude evolves together with the experiment's level of complexity. Once, the leading candidate was found the assessment procedure, in terms of weekly mixes sales, is executed again. Their outputs confirm the efficiency of the proposed solutions. The latter ones are used as input for the upgrading of the final destination of Build to Stock vehicles in production. The problem is settled in terms of the objective function, constraints, and boundaries. It is sought to minimize the square difference between the estimated stock composition and the future demand. In this case, the situation is faced mathematically with convex optimization, instead of employing heuristic algorithms. The simulation is framed within the modification dates for each one of the test periods. Under this scenario, the updating of the compound region destination of the vehicles outperforms the figures achieved by the original situation. The single exceptions are those cases in which there was no production available.

Nevertheless, this simulation comes with its set of limitations as it does not provide a fully accurate picture of reality. The manufacturing policy is influenced not only by commercial considerations but also by private interests. At present, it's challenging to envision a framework where dealerships from one region would be willing to transfer their orders to dealerships in a different location. Consequently, the company is transitioning to an agency model. I mean, the company retains ownership of the vehicles it produces, and dealerships primarily serve as advisors and sales points.

In this context, the optimization protocol is being embraced and expanded to accommodate more restrictive parameters and longer timeframes. The new paradigm would permit to assessment the steps of the proposed solution in a real environment under controlled conditions. Current production would be improved according to the indications of a better forecast fed by online visitors' insights, and the logistics would take place freely, allocating vehicles to the most suitable location. After all the aforesaid, the company views the research conducted in this thesis very positively and is already integrating it into its systems and operations.

Additionally, it is filling a gap in the existing literature, to the best of our knowledge. Despite the study being constrained to one particular car brand, it has faced challenges shared by multiple companies. The timing control in production, despite variations in each manufacturer's production flow, emerges as a central theme. Moreover, the transformation from a sales forecast to a production planning is not straightforward since external factors should be taken into consideration, for instance, the restrictions and limitations in the manufacturing line or with the suppliers. Furthermore, we emphasize the importance of considering both Build-to-Order (BTO) and Build-to-Stock (BTS) strategies in addressing efficient operation in the production and delivery tasks. The acknowledgment of the Dealer effect sheds light on the impact of dealership networks on sales and production dynamics. They are an external agent that assists but influence the customers in their decision-making process. Notably, the ubiquitous presence of Car Configurators across manufacturers highlights the potential for data extraction despite the often opaque nature of this information, on which the thesis mitigates these concerns.

The comparison of all these findings with respect to the current practices in the industry is aligned. For instance, the technological hub of SEAT, named SEAT:CODE, is dedicating its efforts in this direction. Digitising all SEAT operations, from production to marketing, is the first of its challenges. The second is to maximize user experience with brands, from purchase to customer services. With the data of the connected vehicle, they are working on algorithms for searching for use cases [159, 160]. Additionally, our findings extend beyond the automotive sector, offering insights applicable to analogous industries, thus broadening the scope and relevance of our research beyond its immediate context. The use of Artificial Intelligence and Big Data, as they were utilized during the research, together with other technologies, such as Cloud Computing and Smart Factories... is leading the automation and data exchange in manufacturing technologies. This trend is called Industry 4.0. The goals of Industry 4.0 include increasing automation, improving communication and monitoring, enabling self-diagnosis and self-maintenance of machines, and facilitating flexible and efficient production processes. This industrial revolution aims to create more adaptive, efficient, and responsive manufacturing systems that can meet the demands of a rapidly changing market [161, 162, 163, 164]. Lastly, this research has resulted in one publication in a top-quartile indexed journal, as well as presentations at three internationally recognized congresses, besides more documentation waiting to be released.

Appendix A

Correlation Analysis

The information exposed in this appendix supplement the results shown in Subsection 6.2.1.

Table A.1: Pearson correlation coefficient (PCC) between lagged sales record and online visits of SEAT Arona colors (1/4)

	SEAT ARONA											
Lag	0C0C	0C0E	0CF5	0E0C	0E0E	0EF5	0ES7	2Y0C	2Y0E	2Y2Y	2YF5	2YS7
0	0.58	0.65	0.06	0.29	0.54	0.26	0.41	0.71	0.25	0.43	0.08	0.57
1	0.61	0.67	0.12	0.34	0.58	0.21	0.41	0.69	0.3	0.48	0.14	0.58
2	0.62	0.67	0.15	0.4	0.57	0.14	0.44	0.69	0.29	0.48	0.13	0.66
3	0.64	0.68	0.18	0.38	0.53	0.18	0.39	0.68	0.26	0.46	0.15	0.65
4	0.67	0.69	0.1	0.37	0.5	0.16	0.31	0.59	0.28	0.47	0.16	0.7
5	0.7	0.74	0.09	0.41	0.51	0.2	0.26	0.61	0.32	0.51	0.08	0.69
6	0.68	0.74	0.09	0.4	0.46	0.22	0.29	0.55	0.37	0.52	0.07	0.7
7	0.65	0.75	0.15	0.46	0.47	0.29	0.23	0.51	0.33	0.49	0.16	0.66
8	0.61	0.74	0.18	0.49	0.42	0.12	0.28	0.47	0.28	0.47	0.12	0.62
9	0.62	0.74	0.29	0.5	0.48	0.15	0.29	0.44	0.34	0.54	0.13	0.6
10	0.61	0.69	0.39	0.51	0.47	0.25	0.31	0.36	0.34	0.55	0.16	0.62
11	0.62	0.67	0.44	0.47	0.41	0.25	0.27	0.39	0.31	0.5	0.22	0.6
12	0.63	0.66	0.4	0.56	0.4	0.26	0.22	0.35	0.28	0.44	0.1	0.54
13	0.61	0.68	0.37	0.66	0.4	0.15	0.24	0.34	0.32	0.42	0.09	0.54
14	0.61	0.64	0.36	0.6	0.33	0.17	0.23	0.36	0.35	0.44	0.09	0.49
15	0.54	0.58	0.29	0.44	0.33	0.22	0.2	0.36	0.37	0.48	0.14	0.46
16	0.48	0.54	0.32	0.47	0.29	0.32	0.23	0.33	0.3	0.45	0.18	0.44
17	0.43	0.47	0.31	0.48	0.26	0.26	0.21	0.28	0.26	0.46	0.15	0.44
18	0.42	0.41	0.36	0.37	0.25	0.17	0.22	0.25	0.31	0.5	0.3	0.44
19	0.41	0.37	0.32	0.33	0.3	0.18	0.26	0.24	0.35	0.51	0.26	0.38
20	0.43	0.3	0.34	0.32	0.23	0.21	0.26	0.21	0.32	0.5	0.2	0.39
21	0.4	0.3	0.25	0.25	0.17	0.13	0.23	0.19	0.33	0.46	0.22	0.4
22	0.4	0.29	0.29	0.2	0.19	0.2	0.27	0.14	0.42	0.46	0.11	0.37
23	0.38	0.27	0.21	0.25	0.19	0.13	0.26	0.13	0.44	0.52	0.14	0.35
24	0.32	0.22	0.22	0.18	0.17	0.17	0.3	0.11	0.38	0.49	0.11	0.31
25	0.24	0.12	0.16	0.18	0.11	0.16	0.29	0.07	0.34	0.42	0.19	0.22
26	0.25	0.11	0.16	0.16	0.15	0.15	0.29	0.06	0.45	0.48	0.18	0.21
27	0.28	0.05	0.14	0.21	0.18	0.17	0.31	0.08	0.51	0.44	0.11	0.19
28	0.28	0.03	0.17	0.3	0.17	0.16	0.32	0.06	0.5	0.48	0.11	0.2
29	0.26	0.03	0.08	0.17	0.15	0.17	0.26	-0.0	0.27	0.42	0.04	0.17
30	0.24	0.01	0.12	-0.0	0.14	0.14	0.29	-0.01	0.28	0.45	0.12	0.19
31	0.26	-0.01	0.18	-0.06	0.22	0.16	0.33	0.01	0.33	0.48	0.37	0.2
32	0.24	-0.02	0.17	-0.03	0.23	0.14	0.37	-0.03	0.37	0.47	0.22	0.2
33	0.22	-0.01	0.22	-0.1	0.21	0.32	0.4	-0.08	0.32	0.36	0.24	0.19
34	0.17	-0.02	0.17	-0.19	0.06	0.3	0.36	-0.14	0.32	0.32	0.17	0.22
35	0.2	-0.05	0.17	-0.19	0.04	0.18	0.31	-0.15	0.45	0.34	0.11	0.19
36	0.25	-0.04	0.31	-0.19	0.07	0.24	0.31	-0.11	0.39	0.34	0.15	0.15
37	0.25	-0.02	0.19	-0.19	0.03	0.17	0.27	-0.13	0.27	0.31	0.2	0.15
38	0.19	-0.02	0.09	-0.2	-0.04	0.09	0.2	-0.25	0.15	0.26	0.2	0.13
39	0.2	-0.03	0.12	-0.2	0.0	0.11	0.28	-0.26	0.23	0.3	0.12	0.16
40	0.22	-0.05	0.12	-0.2	0.01	0.18	0.23	-0.26	0.27	0.37	0.14	0.13
41	0.18	-0.08	0.19	-0.2	-0.02	0.24	0.26	-0.24	0.32	0.38	0.14	0.13
42	0.13	-0.1	0.06	-0.21	-0.05	0.26	0.26	-0.25	0.26	0.33	0.1	0.16
43	0.11	-0.09	0.16	-0.21	-0.1	0.0	0.15	-0.29	0.24	0.26	0.04	0.11
44	0.15	-0.07	0.25	-0.2	-0.06	0.04	0.18	-0.29	0.29	0.29	0.15	0.1
45	0.16	-0.08	0.13	-0.2	0.0	-0.02	0.06	-0.29	0.3	0.29	0.14	0.09
46	0.11	-0.09	0.22	-0.2	-0.06	0.07	0.04	-0.27	0.24	0.27	0.24	0.06
47	-0.01	-0.13	0.08	-0.2	-0.13	0.04	0.03	-0.29	0.14	0.08	0.13	0.02
48	0.01	-0.13	0.13	-0.2	-0.1	0.13	0.03	-0.31	0.18	0.11	0.17	0.04
49	0.01	-0.13	0.18	-0.2	-0.12	0.01	0.11	-0.3	0.19	0.16	0.18	0.03
50	-0.06	-0.17	0.18	-0.21	-0.11	0.06	0.12	-0.28	0.18	0.12	0.25	-0.01
51	-0.09	-0.19	0.17	-0.21	-0.15	0.03	0.12	-0.3	0.21	0.06	0.12	0.01
52	-0.07	-0.17	0.07	-0.2	-0.11	-0.01	0.16	-0.3	0.31	0.08	0.09	-0.0

Table A.2: Pearson correlation coefficient (PCC) between lagged sales record and online visits of SEAT Arona colors (2/4)

	SEAT ARONA											
Lag	7Y0C	7Y0E	7Y7Y	7YS7	9529	9532	9545	9550	9M9M	B40C	B40E	B4B4
0	0.24	0.45	0.39	0.5	0.34	0.59	0.48	0.66	0.77	0.78	0.41	0.39
1	0.25	0.47	0.39	0.49	0.42	0.56	0.53	0.69	0.8	0.8	0.47	0.44
2	0.28	0.53	0.4	0.46	0.41	0.5	0.6	0.73	0.78	0.8	0.5	0.46
3	0.29	0.43	0.36	0.46	0.34	0.56	0.59	0.7	0.78	0.78	0.5	0.42
4	0.18	0.44	0.45	0.51	0.38	0.56	0.58	0.65	0.76	0.78	0.51	0.4
5	0.23	0.45	0.46	0.5	0.37	0.57	0.57	0.68	0.76	0.8	0.54	0.45
6	0.26	0.48	0.45	0.52	0.39	0.54	0.6	0.66	0.76	0.75	0.51	0.46
7	0.38	0.46	0.42	0.57	0.42	0.59	0.53	0.61	0.78	0.67	0.48	0.44
8	0.4	0.45	0.36	0.51	0.4	0.51	0.48	0.5	0.78	0.61	0.44	0.36
9	0.33	0.51	0.39	0.57	0.37	0.46	0.49	0.56	0.79	0.63	0.49	0.41
10	0.28	0.53	0.36	0.52	0.39	0.48	0.47	0.59	0.82	0.62	0.49	0.43
11	0.35	0.52	0.4	0.59	0.31	0.41	0.43	0.59	0.77	0.59	0.49	0.42
12	0.34	0.49	0.34	0.55	0.31	0.41	0.39	0.56	0.77	0.53	0.46	0.36
13	0.35	0.48	0.37	0.56	0.33	0.37	0.35	0.48	0.76	0.51	0.48	0.39
14	0.28	0.49	0.34	0.55	0.29	0.4	0.31	0.47	0.74	0.49	0.43	0.42
15	0.3	0.52	0.37	0.52	0.3	0.39	0.22	0.41	0.74	0.45	0.38	0.41
16	0.39	0.49	0.4	0.54	0.24	0.38	0.17	0.35	0.76	0.41	0.3	0.36
17	0.36	0.52	0.31	0.54	0.26	0.38	0.18	0.3	0.74	0.39	0.31	0.34
18	0.37	0.53	0.3	0.54	0.25	0.34	0.19	0.29	0.72	0.38	0.3	0.38
19	0.33	0.53	0.35	0.51	0.14	0.31	0.17	0.32	0.65	0.35	0.27	0.36
20	0.22	0.56	0.36	0.52	0.16	0.21	0.16	0.28	0.65	0.32	0.21	0.36
21	0.33	0.49	0.34	0.5	0.22	0.24	0.12	0.23	0.64	0.28	0.19	0.34
22	0.23	0.52	0.36	0.55	0.24	0.27	0.05	0.2	0.61	0.27	0.21	0.38
23	0.3	0.52	0.41	0.58	0.23	0.22	0.01	0.19	0.6	0.24	0.19	0.4
24	0.19	0.51	0.42	0.57	0.2	0.18	-0.01	0.21	0.56	0.19	0.13	0.41
25	0.25	0.44	0.39	0.52	0.08	0.14	-0.0	0.15	0.56	0.14	0.06	0.31
26	0.31	0.47	0.39	0.53	0.09	0.06	-0.03	0.09	0.5	0.13	0.06	0.3
27	0.3	0.48	0.45	0.48	0.05	0.05	-0.05	0.06	0.52	0.13	0.05	0.3
28	0.24	0.47	0.5	0.44	0.04	0.04	-0.08	0.02	0.55	0.1	-0.01	0.29
29	0.18	0.49	0.48	0.41	-0.01	-0.01	-0.14	-0.07	0.46	0.07	-0.05	0.16
30	0.25	0.49	0.44	0.41	-0.01	0.01	-0.21	-0.12	0.46	0.05	-0.07	0.13
31	0.25	0.54	0.43	0.36	-0.04	0.01	-0.27	-0.18	0.43	0.05	-0.05	0.19
32	0.37	0.58	0.52	0.31	0.0	0.01	-0.29	-0.15	0.0	0.03	-0.07	0.22
33	0.14	0.54	0.41	0.27	-0.01	-0.05	-0.24	-0.17	0.0	0.01	-0.09	0.17
34	-0.04	0.46	0.34	0.24	0.01	-0.03	-0.23	-0.19	0.0	0.01	-0.09	0.04
35	-0.01	0.54	0.4	0.2	-0.09	-0.06	-0.25	-0.2	0.0	0.05	-0.03	0.16
36	0.02	0.57	0.42	0.13	-0.14	-0.06	-0.23	-0.15	0.0	0.08	0.02	0.22
37	0.07	0.58	0.41	0.13	-0.16	-0.02	-0.17	-0.16	0.0	0.05	0.01	0.21
38	-0.05	0.46	0.32	0.04	-0.16	-0.08	-0.16	-0.28	0.0	0.01	-0.0	0.05
39	-0.05	0.54	0.18	-0.0	-0.14	-0.05	-0.14	-0.28	0.0	-0.01	0.03	0.14
40	-0.06	0.52	0.23	-0.04	-0.13	-0.04	-0.03	-0.26	0.0	-0.04	0.07	0.21
41	-0.05	0.55	0.32	-0.09	-0.19	-0.03	-0.01	-0.24	0.0	-0.08	0.1	0.19
42	-0.02	0.48	0.26	-0.13	-0.24	-0.03	0.06	-0.24	0.0	-0.08	0.07	0.13
43	-0.12	0.38	0.11	-0.15	-0.25	-0.04	0.11	-0.3	0.0	-0.08	0.06	0.07
44	-0.09	0.46	0.15	-0.15	-0.27	0.04	0.16	-0.29	0.0	-0.07	0.12	0.16
45	-0.12	0.52	0.17	-0.17	-0.28	-0.01	0.2	-0.27	0.0	-0.11	0.13	0.11
46	-0.08	0.43	0.15	-0.17	-0.28	-0.02	0.19	-0.24	0.0	-0.16	0.06	0.08
47	-0.19	0.28	0.08	-0.19	-0.29	-0.14	0.1	-0.31	0.0	-0.22	-0.07	-0.11
48	-0.2	0.32	0.08	-0.19	-0.3	-0.14	0.13	-0.3	0.0	-0.21	0.01	-0.06
49	-0.2	0.34	0.11	-0.2	-0.3	-0.14	0.09	-0.3	0.0	-0.21	0.06	-0.02
50	-0.18	0.38	0.08	-0.2	-0.3	-0.16	0.13	-0.29	0.0	-0.24	0.0	-0.09
51	-0.17	0.34	0.06	-0.2	-0.29	-0.18	0.05	-0.35	0.0	-0.25	-0.13	-0.15
52	-0.15	0.39	0.05	-0.19	-0.29	-0.14	0.11	-0.33	0.0	-0.24	-0.08	-0.15

Table A.3: Pearson correlation coefficient (PCC) between lagged sales record and online visits of SEAT Arona colors (3/4)

	SEAT ARONA											
Lag	B4F5	B4S7	E10C	E10E	E1E1	E1S7	F50C	F50E	F5F5	F5S7	F80C	F80E
0	0.13	0.66	0.41	0.38	0.44	0.55	0.38	0.22	0.32	0.49	0.51	0.31
1	0.17	0.66	0.4	0.43	0.47	0.55	0.46	0.21	0.37	0.51	0.52	0.34
2	0.09	0.68	0.38	0.46	0.5	0.6	0.52	0.26	0.39	0.54	0.53	0.34
3	0.06	0.67	0.37	0.46	0.44	0.58	0.54	0.27	0.4	0.5	0.55	0.33
4	0.12	0.67	0.36	0.45	0.39	0.65	0.55	0.3	0.38	0.47	0.57	0.33
5	0.27	0.68	0.38	0.5	0.42	0.62	0.6	0.3	0.48	0.47	0.55	0.38
6	0.26	0.69	0.34	0.43	0.4	0.65	0.47	0.28	0.47	0.47	0.52	0.39
7	0.3	0.64	0.29	0.39	0.41	0.62	0.53	0.28	0.35	0.49	0.56	0.38
8	0.25	0.63	0.25	0.36	0.38	0.64	0.52	0.27	0.28	0.47	0.56	0.38
9	0.28	0.65	0.23	0.47	0.47	0.67	0.57	0.29	0.33	0.5	0.62	0.43
10	0.25	0.65	0.21	0.52	0.47	0.64	0.53	0.28	0.27	0.48	0.65	0.46
11	0.14	0.67	0.21	0.45	0.44	0.62	0.57	0.25	0.25	0.45	0.68	0.4
12	0.15	0.59	0.18	0.37	0.35	0.49	0.57	0.24	0.18	0.42	0.6	0.44
13	0.25	0.61	0.16	0.39	0.33	0.52	0.48	0.27	0.22	0.46	0.61	0.41
14	0.35	0.6	0.15	0.39	0.34	0.49	0.39	0.27	0.17	0.38	0.56	0.45
15	0.36	0.59	0.12	0.36	0.3	0.45	0.4	0.19	0.16	0.45	0.57	0.43
16	0.3	0.55	0.07	0.27	0.3	0.41	0.42	0.11	0.16	0.45	0.57	0.44
17	0.12	0.55	0.06	0.31	0.28	0.4	0.4	0.15	0.14	0.37	0.59	0.45
18	0.02	0.57	0.05	0.28	0.31	0.39	0.44	0.17	0.17	0.4	0.54	0.41
19	-0.01	0.59	0.02	0.25	0.28	0.4	0.43	0.13	0.13	0.34	0.55	0.36
20	0.11	0.58	0.01	0.19	0.25	0.36	0.38	0.07	-0.01	0.39	0.51	0.29
21	0.15	0.51	0.01	0.12	0.25	0.35	0.3	0.09	0.01	0.32	0.38	0.19
22	0.19	0.54	-0.01	0.14	0.27	0.36	0.3	0.08	0.04	0.34	0.36	0.25
23	0.23	0.53	-0.03	0.13	0.29	0.37	0.27	0.04	0.13	0.38	0.4	0.34
24	0.14	0.55	-0.06	0.11	0.26	0.41	0.21	0.0	-0.01	0.29	0.44	0.32
25	0.04	0.47	-0.09	0.05	0.18	0.35	0.2	-0.01	-0.04	0.35	0.39	0.32
26	-0.0	0.45	-0.11	0.06	0.15	0.39	0.18	0.01	-0.04	0.27	0.34	0.2
27	0.03	0.4	-0.11	0.08	0.14	0.36	0.15	0.0	0.01	0.28	0.31	0.22
28	0.05	0.42	-0.13	0.0	0.12	0.41	0.08	0.01	-0.1	0.19	0.22	0.19
29	0.13	0.35	-0.15	-0.11	0.08	0.38	0.08	0.04	-0.14	0.16	0.18	0.13
30	0.05	0.33	-0.15	-0.15	0.09	0.39	0.07	0.04	-0.16	0.17	0.13	0.18
31	0.09	0.32	-0.16	-0.13	0.18	0.41	0.03	0.03	-0.15	0.17	0.18	0.18
32	0.05	0.32	-0.17	-0.15	0.19	0.41	0.03	0.01	-0.19	0.21	0.18	0.18
33	-0.05	0.34	-0.18	-0.16	0.11	0.43	-0.0	-0.02	-0.24	0.15	0.11	0.07
34	-0.14	0.28	-0.2	-0.24	0.03	0.41	-0.04	-0.01	-0.2	0.18	-0.02	0.02
35	-0.03	0.31	-0.2	-0.11	0.09	0.48	0.02	0.04	-0.13	0.24	-0.05	0.06
36	0.09	0.28	-0.2	-0.03	0.19	0.46	0.07	0.11	-0.12	0.19	-0.04	0.09
37	0.17	0.29	-0.21	-0.07	0.22	0.43	0.14	0.13	-0.14	0.2	-0.01	0.01
38	0.04	0.24	-0.25	-0.19	0.14	0.41	0.04	0.1	-0.14	0.13	-0.1	0.04
39	-0.03	0.27	-0.25	-0.08	0.2	0.45	-0.02	0.15	-0.09	0.18	-0.08	0.04
40	-0.03	0.29	-0.26	0.0	0.18	0.41	-0.04	0.21	-0.14	0.16	-0.07	0.02
41	-0.0	0.3	-0.27	-0.02	0.13	0.37	-0.04	0.17	-0.13	0.23	-0.09	-0.06
42	0.02	0.22	-0.28	-0.1	0.08	0.3	-0.03	0.14	-0.09	0.19	-0.12	0.06
43	0.05	0.22	-0.27	-0.07	0.11	0.27	-0.01	0.14	-0.04	0.09	-0.13	-0.03
44	0.05	0.21	-0.27	0.07	0.25	0.31	0.04	0.15	0.06	0.16	-0.1	0.01
45	0.14	0.18	-0.27	0.04	0.28	0.26	0.03	0.13	-0.06	0.14	-0.09	-0.03
46	0.01	0.19	-0.28	-0.03	0.27	0.25	0.06	0.1	-0.11	0.14	-0.03	-0.01
47	-0.09	0.15	-0.29	-0.16	0.05	0.24	-0.1	0.06	-0.16	0.05	-0.09	-0.06
48	-0.06	0.14	-0.3	0.02	0.04	0.24	-0.03	0.06	-0.12	0.05	-0.09	0.03
49	-0.02	0.15	-0.31	0.04	0.04	0.24	-0.08	-0.02	-0.07	0.07	-0.06	0.01
50	-0.07	0.16	-0.31	-0.02	-0.03	0.24	-0.09	-0.08	-0.22	0.08	-0.08	0.08
51	0.0	0.13	-0.31	-0.12	-0.06	0.22	-0.1	-0.14	-0.22	0.13	-0.09	-0.08
52	-0.0	0.16	-0.3	-0.01	-0.02	0.27	-0.06	-0.11	-0.15	0.1	-0.07	0.01

Table A.4: Pearson correlation coefficient (PCC) between lagged sales record and online visits of SEAT Arona colors (4/4)

	SEAT ARONA									
Lag	F8F8	F8S7	L50C	L50E	L5F5	L5L5	L5S7	S70E	S7F5	S7S7
0	0.48	0.58	0.54	0.52	0.15	0.62	0.44	0.56	0.34	0.59
1	0.5	0.56	0.64	0.54	0.07	0.66	0.42	0.59	0.28	0.6
2	0.5	0.56	0.61	0.5	0.09	0.63	0.38	0.6	0.3	0.61
3	0.5	0.58	0.54	0.49	0.08	0.62	0.47	0.62	0.26	0.61
4	0.58	0.55	0.44	0.48	0.01	0.62	0.5	0.59	0.39	0.6
5	0.63	0.56	0.49	0.51	0.04	0.56	0.49	0.61	0.34	0.61
6	0.58	0.58	0.41	0.53	0.01	0.59	0.52	0.59	0.3	0.61
7	0.61	0.52	0.38	0.51	0.01	0.54	0.49	0.58	0.26	0.63
8	0.57	0.49	0.35	0.53	0.07	0.55	0.47	0.61	0.32	0.62
9	0.58	0.47	0.39	0.57	0.11	0.62	0.5	0.58	0.41	0.65
10	0.53	0.44	0.3	0.57	0.08	0.62	0.49	0.56	0.35	0.64
11	0.51	0.39	0.29	0.61	0.08	0.62	0.49	0.57	0.37	0.65
12	0.49	0.36	0.14	0.59	-0.05	0.51	0.47	0.57	0.28	0.63
13	0.51	0.44	0.14	0.62	-0.05	0.5	0.6	0.6	0.43	0.66
14	0.54	0.33	0.2	0.59	0.1	0.49	0.57	0.58	0.36	0.65
15	0.45	0.26	0.18	0.61	0.12	0.49	0.58	0.58	0.29	0.65
16	0.44	0.3	0.08	0.64	0.03	0.45	0.51	0.54	0.25	0.64
17	0.46	0.22	0.11	0.64	-0.02	0.45	0.51	0.54	0.25	0.67
18	0.41	0.17	0.17	0.65	0.03	0.48	0.44	0.56	0.28	0.71
19	0.4	0.15	0.19	0.65	0.05	0.48	0.41	0.58	0.31	0.7
20	0.31	0.19	0.14	0.63	0.03	0.43	0.39	0.58	0.24	0.71
21	0.29	0.11	0.11	0.55	0.13	0.38	0.34	0.6	0.18	0.72
22	0.3	0.13	0.11	0.59	0.07	0.41	0.39	0.59	0.18	0.75
23	0.32	0.15	0.17	0.59	0.02	0.4	0.44	0.59	0.27	0.75
24	0.3	0.13	0.19	0.61	0.23	0.42	0.53	0.59	0.28	0.74
25	0.21	0.1	0.02	0.61	0.08	0.37	0.43	0.59	0.17	0.69
26	0.2	0.13	0.02	0.64	-0.07	0.35	0.53	0.62	0.18	0.68
27	0.17	0.01	0.01	0.65	-0.02	0.32	0.38	0.6	0.13	0.61
28	0.13	0.01	0.05	0.65	-0.14	0.27	0.37	0.61	0.24	0.59
29	0.08	-0.03	-0.0	0.54	-0.09	0.19	0.35	0.54	0.23	0.54
30	0.04	-0.06	-0.07	0.53	0.02	0.23	0.35	0.59	0.23	0.5
31	0.06	-0.12	-0.07	0.62	0.14	0.23	0.34	0.58	0.21	0.48
32	0.02	-0.14	-0.05	0.58	0.15	0.2	0.41	0.63	0.24	0.47
33	-0.07	-0.15	-0.03	0.55	0.03	0.15	0.34	0.66	0.2	0.42
34	-0.08	-0.18	-0.11	0.48	0.14	0.13	0.34	0.6	0.32	0.36
35	-0.06	-0.18	-0.14	0.56	-0.04	0.16	0.44	0.63	0.26	0.37
36	-0.09	-0.22	-0.13	0.53	-0.11	0.13	0.37	0.6	0.18	0.34
37	-0.06	-0.25	-0.13	0.53	-0.04	0.12	0.37	0.6	0.19	0.3
38	-0.1	-0.24	-0.17	0.48	-0.17	0.03	0.39	0.57	0.25	0.24
39	0.0	-0.24	-0.18	0.52	-0.04	-0.03	0.39	0.56	0.34	0.23
40	-0.0	-0.24	-0.17	0.51	-0.03	0.01	0.26	0.53	0.17	0.23
41	0.04	-0.25	-0.15	0.5	0.06	-0.03	0.33	0.57	0.19	0.21
42	-0.05	-0.29	-0.19	0.46	0.04	-0.08	0.29	0.52	0.28	0.17
43	-0.05	-0.3	-0.22	0.46	-0.04	-0.09	0.34	0.47	0.19	0.16
44	0.01	-0.31	-0.2	0.45	0.07	-0.09	0.29	0.55	0.2	0.17
45	-0.06	-0.31	-0.24	0.41	0.17	-0.07	0.3	0.51	0.15	0.16
46	-0.09	-0.28	-0.24	0.39	0.05	-0.03	0.2	0.53	0.24	0.14
47	-0.1	-0.29	-0.23	0.28	-0.15	-0.18	0.19	0.44	0.06	0.12
48	-0.01	-0.32	-0.23	0.26	-0.04	-0.13	0.21	0.47	0.24	0.13
49	-0.12	-0.31	-0.22	0.27	-0.01	-0.09	0.18	0.47	0.18	0.12
50	-0.13	-0.3	-0.22	0.31	-0.01	-0.1	0.19	0.45	0.25	0.12
51	-0.19	-0.28	-0.21	0.27	0.0	-0.1	0.16	0.46	0.13	0.12
52	-0.18	-0.28	-0.21	0.29	-0.11	-0.09	0.18	0.49	0.25	0.12

Table A.5: Pearson correlation coefficient (PCC) between lagged sales record and online visits of SEAT Ibiza colors

	SEAT IBIZA											
Lag	0C0C	0E0E	2Y2Y	7Y7Y	9550	9M9M	B4B4	E1E1	F5F5	F8F8	L5L5	S7S7
0	0.45	0.33	0.31	0.2	0.61	0.88	0.07	0.19	0.38	0.46	0.47	0.73
1	0.47	0.34	0.34	0.24	0.64	0.89	0.12	0.24	0.34	0.57	0.51	0.77
2	0.49	0.27	0.38	0.13	0.66	0.89	0.13	0.24	0.37	0.6	0.54	0.79
3	0.56	0.28	0.37	0.26	0.63	0.87	0.14	0.24	0.37	0.53	0.55	0.77
4	0.58	0.26	0.37	0.22	0.6	0.87	0.15	0.22	0.37	0.49	0.55	0.76
5	0.59	0.3	0.36	0.21	0.61	0.88	0.15	0.23	0.42	0.48	0.53	0.77
6	0.56	0.14	0.36	0.1	0.54	0.9	0.06	0.14	0.44	0.46	0.52	0.75
7	0.59	0.14	0.32	0.08	0.48	0.89	0.03	0.14	0.4	0.45	0.49	0.74
8	0.61	0.14	0.31	0.09	0.45	0.87	0.05	0.09	0.31	0.35	0.45	0.7
9	0.61	0.09	0.37	0.04	0.44	0.86	0.05	0.09	0.31	0.39	0.43	0.72
10	0.6	0.05	0.37	-0.0	0.44	0.87	0.02	0.11	0.31	0.44	0.41	0.74
11	0.6	-0.01	0.36	-0.04	0.47	0.87	0.03	0.1	0.3	0.41	0.42	0.77
12	0.63	0.04	0.34	0.01	0.5	0.84	0.06	0.14	0.4	0.4	0.43	0.75
13	0.64	0.08	0.41	0.03	0.53	0.8	0.08	0.11	0.45	0.43	0.49	0.75
14	0.65	0.04	0.4	-0.02	0.55	0.78	0.11	0.16	0.45	0.41	0.49	0.78
15	0.64	0.03	0.39	-0.02	0.5	0.79	0.13	0.12	0.4	0.4	0.5	0.79
16	0.68	0.06	0.38	0.01	0.48	0.79	0.18	0.12	0.38	0.43	0.51	0.77
17	0.68	0.06	0.39	0.08	0.47	0.76	0.19	0.14	0.4	0.44	0.45	0.72
18	0.68	0.03	0.35	0.06	0.46	0.71	0.26	0.2	0.4	0.45	0.48	0.72
19	0.7	0.07	0.33	0.11	0.49	0.69	0.33	0.3	0.35	0.44	0.46	0.7
20	0.71	0.04	0.33	0.11	0.52	0.69	0.39	0.31	0.28	0.44	0.48	0.7
21	0.73	0.09	0.33	0.12	0.46	0.68	0.38	0.34	0.29	0.34	0.5	0.66
22	0.7	0.1	0.34	0.06	0.42	0.64	0.34	0.31	0.25	0.34	0.42	0.67
23	0.72	0.09	0.35	0.1	0.41	0.59	0.29	0.3	0.28	0.37	0.41	0.68
24	0.71	0.08	0.35	0.11	0.38	0.57	0.24	0.29	0.33	0.44	0.41	0.66
25	0.69	0.05	0.29	0.04	0.31	0.58	0.15	0.2	0.27	0.26	0.37	0.62
26	0.64	-0.1	0.27	-0.06	0.27	0.52	0.07	0.15	0.17	0.2	0.33	0.62
27	0.64	-0.01	0.29	0.03	0.25	0.45	0.09	0.17	0.12	0.2	0.29	0.59
28	0.63	-0.06	0.34	0.02	0.19	0.46	0.07	0.11	0.17	0.22	0.28	0.57
29	0.62	-0.08	0.29	0.06	0.11	0.43	0.04	-0.01	0.13	0.24	0.2	0.56
30	0.59	-0.12	0.28	-0.03	0.08	0.37	0.06	-0.03	0.22	0.24	0.23	0.53
31	0.6	0.0	0.3	0.09	0.13	0.29	0.16	0.04	0.22	0.31	0.26	0.52
32	0.59	-0.15	0.33	-0.1	0.14	0.0	0.19	0.05	0.16	0.3	0.28	0.5
33	0.57	-0.14	0.31	-0.04	0.16	0.0	0.15	0.06	0.06	0.22	0.25	0.5
34	0.57	-0.19	0.3	-0.01	0.09	0.0	0.11	-0.0	-0.01	0.17	0.17	0.46
35	0.59	-0.19	0.34	-0.03	0.06	0.0	0.14	-0.01	0.06	0.19	0.11	0.45
36	0.59	-0.12	0.29	-0.05	0.03	0.0	0.14	-0.02	-0.01	0.08	0.01	0.44
37	0.59	-0.21	0.24	-0.16	-0.03	0.0	0.1	-0.06	0.02	0.04	-0.05	0.44
38	0.6	-0.22	0.2	-0.12	-0.04	0.0	0.1	-0.05	-0.07	0.03	-0.07	0.43
39	0.6	-0.16	0.26	-0.05	0.02	0.0	0.21	0.04	-0.12	0.08	-0.08	0.41
40	0.62	-0.09	0.29	0.0	0.09	0.0	0.27	0.12	-0.19	0.08	-0.13	0.43
41	0.61	-0.14	0.34	0.02	0.07	0.0	0.24	0.08	-0.19	0.06	-0.13	0.41
42	0.64	-0.06	0.33	0.1	0.08	0.0	0.23	0.05	-0.2	0.09	-0.16	0.4
43	0.63	-0.07	0.29	0.09	0.07	0.0	0.18	-0.02	-0.22	-0.03	-0.18	0.36
44	0.6	-0.14	0.27	0.01	-0.02	0.0	0.11	-0.07	-0.28	-0.03	-0.19	0.37
45	0.57	-0.06	0.31	-0.02	-0.05	0.0	0.05	-0.05	-0.25	0.12	-0.16	0.35
46	0.55	-0.11	0.35	0.08	-0.1	0.0	-0.0	-0.06	-0.25	0.15	-0.16	0.32
47	0.52	-0.17	0.32	0.17	-0.14	0.0	-0.09	-0.11	-0.25	0.01	-0.22	0.3
48	0.5	-0.1	0.28	0.19	-0.22	0.0	-0.06	-0.16	-0.24	-0.01	-0.22	0.28
49	0.47	-0.11	0.31	0.16	-0.24	0.0	-0.07	-0.07	-0.22	0.03	-0.19	0.3
50	0.41	-0.18	0.37	0.03	-0.27	0.0	-0.11	-0.1	-0.2	0.05	-0.21	0.28
51	0.4	-0.21	0.35	0.03	-0.24	0.0	-0.1	-0.13	-0.25	-0.07	-0.23	0.28
52	0.42	-0.2	0.29	0.03	-0.23	0.0	-0.06	-0.07	-0.3	-0.06	-0.2	0.26

Table A.6: Pearson correlation coefficient (PCC) between lagged sales record and online visits of SEAT Leon 5D colors

	SEAT LEON 5D													
Lag	0C0C	0E0E	2Y2Y	7Y7Y	9550	9M9M	B4B4	C0C0	E1E1	F5F5	L5L5	P5P5	S7S7	T4T4
0	0.78	0.08	0.01	0.35	0.54	0.34	0.18	0.35	0.11	0.4	0.26	0.52	0.68	0.39
1	0.79	0.12	0.03	0.37	0.58	0.35	0.19	0.35	0.16	0.37	0.28	0.49	0.71	0.35
2	0.77	0.02	-0.06	0.24	0.54	0.36	0.11	0.31	0.13	0.37	0.25	0.47	0.73	0.36
3	0.77	-0.09	-0.13	0.2	0.46	0.38	0.07	0.2	-0.02	0.32	0.14	0.47	0.71	0.44
4	0.76	-0.13	-0.1	0.22	0.42	0.41	0.0	0.18	-0.01	0.41	0.17	0.43	0.71	0.34
5	0.79	-0.14	-0.08	0.17	0.41	0.47	0.04	0.24	0.02	0.42	0.15	0.41	0.7	0.3
6	0.79	-0.24	-0.1	0.05	0.45	0.48	0.02	0.26	0.04	0.43	0.12	0.28	0.7	0.3
7	0.79	-0.23	-0.12	0.04	0.41	0.5	0.04	0.2	0.02	0.38	0.08	0.27	0.69	0.34
8	0.81	-0.26	-0.1	0.03	0.37	0.54	0.08	0.21	0.01	0.41	0.09	0.36	0.69	0.44
9	0.79	-0.22	-0.07	0.15	0.37	0.54	0.11	0.18	-0.01	0.36	0.08	0.28	0.7	0.42
10	0.81	-0.14	-0.07	0.2	0.42	0.59	0.12	0.12	0.0	0.36	0.11	0.24	0.71	0.42
11	0.79	-0.21	-0.1	0.16	0.47	0.57	0.12	0.08	-0.02	0.41	0.06	0.34	0.72	0.54
12	0.77	-0.18	-0.1	0.16	0.44	0.59	0.13	0.12	-0.05	0.45	0.11	0.47	0.74	0.7
13	0.76	-0.08	-0.07	0.18	0.42	0.61	0.17	0.11	-0.01	0.47	0.12	0.48	0.77	0.63
14	0.78	-0.1	-0.06	0.17	0.42	0.62	0.19	0.22	0.03	0.5	0.1	0.51	0.77	0.61
15	0.77	-0.11	0.01	0.13	0.4	0.61	0.17	0.18	0.08	0.47	0.08	0.51	0.78	0.58
16	0.77	-0.08	0.03	0.13	0.33	0.63	0.16	0.14	0.12	0.38	0.07	0.44	0.76	0.54
17	0.76	-0.02	0.05	0.18	0.31	0.62	0.11	0.1	0.07	0.36	0.1	0.39	0.77	0.46
18	0.73	-0.04	0.05	0.16	0.34	0.57	0.14	0.15	0.06	0.3	0.09	0.28	0.76	0.39
19	0.71	-0.13	0.07	0.09	0.36	0.55	0.09	0.13	-0.04	0.3	0.08	0.2	0.76	0.36
20	0.7	-0.16	0.02	0.08	0.35	0.56	0.04	0.09	0.01	0.32	0.04	0.41	0.77	0.46
21	0.7	-0.15	0.05	0.1	0.32	0.54	0.12	0.14	0.06	0.37	0.01	0.6	0.76	0.57
22	0.65	-0.1	0.06	0.0	0.25	0.48	0.12	0.14	0.11	0.28	0.07	0.56	0.73	0.53
23	0.67	-0.02	0.11	0.03	0.25	0.43	0.19	0.21	0.07	0.29	0.16	0.45	0.73	0.43
24	0.66	-0.08	0.09	0.08	0.27	0.42	0.22	0.16	0.04	0.29	0.11	0.5	0.7	0.43
25	0.65	-0.08	0.03	0.1	0.28	0.44	0.26	0.09	0.06	0.23	0.04	0.55	0.68	0.58
26	0.62	-0.05	-0.0	0.07	0.22	0.42	0.15	0.05	0.06	0.29	0.05	0.42	0.68	0.46
27	0.61	-0.03	0.06	0.16	0.24	0.29	0.17	0.09	0.13	0.34	0.09	0.34	0.67	0.51
28	0.58	-0.03	0.03	0.12	0.19	0.08	0.19	0.07	0.1	0.34	0.05	0.34	0.66	0.49
29	0.57	0.01	-0.01	0.11	0.12	0.18	0.16	0.02	0.03	0.26	-0.07	0.48	0.65	0.59
30	0.58	0.09	0.07	0.12	0.15	0.22	0.2	0.11	0.13	0.31	-0.01	0.5	0.65	0.62
31	0.55	0.1	0.14	0.12	0.12	0.17	0.2	0.01	0.15	0.18	-0.03	0.48	0.63	0.48
32	0.54	0.06	0.07	-0.01	0.04	0.1	0.09	0.02	0.15	0.08	0.02	0.42	0.6	0.52
33	0.51	0.04	0.05	0.03	0.12	0.12	0.12	-0.04	0.11	0.05	-0.02	0.55	0.58	0.65
34	0.47	-0.01	0.06	0.05	0.18	0.11	0.14	-0.1	-0.02	0.14	-0.04	0.63	0.58	0.64
35	0.44	0.03	0.06	-0.0	0.14	0.01	0.08	-0.14	0.04	0.13	-0.07	0.57	0.57	0.69
36	0.42	0.06	0.11	0.02	0.08	-0.05	0.08	-0.06	0.14	0.19	-0.02	0.34	0.55	0.69
37	0.41	0.12	0.03	0.02	-0.0	-0.06	0.12	-0.04	0.08	0.11	0.06	0.42	0.53	0.62
38	0.42	0.16	0.12	0.04	-0.06	-0.08	0.16	-0.03	0.06	0.04	0.04	0.54	0.49	0.79
39	0.39	0.23	0.11	0.03	-0.11	-0.09	0.1	-0.05	0.04	-0.04	0.01	0.34	0.49	0.67
40	0.4	0.22	0.21	0.08	-0.02	-0.1	0.17	-0.08	0.16	-0.01	0.03	0.37	0.47	0.83
41	0.36	0.05	0.08	-0.11	0.0	-0.11	0.09	-0.16	0.05	0.02	-0.03	0.36	0.45	0.56
42	0.36	0.05	0.01	-0.13	0.01	-0.12	0.06	-0.15	0.09	0.01	-0.01	0.36	0.41	0.5
43	0.34	0.09	-0.01	-0.06	-0.01	-0.12	0.08	-0.13	0.09	-0.04	0.03	0.12	0.39	0.49
44	0.31	0.14	-0.13	-0.1	0.02	-0.13	0.12	-0.15	0.06	-0.09	0.02	0.0	0.39	0.2
45	0.33	0.12	-0.01	-0.07	0.07	-0.14	0.11	-0.04	0.02	-0.13	0.0	0.0	0.38	0.16
46	0.29	0.08	-0.05	-0.1	0.04	-0.14	0.08	-0.12	-0.11	-0.19	-0.15	0.0	0.35	0.12
47	0.29	0.07	-0.12	-0.13	-0.01	-0.14	-0.02	-0.09	-0.19	-0.21	-0.17	0.0	0.32	0.18
48	0.28	0.1	-0.22	-0.21	-0.03	-0.14	-0.09	-0.13	-0.14	-0.23	-0.11	0.0	0.31	0.03
49	0.23	0.16	-0.09	-0.11	0.07	-0.14	0.05	-0.13	-0.1	-0.21	-0.04	0.0	0.3	-0.05
50	0.21	0.12	-0.06	-0.15	0.04	-0.14	0.14	-0.18	-0.03	-0.18	-0.08	0.0	0.29	-0.05
51	0.21	0.16	-0.02	-0.19	0.01	-0.15	0.11	-0.18	-0.03	-0.15	-0.13	0.0	0.26	-0.05
52	0.2	0.29	-0.01	-0.05	0.01	-0.15	0.04	-0.08	0.08	-0.19	0.03	0.0	0.25	-0.05

Table A.7: Pearson correlation coefficient (PCC) between lagged sales record and online visits of SEAT Leon ST colors

	SEAT LEON ST													
Lag	0C0C	0E0E	2Y2Y	7Y7Y	9550	9M9M	B4B4	C0C0	E1E1	F5F5	L5L5	P5P5	S7S7	T4T4
0	0.75	0.02	0.46	0.31	0.46	0.46	-0.11	0.41	0.24	0.28	0.35	0.36	0.71	0.19
1	0.75	-0.03	0.48	0.31	0.5	0.49	-0.05	0.38	0.24	0.31	0.31	0.3	0.73	0.16
2	0.75	-0.07	0.45	0.27	0.53	0.49	-0.05	0.43	0.29	0.36	0.19	0.17	0.74	0.16
3	0.78	-0.08	0.48	0.25	0.46	0.49	-0.13	0.43	0.15	0.32	0.16	0.17	0.75	0.26
4	0.76	-0.06	0.41	0.24	0.37	0.46	-0.15	0.41	0.05	0.28	0.22	0.16	0.76	0.33
5	0.76	-0.01	0.4	0.21	0.39	0.46	-0.12	0.35	0.04	0.29	0.23	0.1	0.73	0.32
6	0.76	-0.14	0.38	0.15	0.42	0.44	-0.14	0.38	0.01	0.3	0.22	0.14	0.72	0.35
7	0.79	-0.17	0.4	0.15	0.45	0.43	-0.17	0.44	-0.01	0.27	0.24	0.17	0.74	0.38
8	0.81	-0.12	0.41	0.14	0.44	0.43	-0.12	0.35	-0.19	0.24	0.24	0.22	0.72	0.48
9	0.8	-0.1	0.4	0.23	0.4	0.4	-0.16	0.34	-0.02	0.28	0.33	0.24	0.71	0.35
10	0.8	-0.1	0.44	0.21	0.38	0.43	-0.19	0.36	0.02	0.32	0.34	0.42	0.72	0.36
11	0.8	-0.16	0.46	0.23	0.42	0.46	-0.17	0.38	0.07	0.34	0.28	0.53	0.75	0.41
12	0.8	-0.13	0.47	0.22	0.41	0.45	-0.12	0.36	-0.0	0.31	0.28	0.6	0.73	0.56
13	0.78	-0.02	0.42	0.31	0.33	0.42	-0.19	0.37	0.06	0.33	0.34	0.63	0.72	0.53
14	0.78	-0.04	0.39	0.3	0.36	0.47	-0.2	0.34	0.11	0.3	0.38	0.7	0.72	0.49
15	0.79	-0.09	0.4	0.33	0.31	0.48	-0.14	0.37	0.05	0.3	0.28	0.6	0.69	0.46
16	0.78	-0.06	0.41	0.24	0.33	0.49	-0.16	0.39	0.14	0.34	0.25	0.5	0.69	0.47
17	0.76	-0.11	0.36	0.23	0.33	0.51	-0.19	0.35	0.09	0.36	0.24	0.56	0.72	0.46
18	0.76	-0.15	0.34	0.21	0.32	0.5	-0.16	0.35	0.1	0.34	0.21	0.54	0.72	0.33
19	0.75	-0.14	0.35	0.2	0.31	0.48	-0.1	0.35	-0.03	0.28	0.13	0.45	0.68	0.34
20	0.75	-0.16	0.36	0.18	0.33	0.51	-0.1	0.39	0.05	0.24	0.09	0.4	0.66	0.35
21	0.75	-0.14	0.36	0.15	0.25	0.49	-0.07	0.37	0.03	0.25	0.07	0.33	0.69	0.41
22	0.73	-0.17	0.32	0.18	0.25	0.41	-0.05	0.26	0.05	0.2	0.08	0.31	0.67	0.3
23	0.73	-0.07	0.35	0.17	0.33	0.4	-0.11	0.27	0.13	0.21	0.13	0.18	0.66	0.24
24	0.72	-0.03	0.32	0.21	0.32	0.36	-0.13	0.3	0.12	0.16	0.11	0.34	0.68	0.33
25	0.71	0.01	0.35	0.19	0.29	0.31	-0.15	0.3	0.11	0.15	0.09	0.4	0.66	0.46
26	0.68	0.1	0.29	0.17	0.22	0.25	-0.14	0.21	0.04	0.17	0.13	0.35	0.63	0.39
27	0.67	0.18	0.35	0.23	0.19	0.18	-0.07	0.24	0.07	0.15	0.12	0.33	0.61	0.4
28	0.66	0.19	0.37	0.29	0.22	0.09	-0.06	0.23	0.1	0.16	0.15	0.44	0.61	0.38
29	0.65	0.1	0.35	0.23	0.22	0.1	-0.12	0.21	0.07	0.12	0.14	0.54	0.58	0.52
30	0.63	0.17	0.36	0.21	0.14	0.1	-0.16	0.16	0.01	0.14	0.09	0.34	0.54	0.42
31	0.61	0.18	0.33	0.12	0.12	0.02	-0.15	0.13	0.18	0.15	0.07	0.31	0.52	0.44
32	0.58	0.08	0.29	0.15	0.17	-0.03	-0.09	0.17	0.11	0.17	0.1	0.33	0.47	0.51
33	0.56	0.06	0.28	0.09	0.2	-0.05	-0.06	0.2	0.12	0.19	0.08	0.48	0.45	0.61
34	0.53	0.16	0.27	0.12	0.15	-0.08	-0.02	0.27	0.09	0.17	0.02	0.26	0.43	0.67
35	0.52	0.1	0.21	0.06	0.12	-0.1	0.08	0.22	0.1	0.17	0.05	0.23	0.43	0.52
36	0.51	-0.0	0.26	0.14	0.13	-0.11	0.19	0.23	0.14	0.17	0.05	0.18	0.42	0.57
37	0.51	0.09	0.24	0.08	0.1	-0.11	0.13	0.29	0.15	0.21	0.07	0.21	0.4	0.57
38	0.52	0.03	0.27	0.13	0.15	-0.08	0.08	0.29	0.08	0.19	0.09	0.06	0.41	0.63
39	0.5	0.1	0.17	0.17	0.09	-0.05	-0.02	0.11	-0.04	0.19	0.0	-0.03	0.42	0.56
40	0.49	0.22	0.3	0.11	0.14	-0.04	-0.05	0.15	0.09	0.14	0.01	-0.03	0.43	0.62
41	0.45	0.19	0.27	0.05	0.18	-0.05	-0.06	0.16	0.06	0.09	0.0	0.14	0.42	0.62
42	0.44	0.08	0.26	0.04	0.2	-0.05	-0.1	0.09	0.07	0.12	-0.03	0.31	0.41	0.77
43	0.41	0.2	0.27	0.06	0.19	-0.05	-0.11	-0.01	-0.06	0.09	-0.08	0.28	0.39	0.76
44	0.38	0.17	0.18	-0.06	0.21	-0.09	-0.05	-0.09	-0.07	0.09	-0.07	0.29	0.37	0.63
45	0.41	0.08	0.3	-0.05	0.28	-0.11	-0.08	-0.07	-0.05	0.05	-0.02	0.23	0.38	0.64
46	0.39	0.08	0.26	-0.07	0.31	-0.11	-0.11	0.06	-0.02	0.04	-0.0	0.27	0.35	0.74
47	0.37	0.07	0.2	-0.1	0.39	-0.12	-0.11	0.05	-0.12	0.04	-0.01	0.09	0.3	0.64
48	0.33	0.03	0.07	-0.13	0.27	-0.12	-0.13	-0.14	-0.21	0.12	0.01	-0.03	0.25	0.54
49	0.34	0.12	0.16	-0.08	0.25	-0.12	-0.08	-0.15	-0.14	0.1	-0.03	0.22	0.25	0.49
50	0.31	0.05	0.19	-0.13	0.32	-0.12	-0.13	-0.14	-0.08	0.07	-0.12	0.46	0.24	0.51
51	0.3	-0.01	0.17	-0.1	0.26	-0.12	-0.12	-0.14	-0.16	0.08	-0.09	0.42	0.22	0.42
52	0.27	0.09	0.17	-0.19	0.07	-0.12	-0.11	-0.1	-0.12	0.15	-0.0	0.44	0.2	0.23

Table A.8: Pearson correlation coefficient (PCC) between lagged sales record and online visits of SEAT Arona compound region

	SEAT ARONA					
Lag	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
0	0.39	0.49	0.45	0.49	0.44	0.49
1	0.43	0.54	0.48	0.54	0.51	0.52
2	0.45	0.57	0.49	0.54	0.51	0.55
3	0.45	0.54	0.44	0.51	0.49	0.49
4	0.45	0.54	0.39	0.53	0.49	0.49
5	0.5	0.57	0.45	0.56	0.52	0.58
6	0.5	0.55	0.45	0.56	0.48	0.54
7	0.46	0.47	0.41	0.52	0.45	0.5
8	0.45	0.37	0.33	0.44	0.41	0.38
9	0.49	0.47	0.45	0.5	0.47	0.43
10	0.51	0.5	0.48	0.5	0.5	0.45
11	0.49	0.45	0.43	0.49	0.46	0.45
12	0.46	0.35	0.33	0.47	0.38	0.39
13	0.53	0.35	0.37	0.46	0.37	0.44
14	0.55	0.36	0.39	0.43	0.35	0.43
15	0.52	0.3	0.38	0.38	0.33	0.41
16	0.44	0.21	0.31	0.33	0.3	0.36
17	0.45	0.2	0.32	0.27	0.31	0.31
18	0.42	0.22	0.35	0.32	0.32	0.31
19	0.4	0.21	0.34	0.3	0.33	0.27
20	0.35	0.18	0.31	0.29	0.26	0.26
21	0.35	0.14	0.27	0.24	0.2	0.19
22	0.39	0.16	0.32	0.28	0.22	0.22
23	0.38	0.16	0.32	0.33	0.25	0.21
24	0.42	0.15	0.34	0.3	0.2	0.13
25	0.29	0.04	0.24	0.18	0.09	0.05
26	0.32	0.06	0.22	0.16	0.12	0.0
27	0.31	0.06	0.24	0.17	0.12	0.04
28	0.26	0.03	0.26	0.15	0.05	0.0
29	0.18	-0.06	0.17	0.03	-0.09	-0.08
30	0.17	-0.09	0.17	0.01	-0.12	-0.12
31	0.24	-0.05	0.23	0.06	-0.09	-0.09
32	0.23	-0.02	0.25	0.1	-0.12	-0.06
33	0.15	-0.04	0.17	0.04	-0.16	-0.14
34	0.1	-0.14	0.01	-0.07	-0.17	-0.24
35	0.17	-0.02	0.08	0.01	-0.05	-0.16
36	0.16	0.05	0.16	0.07	-0.02	-0.04
37	0.07	0.01	0.18	0.09	-0.05	-0.01
38	0.03	-0.16	0.04	-0.02	-0.16	-0.09
39	0.14	-0.07	0.17	0.02	-0.12	-0.03
40	0.19	0.05	0.24	0.1	-0.1	0.02
41	0.17	0.07	0.23	0.12	-0.12	0.03
42	0.03	0.05	0.09	0.13	-0.13	0.0
43	0.03	0.03	-0.03	0.09	-0.08	-0.08
44	0.02	0.15	0.04	0.17	0.04	0.04
45	-0.03	0.18	0.07	0.19	0.05	0.05
46	-0.1	0.14	0.08	0.15	0.02	0.12
47	-0.15	-0.09	-0.1	-0.04	-0.1	-0.07
48	-0.02	-0.02	0.01	0.03	-0.04	0.04
49	0.04	0.05	0.03	0.06	0.01	0.04
50	0.05	0.03	-0.02	0.02	-0.01	-0.05
51	-0.08	-0.04	-0.09	-0.05	-0.05	-0.12
52	-0.01	0.06	-0.06	0.02	0.08	-0.09

Table A.9: Pearson correlation coefficient (PCC) between lagged sales record and online visits of SEAT Ibiza compound region

	SEAT IBIZA					
Lag	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
0	0.09	0.27	0.19	0.26	0.14	0.13
1	0.1	0.37	0.22	0.37	0.2	0.16
2	0.05	0.38	0.26	0.37	0.18	0.17
3	0.07	0.37	0.28	0.35	0.16	0.2
4	0.06	0.3	0.26	0.33	0.19	0.16
5	0.06	0.32	0.21	0.28	0.21	0.13
6	-0.04	0.28	0.17	0.16	0.09	0.07
7	-0.04	0.29	0.14	0.12	0.01	0.06
8	-0.02	0.26	0.11	0.14	0.0	0.01
9	-0.03	0.22	0.1	0.15	0.02	-0.04
10	-0.1	0.15	0.11	0.18	0.01	-0.06
11	-0.12	0.15	0.13	0.17	0.02	-0.06
12	-0.06	0.22	0.12	0.18	0.06	-0.04
13	-0.02	0.28	0.14	0.2	0.02	0.05
14	-0.02	0.32	0.16	0.25	0.05	0.08
15	-0.05	0.28	0.18	0.26	0.06	0.12
16	-0.03	0.26	0.19	0.26	0.08	0.14
17	0.08	0.23	0.2	0.23	0.07	0.13
18	0.1	0.21	0.25	0.28	0.08	0.14
19	0.02	0.27	0.33	0.33	0.19	0.2
20	0.03	0.29	0.32	0.36	0.28	0.25
21	0.09	0.32	0.26	0.35	0.23	0.28
22	0.08	0.24	0.22	0.36	0.16	0.18
23	0.13	0.25	0.22	0.33	0.18	0.15
24	0.16	0.26	0.2	0.24	0.18	0.18
25	0.18	0.14	0.12	0.12	0.03	0.13
26	0.12	0.06	0.04	0.01	0.01	0.01
27	0.01	0.11	0.02	0.09	0.06	0.05
28	-0.1	0.12	-0.02	0.06	0.03	0.08
29	-0.12	0.06	-0.08	0.05	-0.01	-0.05
30	-0.08	0.06	-0.01	0.07	-0.01	-0.08
31	-0.03	0.19	0.11	0.17	0.06	0.06
32	-0.14	0.15	0.14	0.19	0.05	0.1
33	-0.17	0.14	0.12	0.15	0.04	0.11
34	-0.21	0.08	0.02	0.12	-0.02	0.05
35	-0.12	0.09	0.01	0.1	-0.02	0.05
36	-0.14	0.1	0.05	-0.02	-0.04	0.02
37	-0.19	-0.04	0.03	-0.09	-0.08	-0.04
38	-0.14	-0.11	-0.01	-0.04	-0.05	-0.05
39	-0.12	-0.02	0.06	0.11	0.07	0.03
40	-0.04	0.06	0.1	0.16	0.09	0.07
41	-0.16	0.05	0.08	0.15	0.08	0.11
42	-0.02	0.01	0.03	0.17	0.14	0.11
43	0.03	-0.0	-0.04	0.1	0.1	0.05
44	0.05	-0.09	-0.04	0.01	-0.02	-0.04
45	0.03	-0.03	-0.04	-0.03	-0.01	-0.0
46	0.05	-0.03	-0.08	-0.03	-0.01	0.02
47	0.1	-0.06	-0.13	-0.13	-0.13	-0.03
48	0.17	-0.15	-0.11	-0.16	-0.09	-0.05
49	0.16	-0.09	-0.12	-0.16	-0.01	-0.04
50	0.09	-0.12	-0.13	-0.15	-0.1	-0.1
51	0.15	-0.16	-0.14	-0.2	-0.2	-0.11
52	0.2	-0.16	-0.11	-0.22	-0.17	-0.14

Table A.10: Pearson correlation coefficient (PCC) between lagged sales record and online visits of SEAT Leon 5D compound region

	SEAT LEON 5D					
Lag	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
0	0.1	0.24	0.11	0.15	0.19	0.13
1	0.11	0.29	0.11	0.22	0.23	0.13
2	0.05	0.2	0.08	0.13	0.13	0.05
3	-0.09	0.08	-0.04	0.04	0.01	-0.03
4	-0.01	0.06	-0.03	0.03	0.0	-0.03
5	-0.0	0.07	0.01	0.07	-0.02	0.03
6	-0.03	0.05	-0.02	0.02	0.01	0.01
7	-0.13	-0.0	-0.03	0.03	-0.01	-0.03
8	-0.11	0.04	-0.04	0.05	-0.04	-0.03
9	-0.12	0.09	0.01	0.01	-0.03	-0.01
10	-0.05	0.15	0.03	0.06	-0.04	0.04
11	-0.01	0.08	0.06	0.03	-0.09	-0.01
12	-0.12	0.06	0.03	0.03	-0.06	-0.01
13	-0.06	0.05	0.05	0.14	-0.02	0.07
14	0.09	0.06	0.09	0.1	-0.0	0.04
15	0.11	0.1	0.1	0.03	0.04	0.05
16	0.1	0.13	0.09	0.03	-0.02	0.12
17	0.16	0.13	0.07	0.09	-0.05	0.18
18	0.04	0.06	0.06	0.11	-0.02	0.14
19	-0.07	0.04	0.01	0.04	0.0	0.05
20	0.01	-0.02	0.05	-0.01	0.03	-0.02
21	0.0	0.02	0.09	0.01	0.05	0.01
22	-0.07	0.03	0.05	0.04	0.05	0.03
23	-0.02	0.1	0.08	0.12	0.03	0.07
24	0.03	0.09	0.07	0.04	0.0	0.03
25	-0.03	0.09	0.06	0.03	-0.07	0.04
26	0.07	0.01	0.02	0.01	-0.08	0.05
27	0.11	0.03	0.12	0.06	0.01	0.15
28	-0.03	-0.03	0.09	0.12	0.01	0.11
29	0.15	-0.07	0.03	0.09	-0.05	0.03
30	0.22	-0.0	0.14	0.1	0.03	0.06
31	0.17	0.05	0.07	0.05	0.01	0.05
32	0.13	0.02	0.06	0.03	-0.08	0.01
33	0.2	-0.01	0.06	0.04	-0.04	0.01
34	0.01	-0.01	0.01	0.07	0.0	0.0
35	0.1	-0.05	-0.03	0.09	0.01	-0.03
36	0.11	-0.01	0.07	0.06	0.07	-0.11
37	0.08	-0.04	0.05	0.07	0.07	-0.12
38	0.04	0.03	0.08	0.12	0.02	-0.11
39	0.12	-0.09	0.07	0.13	-0.02	-0.02
40	-0.01	0.05	0.1	0.1	-0.01	0.13
41	-0.13	-0.08	0.05	-0.01	-0.16	-0.02
42	-0.09	-0.14	0.09	-0.06	-0.22	0.02
43	-0.03	-0.09	-0.03	-0.06	-0.15	0.07
44	-0.04	-0.14	-0.09	0.04	-0.12	0.01
45	0.1	-0.07	-0.01	-0.01	-0.02	-0.01
46	0.07	-0.1	-0.06	-0.11	-0.02	-0.12
47	-0.06	-0.16	-0.13	-0.12	-0.13	-0.09
48	-0.01	-0.18	-0.15	-0.16	-0.14	-0.12
49	-0.06	-0.04	-0.08	-0.16	-0.01	0.05
50	-0.19	0.02	-0.05	-0.1	0.03	0.08
51	-0.12	0.01	-0.05	-0.05	0.06	-0.0
52	0.07	0.04	0.01	-0.03	0.18	0.06

Table A.11: Pearson correlation coefficient (PCC) between lagged sales record and online visits of SEAT Leon ST compound region

	SEAT LEON ST					
Lag	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
0	0.14	-0.07	0.22	0.27	0.28	0.18
1	0.11	-0.01	0.2	0.29	0.33	0.14
2	0.01	-0.04	0.21	0.3	0.31	0.14
3	-0.01	-0.08	0.21	0.35	0.26	0.19
4	0.11	-0.09	0.17	0.29	0.23	0.11
5	0.14	-0.07	0.09	0.27	0.2	0.1
6	-0.0	-0.08	0.1	0.26	0.12	0.06
7	-0.06	-0.04	0.15	0.2	0.1	0.1
8	0.03	-0.05	0.16	0.23	0.1	0.05
9	0.02	-0.14	0.12	0.18	0.16	0.03
10	0.16	-0.11	0.09	0.24	0.18	0.03
11	0.0	-0.08	0.14	0.3	0.18	0.05
12	0.11	-0.11	0.17	0.31	0.17	0.12
13	0.08	-0.1	0.14	0.29	0.2	0.06
14	0.09	-0.05	0.1	0.31	0.2	0.09
15	0.13	0.02	0.08	0.26	0.19	0.09
16	0.13	0.04	0.14	0.19	0.17	0.15
17	0.14	0.06	0.13	0.19	0.11	0.1
18	0.06	0.04	0.09	0.23	0.14	0.11
19	-0.1	0.05	0.07	0.22	0.14	0.09
20	-0.02	0.06	0.09	0.19	0.14	0.13
21	0.06	0.12	0.2	0.16	0.12	0.16
22	-0.04	0.08	0.13	0.08	0.08	0.14
23	-0.04	0.08	0.22	0.18	0.11	0.16
24	-0.02	-0.03	0.21	0.16	0.17	0.2
25	0.1	-0.05	0.24	0.18	0.13	0.21
26	0.01	-0.06	0.13	0.18	0.14	0.13
27	0.13	-0.01	0.15	0.27	0.19	0.24
28	0.14	-0.02	0.23	0.31	0.24	0.25
29	0.09	-0.0	0.21	0.27	0.16	0.24
30	0.04	-0.04	0.21	0.25	0.13	0.21
31	0.01	0.02	0.12	0.2	0.14	0.14
32	0.07	0.04	0.17	0.19	0.07	0.05
33	0.02	0.07	0.17	0.2	0.05	0.13
34	0.05	0.12	0.15	0.17	0.1	0.13
35	0.02	0.07	0.08	0.12	0.1	-0.01
36	-0.01	0.18	0.09	0.02	0.12	0.0
37	0.07	0.16	0.17	0.08	0.1	0.07
38	0.01	0.08	0.24	0.23	0.01	0.1
39	-0.02	-0.14	0.15	0.13	0.0	0.05
40	0.11	-0.07	0.15	0.23	0.05	0.16
41	0.09	-0.11	0.11	0.22	0.05	0.1
42	0.1	-0.21	0.16	0.15	0.09	0.11
43	0.08	-0.18	0.05	0.09	0.07	0.07
44	0.21	-0.21	-0.1	0.16	0.03	-0.05
45	0.12	-0.07	-0.01	0.17	0.17	0.02
46	0.05	-0.01	0.08	0.02	0.18	0.09
47	-0.03	-0.02	0.1	0.03	0.09	0.03
48	0.07	-0.12	-0.02	-0.12	-0.01	-0.11
49	0.13	-0.07	0.06	-0.04	0.07	-0.05
50	-0.18	-0.02	0.12	-0.09	0.07	0.09
51	-0.17	-0.1	0.18	-0.0	0.03	0.16
52	-0.15	-0.14	0.14	-0.09	0.1	0.01

Appendix B

Forecasting Performance

The information presented in this appendix complements the findings detailed in Subsection 6.2.2.

Table B.1: Mean Average Error (MAE) per SEAT Leon ST car variant (car model plus exterior color) and time chunk of each forecasting technique. *Roll.* refers to Rolling, *Uni.* refers to Univariate, *Multi.* refers to Multivariate, and *nan* implies that forecast was not computed because there was not data in both time series.

SEAT LEON ST (Time Chunk 1)																		
Algorithm	0C0C	0E0E	2Y2Y	3W3W	7Y7Y	9019	9550	9M9M	B4B4	C0C0	E1E1	E4E4	F5F5	I4I4	L5L5	P5P5	S3S3	S7S7	T4T4
Roll. ARIMA	4.17	3.17	17.0	nan	4.17	nan	2.17	1.83	7.33	1.33	2.83	nan	0.0	nan	3.33	0.67	nan	0.0	5.17
XGBoost Uni.	4.17	3.33	7.33	nan	3.33	nan	0.67	1.17	5.0	0.67	2.17	nan	0.0	nan	2.0	0.33	nan	0.0	5.5
Roll. ARIMAX	3.0	2.83	16.83	nan	4.0	nan	2.83	1.0	5.17	2.0	2.33	nan	0.0	nan	1.83	0.33	nan	0.0	6.17
XGBoost Multi.	2.67	1.17	7.0	nan	1.83	nan	0.67	1.0	4.83	0.67	1.5	nan	0.0	nan	2.0	0.33	nan	0.0	3.83

SEAT LEON ST (Time Chunk 2)																		
Algorithm	0C0C	0E0E	2Y2Y	3W3W	7Y7Y	9019	9550	9M9M	B4B4	C0C0	E1E1	E4E4	F5F5	I4I4	L5L5	P5P5	S3S3	S7S7	T4T4
Roll. ARIMA	5.17	2.5	4.17	nan	4.0	nan	1.33	0.5	2.0	1.5	6.5	nan	0.33	nan	2.5	0.33	nan	0.0	1.5
XGBoost Uni.	5.0	2.5	2.17	nan	3.83	nan	1.17	0.0	1.83	1.0	3.5	nan	0.33	nan	1.33	0.17	nan	0.0	0.5
Roll. ARIMAX	69.83	2.0	4.17	nan	4.83	nan	1.83	0.33	2.67	1.5	6.83	nan	0.5	nan	1.83	0.33	nan	0.0	1.5
XGBoost Multi.	1.67	1.17	1.83	nan	3.0	nan	0.83	0.0	0.83	1.0	2.67	nan	0.33	nan	1.17	0.17	nan	0.0	0.5

SEAT LEON ST (Time Chunk 3)																		
Algorithm	0C0C	0E0E	2Y2Y	3W3W	7Y7Y	9019	9550	9M9M	B4B4	C0C0	E1E1	E4E4	F5F5	I4I4	L5L5	P5P5	S3S3	S7S7	T4T4
Roll. ARIMA	2.0	5.33	8.83	nan	7.33	nan	1.17	0.0	6.33	1.0	7.5	nan	0.67	nan	1.83	0.0	nan	8.0	0.0
XGBoost Uni.	2.5	5.5	5.83	nan	7.33	nan	0.5	0.0	5.5	1.17	4.67	nan	0.5	nan	2.33	0.0	nan	4.17	0.0
Roll. ARIMAX	3.17	6.17	7.67	nan	8.83	nan	0.83	0.0	4.83	0.67	7.83	nan	0.67	nan	2.67	0.0	nan	5.67	0.0
XGBoost Multi.	1.17	3.33	3.83	nan	6.67	nan	0.17	0.0	4.33	0.83	4.83	nan	0.5	nan	1.83	0.0	nan	2.67	0.0

SEAT LEON ST (Time Chunk 4)																		
Algorithm	0C0C	0E0E	2Y2Y	3W3W	7Y7Y	9019	9550	9M9M	B4B4	C0C0	E1E1	E4E4	F5F5	I4I4	L5L5	P5P5	S3S3	S7S7	T4T4
Roll. ARIMA	0.5	7.5	7.0	nan	4.83	nan	1.83	0.83	10.0	3.33	4.83	nan	0.67	nan	2.17	0.0	nan	8.0	0.0
XGBoost Uni.	0.0	4.17	4.17	nan	2.33	nan	1.0	0.17	7.33	0.5	3.33	nan	0.5	nan	1.33	0.0	nan	5.67	0.0
Roll. ARIMAX	0.0	6.5	9.33	nan	4.67	nan	2.5	1.67	9.83	2.33	5.33	nan	0.33	nan	1.33	0.0	nan	7.83	0.0
XGBoost Multi.	0.0	4.67	3.0	nan	2.5	nan	1.0	0.17	7.17	0.67	2.17	nan	0.5	nan	0.83	0.0	nan	3.5	0.0

SEAT LEON ST (Time Chunk 5)																		
Algorithm	0C0C	0E0E	2Y2Y	3W3W	7Y7Y	9019	9550	9M9M	B4B4	C0C0	E1E1	E4E4	F5F5	I4I4	L5L5	P5P5	S3S3	S7S7	T4T4
Roll. ARIMA	0.0	4.8	7.2	nan	3.8	nan	0.8	2.8	6.4	0.0	1.6	nan	0.4	nan	2.4	0.0	nan	4.2	0.0
XGBoost Uni.	0.0	2.8	3.2	nan	2.0	nan	0.4	1.6	2.4	0.0	2.2	nan	0.0	nan	2.2	0.0	nan	4.8	0.0
Roll. ARIMAX	0.0	4.0	4.8	nan	3.8	nan	0.8	2.6	5.0	0.0	1.8	nan	0.0	nan	2.8	0.0	nan	5.2	0.0
XGBoost Multi.	0.0	2.2	4.0	nan	2.0	nan	0.4	1.6	1.2	0.0	2.2	nan	0.0	nan	2.0	0.0	nan	3.0	0.0

Table B.2: Mean Average Error (MAE) per SEAT Arona car variant (car model plus exterior color) and time chunk of each forecasting technique. *Roll.* refers to Rolling, *Uni.* refers to Univariate, *Multi.* refers to Multivariate, and *nan* implies that forecast was not computed because there was not data in both time series.

SEAT ARONA (Time Chunk 1)																							
Algorithm	0C0C	0C0E	0CF5	0E0C	0E0E	0EF5	0ES7	2Y0C	2Y0E	2Y2Y	2YF5	2YS7	7Y0C	7Y0E	7Y7Y	7YS7	9529	9532	9545	9550	9M0E	9M9M	9MS7	B40C
Roll. ARIMA	0.5	0.0	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.33	nan	0.0	nan	2.17
XGBoost Uni.	0.5	0.0	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.33	nan	0.0	nan	2.17
Roll. ARIMAX	0.5	0.0	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.33	nan	0.0	nan	2.17
XGBoost Multi.	0.5	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.33	nan	0.0	nan	1.83
SEAT ARONA (Time Chunk 1)																								
Algorithm	B40E	B4B4	B4F5	B4S7	E10C	E10E	E1E1	E1S7	F50C	F50E	F5F5	F5S7	F80C	F80E	F8F8	F8S7	L50C	L50E	L5F5	L5L5	L5S7	S70E	S7F5	S7S7
Roll. ARIMA	0.67	0.83	0.0	0.0	2.17	1.67	1.33	0.0	0.67	0.83	0.0	0.0	0.5	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XGBoost Uni.	0.83	1.17	0.0	0.0	2.17	1.83	1.33	0.0	0.33	0.83	0.0	0.0	0.5	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Roll. ARIMAX	0.33	1.0	0.0	0.0	2.5	2.0	1.0	0.0	0.5	0.5	0.0	0.0	0.33	0.67	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XGBoost Multi.	0.83	1.17	0.0	0.0	2.0	1.5	1.0	0.0	0.5	0.67	0.0	0.0	0.67	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

SEAT ARONA (Time Chunk 2)																								
Algorithm	0C0C	0C0E	0CF5	0E0C	0E0E	0EF5	0ES7	2Y0C	2Y0E	2Y2Y	2YF5	2YS7	7Y0C	7Y0E	7Y7Y	7YS7	9529	9532	9545	9550	9M0E	9M9M	9MS7	B40C
Roll. ARIMA	4.83	3.5	0.83	0.67	2.5	0.5	0.17	1.83	2.0	1.17	0.0	1.0	0.5	1.5	0.83	0.0	0.67	1.0	2.33	3.17	nan	0.0	nan	5.17
XGBoost Uni.	4.17	3.0	0.5	0.17	1.67	0.5	0.17	1.17	1.33	1.5	0.0	0.83	0.33	1.5	0.67	0.0	0.5	0.67	0.5	2.5	nan	0.0	nan	4.17
Roll. ARIMAX	4.0	3.17	0.83	0.67	2.5	0.5	0.17	1.67	1.83	1.17	0.0	1.0	0.5	1.67	1.0	0.0	0.5	0.5	1.67	2.33	nan	0.0	nan	4.83
XGBoost Multi.	3.83	1.5	0.33	0.5	1.33	0.33	0.17	1.17	1.33	1.0	0.0	0.83	0.33	1.17	0.83	0.0	0.5	0.5	0.5	2.5	nan	0.0	nan	2.67
SEAT ARONA (Time Chunk 2)																								
Algorithm	B40E	B4B4	B4F5	B4S7	E10C	E10E	E1E1	E1S7	F50C	F50E	F5F5	F5S7	F80C	F80E	F8F8	F8S7	L50C	L50E	L5F5	L5L5	L5S7	S70E	S7F5	S7S7
Roll. ARIMA	6.83	6.33	1.17	1.17	2.17	6.67	4.5	1.17	2.0	1.67	0.5	1.0	1.83	1.83	0.67	0.17	1.17	1.17	0.0	2.5	1.0	0.67	0.0	0.17
XGBoost Uni.	8.0	6.67	1.33	1.0	3.0	7.0	3.67	0.67	1.67	1.33	0.33	0.67	1.17	1.33	0.67	0.0	0.5	0.83	0.0	1.17	0.67	0.67	0.0	0.17
Roll. ARIMAX	11.17	6.33	1.5	1.0	1.83	13.17	4.83	1.17	1.17	1.67	0.33	1.0	1.17	1.83	0.67	0.17	1.0	1.17	0.0	2.67	1.0	0.67	0.0	0.17
XGBoost Multi.	6.5	6.5	1.17	1.0	1.67	4.5	3.5	0.67	1.33	0.83	0.17	0.67	1.33	1.0	0.67	0.0	0.5	0.83	0.0	1.0	0.67	0.67	0.0	0.17

SEAT ARONA (Time Chunk 3)																								
Algorithm	0C0C	0C0E	0CF5	0E0C	0E0E	0EF5	0ES7	2Y0C	2Y0E	2Y2Y	2YF5	2YS7	7Y0C	7Y0E	7Y7Y	7YS7	9529	9532	9545	9550	9M0E	9M9M	9MS7	B40C
Roll. ARIMA	5.5	1.17	0.67	0.0	3.0	0.5	0.67	0.33	3.83	6.83	0.5	3.17	1.0	7.67	1.83	2.17	0.67	1.0	2.5	6.17	nan	0.0	nan	2.83
XGBoost Uni.	2.83	0.33	0.17	0.0	2.5	0.33	0.33	0.33	3.5	6.33	0.33	2.5	0.33	3.67	1.17	2.17	0.5	0.0	1.83	4.0	nan	0.0	nan	2.33
Roll. ARIMAX	5.67	0.67	0.33	0.0	4.17	0.67	0.83	0.5	4.5	6.83	0.5	3.0	0.83	7.67	1.5	2.33	0.83	0.0	1.83	6.17	nan	0.0	nan	2.83
XGBoost Multi.	1.83	0.17	0.17	0.0	1.83	0.33	0.33	0.17	2.33	5.17	0.33	2.33	0.33	3.67	0.67	1.67	0.33	0.0	1.67	3.33	nan	0.0	nan	1.5
SEAT ARONA (Time Chunk 3)																								
Algorithm	B40E	B4B4	B4F5	B4S7	E10C	E10E	E1E1	E1S7	F50C	F50E	F5F5	F5S7	F80C	F80E	F8F8	F8S7	L50C	L50E	L5F5	L5L5	L5S7	S70E	S7F5	S7S7
Roll. ARIMA	8.17	22.67	1.5	6.67	2.0	7.5	10.0	2.67	2.33	3.67	2.0	1.5	1.33	2.0	2.0	1.67	0.83	6.33	0.33	3.0	0.83	2.17	0.67	6.33
XGBoost Uni.	9.17	14.83	0.83	4.33	0.83	6.5	8.67	2.0	1.17	1.17	1.0	0.83	1.0	1.83	1.33	0.83	0.17	4.83	0.17	1.67	0.67	2.17	0.83	4.67
Roll. ARIMAX	8.67	14.5	1.17	7.67	1.83	6.83	7.33	2.67	1.5	2.0	1.17	1.33	1.33	3.0	2.0	1.33	0.17	6.0	0.17	3.83	1.0	2.0	0.5	5.83
XGBoost Multi.	7.5	15.5	1.0	3.83	0.83	5.0	7.17	2.17	0.5	1.5	1.17	0.83	1.17	1.83	1.33	1.0	0.17	4.67	0.17	1.67	0.67	1.67	0.5	3.17

SEAT ARONA (Time Chunk 4)																								
Algorithm	0C0C	0C0E	0CF5	0E0C	0E0E	0EF5	0ES7	2Y0C	2Y0E	2Y2Y	2YF5	2YS7	7Y0C	7Y0E	7Y7Y	7YS7	9529	9532	9545	9550	9M0E	9M9M	9MS7	B40C
Roll. ARIMA	0.83	0.0	0.0	0.0	1.5	0.67	1.17	0.17	2.17	2.5	0.67	1.83	0.0	3.0	1.33	2.5	0.67	0.0	0.5	4.0	nan	1.33	nan	0.5
XGBoost Uni.	0.17	0.0	0.0	0.0	0.67	0.5	0.33	0.17	2.67	1.17	0.5	1.83	0.0	1.67	0.83	1.5	0.33	0.0	0.5	3.33	nan	1.0	nan	0.0
Roll. ARIMAX	0.17	0.0	0.0	0.0	1.67	0.5	1.33	0.17	3.33	2.17	0.5	1.17	0.0	6.83	1.83	2.5	0.67	0.0	0.83	4.33	nan	3.17	nan	0.33
XGBoost Multi.	0.17	0.0	0.0	0.0	0.83	0.0	0.33	0.17	2.5	1.0	0.33	1.5	0.0	1.17	0.83	1.67	0.33	0.0	0.33	2.83	nan	1.0	nan	0.0
SEAT ARONA (Time Chunk 4)																								
Algorithm	B40E	B4B4	B4F5	B4S7	E10C	E10E	E1E1	E1S7	F50C	F50E	F5F5	F5S7	F80C	F80E	F8F8	F8S7	L50C	L50E	L5F5	L5L5	L5S7	S70E	S7F5	S7S7
Roll. ARIMA	7.0	19.17	0.67	6.83	0.0	13.5	6.83	4.33	0.0	1.67	1.33	0.33	0.0	1.33	0.83	1.0	0.0	3.67	0.17	1.83	1.17	1.5	0.5	7.83
XGBoost Uni.	2.5	10.67	0.33	1.67	0.0	8.0	5.17	4.17	0.0	1.5	0.67	0.33	0.0	1.5	0.5	0.5	0.0	4.0	0.17	1.5	0.83	1.0	0.33	6.67
Roll. ARIMAX	5.33	19.83	0.67	7.17	0.0	16.17	6.5	4.67	0.0	1.33	0.83	0.33	0.0	1.0	1.67	0.83	0.0	3.83	0.17	1.67	0.83	1.67	0.33	8.17
XGBoost Multi.	4.17	8.33	0.17	1.67	0.0	10.67	3.83	3.33	0.0	1.17	0.67	0.33	0.0	0.5	0.67	0.33	0.0	3.17	0.0	1.5	0.33	1.5	0.17	7.17

SEAT ARONA (Time Chunk 5)																								
Algorithm	0C0C	0C0E	0CF5	0E0C	0E0E	0EF5	0ES7	2Y0C	2Y0E	2Y2Y	2YF5	2YS7	7Y0C	7Y0E	7Y7Y	7YS7	9529	9532	9545	9550	9M0E	9M9M	9MS7	B40C
Roll. ARIMA	0.0	0.0	0.0	0.0	2.2	0.4	1.0	0.0	5.0	2.8	1.0	4.4	0.0	4.6	2.4	1.0	0.0	0.0	0.0	0.0	nan	4.2	nan	0.0
XGBoost Uni.	0.0	0.0	0.0	0.0	1.6	0.2	0.4	0.0	3.2	2.8	0.4	2.6	0.0	4.2	1.8	0.8	0.0	0.0	0.0	0.0	nan	3.4	nan	0.0
Roll. ARIMAX	0.0	0.0	0.0	0.0	1.8	0.6	0.8	0.0	4.4	3.4	1.0	4.2	0.0	3.8	2.4	0.8	0.0	0.0	0.0	0.0	nan	4.2	nan	0.0
XGBoost Multi.	0.0	0.0	0.0	0.0	1.0	0.2	0.6	0.0	3.0	2.0	0.8	1.8	0.0	3.0	1.0	0.6	0.0	0.0	0.0	0.0	nan	2.8	nan	0.0
SEAT ARONA (Time Chunk 5)																								
Algorithm	B40E	B4B4	B4F5	B4S7	E10C	E10E	E1E1	E1S7	F50C	F50E	F5F5	F5S7	F80C	F80E	F8F8	F8S7	L50C	L50E	L5F5	L5L5	L5S7	S70E	S7F5	S7S7
Roll. ARIMA	7.0	9.0	0.2	5.4	0.0	12.8	4.0	2.2	0.0	1.4	1.2	2.0	0.0	0.6	0.0	0.2	0.0	2.4	0.6	4.0	0.8	4.4	0.6	5.0
XGBoost Uni.	4.2	9.4	0.6	1.4	0.0	9.4	3.2	1.2	0.0	1.2	1.2	2.0	0.0	0.8	0.0	0.2	0.0	2.2	0.6	2.4	0.8	3.8	0.2	4.6
Roll. ARIMAX	11.8	13.2	0.8	4.4	0.0	8.6	3.2	2.6	0.0	1.6	1.2	2.2	0.0	0.8	0.0	0.2	0.0	3.0	1.0	2.6	0.8	4.4	0.8	6.0
XGBoost Multi.	2.0	6.8	0.4	1.0	0.0	7.2	2.4	1.2	0.0	1.2	0.4	1.8	0.0	0.8	0.0	0.2	0.0	0.8	0.4	2.2	0.4	3.2	0.2	3.6

Table B.3: Mean Average Error (MAE) per SEAT Ibiza car variant (car model plus exterior color) and time chunk of each forecasting technique. *Roll.* refers to Rolling, *Uni.* refers to Univariate, *Multi.* refers to Multivariate, and *nan* implies that forecast was not computed because there was not data in both time series.

SEAT IBIZA (Time Chunk 1)												
Algorithm	0C0C	0E0E	2Y2Y	7Y7Y	9550	9M9M	B4B4	E1E1	F5F5	F8F8	L5L5	S7S7
Roll. ARIMA	12.33	9.83	0.0	19.83	8.67	0.0	24.5	6.83	0.0	4.83	0.0	0.0
XGBoost Uni.	7.83	6.83	0.0	9.5	5.0	0.0	15.0	6.83	0.0	5.83	0.0	0.0
Roll. ARIMAX	11.67	8.33	0.0	21.5	9.5	0.0	17.0	7.83	0.0	4.33	0.0	0.0
XGBoost Multi.	7.5	4.5	0.0	9.0	4.5	0.0	12.0	4.17	0.0	3.67	0.0	0.0

SEAT IBIZA (Time Chunk 2)												
Algorithm	0C0C	0E0E	2Y2Y	7Y7Y	9550	9M9M	B4B4	E1E1	F5F5	F8F8	L5L5	S7S7
Roll. ARIMA	9.33	11.17	1.83	15.83	8.17	0.0	10.33	11.0	1.33	6.83	2.0	0.83
XGBoost Uni.	7.83	6.67	2.5	14.17	5.33	0.0	6.33	6.5	1.0	4.0	1.5	0.83
Roll. ARIMAX	14.17	9.33	2.0	11.5	11.5	0.0	22.33	7.5	1.17	22.83	2.67	0.83
XGBoost Multi.	6.0	4.83	1.17	8.5	3.0	0.0	6.67	6.0	0.67	4.0	1.17	0.83

SEAT IBIZA (Time Chunk 3)												
Algorithm	0C0C	0E0E	2Y2Y	7Y7Y	9550	9M9M	B4B4	E1E1	F5F5	F8F8	L5L5	S7S7
Roll. ARIMA	5.83	3.67	4.83	11.0	9.0	0.0	82.83	14.17	2.67	4.17	7.17	12.67
XGBoost Uni.	2.0	2.83	5.17	8.67	5.0	0.0	79.83	9.17	2.17	4.33	5.0	7.5
Roll. ARIMAX	4.33	3.33	6.0	8.17	11.5	0.0	82.83	10.33	2.5	4.33	5.33	9.5
XGBoost Multi.	1.83	2.5	3.67	5.33	5.17	0.0	75.5	9.67	1.83	3.5	3.67	7.83

SEAT IBIZA (Time Chunk 4)												
Algorithm	0C0C	0E0E	2Y2Y	7Y7Y	9550	9M9M	B4B4	E1E1	F5F5	F8F8	L5L5	S7S7
Roll. ARIMA	0.17	2.33	9.83	3.5	6.83	1.83	22.67	6.5	2.17	5.5	3.83	7.83
XGBoost Uni.	0.17	2.17	5.83	4.83	5.0	1.5	16.67	5.5	1.67	3.67	3.33	5.83
Roll. ARIMAX	0.0	3.17	11.0	7.0	9.67	11.67	47.67	8.67	3.0	2.33	4.0	7.83
XGBoost Multi.	0.0	1.67	3.67	3.83	5.0	1.5	17.0	3.67	1.5	2.83	2.83	3.67

SEAT IBIZA (Time Chunk 5)												
Algorithm	0C0C	0E0E	2Y2Y	7Y7Y	9550	9M9M	B4B4	E1E1	F5F5	F8F8	L5L5	S7S7
Roll. ARIMA	0.0	6.2	8.8	9.4	0.4	4.8	34.6	12.8	3.0	0.0	6.2	7.4
XGBoost Uni.	0.0	1.4	5.6	4.0	0.4	3.2	22.6	11.8	0.8	0.0	4.4	4.0
Roll. ARIMAX	0.0	7.2	14.0	8.8	0.4	4.0	28.8	27.6	1.2	0.0	6.2	12.0
XGBoost Multi.	0.0	1.6	4.0	4.0	0.4	2.8	16.0	9.6	0.4	0.0	5.2	4.6

Table B.4: Mean Average Error (MAE) per SEAT Leon 5D car variant (car model plus exterior color) and time chunk of each forecasting technique. *Roll.* refers to Rolling, *Uni.* refers to Univariate, *Multi.* refers to Multivariate, and *nan* implies that forecast was not computed because there was not data in both time series.

SEAT LEON 5D (Time Chunk 1)																		
Algorithm	0C0C	0E0E	2Y2Y	7V7V	7Y7Y	9019	9550	9M9M	B4B4	C0C0	E1E1	F5F5	I4I4	L5L5	P5P5	S3S3	S7S7	T4T4
Roll. ARIMA	14.67	5.5	11.67	nan	13.17	nan	8.5	4.17	17.5	3.17	7.0	0.0	nan	4.5	1.5	nan	0.0	7.83
XGBoost Uni.	7.0	3.5	7.17	nan	6.0	nan	4.33	3.67	4.33	1.83	3.5	0.0	nan	3.67	1.0	nan	0.0	4.0
Roll. ARIMAX	8.0	4.67	9.0	nan	9.33	nan	8.33	4.33	18.17	3.83	8.67	0.0	nan	3.5	1.0	nan	0.0	6.0
XGBoost Multi.	5.17	2.33	7.0	nan	3.0	nan	3.67	2.5	6.0	2.0	4.33	0.0	nan	3.0	0.67	nan	0.0	3.5

SEAT LEON 5D (Time Chunk 2)																		
Algorithm	0C0C	0E0E	2Y2Y	7V7V	7Y7Y	9019	9550	9M9M	B4B4	C0C0	E1E1	F5F5	I4I4	L5L5	P5P5	S3S3	S7S7	T4T4
Roll. ARIMA	2.83	2.17	11.5	nan	14.33	nan	4.5	0.0	7.5	3.5	13.0	0.83	nan	4.33	0.0	nan	0.17	0.0
XGBoost Uni.	2.83	1.17	9.33	nan	7.83	nan	2.83	0.0	4.5	3.17	7.83	1.17	nan	4.33	0.0	nan	0.17	0.0
Roll. ARIMAX	6.0	3.83	11.83	nan	13.33	nan	4.83	0.0	8.33	3.0	13.67	1.0	nan	4.5	0.0	nan	0.17	0.0
XGBoost Multi.	2.17	1.67	9.5	nan	4.17	nan	2.83	0.0	2.67	1.33	7.67	1.17	nan	4.17	0.0	nan	0.17	0.0

SEAT LEON 5D (Time Chunk 3)																		
Algorithm	0C0C	0E0E	2Y2Y	7V7V	7Y7Y	9019	9550	9M9M	B4B4	C0C0	E1E1	F5F5	I4I4	L5L5	P5P5	S3S3	S7S7	T4T4
Roll. ARIMA	2.67	7.17	17.33	nan	13.83	nan	4.0	0.0	23.33	2.0	12.67	1.83	nan	6.5	0.0	nan	10.67	0.0
XGBoost Uni.	1.67	3.5	8.67	nan	6.83	nan	1.5	0.0	5.83	2.0	9.0	1.17	nan	3.5	0.0	nan	6.17	0.0
Roll. ARIMAX	9.0	10.0	16.67	nan	27.17	nan	5.83	0.0	21.67	2.33	15.33	1.83	nan	8.0	0.0	nan	9.0	0.0
XGBoost Multi.	1.33	3.17	7.67	nan	7.33	nan	1.33	0.0	5.83	1.67	7.83	1.0	nan	1.83	0.0	nan	5.17	0.0

SEAT LEON 5D (Time Chunk 4)																		
Algorithm	0C0C	0E0E	2Y2Y	7V7V	7Y7Y	9019	9550	9M9M	B4B4	C0C0	E1E1	F5F5	I4I4	L5L5	P5P5	S3S3	S7S7	T4T4
Roll. ARIMA	0.0	4.5	25.17	nan	10.67	nan	6.0	0.67	15.17	1.0	10.67	1.33	nan	5.33	0.0	nan	13.0	0.0
XGBoost Uni.	0.0	2.67	11.83	nan	5.83	nan	5.67	0.5	10.83	2.0	6.5	0.83	nan	2.33	0.0	nan	8.33	0.0
Roll. ARIMAX	0.0	4.33	22.0	nan	9.33	nan	3.83	0.5	16.17	1.17	12.5	1.17	nan	5.5	0.0	nan	13.5	0.0
XGBoost Multi.	0.0	3.0	11.67	nan	4.17	nan	4.0	0.5	11.0	0.67	4.83	0.67	nan	2.17	0.0	nan	6.83	0.0

SEAT LEON 5D (Time Chunk 5)																		
Algorithm	0C0C	0E0E	2Y2Y	7V7V	7Y7Y	9019	9550	9M9M	B4B4	C0C0	E1E1	F5F5	I4I4	L5L5	P5P5	S3S3	S7S7	T4T4
Roll. ARIMA	0.0	6.4	13.6	nan	8.4	nan	1.4	3.8	11.6	0.4	12.0	1.2	nan	6.4	0.0	nan	14.8	0.0
XGBoost Uni.	0.0	2.0	6.0	nan	6.8	nan	0.8	3.4	13.8	0.0	9.2	0.6	nan	5.6	0.0	nan	7.8	0.0
Roll. ARIMAX	0.0	6.4	20.2	nan	10.8	nan	0.4	4.0	20.0	0.8	11.6	1.0	nan	6.8	0.0	nan	21.6	0.0
XGBoost Multi.	0.0	4.0	7.8	nan	7.8	nan	0.4	3.4	15.8	0.0	5.4	0.4	nan	3.4	0.0	nan	5.2	0.0

Table B.5: Mean Average Error (MAE) per SEAT Arona car variant (car model plus compound region) and time chunk of each forecasting technique. *Roll.* refers to Rolling, *Uni.* refers to Univariate, *Multi.* refers to Multivariate, and *nan* implies that forecast was not computed because there was not data in both time series.

SEAT ARONA (Time Chunk 1)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	1.0	1.83	2.17	1.33	4.5	3.5
XGBoost Uni.	1.0	1.83	2.0	1.17	4.67	2.33
Roll. ARIMAX	0.83	1.33	2.5	1.17	4.0	2.0
XGBoost Multi.	1.0	1.33	1.67	1.17	3.67	2.0

SEAT ARONA (Time Chunk 2)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	7.17	12.0	11.67	10.33	31.0	8.33
XGBoost Uni.	4.5	10.5	10.5	9.17	15.33	7.83
Roll. ARIMAX	6.67	11.33	16.17	10.83	23.67	8.0
XGBoost Multi.	3.17	5.17	6.5	7.33	11.67	7.33

SEAT ARONA (Time Chunk 3)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	12.0	10.67	18.17	14.0	27.5	13.0
XGBoost Uni.	7.17	9.33	13.33	8.5	16.83	11.67
Roll. ARIMAX	5.83	15.5	21.33	17.33	28.5	11.33
XGBoost Multi.	6.33	8.83	13.83	7.83	9.83	9.33

SEAT ARONA (Time Chunk 4)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	5.83	15.83	18.83	7.33	19.5	13.5
XGBoost Uni.	5.0	13.83	14.17	8.33	18.33	10.83
Roll. ARIMAX	6.0	15.33	17.83	6.17	27.33	17.17
XGBoost Multi.	5.17	10.0	10.33	5.67	16.17	8.67

SEAT ARONA (Time Chunk 5)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	8.8	14.0	14.2	12.2	34.2	11.2
XGBoost Uni.	7.6	5.0	6.6	8.8	24.8	8.0
Roll. ARIMAX	6.6	16.4	15.8	12.2	25.8	11.8
XGBoost Multi.	5.2	4.8	5.4	7.0	11.8	7.6

Table B.6: Mean Average Error (MAE) per SEAT Ibiza car variant (car model plus compound region) and time chunk of each forecasting technique. *Roll.* refers to Rolling, *Uni.* refers to Univariate, *Multi.* refers to Multivariate, and *nan* implies that forecast was not computed because there was not data in both time series.

SEAT IBIZA (Time Chunk 1)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	3.33	19.17	10.83	17.0	15.67	13.33
XGBoost Uni.	2.83	15.5	10.33	10.0	17.67	10.0
Roll. ARIMAX	3.83	19.0	9.17	16.67	15.33	15.67
XGBoost Multi.	2.33	12.5	9.17	8.0	15.5	11.17

SEAT IBIZA (Time Chunk 2)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	2.5	17.33	14.83	10.67	16.33	11.5
XGBoost Uni.	4.17	11.33	11.67	6.17	14.5	12.0
Roll. ARIMAX	2.0	16.33	32.67	12.5	15.33	20.17
XGBoost Multi.	3.0	10.17	8.83	8.17	13.33	9.33

SEAT IBIZA (Time Chunk 3)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	6.33	17.5	30.67	25.83	19.0	24.0
XGBoost Uni.	5.5	16.33	23.17	12.5	16.5	16.5
Roll. ARIMAX	5.33	18.33	26.33	25.67	34.33	24.0
XGBoost Multi.	6.5	17.0	22.67	12.67	12.67	15.5

SEAT IBIZA (Time Chunk 4)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	8.67	13.0	11.5	10.0	12.83	12.17
XGBoost Uni.	7.83	8.67	10.5	7.0	8.0	4.33
Roll. ARIMAX	11.17	13.33	18.67	19.0	14.33	11.5
XGBoost Multi.	7.67	6.67	10.17	8.33	6.33	6.5

SEAT IBIZA (Time Chunk 5)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	8.2	6.4	42.4	11.2	11.6	14.8
XGBoost Uni.	3.4	4.2	11.8	14.0	14.8	6.8
Roll. ARIMAX	9.0	10.0	13.6	10.6	13.2	14.2
XGBoost Multi.	4.2	4.2	8.6	8.4	11.2	5.6

Table B.7: Mean Average Error (MAE) per SEAT Leon 5D car variant (car model plus compound region) and time chunk of each forecasting technique. *Roll.* refers to Rolling, *Uni.* refers to Univariate, *Multi.* refers to Multivariate, and *nan* implies that forecast was not computed because there was not data in both time series.

			SEAT LEON 5D (Time Chunk 1)			
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	5.67	12.5	12.17	10.5	18.83	11.17
XGBoost Uni.	4.17	9.67	7.33	8.33	13.5	6.17
Roll. ARIMAX	6.33	12.83	12.17	10.17	10.5	13.33
XGBoost Multi.	3.67	7.17	5.67	5.33	9.67	4.33

			SEAT LEON 5D (Time Chunk 2)			
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	3.5	8.5	8.0	5.67	12.83	9.0
XGBoost Uni.	1.5	7.83	4.33	4.0	9.83	9.17
Roll. ARIMAX	2.83	19.33	10.0	7.67	27.67	14.17
XGBoost Multi.	0.83	7.5	4.17	1.5	12.33	6.83

			SEAT LEON 5D (Time Chunk 3)			
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	5.0	17.17	17.0	12.5	13.83	18.17
XGBoost Uni.	2.5	10.33	11.67	7.33	10.83	11.0
Roll. ARIMAX	5.33	20.17	18.33	10.83	13.5	24.17
XGBoost Multi.	2.5	9.83	7.83	7.17	12.67	10.83

			SEAT LEON 5D (Time Chunk 4)			
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	2.67	18.5	15.83	8.5	10.0	17.17
XGBoost Uni.	1.67	11.67	14.5	6.33	6.5	11.5
Roll. ARIMAX	2.5	12.83	15.83	7.67	9.5	20.83
XGBoost Multi.	0.67	9.5	9.67	4.67	2.67	10.17

			SEAT LEON 5D (Time Chunk 5)			
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	4.6	12.0	14.0	5.6	20.0	16.4
XGBoost Uni.	1.6	7.8	13.4	8.4	16.6	11.8
Roll. ARIMAX	3.0	17.2	12.6	6.2	22.6	16.0
XGBoost Multi.	1.0	9.8	13.2	7.2	11.4	8.0

Table B.8: Mean Average Error (MAE) per SEAT Leon ST car variant (car model plus compound region) and time chunk of each forecasting technique. *Roll.* refers to Rolling, *Uni.* refers to Univariate, *Multi.* refers to Multivariate, and *nan* implies that forecast was not computed because there was not data in both time series.

SEAT LEON ST (Time Chunk 1)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	0.5	9.67	5.0	2.67	8.33	11.17
XGBoost Uni.	0.83	8.0	4.0	1.5	4.0	5.67
Roll. ARIMAX	0.83	9.67	5.17	3.17	8.0	9.33
XGBoost Multi.	0.5	4.5	3.33	1.83	3.5	6.0

SEAT LEON ST (Time Chunk 2)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	0.83	2.83	5.17	1.83	12.0	6.0
XGBoost Uni.	0.33	3.5	3.83	0.67	5.5	3.83
Roll. ARIMAX	0.83	3.17	6.0	2.0	5.67	4.33
XGBoost Multi.	0.33	3.17	3.33	0.5	4.5	3.0

SEAT LEON ST (Time Chunk 3)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	1.17	11.0	6.67	1.83	10.33	13.5
XGBoost Uni.	1.17	5.17	4.5	1.0	8.83	12.5
Roll. ARIMAX	2.0	9.33	6.17	1.67	9.67	13.83
XGBoost Multi.	0.67	4.5	3.83	1.0	7.67	11.67

SEAT LEON ST (Time Chunk 4)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	0.33	13.67	7.67	2.5	6.83	7.33
XGBoost Uni.	0.33	12.0	3.33	1.67	5.33	4.17
Roll. ARIMAX	0.5	16.0	7.17	3.33	8.5	12.67
XGBoost Multi.	0.0	8.17	3.17	1.5	4.5	3.5

SEAT LEON ST (Time Chunk 5)						
Algorithm	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
Roll. ARIMA	0.6	5.0	3.0	2.6	7.4	9.4
XGBoost Uni.	1.0	5.4	1.4	2.2	4.6	4.4
Roll. ARIMAX	0.6	4.6	5.0	2.6	5.8	10.0
XGBoost Multi.	0.6	3.2	0.8	1.4	2.2	4.0

Appendix C

Weekly Mix Sales Assessment

The information presented in this appendix complements the findings outlined in Subsection 6.2.2.

Table C.1: R2 Score (%) of each forecasting technique for the weekly sales mixes of SEAT Arona at exterior color level. *Uni.* refers to Univariate, *Multi.* refers to Multivariate.

SEAT ARONA (Exterior Color)					
Time Chunk	Week	ARIMA	XGBoost Uni.	ARIMAX	XGBoost Multi.
1^{st}	2017-09-17	0.0	0.0	0.0	0.0
	2017-09-24	0.0	0.0	0.0	0.0
	2017-10-01	3.62	0.0	0.87	0.0
	2017-10-08	21.47	15.6	0.69	37.5
	2017-10-15	0.02	0.49	0.83	33.88
	2017-10-22	78.51	78.8	27.43	69.65
2^{nd}	2018-04-15	91.62	95.66	41.74	93.77
	2018-04-22	95.96	97.3	98.1	96.1
	2018-04-29	90.47	89.71	88.46	93.69
	2018-05-06	89.35	91.99	89.78	90.72
	2018-05-13	90.69	91.41	92.4	96.33
	2018-05-20	89.04	92.3	88.75	98.1
3^{rd}	2018-11-11	84.63	96.1	85.85	97.52
	2018-11-18	86.47	90.96	85.34	90.99
	2018-11-25	82.89	92.02	83.2	90.43
	2018-12-02	80.32	88.49	85.47	89.1
	2018-12-09	84.75	84.23	85.61	85.46
	2018-12-16	88.25	90.53	82.9	96.25
4^{th}	2019-06-09	93.98	91.49	87.32	90.85
	2019-06-16	79.63	81.38	72.39	89.98
	2019-06-23	91.68	97.45	85.1	95.64
	2019-06-30	95.61	93.96	91.36	94.89
	2019-07-07	90.41	90.44	88.77	92.68
	2019-07-14	88.92	90.28	69.43	92.52
5^{th}	2020-01-05	87.98	89.51	85.44	86.14
	2020-01-12	81.83	87.89	55.11	94.87
	2020-01-19	95.23	95.13	86.31	96.77
	2020-01-26	93.96	97.11	94.79	98.31
	2020-02-02	93.11	90.8	95.38	95.17

Table C.2: R2 Score (%) of each forecasting technique for the weekly sales mixes of SEAT Ibiza at exterior color level. *Uni.* refers to Univariate, *Multi.* refers to Multivariate.

SEAT IBIZA (Exterior Color)					
Time Chunk	Week	ARIMA	XGBoost Uni.	ARIMAX	XGBoost Multi.
1^{st}	2017-09-17	95.09	88.72	91.28	95.18
	2017-09-24	95.24	93.32	93.74	83.18
	2017-10-01	92.81	85.8	96.95	92.56
	2017-10-08	75.68	84.32	76.46	95.98
	2017-10-15	92.69	93.73	86.21	95.96
	2017-10-22	92.92	88.56	93.96	96.18
2^{nd}	2018-04-15	68.12	92.16	96.63	96.9
	2018-04-22	91.91	97.5	31.64	98.34
	2018-04-29	95.6	86.66	88.64	93.23
	2018-05-06	95.09	97.62	92.05	98.52
	2018-05-13	93.15	92.93	91.57	99.01
	2018-05-20	93.7	94.36	86.54	79.38
3^{rd}	2018-11-11	92.25	92.81	94.34	94.42
	2018-11-18	84.51	92.56	81.77	99.63
	2018-11-25	93.3	97.57	93.91	91.89
	2018-12-02	95.23	95.16	96.1	76.35
	2018-12-09	97.81	89.79	97.86	96.02
	2018-12-16	98.11	98.74	98.01	99.44
4^{th}	2019-06-09	94.85	97.64	66.23	98.75
	2019-06-16	97.74	98.97	80.45	99.35
	2019-06-23	94.48	96.88	51.92	98.59
	2019-06-30	97.51	97.14	97.22	93.76
	2019-07-07	96.26	95.98	93.57	97.37
	2019-07-14	97.95	96.73	46.29	98.93
5^{th}	2020-01-05	76.17	96.11	71.61	94.16
	2020-01-12	91.1	90.51	78.96	88.46
	2020-01-19	80.7	98.14	94.95	97.66
	2020-01-26	89.21	98.79	42.2	99.74
	2020-02-02	81.84	87.65	88.08	91.45

Table C.3: R2 Score (%) of each forecasting technique for the weekly sales mixes of SEAT Leon 5D at exterior color level. *Uni.* refers to Univariate, *Multi.* refers to Multivariate.

SEAT LEON 5D (Exterior Color)					
Time Chunk	Week	ARIMA	XGBoost Uni.	ARIMAX	XGBoost Multi.
1^{st}	2017-09-17	56.5	90.19	63.62	92.07
	2017-09-24	67.58	70.31	60.84	88.75
	2017-10-01	83.12	94.35	88.49	90.84
	2017-10-08	79.34	70.3	79.45	89.59
	2017-10-15	53.16	85.3	74.1	90.91
	2017-10-22	55.13	94.92	72.66	94.05
2^{nd}	2018-04-15	55.5	93.15	90.68	98.06
	2018-04-22	91.17	89.83	74.72	89.92
	2018-04-29	92.68	88.01	83.7	92.43
	2018-05-06	85.24	98.56	80.48	95.01
	2018-05-13	73.86	92.17	68.11	93.61
	2018-05-20	85.64	85.52	71.68	91.51
3^{rd}	2018-11-11	80.04	91.03	39.05	94.75
	2018-11-18	94.98	92.52	88.97	96.89
	2018-11-25	87.45	94.62	53.97	96.33
	2018-12-02	94.51	95.66	91.62	94.46
	2018-12-09	84.96	95.03	75.49	91.75
	2018-12-16	62.47	89.05	75.11	94.52
4^{th}	2019-06-09	97.42	86.38	97.45	92.04
	2019-06-16	91.75	82.33	90.17	98.69
	2019-06-23	84.65	93.91	89.22	90.32
	2019-06-30	93.66	93.19	95.06	99.53
	2019-07-07	86.72	95.47	88.16	95.7
	2019-07-14	96.35	88.64	90.86	88.99
5^{th}	2020-01-05	84.58	97.08	67.47	92.4
	2020-01-12	86.09	63.24	57.32	79.25
	2020-01-19	78.97	83.27	84.18	84.57
	2020-01-26	77.09	95.23	56.31	91.93
	2020-02-02	87.57	84.68	90.4	91.36

Table C.4: R2 Score (%) of each forecasting technique for the weekly sales mixes of SEAT Leon ST at exterior color level. *Uni.* refers to Univariate, *Multi.* refers to Multivariate.

SEAT LEON ST (Exterior Color)					
Time Chunk	Week	ARIMA	XGBoost Uni.	ARIMAX	XGBoost Multi.
1^{st}	2017-09-17	54.73	93.92	77.5	98.03
	2017-09-24	83.55	93.25	91.81	95.5
	2017-10-01	92.64	86.78	84.77	95.51
	2017-10-08	60.74	66.47	59.87	64.11
	2017-10-15	71.25	74.81	63.0	73.71
	2017-10-22	66.31	76.2	50.46	80.77
2^{nd}	2018-04-15	80.84	85.53	91.0	93.34
	2018-04-22	73.96	73.97	91.1	95.57
	2018-04-29	82.88	96.14	89.21	96.19
	2018-05-06	82.92	96.18	71.68	96.6
	2018-05-13	87.02	81.73	74.25	77.74
	2018-05-20	93.05	83.51	10.04	96.2
3^{rd}	2018-11-11	24.74	89.86	63.39	70.15
	2018-11-18	70.48	70.17	74.97	95.57
	2018-11-25	77.32	60.65	71.76	87.29
	2018-12-02	80.65	66.01	73.75	72.54
	2018-12-09	66.66	52.25	65.38	43.14
	2018-12-16	60.82	91.88	70.58	97.48
4^{th}	2019-06-09	55.08	66.66	44.5	75.53
	2019-06-16	81.66	92.25	57.76	95.55
	2019-06-23	85.85	95.27	91.81	93.34
	2019-06-30	81.98	89.68	84.58	93.68
	2019-07-07	82.09	96.43	95.73	97.0
	2019-07-14	87.96	87.88	86.75	81.07
5^{th}	2020-01-05	52.77	93.24	74.26	92.49
	2020-01-12	13.88	59.23	23.66	77.82
	2020-01-19	78.24	83.35	86.61	85.88
	2020-01-26	74.65	90.3	81.49	93.25
	2020-02-02	68.22	90.62	74.52	92.58

Table C.5: R2 Score (%) of each forecasting technique for the weekly sales mixes of SEAT Arona at compound region level. *Uni.* refers to Univariate, *Multi.* refers to Multivariate.

SEAT ARONA (Compound Region)					
Time Chunk	Week	ARIMA	XGBoost Uni.	ARIMAX	XGBoost Multi.
1^{st}	2017-09-17	0.0	0.0	0.0	0.0
	2017-09-24	0.0	0.0	0.0	0.0
	2017-10-01	0.0	0.0	2.94	0.0
	2017-10-08	31.06	31.06	6.09	6.17
	2017-10-15	28.39	4.74	100.0	1.4
	2017-10-22	14.12	1.51	14.41	67.46
2^{nd}	2018-04-15	3.53	55.88	45.37	32.87
	2018-04-22	81.74	70.24	76.98	89.4
	2018-04-29	0.04	62.39	49.26	95.53
	2018-05-06	53.4	68.38	66.81	65.06
	2018-05-13	75.8	52.07	91.89	94.16
	2018-05-20	3.85	75.58	42.87	52.91
3^{rd}	2018-11-11	79.52	7.38	67.13	74.82
	2018-11-18	80.88	71.25	90.0	93.08
	2018-11-25	80.84	86.07	77.86	84.58
	2018-12-02	93.09	86.9	87.39	82.62
	2018-12-09	66.12	45.12	43.82	44.17
	2018-12-16	27.11	36.2	43.79	84.21
4^{th}	2019-06-09	85.52	37.66	74.92	51.44
	2019-06-16	53.01	51.34	3.28	88.59
	2019-06-23	16.36	63.84	77.25	57.47
	2019-06-30	37.85	5.86	66.2	81.86
	2019-07-07	49.7	55.65	53.31	53.7
	2019-07-14	80.86	72.6	70.79	75.59
5^{th}	2020-01-05	90.72	79.06	91.8	97.26
	2020-01-12	37.22	68.35	0.44	85.38
	2020-01-19	89.33	62.12	92.77	99.07
	2020-01-26	85.43	97.47	96.87	97.46
	2020-02-02	38.5	91.42	89.19	92.23

Table C.6: R2 Score (%) of each forecasting technique for the weekly sales mixes of SEAT Ibiza at compound region level. *Uni.* refers to Univariate, *Multi.* refers to Multivariate.

SEAT IBIZA (Compound Region)					
Time Chunk	Week	ARIMA	XGBoost Uni.	ARIMAX	XGBoost Multi.
1^{st}	2017-09-17	35.87	55.26	40.35	58.31
	2017-09-24	64.01	43.43	64.42	32.72
	2017-10-01	60.7	34.73	71.16	51.66
	2017-10-08	90.07	87.32	79.56	98.46
	2017-10-15	30.83	71.46	49.75	41.63
	2017-10-22	93.59	38.39	80.25	81.11
2^{nd}	2018-04-15	46.59	73.23	24.75	45.12
	2018-04-22	82.46	75.24	6.12	73.49
	2018-04-29	57.76	82.78	48.19	93.57
	2018-05-06	45.34	82.63	15.88	72.51
	2018-05-13	77.42	76.09	71.57	75.05
	2018-05-20	88.12	90.64	78.34	92.55
3^{rd}	2018-11-11	73.06	63.55	67.71	82.08
	2018-11-18	51.45	74.61	50.19	95.9
	2018-11-25	93.69	88.34	6.73	94.99
	2018-12-02	69.61	85.46	65.31	59.94
	2018-12-09	93.83	89.5	87.18	65.13
	2018-12-16	72.81	85.34	86.09	89.0
4^{th}	2019-06-09	55.31	32.39	20.03	84.04
	2019-06-16	90.85	71.74	49.51	93.7
	2019-06-23	96.33	82.32	41.81	84.85
	2019-06-30	26.84	72.09	51.78	96.76
	2019-07-07	28.28	73.77	45.1	71.24
	2019-07-14	44.02	95.81	14.96	78.41
5^{th}	2020-01-05	22.96	54.4	66.29	93.69
	2020-01-12	0.1	88.07	32.27	76.5
	2020-01-19	55.06	85.32	40.4	32.04
	2020-01-26	45.12	78.29	93.01	97.11
	2020-02-02	60.51	83.04	64.94	75.81

Table C.7: R2 Score (%) of each forecasting technique for the weekly sales mixes of SEAT Leon 5D at compound region level. *Uni.* refers to Univariate, *Multi.* refers to Multivariate.

SEAT LEON 5D (Compound Region)					
Time Chunk	Week	ARIMA	XGBoost Uni.	ARIMAX	XGBoost Multi.
1^{st}	2017-09-17	84.92	76.89	57.41	94.23
	2017-09-24	88.04	83.71	65.72	96.16
	2017-10-01	98.17	85.03	90.49	86.71
	2017-10-08	91.17	87.32	95.84	85.21
	2017-10-15	31.01	33.94	34.57	82.84
	2017-10-22	54.16	75.63	82.2	80.59
2^{nd}	2018-04-15	45.96	86.99	14.62	85.12
	2018-04-22	91.0	91.47	94.79	98.98
	2018-04-29	86.07	93.11	16.96	88.95
	2018-05-06	72.63	80.45	71.69	94.8
	2018-05-13	63.91	58.18	29.61	81.34
	2018-05-20	87.73	82.83	61.38	83.1
3^{rd}	2018-11-11	78.94	52.59	52.56	58.68
	2018-11-18	14.16	20.28	18.75	30.23
	2018-11-25	74.94	68.58	60.11	87.25
	2018-12-02	71.67	71.94	82.68	83.96
	2018-12-09	50.32	87.51	3.82	76.97
	2018-12-16	34.94	4.8	11.09	23.55
4^{th}	2019-06-09	95.57	77.98	71.48	91.5
	2019-06-16	16.58	62.37	35.95	52.07
	2019-06-23	78.24	75.95	56.96	70.84
	2019-06-30	83.47	76.9	78.09	88.45
	2019-07-07	56.11	32.42	68.32	79.28
	2019-07-14	48.72	88.74	29.37	72.7
5^{th}	2020-01-05	76.21	63.68	96.26	94.33
	2020-01-12	75.89	60.86	76.26	77.75
	2020-01-19	64.32	70.5	69.39	49.43
	2020-01-26	86.98	76.55	79.69	79.0
	2020-02-02	47.3	87.11	67.72	83.23

Table C.8: R2 Score (%) of each forecasting technique for the weekly sales mixes of SEAT Leon ST at compound region level. *Uni.* refers to Univariate, *Multi.* refers to Multivariate.

SEAT LEON ST (Compound Region)					
Time Chunk	Week	ARIMA	XGBoost Uni.	ARIMAX	XGBoost Multi.
1^{st}	2017-09-17	61.28	72.32	55.51	86.84
	2017-09-24	94.28	95.1	94.75	96.37
	2017-10-01	92.71	83.69	86.29	94.9
	2017-10-08	79.92	80.83	94.02	81.85
	2017-10-15	94.35	91.67	90.54	97.5
	2017-10-22	21.71	70.31	37.89	74.8
2^{nd}	2018-04-15	68.66	92.76	90.38	97.93
	2018-04-22	77.5	64.17	80.56	95.96
	2018-04-29	78.61	98.55	88.85	99.08
	2018-05-06	89.83	63.39	62.29	73.33
	2018-05-13	37.55	92.18	87.41	81.67
	2018-05-20	65.62	96.77	87.73	89.13
3^{rd}	2018-11-11	58.03	43.85	72.25	59.45
	2018-11-18	90.44	91.36	85.34	83.97
	2018-11-25	66.07	69.35	39.55	66.36
	2018-12-02	88.02	79.43	78.97	63.31
	2018-12-09	23.07	17.75	25.9	3.47
	2018-12-16	58.18	80.76	79.92	63.56
4^{th}	2019-06-09	80.67	85.5	31.01	96.9
	2019-06-16	91.58	95.79	81.4	98.75
	2019-06-23	91.98	99.05	87.88	99.54
	2019-06-30	93.18	85.54	82.38	82.44
	2019-07-07	83.94	87.08	92.14	91.51
	2019-07-14	93.71	90.73	73.14	98.25
5^{th}	2020-01-05	28.9	94.88	47.76	92.02
	2020-01-12	42.74	62.55	25.86	86.08
	2020-01-19	76.61	76.72	63.51	90.67
	2020-01-26	98.85	96.17	94.63	96.64
	2020-02-02	58.17	83.18	77.43	89.53

Appendix D

Forecast Comparison

The information provided in this appendix complements the results presented in Subsection 6.2.2.

Table D.1: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 50, Population Size 30, Number of Generations 20 - Trial 1.

Experiment: (50, 30, 20) - Trial 1												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.83	1.83	2.0	1.0	3.83	2.33	5.5	18.17	6.17	7.17	23.5	5.33
2^{nd}	2.67	19.33	16.0	13.17	10.67	5.33	2.83	6.83	31.0	3.83	8.0	6.5
3^{rd}	9.33	3.33	28.0	7.17	22.17	6.5	2.67	10.17	16.0	8.17	11.5	11.67
4^{th}	8.17	7.5	6.83	5.0	14.33	5.67	2.33	5.0	6.83	2.0	6.17	3.67
5^{th}	2.4	2.8	3.0	12.2	9.4	19.4	8.0	12.8	28.0	14.8	10.2	20.6
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	6.67	16.33	12.17	9.83	5.83	10.17	0.17	14.17	2.5	1.33	1.67	5.5
2^{nd}	0.67	9.33	12.17	13.5	6.67	23.5	1.17	2.83	9.33	2.67	2.67	8.5
3^{rd}	3.83	22.33	5.33	12.67	7.5	10.0	1.5	11.5	2.67	1.83	2.67	9.33
4^{th}	2.0	8.0	17.0	3.83	16.17	5.83	1.0	12.83	2.33	1.0	3.0	2.0
5^{th}	0.6	8.0	12.8	4.8	23.2	12.4	0.4	3.0	1.6	3.6	6.8	8.2

Table D.2: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 50, Population Size 30, Number of Generations 20 - Trial 2.

Experiment: (50, 30, 20) - Trial 2												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.83	1.83	2.0	1.17	3.17	2.33	5.5	7.83	6.5	6.67	12.83	4.17
2^{nd}	1.83	11.83	16.17	6.17	35.83	4.0	1.17	18.17	28.0	6.17	8.67	5.33
3^{rd}	5.83	3.17	26.83	12.33	21.83	7.17	10.33	14.67	39.5	9.33	11.17	12.17
4^{th}	9.0	7.17	6.33	5.0	14.5	4.17	3.33	4.67	5.0	3.83	22.33	2.83
5^{th}	3.0	2.8	22.0	5.4	11.6	6.0	3.6	14.0	26.0	16.6	7.4	21.2
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	5.0	17.0	14.33	11.33	5.33	10.17	0.17	4.17	8.0	1.0	1.0	5.0
2^{nd}	1.83	20.67	10.17	11.5	7.33	29.0	1.5	12.5	8.17	2.5	2.5	9.67
3^{rd}	2.17	22.33	5.5	16.33	8.5	10.5	1.17	3.83	6.0	0.67	3.67	9.0
4^{th}	1.17	9.17	17.0	3.5	17.83	6.17	1.0	15.17	2.67	1.17	8.5	2.17
5^{th}	0.6	8.4	7.8	6.2	17.4	14.2	1.4	2.0	2.0	2.4	6.4	10.8

Table D.3: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 50, Population Size 30, Number of Generations 20 - Trial 3.

Experiment: (50, 30, 20) - Trial 3												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.83	1.83	2.0	1.17	3.17	2.33	4.83	12.0	8.17	6.33	11.83	6.67
2^{nd}	6.33	18.17	8.5	8.67	35.83	3.83	1.83	5.67	31.67	4.5	7.33	4.33
3^{rd}	5.5	4.83	32.0	5.0	25.67	5.33	2.67	13.5	19.5	6.67	19.17	11.33
4^{th}	7.83	8.17	7.0	3.5	13.67	5.17	4.5	4.67	5.83	2.5	5.0	4.0
5^{th}	2.2	2.4	2.8	5.6	8.6	18.4	9.0	13.8	28.0	15.2	4.6	18.4
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	5.83	16.0	10.5	10.33	5.33	10.17	0.17	4.17	1.33	1.5	1.67	3.5
2^{nd}	0.67	6.5	18.33	13.33	5.17	25.33	1.0	8.5	7.67	1.83	3.33	6.5
3^{rd}	2.0	20.67	5.33	14.83	8.17	8.0	0.5	13.67	3.17	0.83	4.33	7.67
4^{th}	2.67	7.5	13.5	3.83	16.33	5.5	0.83	16.5	2.33	1.17	8.67	2.17
5^{th}	0.4	5.8	6.2	6.2	20.0	13.6	0.4	7.2	1.8	3.4	4.2	9.0

Table D.4: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 50, Population Size 30, Number of Generations 20 - Trial 4.

Experiment: (50, 30, 20) - Trial 3												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	1.0	1.83	2.0	1.17	3.33	1.83	5.5	11.67	7.33	7.83	14.0	5.17
2^{nd}	7.83	19.83	18.0	7.0	11.33	5.33	2.17	7.5	30.67	3.17	3.17	5.33
3^{rd}	6.0	7.33	28.67	5.17	22.5	7.33	4.5	13.67	40.17	9.17	10.67	9.33
4^{th}	4.67	7.33	8.33	14.67	14.5	5.5	3.33	3.67	4.33	3.67	4.83	2.67
5^{th}	2.6	2.8	5.0	6.2	29.8	6.6	8.4	12.0	26.2	11.4	24.4	15.6
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	5.5	16.67	14.5	9.67	5.67	11.17	0.17	9.17	1.0	1.0	1.67	12.83
2^{nd}	2.33	6.0	15.83	9.17	8.17	19.5	1.33	8.17	9.17	2.33	3.17	10.67
3^{rd}	1.83	23.17	6.0	15.5	8.17	7.5	1.5	11.0	3.33	1.83	3.17	8.33
4^{th}	0.5	7.5	8.83	2.33	14.33	8.33	0.5	14.5	2.5	1.0	3.83	1.17
5^{th}	0.4	16.4	12.4	6.6	18.8	12.4	0.6	6.6	0.6	1.0	6.2	9.2

Table D.5: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 50, Population Size 30, Number of Generations 20 - Trial 5.

Experiment: (50, 30, 20) - Trial 3												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	1.0	1.83	2.0	1.17	3.0	2.33	4.67	12.0	3.5	11.33	12.67	5.5
2^{nd}	2.83	19.5	16.5	10.5	10.17	3.5	2.67	7.0	29.67	3.0	3.83	5.33
3^{rd}	8.5	7.17	26.83	7.33	28.5	8.33	5.33	14.17	22.67	9.83	11.83	7.83
4^{th}	5.0	8.83	8.5	5.0	15.5	6.5	4.17	2.5	5.83	2.17	17.17	3.5
5^{th}	3.4	1.8	3.8	6.2	19.6	7.0	10.4	14.2	26.8	8.2	5.8	19.2
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	5.5	16.0	12.5	9.83	5.17	10.5	0.17	11.67	1.83	1.0	1.5	4.83
2^{nd}	1.83	19.5	13.83	9.33	6.5	20.33	0.0	2.83	9.5	1.83	3.5	9.5
3^{rd}	3.33	20.67	6.0	13.5	6.17	6.67	1.33	14.17	1.83	0.83	3.83	7.17
4^{th}	3.5	5.17	16.33	4.0	9.17	6.83	0.5	13.83	2.5	1.17	9.5	10.33
5^{th}	0.4	8.8	8.4	4.0	10.2	13.6	0.4	8.0	2.2	1.0	8.0	11.8

Table D.6: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 30, Number of Generations 20 - Trial 1.

Experiment: (100, 30, 20) - Trial 1												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	1.0	1.83	1.67	1.17	3.0	2.33	5.0	8.67	6.5	6.67	15.0	5.0
2^{nd}	2.67	17.17	10.67	5.5	11.33	4.17	2.67	8.17	21.5	2.33	6.0	6.67
3^{rd}	9.67	3.67	25.67	5.83	26.33	3.67	3.0	10.17	32.17	7.67	33.5	10.5
4^{th}	7.33	7.67	7.17	5.0	12.5	5.5	2.83	2.83	5.17	1.33	16.0	3.0
5^{th}	2.8	1.2	2.6	5.8	9.8	5.6	3.4	13.6	26.6	6.2	3.2	20.8
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	3.33	16.5	11.83	7.83	3.83	10.17	0.17	4.0	2.5	0.83	2.17	10.5
2^{nd}	3.67	6.83	3.33	11.33	6.17	6.0	1.0	2.17	7.67	0.33	3.17	8.67
3^{rd}	2.33	23.0	4.33	15.0	6.17	10.33	1.5	11.33	3.0	2.33	5.17	7.67
4^{th}	3.67	6.5	15.33	3.17	16.33	6.33	0.33	15.33	2.5	1.0	6.33	2.33
5^{th}	0.4	8.4	9.0	5.8	24.4	13.6	0.4	2.2	0.6	1.2	1.8	8.0

Table D.7: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 30, Number of Generations 20 - Trial 2.

Experiment: (100, 30, 20) - Trial 2												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.83	1.83	2.0	1.0	3.0	2.33	6.17	7.17	7.5	7.17	13.17	6.83
2^{nd}	2.67	17.17	4.0	6.67	11.0	6.0	2.83	7.17	23.0	4.17	7.17	5.33
3^{rd}	5.17	7.0	30.67	6.67	28.5	4.0	2.33	11.17	19.0	11.0	26.5	11.33
4^{th}	8.0	6.67	8.67	2.83	15.33	4.17	2.83	2.83	5.17	3.5	23.5	3.0
5^{th}	3.0	1.8	4.2	5.2	5.8	20.6	3.4	13.2	23.4	16.2	6.4	18.4
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	3.5	15.33	5.5	10.33	5.5	10.0	0.17	3.5	2.5	1.5	1.83	5.5
2^{nd}	4.0	6.83	15.5	9.33	7.33	19.5	0.17	2.83	7.5	2.17	3.5	2.83
3^{rd}	2.33	23.83	3.17	12.5	6.17	17.83	1.0	9.5	3.17	2.33	4.0	8.33
4^{th}	4.17	5.67	17.0	3.83	14.67	6.67	0.67	16.5	2.5	1.0	8.0	2.33
5^{th}	0.6	16.0	11.0	4.2	23.0	13.2	0.4	7.0	0.6	0.8	9.8	6.2

Table D.8: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 30, Number of Generations 20 - Trial 3.

Experiment: (100, 30, 20) - Trial 3												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.83	1.83	2.0	1.0	3.0	2.33	5.0	9.17	4.5	7.33	12.83	5.0
2^{nd}	1.17	14.83	5.17	6.0	5.67	4.5	1.83	9.17	30.17	3.67	7.33	5.5
3^{rd}	8.83	7.67	28.5	11.5	26.17	5.67	2.83	10.67	31.83	9.5	26.0	10.33
4^{th}	4.83	6.83	5.67	2.83	14.67	6.0	3.33	4.83	5.67	2.83	5.17	3.5
5^{th}	2.4	3.2	2.4	6.8	5.8	21.2	8.6	15.4	26.8	12.6	2.8	19.8
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	5.67	14.5	8.83	8.0	4.0	10.33	0.17	6.67	3.17	1.17	1.83	5.67
2^{nd}	2.67	16.17	12.0	10.33	5.5	17.67	0.5	2.0	2.67	2.17	4.0	10.33
3^{rd}	2.33	19.83	2.83	16.33	7.33	5.17	0.5	12.17	2.67	2.67	4.33	7.0
4^{th}	3.83	4.5	16.83	3.33	19.83	6.5	0.67	14.67	2.5	0.83	9.83	2.67
5^{th}	0.4	9.4	12.4	5.4	20.6	7.6	0.2	9.4	0.6	1.0	8.4	11.4

Table D.9: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 30, Number of Generations 20 - Trial 4.

Experiment: (100, 30, 20) - Trial 4												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.83	1.83	2.0	1.17	3.17	2.33	5.67	8.17	6.83	7.0	22.5	3.67
2^{nd}	2.5	11.67	6.33	6.17	8.83	4.0	2.67	7.83	25.33	4.17	8.5	5.33
3^{rd}	8.67	5.83	35.0	5.83	27.67	6.17	8.17	12.0	20.17	7.83	24.5	11.0
4^{th}	8.33	6.83	3.83	4.5	12.83	5.5	3.83	3.67	4.5	2.67	6.0	3.83
5^{th}	2.6	2.6	2.4	6.6	6.2	16.4	6.2	13.0	29.8	15.4	21.6	17.4
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	4.67	16.5	12.83	4.33	6.5	10.5	0.33	9.67	1.5	1.5	2.0	15.5
2^{nd}	2.0	14.67	4.0	6.67	6.67	30.67	0.83	3.0	8.0	0.33	3.0	9.67
3^{rd}	4.17	19.5	4.67	14.33	7.83	8.17	0.5	11.5	3.17	0.83	5.0	8.33
4^{th}	2.17	8.83	12.33	3.0	9.33	7.33	0.67	7.83	2.17	0.83	3.17	3.33
5^{th}	0.6	8.2	8.4	3.8	24.8	13.8	0.4	7.0	2.0	0.8	1.6	11.2

Table D.10: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 30, Number of Generations 20 - Trial 5.

Experiment: (100, 30, 20) - Trial 5												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.83	1.83	2.0	1.0	3.0	2.33	5.17	9.5	6.5	6.5	14.83	5.17
2^{nd}	2.67	9.83	24.67	4.33	9.67	4.17	2.33	4.67	32.5	5.17	8.83	6.83
3^{rd}	2.17	6.5	28.67	7.33	19.83	8.5	5.17	10.5	17.33	6.5	38.67	9.83
4^{th}	8.5	8.17	7.5	4.0	15.5	6.33	3.17	4.0	6.0	2.83	3.5	4.5
5^{th}	3.2	2.0	5.0	6.6	5.4	11.0	3.2	9.8	24.4	13.2	1.2	19.0
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	5.5	18.83	12.0	7.0	4.0	10.17	0.17	12.5	2.67	1.33	1.5	5.83
2^{nd}	1.83	5.5	10.5	6.83	7.0	20.5	1.0	7.5	8.83	0.33	2.0	5.83
3^{rd}	2.0	20.0	4.0	10.67	8.0	10.17	1.17	11.17	2.17	0.83	3.83	5.83
4^{th}	4.83	8.5	18.67	2.67	16.0	7.33	1.0	15.0	2.5	1.0	7.83	1.83
5^{th}	0.4	8.8	12.4	5.4	11.2	13.6	0.4	2.6	1.4	4.4	6.8	7.0

Table D.11: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 100, Number of Generations 50 - Trial 1.

Experiment: (100, 100, 50) - Trial 1												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.67	1.33	2.0	1.0	3.0	2.33	5.5	8.33	5.0	5.67	8.67	4.33
2^{nd}	0.5	20.5	2.5	3.17	7.33	3.0	1.83	3.33	35.83	0.83	2.0	1.67
3^{rd}	5.17	2.5	24.83	5.83	40.17	4.33	2.83	7.83	15.33	6.33	10.83	6.33
4^{th}	4.0	4.17	4.0	2.67	8.33	4.33	0.33	1.5	4.33	0.83	1.5	1.0
5^{th}	1.4	1.8	2.2	5.6	1.2	2.8	2.4	13.6	25.6	15.6	3.4	19.0
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	2.33	17.33	1.83	2.17	1.67	10.33	0.0	10.0	0.67	1.0	0.83	3.67
2^{nd}	0.67	4.0	4.0	10.67	2.5	3.17	0.17	0.83	0.67	0.17	1.67	4.5
3^{rd}	1.33	22.0	2.0	3.67	4.17	3.17	0.5	13.0	1.5	0.67	2.0	5.17
4^{th}	0.0	4.17	17.17	1.17	15.5	2.83	0.67	6.67	2.17	0.67	1.5	0.83
5^{th}	0.2	3.6	6.6	1.2	9.6	7.0	0.0	0.4	0.2	0.4	1.0	2.4

Table D.12: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 100, Number of Generations 50 - Trial 2.

Experiment: (100, 100, 50) - Trial 2												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.67	1.17	2.0	1.0	3.0	2.33	5.67	7.83	4.5	5.67	9.5	3.67
2^{nd}	1.0	3.33	4.83	3.5	9.67	2.83	1.33	4.0	7.83	2.33	4.67	3.0
3^{rd}	4.5	2.33	29.0	4.17	26.5	4.5	1.67	9.33	12.5	4.0	8.67	6.0
4^{th}	3.83	5.67	5.67	2.5	6.5	3.17	2.0	2.33	4.33	1.33	1.83	1.67
5^{th}	0.6	0.8	0.8	5.8	1.2	4.8	1.4	15.4	26.8	2.4	2.2	18.4
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	2.0	16.83	4.83	2.0	1.67	10.0	0.0	2.83	1.17	0.67	1.0	4.33
2^{nd}	0.5	4.17	10.0	9.33	1.83	23.33	0.0	2.33	0.67	0.33	1.5	8.33
3^{rd}	1.83	20.33	2.0	6.0	4.5	3.0	0.5	1.67	0.83	0.67	2.33	7.33
4^{th}	3.83	2.67	16.0	2.33	14.67	2.83	0.83	4.5	2.17	0.5	1.33	1.33
5^{th}	0.2	7.0	6.4	2.2	24.6	13.6	0.2	1.0	0.4	0.6	2.0	10.6

Table D.13: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 100, Number of Generations 50 - Trial 3.

Experiment: (100, 100, 50) - Trial 3												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.67	1.17	2.0	1.0	3.0	2.33	5.33	7.17	4.33	5.0	10.5	3.17
2^{nd}	1.5	16.33	3.33	3.5	7.17	1.0	0.33	1.17	4.83	1.5	3.33	2.0
3^{rd}	1.33	2.5	27.67	2.33	23.0	2.5	2.83	4.0	18.83	5.83	4.17	7.33
4^{th}	3.17	4.17	3.67	2.0	10.0	1.5	1.83	1.5	4.17	1.0	3.0	1.83
5^{th}	1.2	1.0	2.0	4.0	2.4	3.4	2.8	3.0	15.0	16.2	1.2	20.0
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	5.67	16.17	2.83	2.5	2.0	8.5	0.0	3.33	0.67	0.67	1.17	1.33
2^{nd}	0.67	5.17	3.17	8.67	2.0	21.33	0.17	2.17	3.0	2.67	1.83	10.17
3^{rd}	1.67	21.17	2.5	4.67	4.83	1.67	1.33	12.83	1.0	0.33	2.83	6.5
4^{th}	0.5	4.5	8.67	2.83	16.67	3.0	0.67	5.17	2.33	0.5	1.17	1.0
5^{th}	0.2	7.2	5.8	2.0	4.6	5.0	0.2	0.4	0.2	0.4	2.0	2.8

Table D.14: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 100, Number of Generations 50 - Trial 4.

Experiment: (100, 100, 50) - Trial 4												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.67	1.83	2.0	1.0	3.0	2.33	5.0	6.67	3.5	6.17	11.5	4.17
2^{nd}	1.83	22.0	5.33	3.0	8.5	2.5	0.33	3.67	8.0	2.33	5.17	2.33
3^{rd}	4.67	2.5	29.0	4.83	17.83	2.67	2.83	8.83	7.5	5.67	39.83	4.33
4^{th}	3.33	4.83	5.0	2.33	12.0	4.5	1.83	1.5	4.17	1.17	1.33	1.0
5^{th}	1.0	0.8	2.0	5.2	2.8	4.2	3.0	2.2	25.8	4.8	3.8	17.0
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	2.17	15.67	11.83	4.0	3.33	13.83	0.0	2.0	1.17	0.5	1.33	2.83
2^{nd}	0.33	3.0	2.83	10.83	3.67	27.0	0.17	1.33	8.67	3.5	1.0	2.83
3^{rd}	1.83	19.5	0.33	3.83	3.17	2.67	0.5	11.33	1.83	0.17	2.0	4.17
4^{th}	2.5	2.17	8.17	2.33	2.17	5.5	0.67	6.33	2.17	0.33	2.83	0.83
5^{th}	0.4	6.4	7.2	3.2	8.4	6.2	0.2	1.6	0.4	0.6	1.0	2.2

Table D.15: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 100, Number of Generations 50 - Trial 5.

Experiment: (100, 100, 50) - Trial 5												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.67	1.17	2.0	1.0	3.0	2.33	5.83	8.5	3.5	5.17	12.0	6.0
2^{nd}	1.67	9.5	1.17	4.5	4.17	2.5	1.0	5.17	31.0	1.0	4.33	1.33
3^{rd}	10.0	3.0	27.33	3.5	3.67	2.5	2.67	7.33	13.33	5.17	6.67	6.83
4^{th}	4.0	5.83	5.5	1.33	9.17	3.17	1.33	1.67	4.0	0.67	2.0	1.33
5^{th}	2.4	1.4	1.6	4.0	2.2	7.2	1.2	15.4	25.0	4.2	1.6	16.6
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	1.67	15.5	9.0	3.33	3.33	10.17	0.0	2.5	0.33	0.5	1.0	3.83
2^{nd}	0.67	5.33	2.5	15.33	3.0	4.67	0.0	1.67	9.17	0.33	2.0	8.17
3^{rd}	1.5	19.33	1.83	5.5	5.0	2.33	0.5	12.17	2.0	0.67	2.5	7.0
4^{th}	0.5	5.5	6.0	2.5	16.0	3.33	1.17	6.5	2.33	0.5	1.5	1.67
5^{th}	0.4	4.4	5.0	3.0	10.0	2.0	0.0	0.2	0.2	0.8	1.0	0.8

Table D.16: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 250, Number of Generations 100 - Trial 1

Experiment: (100, 250, 100) - Trial 1												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.33	1.17	2.0	1.0	3.0	1.67	5.83	7.17	4.17	5.0	11.67	3.17
2^{nd}	1.5	2.83	2.67	1.83	2.5	1.67	0.17	2.83	5.67	0.5	3.0	2.33
3^{rd}	3.33	2.5	26.5	1.0	24.0	2.5	2.17	5.83	10.67	4.0	3.67	3.83
4^{th}	4.0	6.0	3.5	1.33	4.83	3.83	2.0	0.17	3.33	0.5	1.67	1.17
5^{th}	2.0	1.2	1.2	4.2	3.0	2.2	1.6	13.8	6.4	7.4	1.2	19.2
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	2.0	2.0	4.33	2.33	1.17	10.0	0.0	2.67	0.5	0.0	0.5	3.17
2^{nd}	0.5	1.83	1.33	0.33	1.67	4.0	0.17	1.0	1.17	0.0	0.83	2.17
3^{rd}	1.67	17.83	1.83	1.0	3.67	2.83	0.33	13.17	1.33	0.33	0.67	5.5
4^{th}	0.0	5.0	4.0	1.33	18.33	3.67	1.0	5.0	2.0	0.83	0.5	0.33
5^{th}	0.0	4.8	5.2	2.8	8.0	4.2	0.0	0.2	0.4	0.4	0.8	1.8

Table D.17: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 250, Number of Generations 100 - Trial 2

Experiment: (100, 250, 100) - Trial 2												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.67	1.17	2.0	1.0	3.0	1.67	5.67	6.5	2.83	5.83	6.83	3.17
2^{nd}	0.5	2.33	2.67	1.0	6.5	1.17	0.33	1.33	3.5	0.67	2.5	1.5
3^{rd}	1.67	2.5	25.0	1.33	2.33	1.33	2.17	9.0	9.83	4.67	1.83	5.17
4^{th}	3.83	3.33	3.83	0.83	12.33	2.0	1.83	1.67	3.33	1.17	0.33	1.67
5^{th}	1.0	0.8	1.0	4.0	1.4	3.2	2.6	2.6	16.2	2.8	0.6	18.4
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	2.5	5.33	1.83	2.17	1.67	9.83	0.0	1.5	0.33	0.33	0.5	2.5
2^{nd}	0.33	1.67	1.0	1.17	2.33	1.83	0.0	0.5	8.33	0.0	1.33	2.83
3^{rd}	1.5	17.0	1.0	2.83	3.67	1.0	0.33	3.0	1.33	0.17	3.0	5.83
4^{th}	0.17	3.33	6.17	1.0	16.0	2.33	0.83	5.33	2.17	0.67	0.5	0.5
5^{th}	0.2	4.4	5.2	1.4	5.6	5.8	0.2	0.4	0.2	0.4	0.8	1.0

Table D.18: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 250, Number of Generations 100 - Trial 3

Experiment: (100, 250, 100) - Trial 3												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.67	1.17	2.0	1.0	3.0	1.67	5.67	5.67	3.33	5.0	8.83	3.17
2^{nd}	0.17	1.33	2.5	3.17	7.67	1.33	0.5	2.5	6.33	1.33	1.83	1.17
3^{rd}	3.0	0.83	28.17	4.0	5.83	2.17	3.33	6.5	12.0	4.83	4.67	7.0
4^{th}	4.0	2.67	5.67	1.5	4.67	0.67	1.67	2.0	3.83	0.67	1.17	2.33
5^{th}	0.6	1.4	2.0	2.8	1.6	2.2	3.0	13.8	23.6	1.8	1.6	18.2
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	2.83	15.67	1.33	1.67	2.33	10.17	0.0	1.83	0.0	0.33	0.33	3.17
2^{nd}	0.17	3.0	1.33	8.33	1.83	2.33	0.0	0.33	1.33	0.17	0.5	1.83
3^{rd}	1.17	21.17	0.33	3.17	4.17	2.17	0.33	10.5	1.33	0.17	1.0	5.67
4^{th}	0.17	3.83	7.67	1.0	16.67	4.67	0.83	4.83	1.83	0.83	1.33	0.5
5^{th}	0.0	6.4	4.6	2.6	4.6	5.8	0.0	0.4	0.2	0.4	0.4	1.8

Table D.19: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 250, Number of Generations 100 - Trial 4

Experiment: (100, 250, 100) - Trial 4												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.67	1.17	2.0	1.0	3.0	1.67	5.83	6.33	3.17	4.5	9.83	3.33
2^{nd}	0.33	1.33	2.67	2.33	4.83	2.5	0.17	1.83	4.5	1.5	1.67	1.0
3^{rd}	3.17	1.83	26.5	1.5	7.67	0.33	2.17	5.67	7.0	2.33	1.67	7.17
4^{th}	3.83	1.83	5.5	1.17	5.0	3.0	0.67	2.0	3.5	0.67	0.83	1.5
5^{th}	1.4	0.4	1.8	3.8	2.2	2.8	2.0	3.2	27.0	2.2	1.4	21.2
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	2.5	15.0	1.5	2.83	1.67	11.0	0.0	2.33	0.67	0.17	0.5	2.33
2^{nd}	0.17	2.17	0.83	0.83	1.67	1.33	0.0	0.33	0.0	0.17	1.0	2.33
3^{rd}	1.0	22.5	1.83	4.33	3.83	4.83	0.33	1.17	1.17	0.33	1.0	4.17
4^{th}	0.17	1.17	5.83	1.0	16.83	2.17	0.67	4.67	2.0	0.67	2.33	0.83
5^{th}	0.0	3.4	5.6	2.0	8.2	5.6	0.0	0.0	0.2	0.4	0.6	1.8

Table D.20: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 100, Population Size 250, Number of Generations 100 - Trial 5

Experiment: (100, 250, 100) - Trial 5												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.5	1.17	2.0	1.0	3.0	2.33	4.83	5.0	1.5	4.83	6.83	4.33
2^{nd}	0.67	3.5	3.5	1.33	4.67	2.33	0.17	2.17	2.83	1.0	3.33	1.33
3^{rd}	2.5	2.17	33.17	3.33	20.33	3.67	2.0	8.5	7.83	4.33	3.5	5.0
4^{th}	3.5	3.17	4.0	1.67	6.17	2.17	0.67	0.5	3.17	1.33	1.5	2.33
5^{th}	1.2	0.2	1.0	3.0	6.0	2.8	1.4	11.6	18.4	5.0	2.4	18.2
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	1.67	14.83	1.0	2.83	1.33	10.0	0.0	2.83	0.33	0.17	0.83	2.33
2^{nd}	0.17	1.67	1.5	0.33	0.83	3.33	0.0	0.5	0.33	0.0	0.33	1.5
3^{rd}	1.67	22.83	1.17	2.33	2.67	3.0	0.33	2.5	0.83	0.33	1.5	5.83
4^{th}	0.5	2.67	7.67	0.67	22.0	3.33	0.5	4.33	2.0	0.5	0.83	1.0
5^{th}	0.2	4.4	8.0	0.8	6.4	4.6	0.2	1.2	0.2	0.0	1.2	2.0

Table D.21: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 150, Population Size 300, Number of Generations 200 - Trial 1.

Experiment: (150, 300, 200) - Trial 1												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.67	1.17	2.0	1.0	3.0	1.67	5.33	5.5	2.33	5.17	10.33	3.0
2^{nd}	0.83	2.33	3.0	0.83	4.0	0.83	0.0	1.33	3.83	0.83	2.17	1.33
3^{rd}	1.0	2.5	8.17	5.33	1.33	3.0	0.83	5.0	9.5	2.83	3.17	3.67
4^{th}	3.83	1.67	4.5	0.17	5.83	0.83	0.83	0.5	3.17	0.5	2.0	1.33
5^{th}	1.0	1.6	0.4	2.6	2.4	4.8	1.2	1.6	14.6	5.8	0.6	16.8
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	1.67	2.67	0.83	2.0	1.83	9.67	0.0	2.0	0.33	0.0	0.17	2.67
2^{nd}	0.0	2.17	1.0	0.5	0.67	1.0	0.17	0.67	0.33	0.0	0.33	1.33
3^{rd}	1.5	20.5	0.0	2.0	3.33	2.83	0.33	3.0	0.67	0.17	1.0	4.5
4^{th}	0.17	1.5	7.17	0.83	7.17	2.33	0.17	4.83	2.0	0.5	0.67	0.5
5^{th}	0.0	4.2	2.8	1.4	8.6	4.0	0.2	0.2	0.0	0.4	1.0	0.0

Table D.22: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 150, Population Size 300, Number of Generations 200 - Trial 2.

Experiment: (150, 300, 200) - Trial 2												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.67	1.17	2.0	1.0	3.0	1.67	2.17	4.67	1.67	5.0	8.0	2.5
2^{nd}	0.33	3.0	2.33	0.33	4.5	1.17	0.67	2.0	3.83	1.67	1.67	0.5
3^{rd}	2.83	2.0	27.5	2.5	4.5	2.17	0.67	6.33	10.0	1.67	2.67	5.33
4^{th}	3.0	5.0	4.67	0.67	7.33	0.83	1.0	1.5	3.33	0.17	0.67	0.83
5^{th}	1.6	0.6	0.6	3.2	1.8	3.2	1.4	1.0	26.2	3.2	1.0	17.6
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	1.83	4.33	2.17	1.5	1.33	9.83	0.0	1.67	0.0	0.0	0.17	1.5
2^{nd}	0.33	3.5	0.83	0.33	0.67	3.17	0.0	0.5	0.67	0.0	1.0	0.67
3^{rd}	1.17	21.5	0.83	1.83	3.33	0.5	0.33	2.67	0.83	0.0	1.5	3.5
4^{th}	0.0	3.17	4.5	1.17	0.67	2.67	0.83	4.17	1.83	0.5	0.17	0.67
5^{th}	0.0	3.2	3.2	2.4	6.0	5.2	0.0	0.4	0.2	0.2	0.4	0.8

Table D.23: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 150, Population Size 300, Number of Generations 200 - Trial 3.

Experiment: (150, 300, 200) - Trial 3												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.67	1.17	2.0	1.0	3.0	1.67	5.5	7.17	4.33	4.5	8.67	4.5
2^{nd}	0.5	18.17	1.0	1.5	3.83	0.67	0.33	2.0	3.5	0.33	2.17	1.5
3^{rd}	0.67	2.17	27.0	3.83	2.83	2.17	2.17	4.33	4.83	1.17	3.33	3.0
4^{th}	3.5	3.0	2.5	0.33	5.5	1.67	0.67	1.0	3.5	0.83	1.5	0.5
5^{th}	1.0	1.8	1.0	2.4	0.2	2.2	1.8	0.8	26.6	1.8	0.6	18.0
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	2.17	1.83	1.5	3.17	1.83	10.33	0.0	2.0	0.33	0.17	0.33	3.17
2^{nd}	0.17	1.0	0.33	0.17	0.83	1.0	0.0	0.17	0.0	0.17	0.5	0.5
3^{rd}	1.33	5.5	1.0	1.33	4.17	4.83	0.33	2.0	1.5	0.0	0.67	5.17
4^{th}	0.17	3.5	5.67	1.0	10.0	2.17	1.0	1.67	2.17	0.67	1.17	0.33
5^{th}	0.0	4.8	6.0	1.2	5.6	2.2	0.0	0.2	0.2	0.0	0.0	1.2

Table D.24: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 150, Population Size 300, Number of Generations 200 - Trial 4.

Experiment: (150, 300, 200) - Trial 4												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.67	1.17	2.0	1.0	3.0	1.67	6.17	7.0	2.33	4.67	6.83	2.5
2^{nd}	0.67	3.0	3.17	0.83	2.5	0.83	0.0	0.67	3.5	1.0	1.5	2.5
3^{rd}	0.17	2.5	27.17	2.33	2.83	4.67	2.33	5.67	5.67	3.67	1.33	4.17
4^{th}	3.17	1.5	2.83	0.17	5.67	1.5	1.17	0.5	3.33	0.33	0.83	1.67
5^{th}	1.2	0.8	0.6	3.0	1.8	3.8	1.4	15.2	27.2	3.8	1.0	16.4
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	2.0	3.83	2.67	2.0	2.5	10.33	0.0	2.0	0.67	0.17	0.33	2.17
2^{nd}	0.17	3.0	0.83	0.33	1.17	2.0	0.17	0.5	0.0	0.0	0.33	1.33
3^{rd}	1.5	7.0	0.17	3.83	4.0	0.83	0.33	1.5	1.67	0.0	1.17	6.17
4^{th}	0.17	1.67	6.33	1.33	14.33	1.33	0.5	4.5	2.17	0.33	1.17	0.0
5^{th}	0.0	3.8	1.4	2.0	6.4	4.0	0.0	0.4	0.2	0.4	0.2	1.2

Table D.25: Mean Average Error (MAE) of the genetic prediction for all car models and time chunks done under the following conditions: Number of rules 150, Population Size 300, Number of Generations 200 - Trial 5.

Experiment: (150, 300, 200) - Trial 5												
	SEAT Arona						SEAT Ibiza					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	0.67	1.17	2.0	1.0	3.0	1.67	5.5	4.67	4.33	5.0	7.17	2.33
2^{nd}	0.17	0.5	0.5	2.67	2.67	0.67	0.33	1.67	3.17	0.83	1.5	1.17
3^{rd}	1.83	0.67	26.0	1.67	4.5	2.0	1.33	3.0	6.0	2.83	3.33	3.5
4^{th}	3.67	3.67	3.5	1.0	4.0	2.0	2.83	1.0	4.17	1.17	1.33	0.67
5^{th}	1.2	0.6	1.2	3.0	3.2	3.6	1.8	1.8	23.2	5.2	0.6	20.4
	SEAT Leon 5D						SEAT Leon ST					
Time Chunk	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER	CMC	MADRID	LA RODA	CHESTE	LLAGOSTA	SANTANDER
1^{st}	1.67	4.0	0.83	2.0	2.67	10.0	0.0	1.67	0.33	0.17	0.5	2.5
2^{nd}	0.17	2.67	0.33	0.5	0.67	0.67	0.0	0.5	0.33	0.0	0.5	0.17
3^{rd}	1.5	22.0	0.83	1.83	2.33	1.33	0.33	2.17	1.17	0.17	1.17	7.67
4^{th}	0.0	2.17	6.0	2.17	0.83	1.5	0.33	4.83	2.17	0.33	1.17	0.83
5^{th}	0.0	4.2	4.6	1.0	3.4	5.6	0.0	0.4	0.2	0.2	0.0	2.2

Table D.26: R2 Score (%) of the best candidate derived from the genetic forecast for each time chunk, week, and car model.

		Experiment: (150,300,200) - Trial 2			
Time Chunk	Week	SEAT Arona	SEAT Ibiza	SEAT Leon 5D	SEAT Leon ST
1^{st}	2017-09-17	0.0	97.12	81.14	99.85
	2017-09-24	0.0	90.23	94.23	99.96
	2017-10-01	2.94	91.74	99.07	100.0
	2017-10-08	11.07	82.29	57.04	90.98
	2017-10-15	5.62	99.15	79.54	99.69
	2017-10-22	62.66	99.75	96.07	85.45
2^{nd}	2018-04-15	77.73	99.67	99.68	98.77
	2018-04-22	98.59	99.56	99.57	99.67
	2018-04-29	97.2	95.78	95.98	99.64
	2018-05-06	99.14	99.05	98.91	95.63
	2018-05-13	96.49	99.24	76.69	99.52
	2018-05-20	99.07	100.0	99.83	99.89
3^{rd}	2018-11-11	54.76	92.72	92.52	99.67
	2018-11-18	81.89	97.2	68.36	98.45
	2018-11-25	67.21	99.26	82.51	83.39
	2018-12-02	45.73	94.0	67.58	98.26
	2018-12-09	5.59	97.73	72.61	48.49
	2018-12-16	99.81	99.39	52.24	99.66
4^{th}	2019-06-09	60.04	99.87	90.7	99.49
	2019-06-16	90.11	99.37	99.67	99.85
	2019-06-23	99.72	99.94	99.76	99.93
	2019-06-30	71.3	82.93	95.92	93.6
	2019-07-07	90.81	99.68	84.64	99.2
	2019-07-14	100.0	99.83	99.9	99.95
5^{th}	2020-01-05	99.95	25.74	88.47	99.6
	2020-01-12	95.21	37.98	93.3	98.93
	2020-01-19	99.79	91.29	93.87	99.61
	2020-01-26	99.89	88.22	91.63	99.25
	2020-02-02	98.18	51.55	98.87	99.89

Bibliography

[1] Behzad Saberi. "The role of the automobile industry in the economy of developed countries". In: *International Robotics & Automation Journal* 4 (July 2018). DOI: 10.15406/iratj.2018.04.00119.

[2] David Brown et al. *The Future of the EU Automotive Sector*. Accessed: 4 April 2024. 2021. URL: https://www.europarl.europa.eu/RegData/etudes/STUD/2021/695457/IPOL_STU(2021)695457_EN.pdf.

[3] Kim Hill et al. *Contribution of the Automotive Industry to the Economies of All Fifty States and the United States*. Accessed: 4 April 2024. 2015. URL: https://www.cargroup.org/wp-content/uploads/2017/02/Contribution-of-the-Automotive-Industry-to-the-Economies-of-All-Fifty-States-and-the-United-States2015.pdf.

[4] Judit Montoriol-Garriga and Sergio Díaz. *Spain's automotive industry: strategic and undergoing a transformation*. Accessed: 4 April 2024. 2021. URL: https://www.caixabankresearch.com/en/sector-analysis/industry/spains-automotive-industry-strategic-and-undergoing-transformation.

[5] Augustin Friedel, Nils Schaupensteiner, and Marcus Willand. *The Software Race: Are Chinese Automakers Taking the Lead?* Accessed: 4 April 2024. 2023. URL: https://www.mhp.com/fileadmin/www.mhp.com/downloads/studien/MHPStudy_The-Software-Race_EN.pdf.

[6] Bill Peng et al. *Winning the race: China's auto market shifts gears*. Accessed: 4 April 2024. 2019. URL: https://www.mckinsey.com/~/media/mckinsey/industries/automotive%20and%20assembly/our%20insights/winning%20the%20race%20chinas%20auto%20market%20shifts%20gears/winning-the-race-chinas-auto-market-shifts-gears.ashx.

[7] John Letzing and Minji Sung. *Mapping the rise of China's autos and other global trade trends*. Accessed: 4 April 2024. 2024. URL: https://www.weforum.org/agenda/2024/02/mapping-the-state-of-global-trade-starting-with-the-rise-of-china-s-autos/.

[8] Dae-Ho Byun. "The AHP approach for selecting an automobile purchase model". In: *Information & Management* 38 (Mar. 2001), pp. 289–297. DOI: 10.1016/S0378-7206(00)00071-9.

[9] D.V.V. *SEAT presenta Fast Lane, un innovador método para acortar los plazos de entrega*. Accessed: 15 January 2024. 2017. URL: https://www.elespanol.com/motor/20171128/265473605_0.html.

[10] *SEAT Fast Lane*. Accessed: 15 January 2024. URL: https://www.seat.com/carworlds/fast-lane-express-car-delivery.

[11] Juan Manuel García Sánchez, Xavier Vilasís Cardona, and Alexandre Lerma Martín. "Influence of Car Configurator Webpage Data from Automotive Manufacturers on Car Sales by Means of Correlation and Forecasting". In: *Forecasting* 4.3 (2022), pp. 634–653. ISSN: 2571-9394. DOI: 10.3390/forecast4030034. URL: https://www.mdpi.com/2571-9394/4/3/34.

[12] Juan Manuel García-Sánchez, Xavier Cardona, and Alexandre Martín. "Binary Delivery Time Classification and Vehicle's Reallocation Based on Car Variants. SEAT: A Case Study". In: Frontiers in Artificial Intelligence and Applications. IOS Press, Oct. 2022, pp. 147–150. ISBN: 9781643683263. DOI: 10.3233/FAIA220329.

[13] Juan Manuel García-Sánchez et al. "Data Mining Car Configurator Clickstream Data to Identify Potential Consumers: A Genetic Algorithm Approach". In: *Artificial Intelligence and Soft Computing*. Ed. by Leszek Rutkowski et al. Cham: Springer Nature Switzerland, 2023, pp. 375–384. ISBN: 978-3-031-42505-9.

[14] Juan Manuel García-Sánchez, Xavier Cardona, and Alexandre Martín. "Analyzing Car Configurator Impact Through Genetic Algorithm from a Regional Perspective". In: Frontiers in Artificial Intelligence and Applications. IOS Press, Oct. 2023, pp. 106–109. DOI: 10.3233/FAIA230666.

[15] Joel R. Spiegel et al. "Method and System for Anticiparoty Package Shipping". US8615473B2. Worldwide applications. Dec. 2013. URL: https://patents.google.com/patent/US8615473B2/en.

[16] Amazon. *Los más vendidos de Amazon*. Accessed: 4 April 2024. 2024. URL: https://www.amazon.es/gp/bestsellers/?ref_=nav_cs_bestsellers.

[17] Arishekar N. *Top Amazon Product Categories: Trends, Tips, and Tricks*. Accessed: 4 April 2024. 2023. URL: https://www.sellerapp.com/blog/best-selling-products-on-amazon/#1.

[18] Yanshuo Sun, Sajeeb Kirtonia, and Zhi-Long Chen. "A survey of finished vehicle distribution and related problems from an optimization perspective". In: *Transportation Research Part E: Logistics and Transportation Review* 149 (2021), p. 102302. ISSN: 1366-5545. DOI: https://doi.org/10.1016/j.tre.2021.102302. URL: https://www.sciencedirect.com/science/article/pii/S1366554521000764.

[19] Matthias Holweg and Joe Miemczyk. "Delivering the '3-day car'—the strategic implications for automotive logistics operations". In: *Journal of Purchasing and Supply Management* 9.2 (2003), pp. 63–71. ISSN: 1478-4092. DOI: https://doi.org/10.1016/S1478-4092(03)00003-7. URL: https://www.sciencedirect.com/science/article/pii/S1478409203000037.

[20] Matthias Holweg and Joe Miemczyk. "Logistics in the "three-day car" age: Assessing the responsiveness of vehicle distribution logistics in the UK". In: *International Journal of Physical Distribution & Logistics Management* 32.10 (2002), pp. 829–850. ISSN: 0960-0035. DOI: 10.1108/09600030210455438. URL: https://doi.org/10.1108/09600030210455438.

[21] Matthias Holweg and Frits K. Pil. *The Second Century: Reconnecting Customer and Value Chain through Build-to-Order Moving Beyond Mass and Lean in the Auto Industry*. Vol. 1. MIT Press Books 0262582627. The MIT Press, Feb. 2005. ISBN: ARRAY(0x4a29bbf8). URL: https://ideas.repec.org/b/mtp/titles/0262582627.html.

[22] Thomas Volling et al. "Planning of capacities and orders in build-to-order automobile production: A review". In: *European Journal of Operational Research* 224 (2013), pp. 240–260. DOI: 10.1016/j.ejor.2012.07.034.

[23] Felipe Luiz, Luiz Scavarda, and Tiago Fernandes. "008-0447 Handling Product Variety and its Effects in Automotive Production". In: *Proceedings of the 19th Annual Production and Operations Management Society (POMS) Conference*. POMS. La Jolla, California, USA, 2008.

[24] Sheena S. Iyengar and Mark R. Lepper. "When choice is demotivating: Can one desire too much of a good thing?" In: *Journal of Personality and Social Psychology* 79.6 (2000), pp. 995–1006. ISSN: 1939-1315 (Electronic), 0022-3514 (Print). DOI: 10.1037/0022-3514.79.6.995.

[25] Barry Schwartz. "The Paradox of Choice". In: *Positive Psychology in Practice*. John Wiley & Sons, Ltd, 2015. Chap. 8, pp. 121–138. ISBN: 9781118996874. DOI: https://doi.org/10.1002/9781118996874.ch8. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118996874.ch8. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118996874.ch8.

[26] Barbara Kahn. "Variety: From the Consumer's Perspective". In: *Product Variety Management: Research Advances*. Ed. by Teck-Hua Ho and Christopher S. Tang. Boston, MA: Springer US, 1998, pp. 19–37. ISBN: 978-1-4615-5579-7. DOI: 10.1007/978-1-4615-5579-7_2. URL: https://doi.org/10.1007/978-1-4615-5579-7_2.

[27] S. Rajagopalan and Nan Xia. "Product variety, pricing and differentiation in a supply chain". In: *European Journal of Operational Research* 217.1 (2012), pp. 84–93. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2011.08.016. URL: https://www.sciencedirect.com/science/article/pii/S0377221711007570.

[28] Mark Heitmann, Andreas Herrmann, and Christian Kaiser. "The effect of product variety on purchase probability". In: *Review of Managerial Science* 1 (July 2007), pp. 111–131. DOI: 10.1007/s11846-007-0006-6.

[29] Lechner Annika, Katja Klingebiel, and Axel Wagenitz. "Evaluation of Product Variant-driven Complexity Costs and Performance Impacts in the Automotive Logistics with Variety-driven Activity-based Costing". In: *Lecture Notes in Engineering and Computer Science* 2 (Mar. 2011).

[30] Edward H. Bowman and Bruce Kogut. "Redesigning the firm". In: 1995. URL: https://api.semanticscholar.org/CorpusID:166880666.

[31] *Volkswagen AG*. Available at https://www.ai4europe.eu/ai-community/organizations/company/volkswagen-ag (2023/12/31).

[32] *Volkswagen Data:Lab*. Available at https://www.munich-startup.de/startups/volkswagen-datalab/ (2024/01/09).

[33] Frank Van Rijnsoever, Jacco Farla, and Martin Dijst. "Consumer car preferences and information search channels". In: *Transportation Research Part D Transport and Environment* 14 (Mar. 2009). DOI: 10.1016/j.trd.2009.03.006.

[34] Yihong Zhang and Takahiro Hara. "Predicting E-commerce Item Sales With Web Environment Temporal Background". In: *24th International Conference on Business Information Systems, BIS 2021, Hannover, Germany, June 15-17, 2021*. Ed. by Witold Abramowicz, Sören Auer, and Elzbieta Lewanska. 2021, pp. 233–243. DOI: 10.52825/bis.v1i.37. URL: https://doi.org/10.52825/bis.v1i.37.

[35] Yi-Ting Huang and Ping-Feng Pai. "Using the Least Squares Support Vector Regression to Forecast Movie Sales with Data from Twitter and Movie Databases". In: *Symmetry* 12 (2020), p. 625. DOI: 10.3390/sym12040625.

[36] Sharad Goel et al. "Predicting consumer behavior with Web search". In: *Proceedings of the National Academy of Sciences* 107.41 (2010), pp. 17486–17490. DOI: 10.1073/pnas.1005962107. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1005962107.

[37] Liwen Ling et al. "Can online search data improve the forecast accuracy of pork price in China?" In: *Journal of Forecasting* 39 (2020). DOI: 10.1002/for.2649.

[38] Tomas Havranek and Ayaz Zeynalov. "Forecasting tourist arrivals: Google Trends meets mixed-frequency data". In: *Tourism Economics* 27 (2019), p. 135481661987958. DOI: 10.1177/1354816619879584.

[39] "Web Search Queries Can Predict Stock Market Volumes". In: *PLOS ONE* 7.7 (2012), pp. 1–17. DOI: 10.1371/journal.pone.0040014. URL: https://doi.org/10.1371/journal.pone.0040014.

[40] Dai Wei et al. "A prediction study on e-commerce sales based on structure time series model and web search data". In: *The 26th Chinese Control and Decision Conference (2014 CCDC)*. 2014, pp. 5346–5351. DOI: 10.1109/CCDC.2014.6852219.

[41] Jessie Sujo, Elisabet Ribé, and Xavier Cardona. "CAIT: A Predictive Tool for Supporting the Book Market Operation Using Social Networks". In: *Applied Sciences* 12 (2021), p. 366. DOI: 10.3390/app12010366.

[42] Cheng Chen et al. "Optimal decision of multiobjective and multiperiod anticipatory shipping under uncertain demand: A data-driven framework". In: *Computers & Industrial Engineering* 159 (2021), p. 107445. DOI: 10.1016/j.cie.2021.107445.

[43] Eli Beracha and M. Babajide Wintoki. "Forecasting residential real estate price changes from online search activity". In: *Journal of Real Estate Research* 35.3 (2013), pp. 283–312. ISSN: 08965803. DOI: 10.1080/10835547.2013.12091364. URL: https://www.researchgate.net/publication/267826174_Forecasting_Residential_Real_Estate_Price_Changes_from_Online_Search_Activity.

[44] Daoyuan Sun et al. "Combining Online News Articles and Web Search to Predict the Fluctuation of Real Estate Market in Big Data Context". In: *Pacific Asia Journal of the Association for Information Systems* (2013), pp. 19–37. DOI: 10.17705/1pais.06403.

[45] Marian Dietzel, Nicole Braun, and Wolfgang Schäfers. "Sentiment-based commercial real estate forecasting with Google search volume data". In: *Journal of Property Investment and Finance* 32 (2014). DOI: 10.1108/JPIF-01-2014-0004.

[46] Yu Wei and Yang Cao. "Forecasting house prices using dynamic model averaging approach: Evidence from China". In: *Economic Modelling* 61 (2017), pp. 147–155. DOI: 10.1016/j.econmod.2016.12.002.

[47] Madalasa Venkataraman, Venkatesh Panchapagesan, and Ekta Jalan. "Does internet search intensity predict house prices in emerging markets? A case of India". In: *Property Management* 36 (2018). DOI: 10.1108/PM-01-2017-0003.

[48] Nina Rizun and Anna Baj-Rogowska. "Can Web Search Queries Predict Prices Change on the Real Estate Market?" In: *IEEE Access* 9 (2021), pp. 70095–70117. DOI: 10.1109/ACCESS.2021.3077860.

[49] Sunil Punjabi et al. "Sales Prediction using Online Sentiment with Regression Model". In: 2020, pp. 209–212. DOI: 10.1109/ICICCS48265.2020.9120936.

[50] Ping-Feng Pai and Chia-Hsin Liu. "Predicting Vehicle Sales by Sentiment Analysis of Twitter Data and Stock Market Values". In: *IEEE Access* PP (2018), p. 1. DOI: 10.1109/ACCESS.2018.2873730.

[51] Hal Varian and Hyunyoung Choi. "Predicting the Present with Google Trends". In: *Economic Record* 88 (2009). DOI: 10.2139/ssrn.1659302.

[52] Dongha Kim et al. "Can search engine data improve accuracy of demand forecasting for new products? Evidence from automotive market". In: *Industrial Management & Data Systems* 119 (2019). DOI: 10.1108/IMDS-08-2018-0347.

[53] Philipp Wachter, Tobias Widmer, and Achim Klein. "Predicting Automotive Sales using Pre-Purchase Online Search Data". In: 2019, pp. 569–577. DOI: 10.15439/2019F239.

[54] Dean Fantazzini and Zhamal Toktamysova. "Forecasting German car sales using Google data and multivariate models". In: *International Journal of Production Economics* 170 (2015). DOI: 10.1016/j.ijpe.2015.09.010.

[55] Georg Graevenitz et al. "Does Online Search Predict Sales? Evidence from Big Data for Car Markets in Germany and the UK". In: *SSRN Electronic Journal* (2016). DOI: 10.2139/ssrn.2832004.

[56] Fons Wijnhoven and Olivia Plant. "Sentiment Analysis and Google Trends Data for Predicting Car Sales". In: 2017.

[57] Chuan Zhang, Yu-Xin Tian, and Ling-Wei Fan. "Improving the Bass model's predictive power through online reviews, search traffic and macroeconomic data". In: *Annals of Operations Research* 295 (2020). DOI: 10.1007/s10479-020-03716-3.

[58] William Darley, Charles Blankson, and Denise Luethge. "Toward an Integrated Framework for Online Consumer Behavior and Decision Making Process: A Review". In: *Psychology and Marketing* 27 (Mar. 2010), pp. 94–116. DOI: 10.1002/mar.20322.

[59] George Lawton. *10 big data challenges and how to address them*. 2022. URL: https://www.techtarget.com/searchdatamanagement/tip/10-big-data-challenges-and-how-to-address-them (visited on 01/23/2023).

[60] Panu Turcot and David G Lowe. "Better matching with fewer features: The selection of useful features in large database recognition problems". In: *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. 2009, pp. 2109–2116. DOI: 10.1109/ICCVW.2009.5457541.

[61] Ting Li, Yuhua Lin, and Haiying Shen. "A locality-aware similar information searching scheme". In: *International Journal on Digital Libraries* 17.2 (2016), pp. 79–93. ISSN: 1432-1300. DOI: 10.1007/s00799-014-0128-9. URL: https://doi.org/10.1007/s00799-014-0128-9.

[62] Xiaofeng Chen et al. "Verifiable Computation over Large Database with Incremental Updates". In: *IEEE Transactions on Computers* 65.10 (2016), pp. 3184–3195. DOI: 10.1109/TC.2015.2512870.

[63] Mahyuddin K. M. Nasution and Marischa Elveny. "Data Modeling as Emerging Problems of Data Science". In: *Data Science with Semantic Technologies: Theory, Practice, and Application*. Ed. by Archana Patel, Narayan C. Debnath, and Bharat Bhusan. Wiley-Scrivener, 2022. Chap. 3, pp. 71–90. ISBN: 9781119864981. DOI: https://doi.org/10.1002/9781119865339.ch3.

[64] Mahyuddin Nasution and B R Syah. "Data Management as Emerging Problems of Data Science". In: *Data Science with Semantic Technologies: Theory, Practice, and Application*. Ed. by Archana Patel, Narayan C. Debnath, and Bharat Bhusan. Wiley-Scrivener, 2022. Chap. 4, pp. 91–104. ISBN: 9781119864981. DOI: 10.1002/9781119865339.ch4.

[65] Diego García-Gil et al. "Enabling Smart Data: Noise filtering in Big Data classification". In: *Information Sciences* 479 (2019), pp. 135–152. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2018.12.002. URL: https://www.sciencedirect.com/science/article/pii/S0020025518309460.

[66] Julián Luengo et al. "Data Reduction for Big Data". In: *Big Data Preprocessing: Enabling Smart Data*. Cham: Springer International Publishing, 2020, pp. 81–99. ISBN: 978-3-030-39105-8. DOI: 10.1007/978-3-030-39105-8_5. URL: https://doi.org/10.1007/978-3-030-39105-8_5.

[67] Christina O'Connor and Stephen Kelly. "Facilitating knowledge management through filtered big data: SME competitiveness in an agri-food sector". In: *Journal of Knowledge Management* 21 (2017), pp. 156–179. DOI: 10.1108/JKM-08-2016-0357.

[68] Gabriele Santoro et al. "Big data for business management in the retail industry". In: *Management Decision* 57.8 (Jan. 2019), pp. 1980–1992. ISSN: 0025-1747. DOI: 10.1108/MD-07-2018-0829. URL: https://doi.org/10.1108/MD-07-2018-0829.

[69] Kandarp P. Shroff and Hardik H. Maheta. "A comparative study of various feature selection techniques in high-dimensional data set to improve classification accuracy". In: *2015 International Conference on Computer Communication and Informatics (ICCCI)*. 2015, pp. 1–6. DOI: 10.1109/ICCCI.2015.7218098.

[70] Cheng-Lung Huang and Chieh-Jen Wang. "A GA-based feature selection and parameters optimizationfor support vector machines". In: *Expert Systems with Applications* 31.2 (2006), pp. 231–240. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2005.09.024. URL: https://www.sciencedirect.com/science/article/pii/S0957417405002083.

[71] Kenji Kira, Larry A Rendell, et al. "The feature selection problem: Traditional methods and a new algorithm". In: *Aaai*. Vol. 2. 1992a. 1992, pp. 129–134.

[72] Ian H. Witten et al. *Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques*. 4th. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2016. ISBN: 0128042915.

[73] Shital Shah and Andrew Kusiak. "Cancer gene search with data-mining and genetic algorithms". In: *Computers in Biology and Medicine* 37.2 (2007), pp. 251–261. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2006.01.007. URL: https://www.sciencedirect.com/science/article/pii/S0010482506000217.

[74] Rajdev Tiwari and Manu Singh. "Correlation-based Attribute Selection using Genetic Algorithm". In: *International Journal of Computer Applications* 4 (Aug. 2010). DOI: 10.5120/847-1182.

[75] Lei Yu and Huan Liu. "Feature selection for high-dimensional data: A fast correlation-based filter solution". In: *Proceedings of the 20th international conference on machine learning (ICML-03)*. 2003, pp. 856–863.

[76] Z. Chuanlei et al. "Apple leaf disease identification using genetic algorithm and correlation based feature selection method". In: 10 (Jan. 2017), pp. 74–83. DOI: 10.3965/j.ijabe.20171002.2166.

[77] Krzysztof Drachal and Michał Pawłowski. "A Review of the Applications of Genetic Algorithms to Forecasting Prices of Commodities". In: *Economies* 9.1 (2021). ISSN: 2227-7099. DOI: 10.3390/economies9010006. URL: https://www.mdpi.com/2227-7099/9/1/6.

[78] Mohammed Farsi et al. "Parallel genetic algorithms for optimizing the SARIMA model for better forecasting of the NCDC weather data". In: *Alexandria Engineering Journal* 60.1 (2021), pp. 1299–1316. ISSN: 1110-0168. DOI: https://doi.org/10.1016/j.aej.2020.10.052. URL: https://www.sciencedirect.com/science/article/pii/S1110016820305706.

[79] Zeynep Idil Erzurum Cicek and Zehra Kamisli Ozturk. "Optimizing the artificial neural network parameters using a biased random key genetic algorithm for time series forecasting". In: *Applied Soft Computing* 102 (2021), p. 107091. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2021.107091. URL: https://www.sciencedirect.com/science/article/pii/S1568494621000144.

[80] Shi-Yuan Pan, Qi Liao, and Yong-Tu Liang. "Multivariable sales prediction for filling stations via GA improved BiLSTM". In: *Petroleum Science* 19.5 (2022), pp. 2483–2496. ISSN: 1995-8226. DOI: https://doi.org/10.1016/j.petsci.2022.05.005. URL: https://www.sciencedirect.com/science/article/pii/S1995822622001042.

[81] Mamluatul Hani'ah, Ika Kusumaning Putri, and Ariadi Retno Tri Hayati Ririd. "Genetic Algorithms for Holt Winter Exponential Smoothing Parameter Optimization in Indonesian Car Sales Forecasting". In: *Proceedings of the 2022 Annual Technology, Applied Science and Engineering Conference (ATASEC 2022)*. Atlantis Press, 2022, pp. 159–171. ISBN: 978-94-6463-106-7. DOI: 10.2991/978-94-6463-106-7_15. URL: https://doi.org/10.2991/978-94-6463-106-7_15.

[82] Zhi Li, Hang Fan, and Shuyan Dong. *Electric Vehicle Sales Forecasting Model Considering Green Premium: A Chinese Market-based Perspective*. 2023. arXiv: 2302.13893 [eess.SY].

[83] Shile Chen and Changjun Zhou. "Stock Prediction Based on Genetic Algorithm Feature Selection and Long Short-Term Memory Neural Network". In: *IEEE Access* 9 (2021), pp. 9066–9072. DOI: 10.1109/ACCESS.2020.3047109.

[84] Hani El-Chaarani et al. "Forecasting a Stock Trend Using Genetic Algorithm and Random Forest". In: *Journal of Risk and Financial Management* 15 (Apr. 2022), pp. 1–18. DOI: 10.3390/jrfm15050188.

[85] Elias Kalapanidas and Nikolaos Avouris. "Feature selection for air quality forecasting: A genetic algorithm approach". In: *AI Commun.* 16 (Jan. 2003), pp. 235–251.

[86] Alen Costa Vieira et al. "Improving flood forecasting through feature selection by a genetic algorithm – experiments based on real data from an Amazon rainforest river". In: *Earth Science Informatics* 14.1 (Mar. 2021), pp. 37–50. ISSN: 1865-0481. DOI: 10.1007/s12145-020-00528-8. URL: https://doi.org/10.1007/s12145-020-00528-8.

[87] Ankit Kumar Srivastava et al. "A Day-Ahead Short-Term Load Forecasting Using M5P Machine Learning Algorithm along with Elitist Genetic Algorithm (EGA) and Random Forest-Based Hybrid Feature Selection". In: *Energies* 16.2 (2023). ISSN: 1996-1073. DOI: 10.3390/en16020867. URL: https://www.mdpi.com/1996-1073/16/2/867.

[88] Jiezhen Li. "A Feature Engineering Approach for Tree-based Machine Learning Sales Forecast, Optimized by a Genetic Algorithm Based Sales Feature Framework". In: *2022 5th International Conference on Artificial Intelligence and Big Data (ICAIBD)*. 2022, pp. 133–139. DOI: 10.1109/ICAIBD55127.2022.9820532.

[89] Waleed Ali and Adel. A Abdullah. "Hybrid Intelligent Phishing Website Prediction Using Deep Neural Networks with Genetic Algorithm-based Feature Selection and Weighting". In: *IET Information Security* (Nov. 2019). DOI: 10.1049/iet-ifs.2019.0006.

[90] Scott Kirkpatrick, Charles D. Gelatt, and Mario P. Vecchi. "Optimization by Simulated Annealing". In: *Science* 220 (1983), pp. 671–680. URL: https://api.semanticscholar.org/CorpusID:205939.

[91] P. J. M. van Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications*. Mathematics and its applications (D. Reidel Publishing Company). Dordrecht, Boston: D. Reidel ; Sold, distributed in the U.S.A., and Canada by Kluwer Academic Publishers, 1987. ISBN: 9027725136; 9789027725134. URL: https://worldcat.org/title/15548651.

[92] Fred Glover. "Tabu Search—Part I". In: *ORSA Journal on Computing* 1.3 (1989), pp. 190–206. DOI: 10.1287/ijoc.1.3.190. URL: https://doi.org/10.1287/ijoc.1.3.190.

[93] Fred Glover. "Tabu Search—Part II". In: *ORSA Journal on Computing* 2.1 (1990), pp. 4–32. DOI: 10.1287/ijoc.2.1.4. URL: https://doi.org/10.1287/ijoc.2.1.4.

[94] Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. 1st. Cambridge: Cambridge University Press, 2005. ISBN: 0521833787; 9780521833783. URL: https://worldcat.org/title/1025255419.

[95] Akshay Agrawal, Steven Diamond, and Stephen Boyd. "Disciplined geometric programming". In: *Optimization Letters* 13.5 (2019), pp. 961–976.

[96] Steven Diamond and Stephen Boyd. "CVXPY: A Python-embedded modeling language for convex optimization". In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.

[97] Akshay Agrawal et al. "A rewriting system for convex optimization problems". In: *Journal of Control and Decision* 5.1 (2018), pp. 42–60.

[98] Piers Ward. *How the agency model is shaking up the car retail industry*. 2022. URL: https://www.autocar.co.uk/car-news/business-dealership%5C%2C-sales-and-marketing/how-agency-model-shaking-car-retail-industry.

[99] Maximilian Holtgrave, Axel Schmidt, and Johannes Trenka. *The agency model is coming: Why this is good news for dealers*. 2021. URL: https://europe.autonews.com/guest-columnist/agency-model-coming-why-good-news-dealers.

[100] John Possumato. *Agency Model: Developing Trend in New Car Retailing?* 2021. URL: https://www.wardsauto.com/dealers/agency-model-developing-trend-new-car-retailing.

[101] Sebastian Tschödrich et al. *AGENCY SALES MODEL: ACCELERATING THE FUTURE OF AUTOMOTIVE SALES*. Tech. rep. Capgemini invent, 2020. URL: https://www.capgemini.com/wp-content/uploads/2020/11/Automotive-Agency-Sales-Model_POV_Capgemini-Invent.pdf.

[102] Benjamin Balensi. *New trends in the sales model of the automobile industry*. Tech. rep. Deloitte, 2021. URL: https://www2.deloitte.com/global/en/pages/legal/articles/sales-model-of-automobile-industry.html.

[103] Jan-Philipp Hasenberg. *Our Automotive Sales News series – Part 2*. 2021. URL: https://www.rolandberger.com/en/Insights/Publications/How-agency-sales-models-can-benefit-manufacturers-and-dealers.html (visited on 05/24/2022).

[104] Gérard Cachon, Santiago Gallino, and Marcelo Olivares. "Does Adding Inventory Increase Sales? Evidence of a Scarcity Effect in U.S. Automobile Dealerships". In: *Management Science* 65 (2018). DOI: 10.1287/mnsc.2017.3014.

[105] Xihong Fei, Yi Fang, and Qiang Ling. "Discrimination of Excessive Exhaust Emissions of Vehicles based on Catboost Algorithm". In: *2020 Chinese Control And Decision Conference (CCDC)*. 2020, pp. 4396–4401. DOI: 10.1109/CCDC49329.2020.9164224.

[106] Sami Ben Jabeur et al. "CatBoost model and artificial intelligence techniques for corporate failure prediction". In: *Technological Forecasting and Social Change* 166 (2021), p. 120658. ISSN: 0040-1625. DOI: https://doi.org/10.1016/j.techfore.2021.120658. URL: https://www.sciencedirect.com/science/article/pii/S0040162521000901.

[107] Rami Harb et al. "Exploring precrash maneuvers using classification trees and random forests". In: *Accident Analysis & Prevention* 41.1 (2009), pp. 98–107. ISSN: 0001-4575. DOI: https://doi.org/10.1016/j.aap.2008.09.009. URL: https://www.sciencedirect.com/science/article/pii/S0001457508001887.

[108] Denis Torgunov et al. "Vehicle Warranty Claim Prediction from Diagnostic Data Using Classification". In: *Advances in Computational Intelligence Systems*. Ed. by Zhaojie Ju et al. Cham: Springer International Publishing, 2020, pp. 483–492. ISBN: 978-3-030-29933-0.

[109] Mark Schwabacher, Robert Aguilar, and Fernando Figueroa. "Using decision trees to detect and isolate simulated leaks in the J-2X rocket engine". In: 2009, pp. 1–7. DOI: 10.1109/AERO.2009.4839691.

[110] Kürşat İnce and Yakup Genç. "Data Analysis for Automobile Brake Fluid Fill Process Leakage Detection using Machine Learning Methods". In: *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*. 2019, pp. 1–5. DOI: 10.1109/ASYU48272.2019.8946399.

[111] Dahai Zhang et al. "A Data-Driven Design for Fault Detection of Wind Turbines Using Random Forests and XGboost". In: *IEEE Access* 6 (2018), pp. 21020–21031. DOI: 10.1109/ACCESS.2018.2818678.

[112] Daniel Alvarez-Coello et al. "Modeling dangerous driving events based on in-vehicle data using Random Forest and Recurrent Neural Network". In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 165–170. DOI: 10.1109/IVS.2019.8814069.

[113] Shutao Wang et al. "A new method of diesel fuel brands identification: SMOTE oversampling combined with XGBoost ensemble learning". In: *Fuel* 282 (2020), p. 118848. ISSN: 0016-2361. DOI: https://doi.org/10.1016/j.fuel.2020.118848. URL: https://www.sciencedirect.com/science/article/pii/S0016236120318445.

[114] Abdul Rauf Khan, Henrik Schiøler, and Murat Kulahci. "Selection of objective function for imbalanced classification: an industrial case study". In: *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2017, pp. 1–4. DOI: 10.1109/ETFA.2017.8396223.

[115] Shuo Zhao, Xin Li, and Ying-Chi Chen. "A Classification Framework Using Imperfectly Labeled Data for Manufacturing Applications". In: *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Vol. 1. 2020, pp. 921–928. DOI: 10.1109/ETFA46521.2020.9211878.

[116] João Gonçalves et al. "A multivariate approach for multi-step demand forecasting in assembly industries: Empirical evidence from an automotive supply chain". In: *Decision Support Systems* 142 (2020), p. 113452. DOI: 10.1016/j.dss.2020.113452.

[117] Tingliang Huang and Jan Van Mieghem. "Clickstream Data and Inventory Management: Model and Empirical Analysis". In: *Production and Operations Management* 23 (2014), pp. 333–347. DOI: 10.2139/ssrn.1851046.

[118] Barbara Illowsky and Susan Dean. *Introductory Statistics*. Ed. by OpenStax. Houston, 2014. ISBN: 978-1-938168-20-8. URL: https://openstax.org/details/books/introductory-statistics.

[119] Mahmoud Akrin. *How Consumers Compare Prices To Make Purchase Decisions*. 2021. URL: https://www.forbes.com/sites/forbesbusinesscouncil/2021/06/18/how-consumers-compare-prices-to-make-purchase-decisions/?sh=244651864865 (visited on 12/12/2022).

[120] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[121] Jason Brownlee. *Classification and Regression Trees for Machine Learning*. Accessed: 08 January 2024. 2020. URL: https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/.

[122] Jason Brownlee. *A Gentle Introduction to Ensemble Learning Algorithms*. Accessed: 09 January 2024. 2021. URL: https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/.

[123] Jason Brownlee. *How to Develop a Bagging Ensemble with Python*. Accessed: 09 January 2024. 2021. URL: https://machinelearningmastery.com/bagging-ensemble-with-python/.

[124] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324.

[125] Jason Brownlee. *How to Implement Random Forest From Scratch in Python*. Accessed: 09 January 2024. 2020. URL: https://machinelearningmastery.com/implement-random-forest-scratch-python/.

[126] AIML.com. *What are the advantages and disadvantages of Random Forest?* Accessed: 09 January 2024. 2023. URL: https://aiml.com/what-are-the-advantages-and-disadvantages-of-random-forest/.

[127] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794. ISBN: 9781450342322. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

[128] Yoav Freund and Robert E Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139. ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.1997.1504. URL: https://www.sciencedirect.com/science/article/pii/S002200009791504X.

[129] Jason Brownlee. *Essence of Boosting Ensembles for Machine Learning*. Accessed: 10 January 2024. 2021. URL: https://machinelearningmastery.com/essence-of-boosting-ensembles-for-machine-learning/.

[130] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine." In: *The Annals of Statistics* 29.5 (2001), pp. 1189–1232. DOI: 10.1214/aos/1013203451. URL: https://doi.org/10.1214/aos/1013203451.

[131] Jason Brownlee. *A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning*. Accessed: 10 January 2024. 2020. URL: https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/.

[132] Liudmila Prokhorenkova et al. *CatBoost: unbiased boosting with categorical features*. 2019. arXiv: 1706.09516 [cs.LG].

[133] Prince Grover. *Getting Deeper into Categorical Encodings for Machine Learning*. Accessed: 14 January 2024. 2019. URL: https://towardsdatascience.com/getting-deeper-into-categorical-encodings-for-machine-learning-2312acd347c8.

[134] Daniele Micci-Barreca. "A Preprocessing Scheme for High-Cardinality Categorical Attributes in Classification and Prediction Problems". In: *SIGKDD Explor. Newsl.* 3.1 (July 2001), pp. 27–32. ISSN: 1931-0145. DOI: 10.1145/507533.507538. URL: https://doi.org/10.1145/507533.507538.

[135] Mario Filho. *How to Get Feature Importance in CatBoost in Python*. Accessed: 15 January 2024. 2023. URL: https://forecastegy.com/posts/catboost-feature-importance-python/.

[136] Eric Brochu, Vlad M. Cora, and Nando de Freitas. *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*. 2010. arXiv: 1012.2599 [cs.LG].

[137] Robert B. Gramacy. *Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences*. http://bobby.gramacy.com/surrogates/. Boca Raton, Florida: Chapman Hall/CRC, 2020.

[138] Kevin P Murphy. *Machine learning: a probabilistic perspective*. Cambridge, MA, 2012.

[139] David Kristjanson Duvenaud. "Automatic Model Construction with Gaussian Processes". Available at https://www.cs.toronto.edu/~duvenaud/thesis.pdf. PhD thesis. University of Cambridge, June 2014.

[140] Stathis Kamperis. *Acquisition functions in Bayesian Optimization*. Accessed: 2 February 2024. 2021. URL: https://ekamperi.github.io/machine%20learning/2021/06/11/acquisition-functions.html.

[141] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. arXiv: 1705.07874 [cs.AI].

[142] L. S. Shapley. "17. A Value for n-Person Games". In: *Contributions to the Theory of Games (AM-28), Volume II*. Ed. by Harold William Kuhn and Albert William Tucker. Princeton: Princeton University Press, 1953, pp. 307–318. ISBN: 9781400881970. DOI: doi:10.1515/9781400881970-018. URL: https://doi.org/10.1515/9781400881970-018.

[143] A Data Odyssey. *The mathematics behind Shapley Values*. Accessed: 8 February 2024. 2023. URL: https://www.youtube.com/watch?v=UJeu29wq7d0&list=PLqDyyww9y-1SJgMw92x90qPYpHgahDLIK&index=3.

[144] Christoph Molnar. *9.5 Shapley Values*. Accessed: 8 February 2024. 2023. URL: https://christophm.github.io/interpretable-ml-book/shapley.html.

[145] Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. *Consistent Individualized Feature Attribution for Tree Ensembles*. 2019. arXiv: 1802.03888 [cs.LG].

[146] Marco Peixeiro. *Time Series Forecasting in Python*. Manning Publications, Oct. 2022. ISBN: 9781617299889.

[147] *statsmodels.tsa.arima.model.ARIMA*. Available at https://www.statsmodels.org/devel/generated/statsmodels.tsa.arima.model.ARIMA.html (2022/06/29).

[148] W.A. Fuller. *Introduction to Statistical Time Series*. Wiley Series in Probability and Statistics. Wiley, 1995. ISBN: 9780471552390. URL: https://books.google.es/books?id=wyRhjmAPQIYC.

[149] *statsmodels.tsa.stattools.adfuller*. Available at https://www.statsmodels.org/dev/generated/statsmodels.tsa.stattools.adfuller.html (2022/06/29).

[150] *statsmodels.tsa.stattools.acf*. Available at https://www.statsmodels.org/stable/generated/statsmodels.tsa.stattools.acf.html (2022/06/29).

[151] *statsmodels.tsa.stattools.pacf*. Available at https://www.statsmodels.org/stable/generated/statsmodels.tsa.stattools.pacf.html (2022/06/29).

[152] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, Apr. 1992. ISBN: 9780262275552. DOI: 10.7551/mitpress/1090.001.0001. URL: https://doi.org/10.7551/mitpress/1090.001.0001.

[153] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley series in artificial intelligence. Addison-Wesley, 1989. ISBN: 9780201157673. URL: https://books.google.es/books?id=2IIJAAAACAAJ.

[154] Stephen Smith. "Flexible Learning of Problem Solving Heuristics Through Adaptive Search." In: Jan. 1983, pp. 422–425.

[155] J.H. Holland. "Escaping brittleness: The possibilities of general purpose learning algorithms applied to parallel rule-based systems". In: *Machine learning: An artificial intelligence approach*. Ed. by R.S. Michalski, J.G. Carbonell, and T.M. Mitchell. Vol. 2. Los Altos, CA: Morgan Kaufmann, 1986. Chap. 20, pp. 593–623.

[156] B. P. Murphy. "Handbook of Methods of Applied Statistics". In: *Journal of the Royal Statistical Society Series C* 17.3 (Nov. 1968), pp. 293–294. DOI: 10.2307/2985652. URL: https://ideas.repec.org/a/bla/jorssc/v17y1968i3p293-294.html.

[157] National Institute of Standards and Technology. *NIST/SEMATECH e-Handbook of Statistical Methods*. Accessed: 18 February 2024. 2012. URL: https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm.

[158] Felipe de Pontes Adachi. *Understanding Kolmogorov-Smirnov (KS) Tests for Data Drift on Profiled Data*. Accessed: 18 February 2024. 2022. URL: https://towardsdatascience.com/understanding-kolmogorov-smirnov-ks-tests-for-data-drift-on-profiled-data-5c8317796f78.

[159] CBC$_L$*ASALLE*. *CBC News. Entrevista: Isaac Partal, CEO de SEAT CODE*. Accessed: 12 June 2024. 2024. URL: https://issuu.com/cbc_lasalle/docs/cbc_news-6.

[160] SEAT:CODE. *SEAT CODE Home webpage*. Accessed: 12 June 2024. 2024. URL: https://code.seat/.

[161] Mohammad H. Eslami et al. "Knowledge-sharing across supply chain actors in adopting Industry 4.0 technologies: An exploratory case study within the automotive industry". In: *Technological Forecasting and Social Change* 186 (2023), p. 122118. ISSN: 0040-1625. DOI: https://doi.org/10.1016/j.techfore.2022.122118. URL: https://www.sciencedirect.com/science/article/pii/S0040162522006394.

[162] Soujanya Mantravadi, Jagjit Singh Srai, and Charles Møller. "Application of MES/MOM for Industry 4.0 supply chains: A cross-case analysis". In: *Computers in Industry* 148 (2023), p. 103907. ISSN: 0166-3615. DOI: https://doi.org/10.1016/j.compind.2023.103907. URL: https://www.sciencedirect.com/science/article/pii/S016636152300057X.

[163] Catherine Marinagi et al. "The Impact of Industry 4.0 Technologies on Key Performance Indicators for a Resilient Supply Chain 4.0". In: *Sustainability* 15.6 (2023). ISSN: 2071-1050. DOI: 10.3390/su15065185. URL: https://www.mdpi.com/2071-1050/15/6/5185.

[164] Zohaib Jan et al. "Artificial intelligence for industry 4.0: Systematic review of applications, challenges, and opportunities". In: *Expert Systems with Applications* 216 (2023), p. 119456. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.119456. URL: https://www.sciencedirect.com/science/article/pii/S0957417422024757.