Order cycle optimization in a luxury fashion marketplace

Tiago Manuel Medeiros Furtado

Master's Dissertation

Supervisor: Prof. Bernardo Sobrinho Simões de Almada-Lobo

U.PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Integrated Master in Industrial Engineering and Management

June 2019

Abstract

The luxury fashion industry presents itself as fast-paced, unpredictable and with ever increasing competition. When integrating it in an e-commerce environment, the demanding customer expectations are especially linked to considerations regarding order to delivery time (*Lead Time*). Looking at *Farfetch* as a case study, the main operational obstacles to meet these external requirements reside in its greatest virtue - the absence of stock ownership, and externalization of order processing.

This project aims to tackle this marketplace's complexity by performing an end-to-end analysis over the *Lead Time* and its explanatory drivers, ending the current scattered vision on order cycle performance, and identifying crucial opportunities to make it more agile and flexible. Through data mining techniques, it was possible to aggregate all relevant data that was leveraged to build a comprehensive monitoring tool. This way, a holistic view on the performance of each fulfillment step was provided. Additionally, analytical methods were employed in developing predictive models for the two main segments of *Lead Time*: one regarding partner processing and the other concerning carrier delivery. These predictive instruments enable the quantification of each drivers' importance on the different order cycle stages and the investigation of improvement scenarios.

From the insights generated along this research, three main improvement opportunities were studied, the increase of fraud automation level, the alignment of the partner daily cutoff hour with the respective pickup, and the boost of domestic routing. It was observed that optimizing these multi-participant activities had a predicted reduction of *Lead Time* in order cycle processes of more than 20%.

As a result, for the first time, order cycle stakeholders have access to *Lead Time* tools that enhance monitoring and support strategic decisions, paving the way for a better customer experience through operational excellence.

Resumo

Atualmente, a indústria da moda de luxo apresenta um ambiente de mudança constante, imprevisivel e com grande concorrência. Integrando-a num contexto de e-commerce, as expectativas dos consumidores enaltecem-se e prendem-se em grande parte com o tempo decorrido entre o momento de encomenda e a entrega (*Lead Time*). Usando a *Farfetch* como caso de estudo, os principais obstáculos operacionais para atingir as exigências dos consumidores estão refletidos nas suas maiores virtudes - a ausência de controlo de stock, e a externalização do processamento de encomendas.

Este projeto visa abordar a complexidade deste marketplace, realizando uma análise completa sobre o *Lead Time* e as suas variáveis mais relevantes, acabando com a visão dispersa presente atualmente sobre o desempenho do order cycle e identificando oportunidades para torná-lo mais ágil e flexível. Através de técnicas de data mining, foi possível agregar todos os dados relevantes, que foram aproveitados para construir uma ferramenta de monitorização. Desta forma, uma visão holística sobre o desempenho de cada etapa do processo foi desenvolvida. Além disso, métodos analíticos foram usados no desenvolvimento de modelos preditivos para os dois principais segmentos do *Lead Time*: um referente ao tempo de processamento do parceiro e outro referente ao tempo de entrega por parte do transportador. Estes instrumentos preditivos permitem quantificar a importância de cada variável nas diferentes etapas do order cycle e a investigação de cenários de melhoria.

A partir do conhecimento gerado ao longo desta pesquisa, três principais oportunidades de melhoria foram estudadas, o aumento do nível de automação de fraudes, o alinhamento da hora de término de processamento do parceiro com o respetivo pickup, e o aumento de rotas domésticas. Observou-se que a otimização destes processos tem uma redução prevista de *Lead Time* em processos do order cycle em mais de 20%.

Como resultado, pela primeira vez, os intervenientes do order cycle têm acesso a ferramentas que aprimoram o monitoramento do order cycle e apoiam decisões estratégicas ligadas a *Lead Time*, permitindo atingir uma melhor experiência para o cliente através de excelência operacional.

Acknowledgements

I would like to thank *Farfetch* for the opportunity of developing this project and for providing me with all the necessary resources and conditions to succeed. A special gratitude to the Operations Strategy team for embracing me and positively shaping my early professional development. In particular, to Filipe Aguilar and Fábio Moreira, for all the mentoring and help with all my struggles. And to Jorge Carmo and Rodrigo Veríssimo, for personalizing the selflessness of this organization, by voluntarily supporting me and enhancing my curiosity. I hope you all continue to be a part of my growth.

I would also like to acknowledge my dissertation supervisor, Professor Bernardo Almada-Lobo, whose guidance and feedback were vital in the completion of this challenge presented to me. My appreciation extends to all my professors at FEUP, for all the transmitted knowledge that enriched me as a student.

To my friends, with whom I have been sharing amazing experiences throughout my life, thank you for inspiring me to be the best version of myself.

For being my unconditional source of motivation, a huge thank you to my parents and brother.

"Motivated by curiosity"

Stephen Hawking

Contents

Acronyms and Symbols

KPI	Key Performance Indicator
ISE	In-Sample Error
OSE	Out-of-Sample Error
API	Application Programming Interface
AWB	Air Way Bill
GMT	Greenwich Mean Time
LT	Lead Time
TT	Transit Time
SoS	Speed of Sending
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentual Error
RMSE	Root Mean Squared Error
IQR	Interquartile Range Method
FIFO	First-In-First-Out

List of Figures

List of Tables

Chapter 1

Introduction

This chapter focuses on providing an introductory context of the project. It presents the e-commerce marketplace business setting used as a case study to investigate the operations behind an order-to-delivery process. Moreover, the motivation for this thesis is understood and the goal of optimizing the order cycle *Lead Time* is set. Finally, the main project stakeholders are identified and the methodology followed throughout the document is displayed.

1.1 Company Description

This thesis will have its central point in the marketplace operations of *Farfetch*. This is the branch of the *Farfetch Group* where luxury merchants across the world are faced with the opportunity to sell luxury items to the millions of monthly website visitors. These high-end customers have the possibility of browsing and acquiring the more than 300 000 items in the catalogue, and receiving them wherever they prefer.

Farfetch focuses on connecting innovation with fashion industry stakeholders across the globe, and consequently is positioned as one of the biggest players in the global fashion industry. This means being able to cope with the increasing empowerment of luxury fashion customers, that reveal themselves with high expectations, instant access to information and low switching costs, as stated in Tauriello et al. (2016). These independent individualities are less prone to develop specific brand loyalties, and thus the opportunity arises for small multi-brand boutiques to expand their client reach without significant investment and know-how in e-commerce operations. Larger luxury brands are also acknowledging the opportunity of a stronger digital presence, as the distress of mainstream brand dilution is being re-evaluated - Arienti (2019).

Furthermore, since a stock-less business model is in place, *Farfetch* is responsible for the allocation of customer orders to partner-owned stock points, which means it functions as an intermediary in the transaction, guaranteeing an effective communication between all parties involved and streamlining all the processes. This business model allows the company to avoid the risk of owning products that have high sales variability and seasonality. As of September of 2018, *Farfetch* is a public company, with an ever-expanding geographic presence and business diversification.

1.2 Motivation and Goals

Being a prominent player in the luxury fashion industry, *Farfetch* aims to provide the best customer experience in its marketplace through operational excellence. This customer-centric approach has its backbone in the order cycle, which is the workflow that defines the execution of daily orders, connecting more than 50 supplying countries to around 200 customer countries.

The order cycle is as old as the company itself, with its idealization and implementation dating back to 2008. It is composed of six steps that detail the operational journey beginning with the customer purchase in *Farfetch*'s website and ending with a third party carrier being responsible for picking and delivering the order to the final customer. *Farfetch* has no direct responsibility over operations, and lacks the comprehensive visibility over performance that is crucial to have, in order to adapt and streamline every process with all the different participants involved.

Furthermore, recent years have brought an increase in operational challenges, with the exponential growth in partners and number of orders, as shown in Figure 1.1. This complexity is inflated if the number of steps and intervenients are taken into consideration since massive coordination is required. In addition, the intervenients themselves are sources of variability, due to their different operating efficiencies and levels of integration with the internal systems. The absence of adaptability in the process makes it under-optimized, with numerous improvement opportunities.

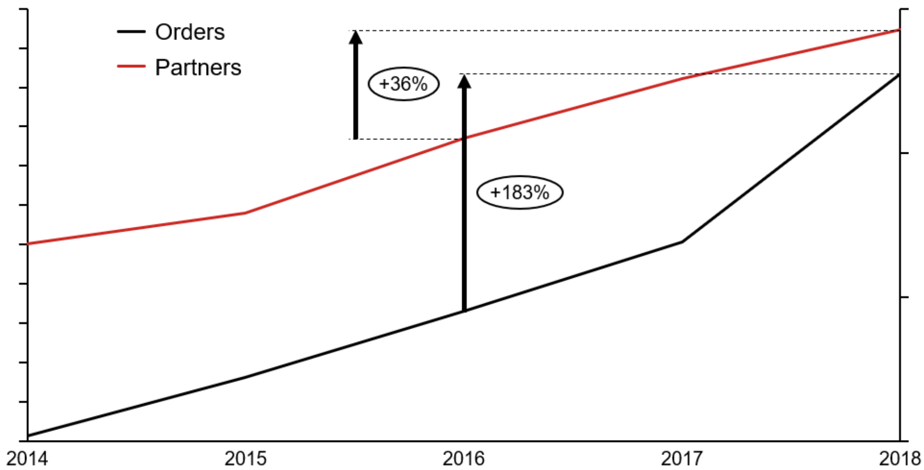

Figure 1.1: Evolution of number of partners and orders in *Farfetch*'s lifetime

Subsequently, the project presented in this thesis has two main goals. Firstly, to develop a tool that accurately assesses the end-to-end performance of the order cycle in terms of lead time and its variability - the Order Cycle Lead Time Dashboard. Secondly, to develop a Lead Time Predictive Model that quantifies the impact of the variables identified in the assessment phase and to use that model to test lead time improvement hypothesis, pinpointing the biggest opportunities to develop current operations. In short, enhancing the order cycle efficiency makes *Farfetch* more flexible and agile towards satisfying the high-end needs of luxury customers.

1.3 Project Stakeholders

This project was developed by the Operations Strategy team, working in close relationship with the Supply Chain and Fulfillment teams. These are the operational teams whose day-to-day deals with the problems addressed in this thesis. Supply Chain focuses on monitoring every process related to order management and coordinating all entities within the logistic environment of *Farfetch*. Fulfillment has responsibility over partner related issues. They are the contact point regarding operational efficiency and perform continuous evaluations, to allow every kind of partner to achieve its optimal execution capabilities. The Operations Strategy team has the support role of articulating all the internal teams and helping in the development of innovating solutions.

The relevance of this project begins in building an end-to-end view over the lead time performance of the order cycle. This way, the Supply Chain team manages to aggregate and standardize lead time metric reporting for the first time, utilizing the tool developed in this project on a weekly basis.

This project also enhances the ventures of the Fulfillment team. The Order Cycle Lead Time Dashboard helps identifying pain points in partner order processing. Additionally, the Lead Time Predictive Model unlocks the possibility of quantifying the impacts of new approaches. The main improvement opportunities are studied around *Farfetch* and partners processes, considering that these present the easier access points to facilitate future implementation.

1.4 Methodology

To better guide the project realization, the *Cross-Industry Standard Process for Data Mining* (CRISP-DM) was followed - Wirth and Hipp (2000). This methodology draws the framework for data mining projects, and this thesis goes through the proposed phases in Figure 1.2. This approach matches the development of the two goals presented: the Order Cycle Dashboard and the Lead Time Model.

The Order Cycle Dashboard development phase required an initial study of current operations, through a series of inductions and interviews with several stakeholders across the organization. This enabled a perception of the overall operational endeavours, as well as sharpened some specific details, paving the way to map the current order cycle. Afterwards, it was necessary to organize the data gathering procedures, since the information needed was sparse. This real sales data was recovered from different databases using *SQL*, with the main ones regarding order and partner information. The amount of data and its complexity forced an exhaustive preparation and treatment steps, removing unnecessary or not accurate entries. With the data-set ready for use and the order cycle already mapped, it was necessary to identify the most relevant drivers - for the global analysis and each of the steps. Finally, taking advantage of *Tableau* - a data manipulation and visualization tool - the Order Cycle Dashboard was developed iteratively, as constant feedback from several stakeholders was always taken into consideration.

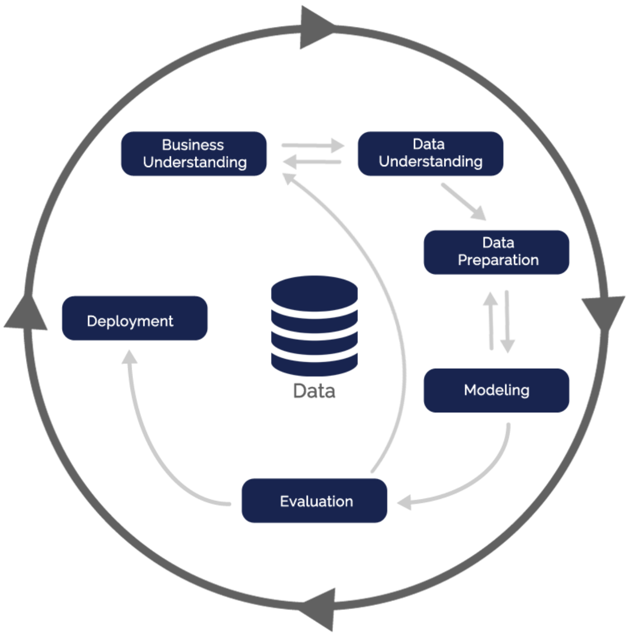

Figure 1.2: CRISP-DM Methodology. Source: Otaris (2011)

The Lead Time Model is built using the same data-set prepared earlier. By modelling the lead time according to the investigated and prepared variables, it was possible to assess and quantify their impact on the lead time performance, using *Python 3.7* as the programming software. After completing this phase, it was necessary to identify the different hypothesis to be tested using the model, which means investigating how to change the different variables and measure its lead time impact, completing a what-if analysis.

1.5 Thesis Outline

This document is organized in six chapters:

Chapter 1 gives the initial insights on the basis on which the project occurs, by presenting the company situation, the motivation of the project, the end objective with its development and the methodologies that were employed.

Chapter 2 evaluates the current state-of-the-art, regarding the industry the company is inserted in, as well as the way raw data can be treated and visualized to extract the most reliable conclusions. Moreover, an introduction to predictive models is done.

Chapter 3 dissects the current order cycle, including analysis of every order cycle step. Additionally, the monitoring and visualization tool is disclosed.

Chapter 4 introduces the use of machine learning to model the lead time, and the results obtained.

Chapter 5 proposes optimizations opportunities and evaluates their impact.

Chapter 6 assesses the conclusions of the project and presents future research.

Chapter 2

State-of-the-Art

The State-of-the-art chapter provides the theoretical background that helps understanding the conditions surrounding the development of this project. This overview analyzes both external environment of the business and the tools and methodologies that support the technical aspects presented further in the document.

2.1 E-commerce luxury fashion industry

This project is set around the e-commerce luxury fashion industry, and so it is important to evaluate its main features, in order to understand how the developments of this thesis fit with external market requirements. First of all, this industry has been significantly evolving over the last 20 years, since it has been hard to unanimously define what is valuable in aesthetic terms - Djelic and Ainamo (1999). Therefore, luxury fashion products are associated with intangible dimensions of value such as power and exclusivity (Li et al. (2012)), which leads this industry to be highly unpredictable, fast-changing and extremely competitive.

Despite the uncertainty associated with the fashion industry, its appeal has been steadily translated into growth in sales, especially in the luxury segment - Berg and Amed (2019). This fact is confirmed by Figure 2.1, where the upward growth trend is clear for the upcoming years. This figure also points out the increasing share of online sales. It is expected that by 2025, almost one in every five sales will happen on online and digital channels.

Digital is not only having an impact on the retail channels, but it is also massively influencing how luxury shoppers choose their goods - Marchessou et al. (2018). About 80% of luxury sales are "digitally influenced", which means that there is an increasingly close connection between the offline and online customer journey, in terms of inspiration, discovery and purchase. Consequently, e-commerce is presenting itself as a reliable part of omnichannel solutions to the industry's players, where e-tailers tuned for flexibility and technology have been emerging. Moreover, despite mono-brand websites still having the majority of the market's online sales, multi-brand marketplaces as *Farfetch.com* are presenting a more acute growth.

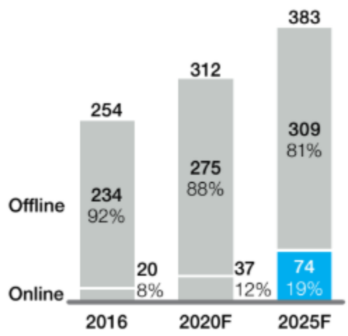

Figure 2.1: Luxury fashion industry sales evolution. Source: Marchessou et al. (2018)

The luxury fashion environment is being defined by the new generation of customers, who are being described as HENRYs (High-Earners-Not-Rich-Yet) - Arienti (2019). Moved by instant satisfaction, high information access and awareness, the industry is dealing with increasingly demanding consumers, who are seeking self-differentiating multi-brand experiences and thus, shifting from the traditional approach of brand loyalty through heritage. This new type of shopper is enabling the paradox of having a luxurious and customized experience through traditionally undifferentiated e-channels - Okonkwo (2010).

Advances in more analytical methods, such as big data and machine learning, are empowering businesses to develop a more customized relationship with the consumers, introducing the concept of omni-personal luxury, where the personalization of customer demands plays a central role, allowing for a more holistic experience - Socol (2018).

With the challenging environment imposed by consumers and competitors, every business focuses on having order-winner operational features, translated mainly in fast order to delivery times (Heim and Sinha (2001) and Sheng and Liu (2010)) and a wide product range. Giving the customer what he wants, when he wants it, has a major impact on his retention, and thus an increase in future revenue. This meets the investigation done in this project, as an efficient order cycle with reduced lead time is a step towards satisfying customers.

Supported by what was previously stated, the luxury fashion industry presents itself as a high potential opportunity for suiting contenders, that need to be extremely flexible and have a great focus on customer-centricity. To achieve that, one can leverage this data-immerse environment and continuously apply state-of-the-art analytic methods to generate the necessary business insights. Marchessou et al. (2018) is inspired by Industry 4.0 to suggest the trend for a fully digitized luxury fashion model - Luxury 4.0.

2.2 Big Data: Preparation and Visualization

Nowadays, information is an intrinsic part of the way our life and businesses operate, from its generation to its accessibility and use - Khan and Khan (2011). Especially in e-commerce environments, such as the one evaluated in this thesis, a great amount of data is being created from all the available customer touch-points and the relative absence of difficulty of gathering its outputs - Chen et al. (2012). This massive amount of available data follows the HACE theorem - *Big Data starts with large-volume,* ***H****eterogeneous,* ***A****utonomous sources with distributed and decentralized control, and seeks to explore* ***C****omplex and* ***E****volving relationships among data.* This translates in an estimate of doubled worldwide data volume every two years - Gantz and Reinsel (2011). In the scope of this project, it is important to understand how to leverage all the potential data into relevant insights.

As might be expected, this kind of disperse knowledge has a huge potential to be gathered and applied to develop intelligence that can impact many different areas. Naturally, each area collects its specific effects, but in general analysis, a good data understanding leads to higher output efficiencies. Specifically, it is reveal that e-commerce companies have experienced a 5 to 6% productivity surplus through effective data analysis - Akter and Wamba (2016).

However, big data is accompanied by some setbacks. Among them is the inherent storage problem for all the necessary information that needs a repository. Additionally, since it is difficult to have a complete and in-detail comprehension of the data-set, redundancy and misrepresentation can cause trouble. All this is complemented by the intrinsic high processing power required to manipulate said data, and adjacent costs. Moreover, data in its raw state is not reliable, as it can contain inaccurate entries that influence the outcome of the analysis, and therefore should be removed or its occurrences minimized. Data pollution can arise mainly from two different sources: data errors and data variability - Anscombe (1960). Data errors can be avoided by reducing the utilization of manually introduced variables and maximizing the use of standardized and consistent sources of information. Osborne and Overbay (2004) perform a deep dive in data error types and their different causes.

Another key feature of data treatment is outlier detection and removal. Outliers are a source of data variability and are defined by Hawkins (1980) as an observation that "deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism". Although some authors claim outlier removal brings unwanted outcomes, the majority of the literature argues otherwise. Osborne and Overbay (2004) confirm this, as they show an increase in accuracy and reduction in inference error by successfully dealing with outliers in their estimates.

Outlier identification depends heavily on the data statistical distribution and number of observations in the data-set. To address the most general of problems, one can use the Interquartile Range Method (IQR). This method does not require a data-set with specific statistical distribution and is not influenced by the mean and standard deviation - two parameters highly affected by outliers - Leys et al. (2013). This outlier removal procedure can be visualized in Figure 2.2 and works as follows (Walfish (2006)):

1. Identify the 50th percentile (median), the 25th percentile (Q1) and the 75th percentile (Q3) of the data-set.

2. Compute IQR as being the difference between Q3 and Q1.

3. Compute maximum as being (α*IQR) times greater than Q3.

4. Compute minimum as being (α*IQR) times lesser than Q1.

5. Remove data that outside the boundaries set by the minimum and the maximum.

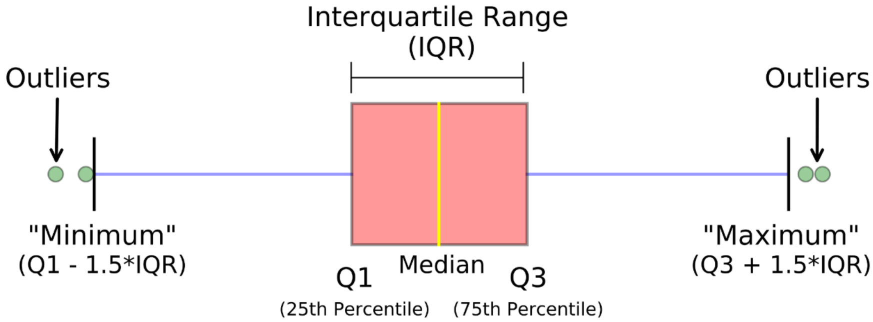

Figure 2.2: Interquartil Range for outlier identification. Source: Galarnyk (2018)

The parameter α is generally 1.5. However, this value can be adjusted given the specific characteristics of each data-set, allowing the user to choose how conservative to be in the approach. Once identified, one should try to recognize the root causes of the outliers. If a cause cannot be assigned, the data-point should be kept for future evaluation with more data. If it is possible, one has to decide if that root cause should be influencing the data-set or not.

To complement a good data structure one needs robust visualization methods to enhance data analysis, decision-making and information sharing (Wang et al. (2015)). These techniques can be thought of as the "front end of big data". To effectively apply them, one must guarantee that only pertinent data must be displayed, and that this data must be well organized and easily perceptible according to the available space. This ensures simplicity and enhances user comprehension.

The big data challenges presented previously continue to happen when analyzing visualization techniques, translated in visual noise, information loss and high-performance requirements. These can be mitigated by using specially designed software. In the context of the project presented in this document, *Tableau* is used - a business intelligence software that supports interactive and visual analysis of data - Wang et al. (2015). It is specialized in large-scale data-sets and is capable of providing a smooth user experience, due to its interactive architecture and effortless data integration.

2.3 Predictive Models

After evaluating how data behaves in current state-of-the-art situations, it is fundamental to understand how one can use analytic methods to leverage it. Predictive models are an extremely powerful analytic tool to infer future uncertainties from historical records, by analyzing the underlying structure of data and its patterns - Friedman (2006).

This project aims to build a model that is a supervised learning application to a regression problem, since the data contains descriptive labels and the output variable is a continuous measure

- *Lead Time*. These kind of predictive models are commonly generated by machine learning algorithms, that can arise from two main approaches (Qiu et al. (2016)). Unsupervised learning, where the algorithm aims to learn the data structure without knowing the inherent data labels. The second one is supervised learning, where the model is being fed the data labels and the real outputs to grasp patterns. Supervised learning models can broadly fall into two categories - classification models, where the output variables are categories or a class; and regression models, where the output variable is a continuous real-value (e.g. amount, quantity).

The programming tool that enabled all the investigation was *Python*. This open-source computing language stands out for its accessibility and convenience - McKinney (2011). Its wide range of stable numerical libraries and quality documentation provides a solid data analysis environment.

Sections 2.3.1 through 2.3.4 describe the necessary modelling steps to be followed, investigating which algorithms can be used in the studied case and how to perform its validation, evaluation and hyperparameter tuning.

2.3.1 Model Validation

Predictive models have a goal of being able to generalize outputs from inputs that they have never received. To effectively validate the results from the model, one has to divide the data-set in three components. The training set is the data from which the model will learn and iterate its best fit. The validation set is used to evaluate the training results and to tune parameters. Finally, the test set is the data the model receives once is fully trained, to understand its behaviour in unforeseen conditions. Figure 2.3 provides an intuitive understanding of this split. The training set consists of the majority of the data set, and the validation and test sets are divided equally from the remaining data. If this is not done, the model will be evaluated using data it was trained on, giving false conclusions about its performance. This split is done in a sequential order, in order to capture the temporal relationships between the observations.

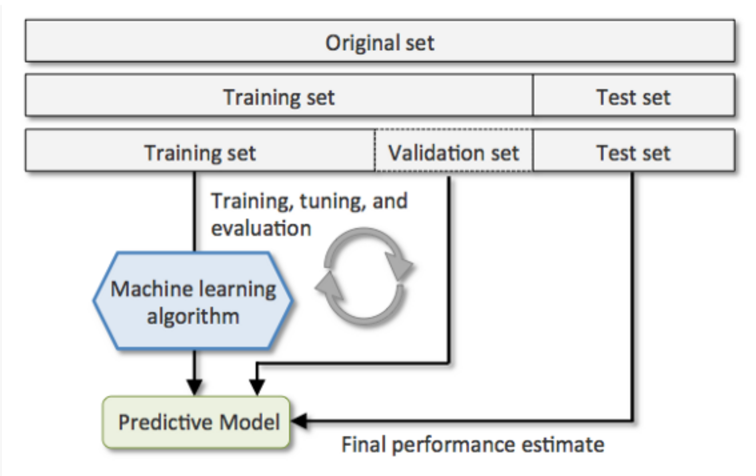

Figure 2.3: Data-set split. Source: Patel (2018)

2.3.2 Model Evaluation

Model evaluation techniques need to be comparable, measurable and reproducible across multiple data-sets. To correctly assess the quality of the proposed regression model, one has to measure two kinds of errors: in-sample errors (ISE) and out-of-sample errors (OSE). The evaluation metrics can be equally applied to both errors and are done by comparing the real output to the predicted output. The difference between them is the data used in the calculation: ISE uses the training set and OSE uses the test set.

Prediction errors are originated by three factors, according to Singh (2018b):

- Noise - intrinsic difficulty of regression problems;
- Bias - how far away is the best learner from the correct one;
- Variance - how variable is the model predictions for a given data point.

High variance means the model is including too much noise from the data-set and ends being too flexible in the predictions - commonly refered as *over-fitting*. High bias means the model is predicting values significantly different from the real ones - also known as *under-fitting* -, caused by an over-simplistic approach. The goal in model construction is to maximize the addition of explanatory power without over adjusting to a limited set of data, which means achieving a low bias and low variance. Figure 2.4 presents a visual representation of the above statements.

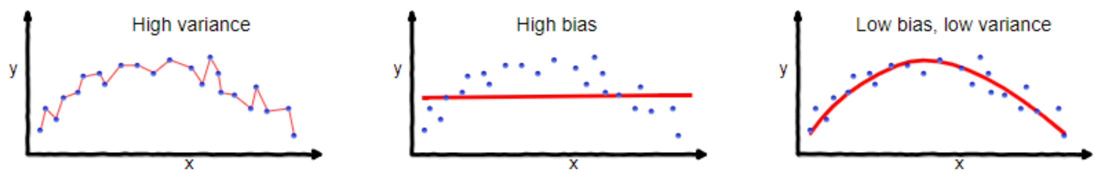

Figure 2.4: Model output impact of different types of prediction errors. Source: Singh (2018b)

Moreover, in chapter 1, for a model to be robust, it must be carefully evaluated. Hence, in order to assess the proposed regression models, the three metrics presented in equations (2.1), (2.2) and (2.3) are considered.

$$Mean\ Absolute\ Error = \frac{1}{N} \times \sum_{i=1}^{N} |y_i - \widehat{y_i}| \tag{2.1}$$

The Mean Absolute Error (MAE) measures the average magnitude of the errors, without considering their direction. It compares the predicted value ($\widehat{y_i}$) with the real one (y_i), and averages all the deviations. Additionally, it is an absolute measure, so it does not differ between negative and positive errors. This makes sense in the models of this project, since it is not desirable to know if the prediction is above or below the real value.

$$Mean\ Absolute\ Percentual\ Error = \frac{1}{N} \times \sum_{i=1}^{N} \left| \frac{y_i - \widehat{y}_i}{y_i} \right| \tag{2.2}$$

The Mean Absolute Percentual Error (MAPE) presents similar characteristics and conclusions to MAE. It quantifies the magnitude of the prediction error and not their orientation. Nonetheless, it is a relative measure that facilitates comparisons of models with results with different magnitudes.

$$Root\ Mean\ Squared\ Error = \sqrt{\frac{1}{N} \times \sum_{i=1}^{N} (y_i - \widehat{y}_i)^2} \tag{2.3}$$

The Root Mean Squared Error (RMSE) is the standard deviation of the prediction errors, which discloses how spread the error is from the model fit. In accordance with MAE and MAPE, it is indifferent to the direction of errors. Despite that, and unlike the previous ones, it is more sensitive to outliers, as it gives a relatively high weight to large errors. One important advantage is that both RMSE and MAE express the error in the same units as the model's results.

2.3.3 Algorithms

Within the supervised learning and regression algorithms, Decision Trees are one of the most popular and intuitive predictive learning methods, where tree-like data structures are used to perform binary decisions. Each tree consists of a root node, decision nodes and a terminal node - Chepenko (2013). Figure 2.5 shows an example of the typical decision tree architecture. Starting from the root node, all possible feature splits are calculated and the selection is made in a greedy manner - it is chosen the one that maximizes loss function improvement. Subsequent splits are done based on the previous ones, until there is a terminal node where the target variable is predicted.

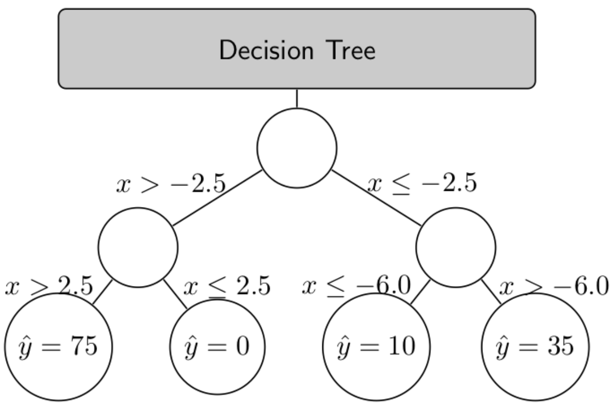

Figure 2.5: Example of the inherent structure of a Decision Tree

No matter how many dimensions the predictor variable space has, the decision tree model can be plotted and examined for interpretation. Moreover, there are no restriction in variable types, since both numerical and categorical variables are supported.

To develop a tree, one should study and decide which features to use in the splits, which splitting conditions should be in place and the stopping conditions. Inherently to their constructing nature, decision trees tend to be over-fitted, presented with low training error (ISE), but high test error (OSE). Adequate stopping conditions, such as maximum number of leaves, maximum tree depth and maximum iterations, can prevent this occurrence, by limiting the tree's adaptability to a specific part of the data-set.

To overcome the high model variance, ensemble methods were developed for decision trees. The principle behind them is that a linear combination of several weak decision trees produce a better predictive performance than a single low-bias, high-variance decision tree.

The models in this project are based in the decision trees ensemble algorithm named Gradient Boosting. This is one of the most efficient ways to build ensemble models, with state-of-the-art results in structured data applications - Ershov (2018). In an iterative way, the first tree is set to minimize the training error, which is only slightly better than random chance. Then, the subsequent trees optimize their learning by minimizing the error from the previous trees, using gradient descent to optimize the loss function, i.e. looking at the derivatives of the function with respect to each of its parameters, and seeing which step, via which parameter, is the next best step to minimize the function. Considering that the previous tree is being iteratively evaluated, it is possible to weight the observations accordingly, allowing the following trees to grow on weighted data, as stated by Singh (2018a).

To implement the machine learning approach to gradient boosting on decision trees, this project took advantage of an open source library named CatBoost. This algorithm thrives in huge data-sets with diverse data-sources, and has an innovative approach to dealing with categorical features that prevents the need for one-hot-encoding, a major challenge in machine learning problems. In Dorogush et al. (2018) a benchmark with other gradient boosting algorithms is made, revealing that CatBoost has a superior performance in terms of accuracy and processing time. For further study and curiosity, consider reading the article by Prokhorenkova et al. (2018), as it explains in detail all the theoretical nuances of this algorithm.

In order to validate the use of gradient boosting in this project, it is necessary to study alternative approaches to solve regression problems. It is important to keep in mind that the data-set in study has millions of data points, and its variables are mainly categorical. Therefore a benchmark was made with other algorithms design for regression problems with continuous outputs.

Multiple Linear Regression is the method of discovering a linear equation that explains the relationships between independent variables and a dependent one. It is a rather simplistic approach, as it only captures linear dependencies. Consequently it is not able to represent more complex, non-linear associations. However, it can be a good alternative when the data has a lot of features and low noise - Tranmer and Elliot (2008).

Random Forest is a decision trees algorithm, much like gradient boosting. The main difference is that Random Forest ensembles on fully grown decision tress, while in gradient boosting new shallow and weak trees complement previous ones. This allows Gradient Boosting to achieve better accuracy with less trees. Random Forest approaches are based on random portions of the data-set, which can lead to high model bias. Additionally, categorical variables are handled in a way that the model is biased towards categories with more levels - Goel et al. (2017). Carvalho (2016) used *Farfetch* as a case study to develop a conditional inference decision tree model for each step of the order cycle processes. Her work will be used in further chapters as a comparison to the models in this thesis.

Overall, Gradient Boosting provides a lot of flexibility, since it can optimize different loss functions and provide several hyperparameter tuning alternatives. Moreover, no specific data pre-processing is required for categorical variables and it easily handles missing data without requiring imputation.

2.3.4 Hyperparameter Tuning

One of the challenges in predictive modelling is the correct tuning of the hyperparameters. This is the process of optimizing the model's configurations in order to get the best results possible. These hyperparameters are used as models inputs (e.g. learning rate, search space), and can affect the outputs. Since there are multiple possible alternatives for its values, this is an highly time-consuming process. Manual *ad hoc* optimization heuristics augment this problem, specially in awkward search spaces.

Grid Search and *Randomized Search* are more efficient methods and were evaluated. The first assesses every possible combination of parameters. The latter chooses to analyze a reduced amount of combinations and outputs the best one. There is an obvious trade-off between run time and accuracy, that needs to be considered depending on the specific model application. On one hand, *Grid Search* has a more extensive range, but exponentially grows computing time as the dimensions increase. On the other hand, *Randomized Search* has a lower computing time and generally outputs results close to optimal - Bergstra and Bengio (2012).

Probst et al. (2019) show various experiments on algorithms based on decision trees, and reveal that tuned models perform better than models with the default settings. This configurations are dependent on properties of the data-set at hand and are specially important to prevent overfitting, a common problem in decision trees algorithms.

2.4 Final Considerations

The investigation presented in this chapter has a supportive role in the realization of this project. Understanding the e-commerce luxury fashion industry reveals the importance of optimizing the order cycle lead time, as to meet the customer's requirements. After reinforcing the inherent motivation, the study on big data was used in the diagnosis of the order cycle processes, aiding in data gathering, manipulation and visualization. This way, there is an comprehensive baseline to enable accurate insight generation. Finally, the investigation on predictive models is useful to complement the big data processes previously described. Not only is it possible to define which is the best model to use for this situation, but also there is an identification of the methodologies needed to correctly develop a precise prediction tool.

Chapter 3

Order Cycle Diagnosis

As discussed in the first chapter, *Farfetch* runs a marketplace business model, characterized by the absence of stock ownership. Its customers' expectations are a continuous challenge to company's operations, which means the responsible teams are permanently finding solutions to provide a world-class experience to them. Throughout this document, the process beginning with the customer purchase in *Farfetch*'s website and ending with the order delivery is named Order Cycle.

In the first section of this Chapter, it is presented a description of the order processing steps. Subsequently, it is exhibited the data collection process and an exploratory analysis on the identified variables. The goal of this Chapter is fulfilled in the last section, where are the aggregated information is used to develop a monitoring tool.

3.1 Order Processing

The operational architecture of *Farfetch* allows a customer to browse thousands of products in one website and select several of them to make a purchase, even though they can be sourced by different partners. When a customer places an order (*Portal Order*), multiple items can be chosen - each one is a specific product order - and they can belong to different boutiques. Each boutique that fulfills part of a *Portal Order* has a *Boutique Order* attribution. In the example of Figure 3.1, the *Portal Order* is composed by three products, two of them coming from the same merchant and the last from a different partner, meaning two *Boutique Orders* are placed.

There are two sequential algorithms responsible for the allocation of orders to partners, if there are multiple stock points that can satisfy the same customer request. One decides the stock point price to be displayed in the website, and the other selects the final stock point which will ship to the customer. After the allocation, the order cycle process goes as seen in Figure 3.2. There are six main steps in the order cycle process: *Check Stock*, *Approve Payment*, *Decide Packaging*, *Create AirWayBill*, *Send Parcel* and *In Transit*. Each one will be subsequently presented.

Step 1 - Check Stock

The first baseline that the partner must meet, after having an order allocated, is ensuring stock availability. This process depends on the integration the partner has with *Farfetch*. Either the

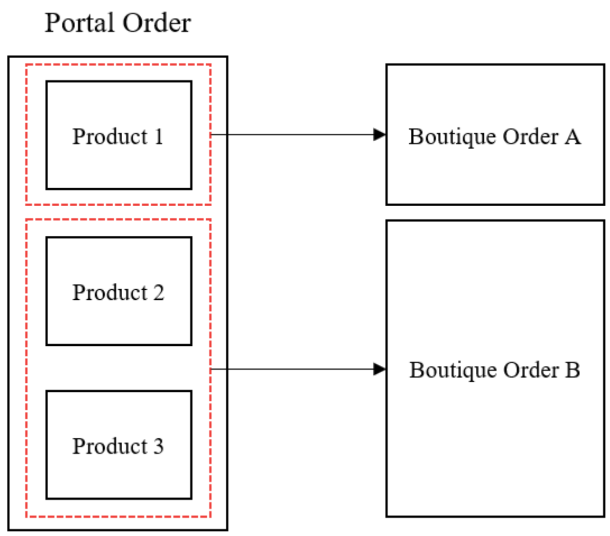

Figure 3.1: *Portal Order* and *Boutique Order* distinction

information is manually introduced using the provided software - *STORM*, or is automatically communicated though an Application Programming Interface (API). Smoother data synchronization is achieved when partners have a higher degree of integration with the internal systems. If the allocated partner does not comply with the stock solicitation, the algorithm re-allocates the order to the next best partner, until stock is guaranteed. In case of no partner is able to satisfy the order, the customer receives a refund, or a similar available item is suggested.

Step 2 - Approve Payment

At the same time the partner is validating the stock, there is an internal *Farfetch* process to check if the purchase is fraudulent. Due to the high number of daily orders, this is mainly an automatic process. There is a tool that classifies every order in one of three types: non-suspicious, suspicious or very suspicious. Non-suspicious and very-suspicious orders are automatically cleared or refused, accordingly. Suspicious orders are passed to the internal Fraud team that further investigates them. Although Step 2 occurs simultaneously with Step 1, fraud validation is faster in around 80% of orders and thus, Check Stock is the bottleneck in the initial order processing phases.

Step 3 - Decide Packaging

After stock and fraud validations of the order, the partner is able to decide in which box the order will be packed. Boxes are provided by *Farfetch* and there is a box recommendation algorithm based on available stock and order size that the partner can choose to take into consideration.

Step 4 - Create AirWayBill

Subsequent to packaging, there is an automatic generation of the AirWayBill (AWB), that contains shipping-related information to identify the parcel. Internal teams only intervene if there is any missing or incorrect information, or if there is a pending legal matter.

Step 5 - Send Parcel

As Step 4 is completed, the package is signalled as ready to be picked up by the courier. Partners can either have *ad hoc* scheduled pickups, or more predictable pickups performed daily. Considering that there is no complete visibility over the order preparation, there is shared responsibility between the partner and the carrier to smoothen pickups. This step denotes the end of partner interaction, whose time-span is referred as *Speed of Sending* (SoS). This metric is computed as follows:

$$Speed\ of\ Sending = \max\{Step_1\ Time,\ Step_2\ Time\} + \sum_{i=3}^{5} Step_i\ Time, \tag{3.1}$$

where Step_i denotes the time taken by each processing Step i.

Step 6 - In Transit

Finally, Step 6 reflects *Transit Time* (TT) between the merchant location and the client's delivery point. As around 70% of orders have to cross borders, there is time spent in customs added to the actual time spent travelling. This means high variability, because border control authorities are significantly different depending on the receiving country and the nature of the product being dealt with.

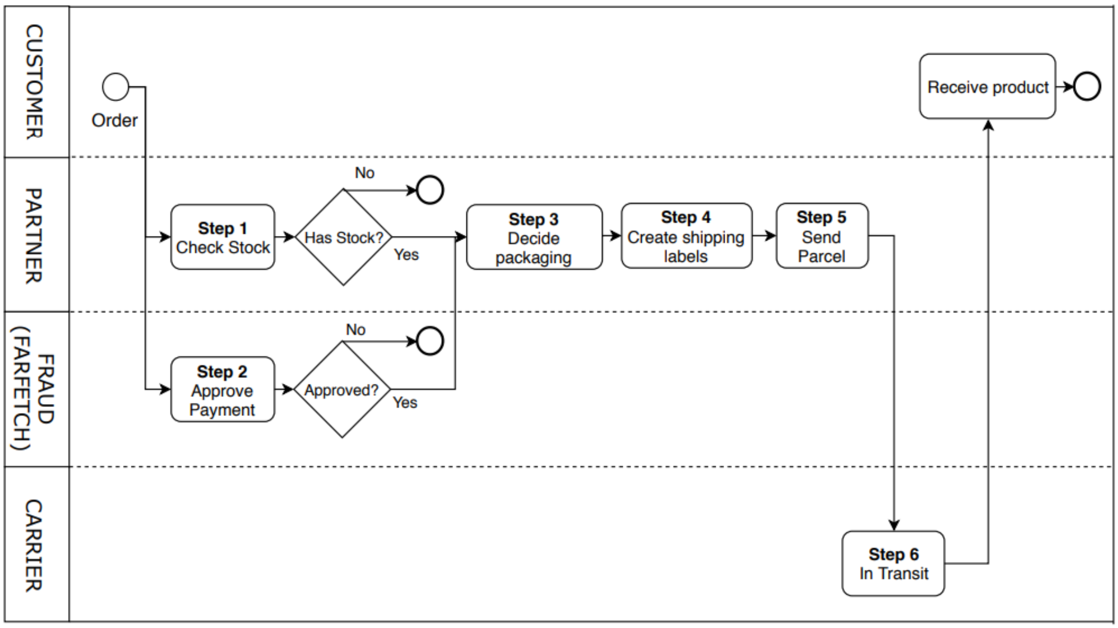

Figure 3.2: Order Processing Steps

The order cycle process is thus concluded, and the *Lead Time* (LT) is calculated by equation (3.2), differentiating the time spent processing the order in the partner and the time spent transporting the order.

$$Lead\ Time = Speed\ of\ Sending + Transit\ Time \tag{3.2}$$

3.2 Data

As it was introduced in Chapter 1, the data gathering process was based on internal databases' information. In order to identify the correct data sources, several stakeholders were contacted in order to assess impactful drivers of lead time and its data availability. In addition, due to the amount of data to collect, several tools were used to extract and process it, including SQL, Python and Tableau. The data regarding orders used in this project was collected at the *Boutique Order* level, since significantly different operational behaviours are expected of orders that are fulfilled by different partners, and each one of them follows an independent order cycle process.

To assemble the data-set, only completed orders were considered (i.e., no cancellations). Moreover, the temporal data-range covers ten months, as to only include orders with full integration of the new fraud automation provider. Furthermore, only three different carriers supply reliable information, so the rest of orders were discarded. The collected variables are presented in Table 3.1, where there is a distinction between variables affecting the general process, variables affecting the *Speed of Sending* and variables mainly influencing the *Transit Time*. These will be further scrutinized in sections 3.2.1, 3.2.2 and 3.2.3.

Regarding data processing, all date-time fields were deconstructed into variables comprising the corresponding *hour*, *day*, *weekday*, *month* and *year*. Moreover, extra data manipulation was needed, considering that dates could be harvested in any timezone. Therefore, a normalization was done, to enable dates to be analyzed in three different timezones: GMT (Greenwich Mean Time), Partner Timezone and Customer Timezone.

Table 3.1: Collected variables regarding order processing

	Variable	Type	Description
General	Partner ID	Categorical	Identification number of each partner
	Partner Country	Categorical	Country of the partner
	Daily Pickup	Categorical	Indicator if the partner has a scheduled daily pickup
SoS	Order Step Date	Numeric	Date-time of each step of order processing
	Approval Type	Categorical	Type of fraud validation (Automatic vs Manual)
TT	Order Pickup Date	Numeric	Date-time of pickup
	Customer Country	Categorical	Country of customer
	Customer Region	Categorical	Region of customer
	Service Type	Categorical	Type of carrier service (Express vs Standard)
	Crossborder	Categorical	Route classification regarding customs clearance

Considering that the end goal is to evaluate order cycle lead time - a continuous variable that is supported by date time data-fields - , it is necessary to perform an outlier identification and removal, as introduced in Chapter 2. As described in Figure A.1 in Appendix A, this process was applied to the main time interval variables: *Lead Time*, *Speed of Sending* and *Transit Time*, removing 1,16% of data-points. These were the chosen variables because of the impact they have in the outcome of the analysis. *Lead Time* is the overall dependent variable; *Speed of Sending* and *Transit Time* represent a significant order processing segmentation, with mutually independent process characteristics and dynamics. The values that were removed represented extremely unusual and irrelevant situations that resulted from system failures or manually introduced data.

An overview of the descriptive statistics of the continuous variables is presented in Table 3.2. The frequency distribution graphs in Figure 3.3 and in Appendix B endorses what is presented in Table 3.2. For all measures, the median is lower than the mean, indicating a right skewed distribution. *Transit Time* represents around 70% of *Lead Time* and has the major contribution to the overall variation of results. This is a more dynamic process and thus, more prone to unpredictable events.

Table 3.2: Global data-set descriptive statistics

Metric	Median (days)	Mean (days)	Std. Dev. (days)	Coef. Var.
Lead Time	4.1	4.6	2.5	0.5
Transit Time	2.6	3.2	2.4	0.7
Speed of Sending	1.0	1.4	1.1	0.8
Step1	0.4	0.6	0.7	1.2
Step2	0.0	0.1	0.4	4.0
Step3	0.0	0.2	0.5	2.5
Step4	0.0	0.0	0.1	8.5
Step5	0.3	0.5	0.8	1.6

Within the *Speed of Sending* steps, it is clear that the most time consuming steps are the ones where there is the need for information flow or communication between the responsible entities (Step 1 and Step 5), whereas the middle steps are substantially quicker (Step 2, Step 3 and Step 4). Step 1 takes the longest time because partners tend to accumulate orders to later perform batch fulfillment. This way, they also maximize the probability of fraud validation being done before they begin processing it. Step 5 duration reflects the pickup coordination complexity between carrier and partners, as described in section 3.1. Step 4 has the lowest mean duration because it is performed by the partner almost immediately after Step 3, removing all friction related with coordination and communication. The most uncertain steps are 2 and 4, because although they are generally automatic - low mean duration -, when there is a punctual disruption of the normal motion of the process, the impacts are severe.

It is also important to note that these are overall calculations rounded at the decimal level, and that performance depends highly on several variables that will be subsequently presented.

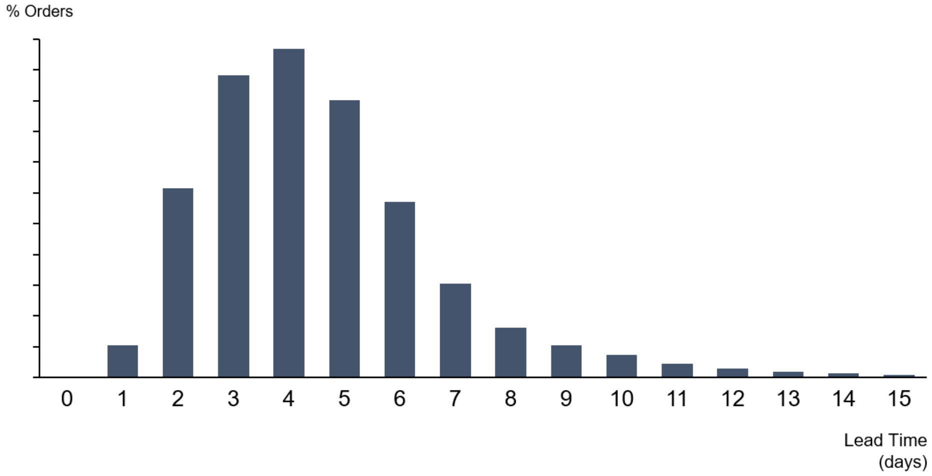

Figure 3.3: Lead Time frequency distribution (days)

3.2.1 General Variables

In this section, it is presented an exploratory analysis of the general variables influencing both *Speed of Sending* and *Lead Time* - *PartnerID*, *Partner Country* and *Daily Pickup*.

PartnerID

Figure 3.4 demonstrates that there is significant differences in *Lead Time* between partners. Each of them has an independent processing configurations and backlog control policies when it comes to order fulfillment that influences their efficiency. Additionally, each unique set of locations impacts the carrier routes, and consequently, the *Transit Time*.

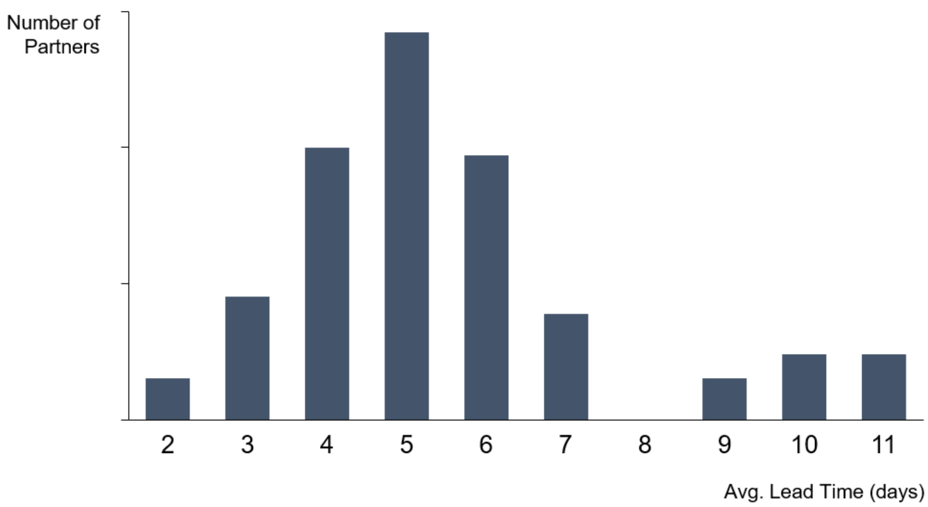

Figure 3.4: Number of partners per average *Lead Time*

Partner Country

Partner Country distribution - Figure 3.5 - reveals a great dependency with suppliers. Around half of orders are satisfied by the most relevant country, and almost 90% of them are satisfied by only 10 of supplying countries. Geography can be a significant indicator not only for working hours, but also for carrier related optimizations, such as route and pickup timings.

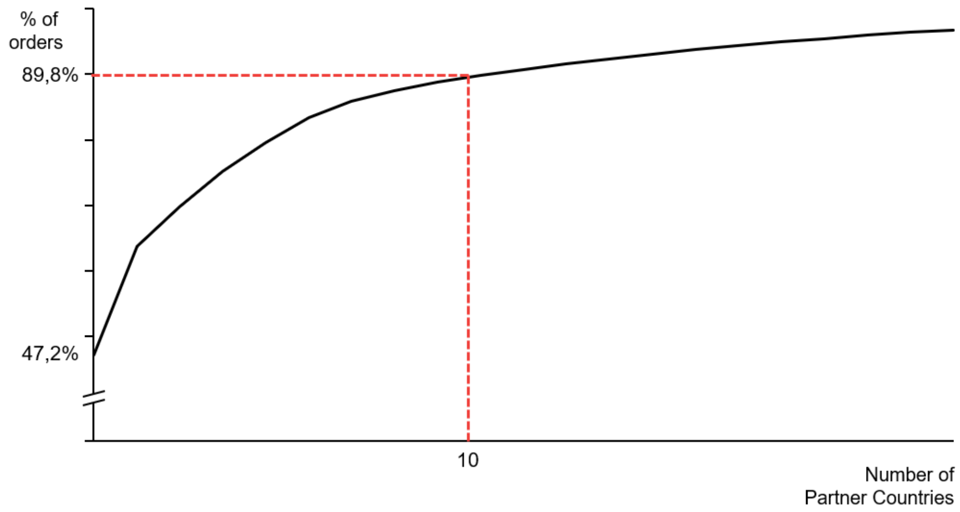

Figure 3.5: Partner Country cumulative order distribution

Daily Pickup

Analyzing Table 3.3, one can observe no differences in *Lead Time* between partners that have *Daily Pickup* and those who schedule *ad hoc* pickups. The low percentage of partners flagged with *Daily Pickup* was not expected and may indicate a lack of update in the system databases. Therefore, these numbers can me misleading and are analyzed critically.

Table 3.3: *Daily Pickup* descriptive statistics

Daily Pickup	Frequency (%)	Mean LT (days)
Yes	38%	4.6
No	62%	4.6

3.2.2 Speed of Sending Variables

After evaluating the overall settings, this section introduces specific variables that influence the time partners take to fulfill an order. One is related to how payments are approved in Step 2 - *Approval Type* - and the others are associated with the order processing of each step - *Order Creation Date* and *Order Processing Date*. *Speed of Sending* does not exceed five days in this data-set, and more than 70% of orders is dispatched in less than two days, as seen in Figure B.2.

Approval Type

As discussed in the Step 2 description, the customer payment can be processed in two possible ways. It can be an automatic process, done by a third-party provider, or it can be a manual process, concluded by *Farfetch*'s internal teams, depending on the uncertainty associated with the order.

Table 3.4 reveals the predominance of automatic fraud validations, with a significant impact on performance. In manual approved orders, partners reveal a 60% delay in overall processing efficiency.

Table 3.4: *Approval Type* descriptive statistics

Approval Type	Frequency (%)	Mean SoS (days)
Automatic	79%	1.3
Manual	21%	2.1

Order Creation Date

As stated previously, the date variables were dissected - in *Hour*, *Day*, *Weekday*, *Month* and *Year* - to allow for better inferences. The *Year* component was abandoned, because the data range has less than one year, so its impact cannot be expressed. From the remaining four, *Day* and *Month* have a similar seasonality interpretation - see Figure C.1 in Appendix C. The main impact is promotions and peak sales seasons, when demand rises and the workload is increased, leading to an operational performance reduction. Finally, *Weekday* presents an evident and continuous impact on *Speed of Sending* performance, as proven by Figure 3.6.

In this Figure one can understand that *Monday* is the weekday with the most processing needs, opposed to the weekend, when some partners are absent of order processing. This can cause partner processing delays, due to backlog accumulated from non-working weekdays. Other *Weekday* influence results from carriers not being available in certain weekdays, limiting pickups. Thus, *Friday* and weekend under-performance was expected. On the other hand, *Monday* through *Thursday* present the optimal combination of backlog minimization and carrier availability maximization, where the *Speed of Sending* is expected to be close to one day. The *Order Creation Hour* analysis will be detailed in section 3.3.

Order Processing Date

Taking the previous date decomposition in consideration, one wonders the impact of step processing dates on partner's *Speed of Sending* performance. Every analysis in this section uses the timezone of the partner, to guarantee actionable results.

Order Processing Day, *Month* and *Weekday* influence have similar interpretations as in *Order Creation*. Moreover, in order to to assess the *Order Processing Hour* impact, the chosen method was to measure, for every step, the time until fulfillment, as evidenced by equation (3.3). This equation reveals how long it takes for the order to get picked by the carrier, depending on each processing step.

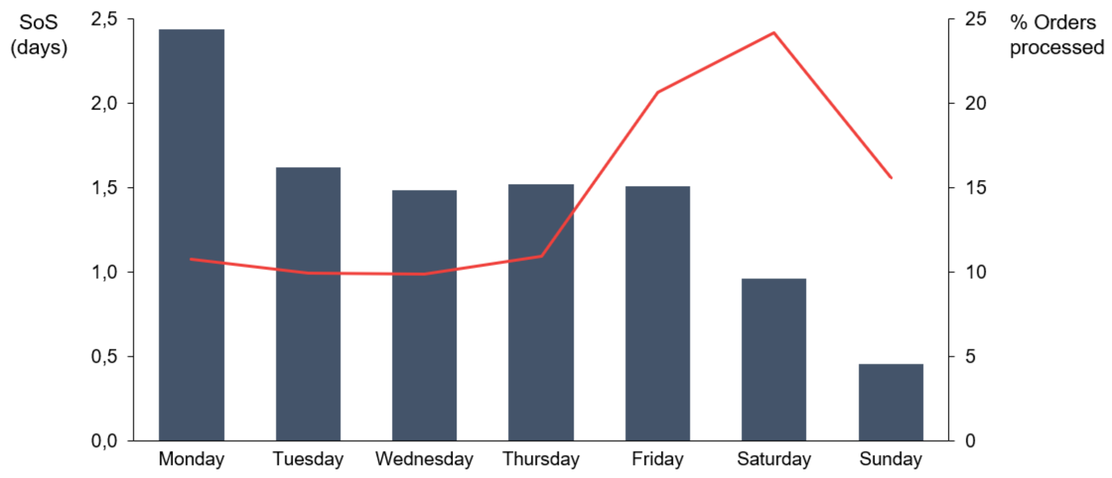

Figure 3.6: *Speed of Sending* weekday seasonality and order processing frequency

$$Time\ Until\ Fulfillment_i = Speed\ of\ Sending - \sum_{j=1}^{i-1} Step_j\ Time \tag{3.3}$$

This analysis was conducted only from Step 1 to Step 4, since Step 5 corresponds to the fulfillment closure. Additionally, only hours containing significant records are considered. Figure 3.7 shows, as an example, the Step 3 influence on time until fulfillment - graphs for Step 1, 2 and 4 are exhibited in Appendix D.

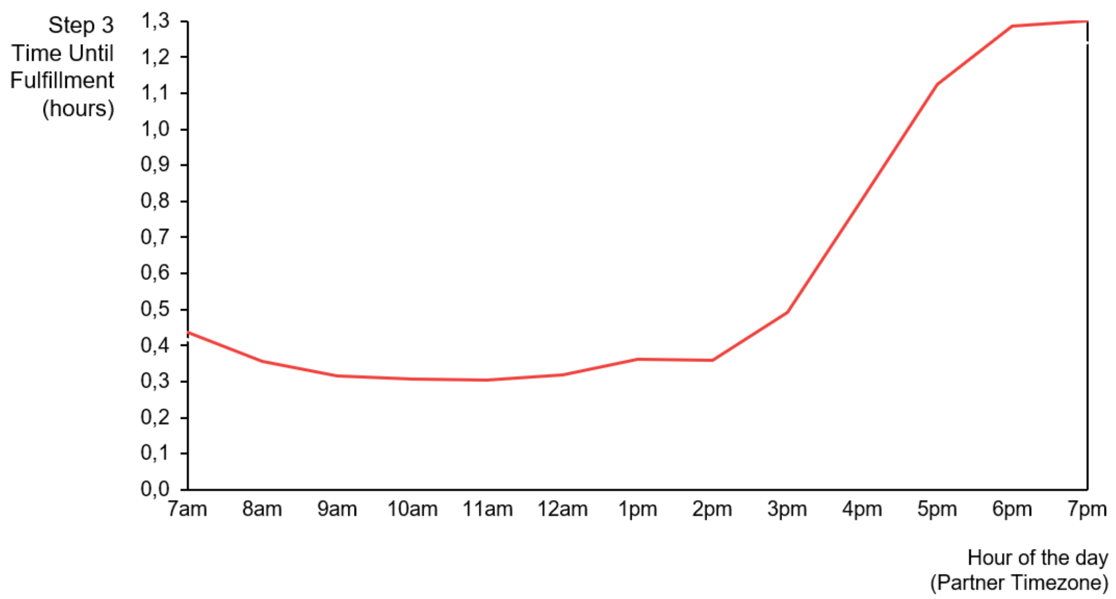

Figure 3.7: Step 3 *Hour* impact on Time Until Fulfillment (hours)

In these graphs, it is important to evaluate the inflexion point, where the fulfillment time starts rapidly escalating. Generally, this points to the daily threshold of order fulfillment, meaning that, on average, there is a time of the day that, for each step, the order will only be shipped in the next day. This is tightly connected with pickup timings that are variable for each partner stock point. For example, if Step 3 is concluded before 3pm, there is a large probability the order is shipped in that day. The thresholds for each step are increasingly later in the day, as one would expect when analyzing dependent and sequential processes.

Step 3 and 4 have similar conclusions, since most of the times these can be done almost simultaneously by the partner. There is no interaction needed between entities, reducing communication friction and such entropies, leading to a smoother process.

3.2.3 Transit Time Variables

Regarding *Transit Time*, the performance can be affected by specific route characteristics - *Customer Country*, *Service Type* and *Crossborder* - or by specific order characteristics - *Order Pickup Date*. Since so many combinations can occur, it is expected that this step will have the greatest contribution on LT variation, as proven by Figure B.1. There, one can observe that order transportation can be concluded either in the same day it is expedited, or it can also take more than one week.

Customer Country

As the curve in Figure 3.8 demonstrates, there is a long tail of countries that contribute with a small fraction of the orders. In fact, the top ten *Customer Countries* are responsible for 70% of placed orders. This impacts order allocation to the partners, and consequently, the route the parcel will travel.

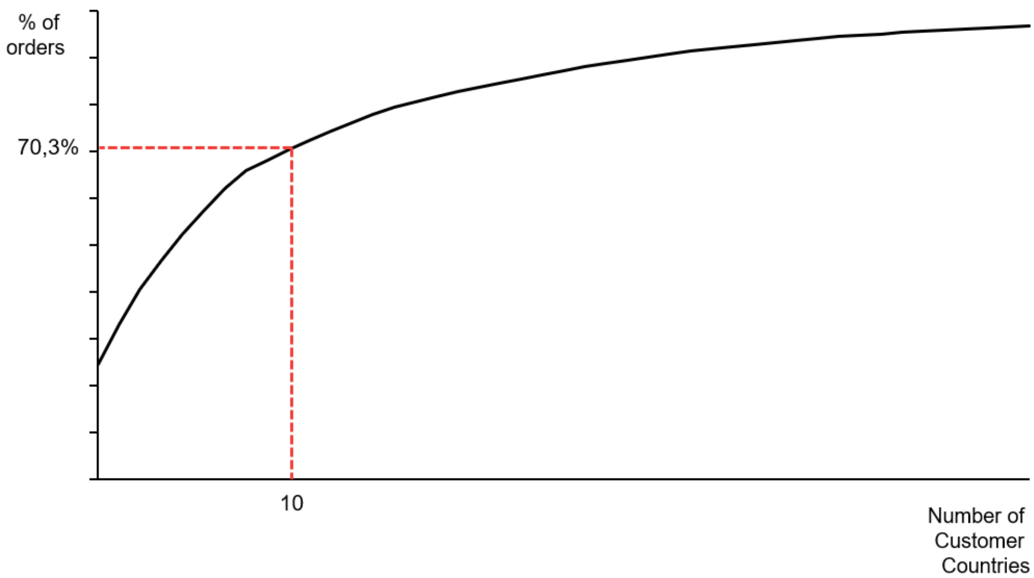

Figure 3.8: Customer Country cumulative order distribution

Crossborder

Depending on the Customer Country and Partner Country combination, the route can be classified in: *Crossborder*, if there is need for a customs clearance; *Domestic*, if it is an order fulfilled within the country it was placed on; *Intra-EU*, if an order that will move across borders, but inside the European Union commercial space. Due to the worldwide combinational presence of stock-points and clients, *Crossborder* routes are the most common, and at the same time, the most time consuming, as evidenced by Table 3.5. This is caused not only by the large distances between the fulfillment points, but also by the need of customs clearance, that, depending on country regulations, can cause considerable delays.

Table 3.5: *Crossborder* descriptive statistics

Crossborder	Frequency (%)	Mean TT (days)
Crossborder	86%	3.4
Domestic	4%	1.6
Intra-EU	10%	1.7

Service Type

When placing an order, every customer is empowered to select the urgency of delivery - either *Express* or *Standard*. For each route, this influences the carrier that performs the parcel transportation. Table 3.6 reveals that most orders are fulfilled through *Express* services, since this is the only applicable service for non-domestic routes. The mean difference of *Transit Time* is apparently not very significant, but a deep-dive for each route discloses different conclusions. In the US-US route there is an uplift of 76% of time efficiency by using the *Express* service - as seen on Figure E.1.

Table 3.6: *Service Type* descriptive statistics

Service Type	Frequency (%)	Mean TT (days)
Express	82%	3.1
Standard	18%	3.5

Order Pickup Date

Once again, time-driven order features should be investigated regarding their impact on *Transit Time*. Thus, it is relevant to evaluate Step 5 information that corresponds to the carrier pickup. *Order Pickup Day* and *Month* have similar seasonalities as in *Order Creation* and *Order Processing*, as the main influence is demand fluctuations. More demand creates order accumulation that can lead to extended waiting periods. *Order Pickup Hour* appears to have a reduced influence since it remains mostly constant for every partner-carrier relationship. *Weekday* heavily impacts the number of pickups, specially because most carriers are not available on weekends. An order picked up on *Friday* will take almost four days to reach the client, whereas one picked up on *Monday* takes on average 2.7 days - Figure 3.9. This relationship is similar to the one presented for *Order Creation Date*, where *Monday* has the greatest amount of activity, and the time performance is worse as processing is done later in the week.

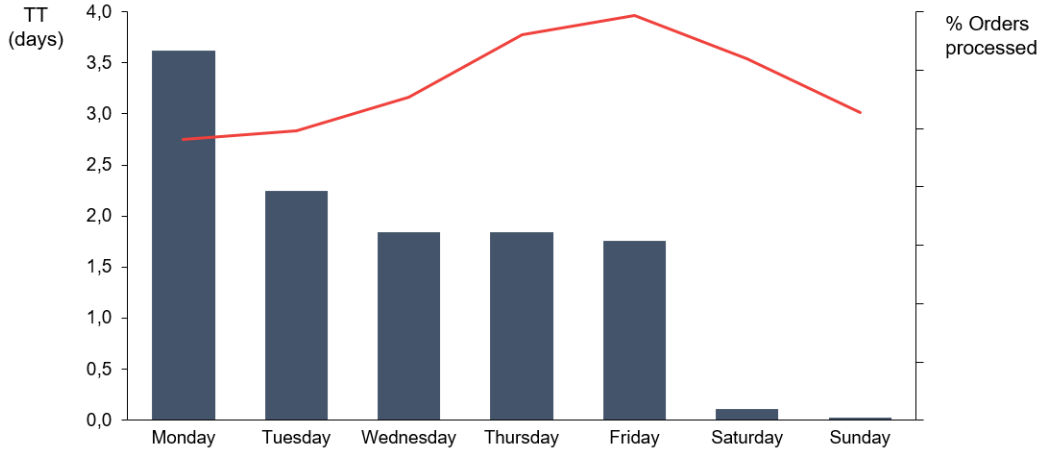

Figure 3.9: Pickup weekday impact on *Transit Time*

3.3 Order Cycle Lead Time Dashboard

Being capable of defining and gathering all the data previously shown facilitates the dashboard development phase. However, firstly, one needs to explicitly clarify the main purposes of this tool. This visual representation allows an end-to-end overview of lead time performance, aggregating knowledge that was previously scattered throughout several data-bases and individual team reports. This way, there is an interactive method to expose and monitor time-related metrics, scrutinized at the most detailed levels available. Additionally, this dashboard is a straightforward technique to originate and test lead time optimization drivers and hypothesis.

Tableau was the software utilized for the dashboard construction. This tool fully integrates with SQL servers, automating the data synchronization updates. Moreover, it provides an intuitive and interactive environment that every user in *Farfetch* has access to. In Appendix H, the various pages of this tool are displayed. All data can be filtered according to the parameters displayed in Table 3.7. The two last parameters, *Season* and *Partner Volume*, were not found in any database, and so had to be computed.

After describing the dynamic parameters, the visualization tool can be detailed. Figure 3.10 shows the main page, where one can have the general overview of the order cycle lead time and its components, given the chosen filters. There is the possibility to evaluate their temporal behaviour, absolute values and distribution. Additionally, there is a breakdown of lead time, where one can assess the contribution of each step to the overall performance. Furthermore, an overview of the *Order Creation Hour* impact on *Speed of Sending* is given, where it is reflected the daily threshold for same day fulfillment. This way, every user can intuitively check each step's absolute and relative impact on the order cycle. Finally, one can check which are the top partners (order volume wise) and how they perform in terms of SoS. This provides a more strategic view, as users are able to target specific partners to evaluate in further detail.

Table 3.7: Parameters available to filter data

Parameter	Values	Description
Time Range	{date interval}	Time range of order creation
Customer Geography	{region list}	Region of customer
	{country list}	Country of customer
Partner Country	{country list}	Country of partner
Partner Name	{name list}	Individual partner identifier
Daily Pickup	Yes	Has scheduled daily pickup
	No	Schedule ad hoc pickups
Season	Peak	Months with peak demand
	Normal	Months with normal demand
Partner Volume	High	Partner fulfills >10% of total orders
	Medium	Partner fulfills >1% and <10% of total orders
	Low	Partner fulfills <1% of total orders

Henceforward, the Order Cycle Dashboard dedicates one page to each one of the steps of the order cycle - see Appendix H. They follow a similar structure, where a first descriptive presentation is done - average step duration over time, weekday influence and hour impact on fulfillment. Then, specific improvement hypothesis for each step can be monitored and tested.

The overview of Step 1 is presented in Figure H.2. One can evaluate how different processing hours and processing volumes affect the fulfillment time. Additionally, it is possible to test different demand timezones. The main purpose is to determine if orders are being assigned to partners that are not in their working hours (9am - 6pm), and to assess the impact on fulfillment efficiency. With the default filters, almost half of orders are placed after the partners are closed. Figure 3.11 shows an example of a screenshot of the hourly demand in the partner's timezone, in one of the most relevant customer regions. It is evident that most orders are allocated to partners that are not within working hours.

Regarding Step 2, the same processing hours and volumes evaluation is done. This step is performed simultaneously with the first one, so it is important to assess what step usually ends faster. Figure H.3 monitors the percentage of times that Step 2 is quicker. This could be caused by the greater automation of this step - which is also being monitored -, but also by the fact that some partners choose not to do Step 1 without having fraud validation from *Farfetch*, since there is a chance they are using resources in an order that will later be invalid. When Step 1 is faster, it usually means that fraud validation is done manually, which translates in processing delays.

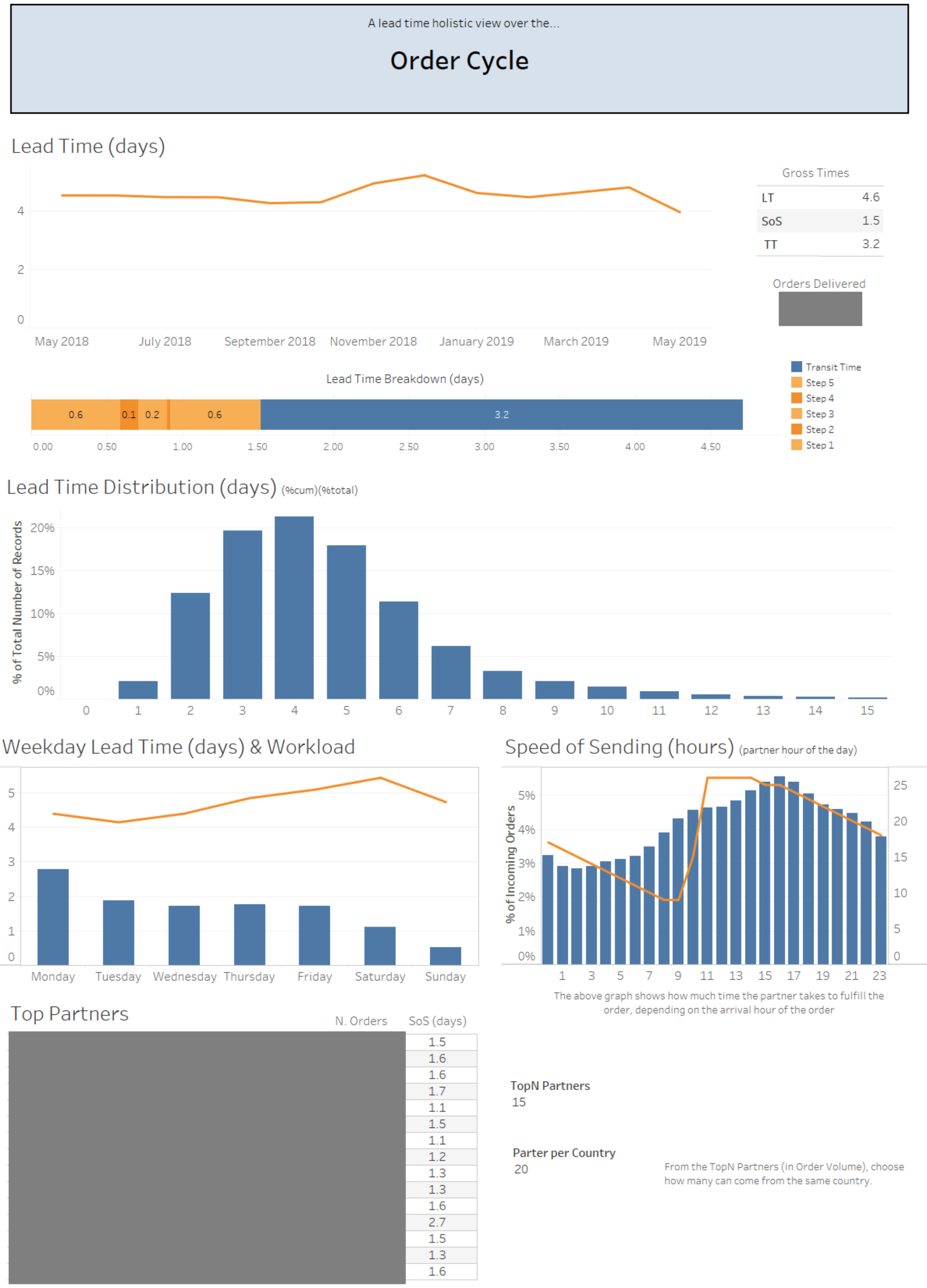

Figure 3.10: Order Cycle overview dashboard page

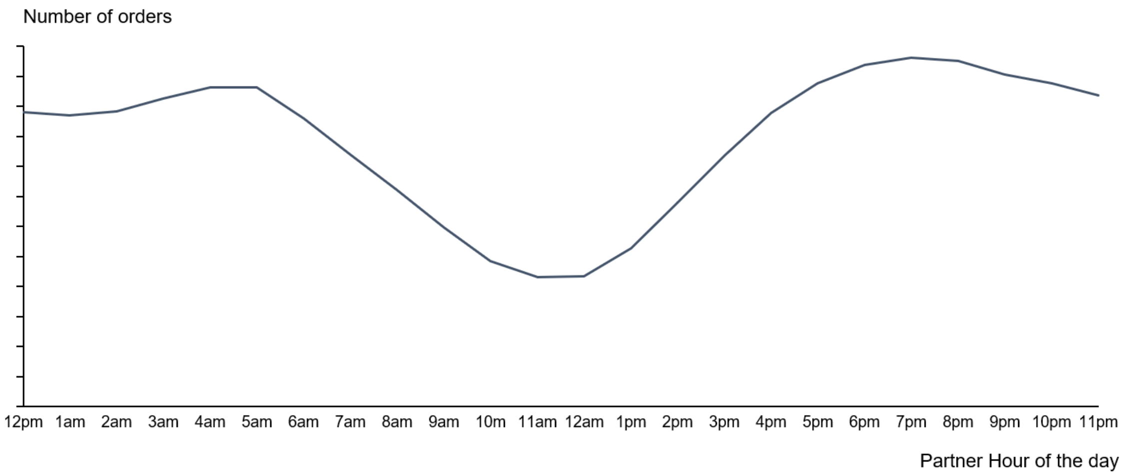

Figure 3.11: Demand by hour of the day (Partner Timezone) of one customer region.

Step 3 has many similarities to Step 4 in terms of efficiency analysis, so these steps are described together. The most interesting part is understanding how partners deal with the end of Step 2. Some are able to rapidly continue to Step 3. Others, due to a break in order responsibility, concentrate their efforts on adjacent tasks, and only return to order fulfillment for the *Farfetch* channel when it is more convenient to them.

Since the partner also has responsibility over Step 4, it usually happens immediately after Step 3. However, there are problems that can occur with the AirWayBill (AWB) due to incorrect shipping information. This way, it is vital to control how many orders cause these problems, and to assess their impact on time efficiencies. AWB problems in the same *Customer Geography* tend to have similar root causes, so having this data breakdown allows the internal teams to act promptly to resolve any issues, specific to a certain area.

Step 5 is important because it sets the boundaries between the processing done by the partner, and the transportation by the carrier. This is a friction point that needs careful investigation. One interesting metric to evaluate is orders that are ready in a certain day, and are not expedited in that same day. This *Failed Pickup* rate can have its root causes on the carrier, by changing the regular pickup hour or by not accounting for the required volume of orders. At the same time, partners can also contribute by not having the orders completely ready as the carrier arrives. The internal teams have no clear indication from the partner that an order is already packed, there is only information that all the conditions to pack are met, so the partner has the responsibility of ensuring completion before pickup time. The optimal goal is for partners to ship every order in the same day it is processed. Figure 3.12 evidences the struggle in pickup coordination faced by one of the most relevant partners. In this example, the partner has the most of its pickups around 2pm, although order preparation follows a somewhat regular pattern during the day. This will cause an increase in orders that are ready, but are not shipped in that day. By not being *In Transit* as soon as possible, the *Lead Time* delay could be greater than one day, specially if the pickup delay pushes the order to be shipped close to the weekend. This analysis can alert partner facing teams to optimize pickup timings, through scheduling later pickups or anticipating order fulfillment.

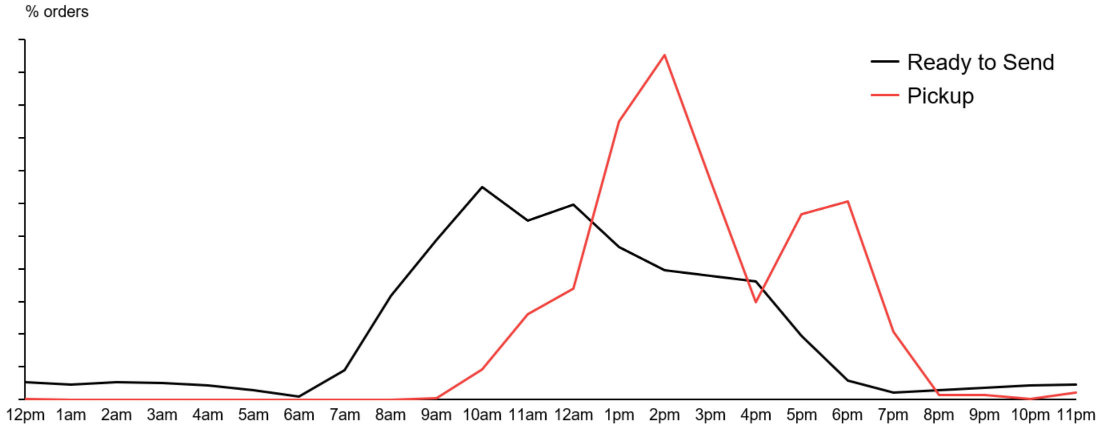

Figure 3.12: Comparison between hour of 'Ready to Send' and hour of 'Pickup'

All analysis so far mostly concerned the partner and *Farfetch*'s internal teams. Figure H.6 is the dashboard page on Step 6 that exclusively monitors carrier performance. One can assess the seasonal trends of *Transit Time*, and the aforementioned weekday, service type and crossborder influences. On top of that, it is possible to investigate the distribution of orders per TT duration. To conclude, the top 20 Routes by order volume are presented. With the default filters, the top 3 routes account for around 20% of order volume. For each route, one can determine the *Transit Time* breakdown in *Time First Mile*, *Time in Customs* and *Time Last Mile*. This can divide the potential delays into two segments: delays regarding travelling distance and delays regarding customs processing. One can strategically analyze important routes and even redefine order allocation and fulfillment accordingly.

After presenting all the explanatory analysis, the basis are set to quantify the *Lead Time* effects of the described variables in the following chapter.

Chapter 4

Lead Time Predictive Model

Building the Order Cycle Dashboard required the creation of an unified and consistent data-set to effectively monitor the lead time performance of order processing. Having this structured data available allows the use of analytical methods to model the operational behaviour of the Order Cycle. This section reflects the creation of a model that is able to predict the *Speed of Sending* and *Transit Time* values, from specific order events. The main goal is to achieve visibility in order execution duration and to develop a tool that enables what-if analysis around the operational dynamics. The Lead Time Predictive Model chapter is divided to reflect data processing, modelling, and results.

4.1 Data Processing

Recapping the methodology followed in section 3.2, data was gathered at the *Boutique Order* level, which means the data-set is a table in which every row represents a different *Boutique Order*, and the columns contain all the different features to evaluate. As not to damage the model's performance, all records containing manual inputs or non-trustworthy sources were removed, and an outlier identification and removal was done to the three main time-related metrics - *Lead Time*, *Speed of Sending* and *Transit Time*. Additionally, variable deconstruction was needed in date-time fields, finalizing the data-set pre-composition with more than 4 million orders.

Moreover, in order to have a close evaluation of partner segmentation regarding backlog control performance, a new variable was introduced - *Pipeline*. This variable tracks, for each order, in that specific partner, the number of additional orders currently being processed. The computing is presented in Figure F.1 of Appendix F. This allows for the assessment of how different partners deal with different levels of workload, and their efficiencies should be a reflection of their fulfillment configurations. Figure 4.1 displays the distribution of Pipeline values. Most orders are processed with less than 100 simultaneous orders, but there is an exponential difference to the most extreme cases. There are partners that endure a pipeline of up to 7 000 synchronous orders.

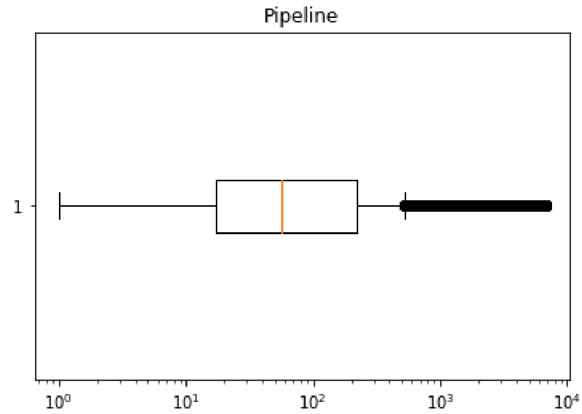

Figure 4.1: Boxplot of number of orders in Pipeline

Speed of Sending and *Transit Time* have significantly independent impacts on overall *Lead Time* performance. Therefore, it is necessary to separately define the variables that impact each one of the components. This will lead to the construction of different models, as to maximize the output explanation through the selected variables. Table 3.1 contains the previously identified data-set variables that can be relevant to model each of the components.

Every described variable expresses a specific attribute of each order. However, not all of them will have the same relevance as predictors of the models. Therefore, it is important to assess the association between those variables, in order to determine if there is redundant information. Avoiding redundant information increases the efficiency of the model, without hurting its explanatory power. Consequently, a correlation analysis was done for the non-categorical variables, using the Pearson method. For categorical variables, the proposed method was the Chi-squared test.

Appendix G presents the final test results. In spite of *Partner Country* revealing a close relationship with *Partner ID*, the decision was to keep both variables, mainly to ease the final model interpretation. Moreover, *Daily Pickup* was discarded due to its connection with *Partner ID*. This was a rather specific attribute that was not adding predictive value to the model. *Customer Country* is not only a good proxy for *Customer Region*, but also manages to contribute with more detailed information, making the geographic *Region* irrelevant. Furthermore, *Crossborder* was eliminated because its influence can be substituted by other route characteristics: *Customer Country* and *Service Type*.

Regarding non-categorical variables, the only dependence was found between the different *Order Step Dates*. This was already expected, since they represent sequential steps. With that in mind, two different *Speed of Sending* models are developed. One that contains the *Order Step Dates*, and other that only contains the *Order Creation Date*. Further explanation regarding this division is given in section 4.2. Table 4.1 shows the final compilation of variables that were considered in the development of the models of *Speed of Sending* and *Transit Time*.

Table 4.1: Final variables considered to model *Speed of Sending* and *Transit Time*

	Variable	Type	Description
General	Partner ID	Categorical	Identification number of each partner
	Partner Country	Categorical	Country of the partner
	Pipeline	Quantitative	How many orders are being processed at a certain time
SoS	Order Step Date	Numeric	Date-time of each step of order processing
	Approval Type	Categorical	Type of fraud validation (Automatic vs Manual)
TT	Order Pickup Date	Numeric	Date-time of pickup
	Customer Country	Categorical	Country of customer
	Service Type	Categorical	Type of carrier service (Express vs Standard)

4.2 Modelling

Using the gradient boosting approach introduced in section 2.3.3, three different models were developed: two regarding *Speed of Sending* and one related to *Transit Time*. The two *Speed of Sending* models are not complementary, yet either one of them completes the *Transit Time* model to predict the final *Lead Time* value. This section is divided to reflect the modelling methodology. First, the different models are explained. Afterwards, the split validation, hyperparameter tuning and evaluation are presented in sections 4.2.1, 4.2.2 and 4.2.3, respectively.

The main difference between the two SoS models resides in the variables used. The first one leaves out the *Order Step Date* variable - model *SoSNoLogs*. The second one applies all the significant variables discussed in the previous section - model *SoSWithLogs*. The goal is to capture two distinct ways of making the prediction. *SoSNoLogs* mainly considers *a priori* information available at the moment the order is placed, whereas *SoSWithLogs* predicts the outcomes using information from each step that is updated throughout order processing. *Order Step Date* is expected to have a preponderant influence on predictions, so this distinction was made to capture in greater detail the impact of the remaining variables. Basically, *SoSNoLogs* is more reliable as an importance evaluator for the various variables, except *Order Step Date*, while *SoSWithLogs* provides more accurate and complete predictions.

For each model, three pools were created, one for each of the three data-sets (training, validation and testing). A pool is a structure that can contain three features: the real output values; the input variable data; and the identification of which variables are categorical. MAE is the measure used as the regression loss function for the models. This loss function quantifies the failure of achieving the desired result, and is minimized by the optimization algorithms, as to converge to the most favourable solution.

4.2.1 Model Split Validation

After setting the significant independent variables for the models, one has to structure the data-set for future validation. As studied in section 2.3.1, the original data-set is divided in three components. The first one, consisting of 60% to 90% of data is the training set. This set is used to train the model, and should represent the majority of possible situations, for the model to have a learning phase as broad as possible. The second and third ones are the validation set and the testing set, that equally divide the remaining 10% to 40% of data. The validation set performs the tuning of the model's predictors. Finally, the testing set uses completely new and unseen data on the model, to test how it applies the assumptions it was trained on. This data-set split is done in a temporal sequence to capture seasonalities and dependencies between observations. Due to time constraints, the *Transit Time* model was used to test the data-set split that was replicated in the other models. This choice was made because this was the model anticipated to have the highest error, and thus needed a careful assessment of the trade-off between training data-set size and overfitting.

Different data-set splits were tested to evaluate the optimal division for the models. For each combination, the performance was defined by how fast was the learning, and how significant was the testing error. The training error (ISE) shows how the model is fitting with the training data, and low values can indicate over-fitting. The testing error (OSE) measures how the mode applies the trained assumptions. Analyzing Table 4.2, the 60-20-20 split is excluded, due to having the highest testing error. From the remaining ones, that have significantly similar testing errors, the 70-15-15 split was chosen because it has the lowest running time and has the smallest gap between training and testing error, meaning that it is less likely to overfit.

Data consistency was guaranteed across the created sets. There are no significant variable differences that are expected to have a negative influence on the model's outcomes. However, the sets are not completely homogeneous because of the demand seasonality impact on operational efficiency.

Table 4.2: Data-split sets performance

Splits (%)	Training Error (hours)	Testing Error (hours)	Running Time (s)
60-20-20	10.1	13.0	3 286
70-15-15	10.2	10.4	2 129
80-10-10	9.0	10.1	4 417
90-5-5	7.8	10.0	23 646

4.2.2 Hyperparameter Tuning

Concentrating even further on model efficiency, it is relevant to define its optimal hyperparameters. This process is called hyperparameter tuning and the expected outcome is the best combination of model configurations to minimize the model's error. The first move is to identify which parameters to optimize. The selected ones are presented in Table 4.3.

Table 4.3: Optimizable model parameters

Parameter	Search Space	Description
Iterations	[1, +∞]	Maximum number of trees that can be built
Learning Rate	[0.01, 0.3]	Used to define the reduction in gradient variation
Depth	[6, 16]	Length of the longest path between root node and leaf node
l2-leaf-reg	[0, 100]	Overfitting regulator

These are the parameters that govern the entire model training, so when evaluating them, one has to consider the trade-off between time consumption and model accuracy. Normally, more accuracy comes at the cost of greater processing times.

The number of iterations sets the limit of trials in the learning process. Generally a large initialization value is given, to increase the chances of better results. However, it is important to note that the marginal gain with each iteration reduces and computing time grows exponentially. The learning rate is the weight given to corrections from new data. As these weights increase, the model tens to converge quicker, but can lead to a sub-optimal solutions (overfitting). Lower learning rates are dependent on more iterations to display better results. Depth controls how much information about the data is captured. Increasing this value will make the model more complex and more likely to overfit. L2-leaf-reg is an important parameter to balance the learning process, as it is a coefficient that operates as a penalty term to the loss function. As this value increases, the probability of underfitting grows, which means it will make the model more conservative.

The approach to define these values was common to the three models. Based on preliminary computational experiments, the number of iterations was set to 10 000 and the learning rate to 0.1. These where the necessary initialization values. Using the *HyperOpt* (Bergstra et al. (2013)) package in *Python*, several tests were done to the models. In each test, a random search for each parameter was performed, within the search space. The error minimization parameters for each model are presented in Table 4.4.

Table 4.4: Optimal model hyperparameters

Parameter	*SoSNoLogs*	*SoSWithLogs*	*Transit Time*
Iterations	1 000	2 000	5 000
Learning Rate	0.29	0.22	0.19
Depth	13	13	14
l2-leaf-reg	58.5	61.9	56.6

The *Transit Time* model has the highest iterations threshold. This indicates that the variables used to explain the TT values lack explanatory power and it should take longer to converge. The learning rate is intrinsically connected to the number of iterations. Lower learning rates need more iterations, leading the models to be more complex. These learning rates, together with high depth values increase the tendency to over explain a limited set of data points. This is mitigated by large regulation values (l2-leaf-reg).

As discussed in Probst et al. (2019), hyperparameter tuning translates in better model accuracy and efficiency. In the developed models of this project, this optimization resulted in a reduction of, on average, 6% in error.

4.2.3 Model Evaluation

Looking at the results in Table 4.5, one can conclude that the three different models appear to have similar behaviours in the training and testing phases. *SoSWithLogs* presents the biggest error gap, either in relative and absolute terms, but at the same time, the magnitude of the error itself is the less significant. Translating the values from days to hours, this model has an expected value of 33 hours, while presenting an MAE of around 2 hours. *SoSNoLogs* has a higher associated error than its SoS predictor counterpart. This model is basing its predictions on a reduced amount of regressors, and so it is more prone to misattribute the values of certain unseen features.

The *Transit Time* model has the highest MAE. This is explained by the greater variability associated with the process this model is trying to predict and its timespan preponderance relative to the overall *Lead Time*. In this model, the OSE is smaller than the ISE, an occurrence that is nothing but unusual. Possible explanations can arise from the data set split composition, where unidentified noise could have passed unnoticed through the data processing phases, but overall this is an indication that the proposed model has a good fit to the presented data. All in all, these models present an estimate of the *Lead Time* with an associated error comprised between 0.9 and 1.3 days.

Table 4.5: Final model evaluation metrics using gradient boosting

	SoSNoLogs		*SoSWithLogs*		*Transit Time*	
Metric	Train	Test	Train	Test	Train	Test
MAE (days)	0.40	0.43	0.02	0.08	0.90	0.86
RMSE (days)	-	0.82	-	0.20	-	1.86
MAPE (%)	-	42%	-	9%	-	26%

As discussed in section 2.3.3, Carvalho (2016) also proposed a modelling methodology to predict the delivery dates upon the customer order placement, using *Farfetch* as a case study. In her work, a conditional inference decision trees model was developed to predict the timespan of each step. When comparing Tables 4.5 and 4.6, it is revealed that the *Speed of Sending* predictions are more robust using the approach in this project. Carvalho (2016) had the goal of predicting the delivery date with *a priori* information, the same logic behind *SoSNoLogs* that performed almost 50% better in terms of MAE. The *Transit Time* model has a slightly superior MAE, but inferior MAPE, meaning the overall transit time prediction values are superior in the data-set used in this project. These comparisons should be interpreter carefully, because the data collection and preparation was different for both models. Nevertheless, the gradient boosting approach brought apparent improvements.

Table 4.6: Best model results from Carvalho (2016)

Metric	Speed of Sending	Transit Time
MAE (days)	0.8	0.7
MAPE (%)	-	30%

4.3 Results

After the construction and evaluation of the models, it is necessary to assess its outputs. Thus, this section is dedicated to identifying and analyzing the main drivers of variability in each one of three models. Section 4.3.1 for model *SoSNoLogs*, section 4.3.2 for model *SoSWithLogs* and section 4.3.3 for model *TransitTime*. Each section is supported estimations of relative and absolute importance of features to be done - *Feature Importances* and *SHAP Values*.

SHAP Values is a technique that explains individual impacts of each feature. Basically, it evaluates the importance of each value of the features, in comparison to its baseline values - Wild (2018). Figure F.2 in Appendix F demonstrates an example, where the base value for the SoS prediction was around 26 hours, but the predicted output value was 36 hours. This difference is given by a combination of influences from the explanatory variables. The attributes in the variables in red are increasing the SoS prediction (in hours) and the blue ones are contributing for a reduction in SoS. For example, *Manual Approval Type* is causing a significant delay in this order, whereas *Monday* as the *Order Creation Weekday* is leading to lower SoS forecasts.

4.3.1 *Speed of Sending - No Logs*

This is the *Speed of Sending* model that disregards *Order Processing Dates*. As seen in Table 4.7, the variables represent the information existent at the moment the order is placed. The main evident driver of *Speed of Sending* is the *Weekday* of *Order Creation*, with a relative importance of 58%. Remembering Figure 3.6, from *Monday* until *Thursday*, orders are typically dispatched by partners in around one day, although these are the days with greater order processing frequency. From *Friday* until *Sunday*, on average, the SoS doubles.

Order Creation Hour appears as the second most important feature. Consulting Figure 4.2, its influence on *Speed of Sending* is not linear, but one can conclude that has the order is placed later in the day, its fulfillment time increases. This could be caused by two reasons. First, orders may be allocated to partners that are not within working hours - monitored in Figure 3.11. Second, orders that are placed earlier are naturally going to be processed first, since most partners follow a First-In-First-Out (FIFO) policy.

Table 4.7: SoSNoLogs feature importances

Feature	Importance
OrderCreation_Weekday	58%
OrderCreation_Hour	16%
SiteID	8%
Pipeline	5%
OrderCreation_Month	4%
Site Country	3%
OrderCreation_Day	3%
Approval Type	3%

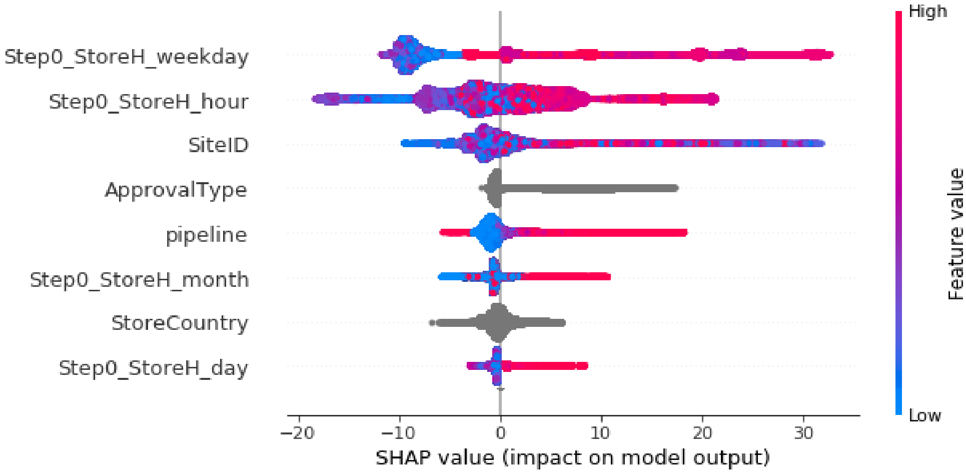

Figure 4.2: SoSNoLogs SHAP Values

In this model, *Site ID* is confirming the premise that different partners have distinct processing configurations. Although its relative impact is not large, Figure 4.2 reveals that the absolute impact is almost as important as the *Weekday*.

Each one of the rest of the features presents less that 5% of relative importance. *Order Creation Month* and *Day* are a proxy for demand seasonality impacts. *Store Country* presents a well distributed impact, and its contained results may be caused by the close correlation with *Site ID*.

Approval Type and *Pipeline* disclose significant absolute importances, despite its relative ones. The positive impact of automation in fraud validation was already assessed in section 3.2.2. *Pipeline* presents curious results. High pipeline values can both have a positive or a negative impact on *Speed of Sending*. This is explained by the dependence of this variable with *Site ID*. Some partners are used to dealing with high volumes of orders, and are not affected by an amount of order variation that for other partners would be problematic.

4.3.2 *Speed of Sending - With Logs*

In this model, all processing information was considered, performing an evaluation over a total of 28 features. For interpretation purposes, only the top ten are presented in Table 4.8. *Order Creation Weekday* and *Hour* are once again among the most relevant variable, but its relative preponderance is less significant, when comparing with the model in section 4.3.1. *Order Pickup Weekday* and *Hour* follow as the next most important features. Together with the *Order Creation* information, these fields determine the timespan of the SoS, justifying its high prediction value. This premise is validated by Figure 4.3, where it is possible to understand the absolute impact of these features. However, the SHAP values do not provide clear indication of the linearity of their relationship with the *Speed of Sending* values.

The relative impact of the remaining variables has a low significance. However, one can note the sequential importance of the fulfillment steps in *Speed of Sending*. Additionally, when considering these impacts, one has mainly to evaluate the *Hour* in which the steps are made. This way, it can be concluded that the time *Farfetch* concludes Step 2 (Approve Payment) and the partner prints the shipping label (Step 4) are contributing more than the partner's *Hour* of Step 1 and Step 3. This relative importance exists on one hand because Step 2 is a processing milestone that signals the partner that it can continue with fulfillment. On the other hand, Step 4 is an indication of when the order is ready to be collected by the partner. Consequently, the *Hour* of these two steps is a determinant factor on whether the order is shipped in the pickup of that day.

It is important to note that in the top 10 most important features listed in Table 4.8, only *Site ID* is not related with order processing dates. The impact of *Pipeline*, *Approval Type* and *Site Country* is substituted by the absolute times of the processing steps they influence. This leads to another relevant conclusion. This model is affected by the correlation between processing steps, due to their sequential nature. Hence, the main potential of this analysis lies with the evaluation of relative importance between fulfillment steps.

Table 4.8: SoSWithLogs top 10 feature importances

Feature	Importance
OrderCreation_Weekday	27%
OrderPickUp_Weekday	20%
OrderPickUp_Hour	14%
OrderCreation_Hour	13%
OrderStep4_Hour	6%
OrderStep2_Hour	6%
SiteID	4%
OrderStep2_Weekday	2%
OrderStep3_Hour	2%
OrderStep1_Hour	2%

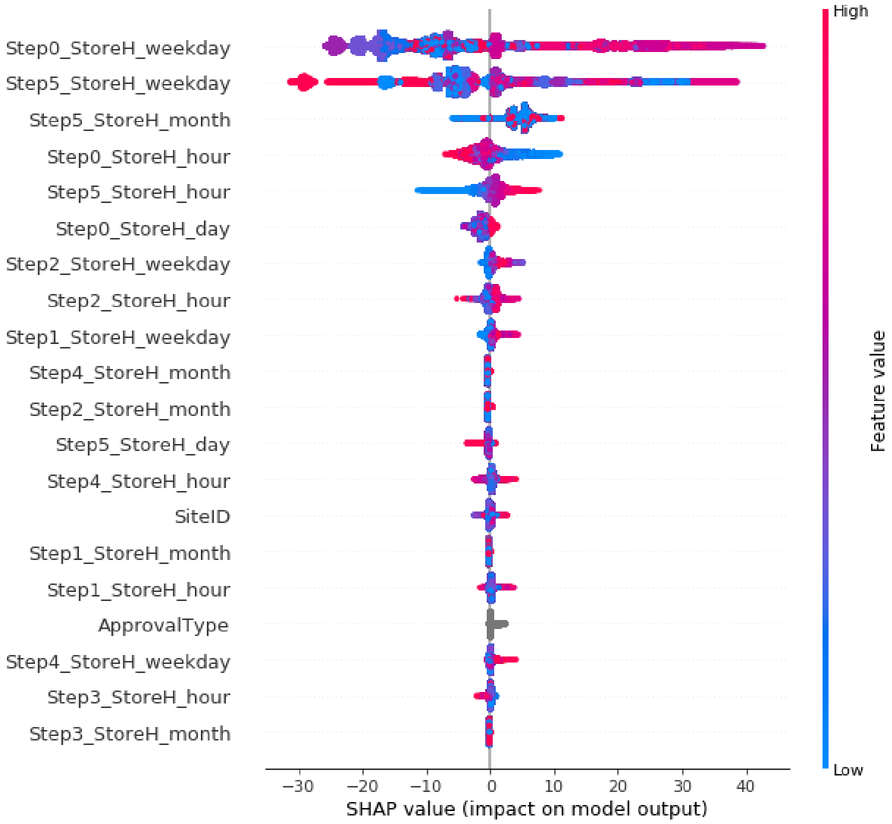

Figure 4.3: SoSWithLogs SHAP Values

4.3.3 *Transit Time*

Table 4.9 displays the relative importance of the *Transit Times* features. Analyzing that table, one can differentiate between two kinds of features that impact *Transit Time*. On the one hand there are route specific features, such as *Customer Country*, *Store Country* and *Service Type*, that determine the distance the parcel will travel and the customs clearance needed to import the item. On the other hand, there are *Order Pickup Date* specific features that define when an order is picked up at the partner. Especial relevancy is given to *Weekday*, whereas *Month*, *Day* and *Hour* present decreased importance. The relative dominance of *Weekday* is due to carriers' limited weekend processing frequency. For example, an order that is shipped on Friday will, on average, have a one day delay when compared to orders shipped on Tuesday - as seen in Figure 3.9. Figure 4.4 corroborates this explanation, as high values for the *Weekday* feature are related with an increase with the overall Transit Time.

Site ID has a relevant importance and it is not straightforward to interpret. The most viable hypothesis is that the model is using this feature as a proxy to detail different *Transit Times* within the same *Partner Country*, being used like a *Partner City* variable.

Pipeline has an almost non-existent significance, which insinuates that its impact is mostly on partner performance that only indirectly influences *Transit Time*. This indirect influence is transmitted through the capacity of each partner to have the order ready to be picked up as soon as possible.

Table 4.9: Transit Time feature importances

Feature	Importance
Customer Country	27%
OrderPickUp_Weekday	24%
Store Country	13%
SiteID	13%
Service Type	10%
OrderPickUp_Month	5%
OrderPickUp_Day	4%
OrderPickUp_Hour	3%
Pipeline	1%

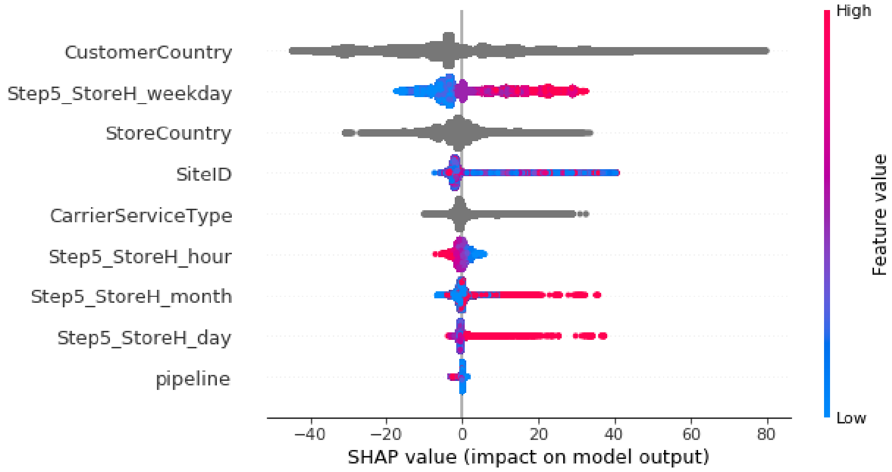

Figure 4.4: Transit Time SHAP Values

Chapter 5

Improvement Opportunities

The previous chapters presented a deep explanatory analysis over *Farfetch*'s order cycle, and its prominent lead time drivers. Over this journey, several insights were gathered from each of the steps. In this chapter, all the aggregated knowledge is applied in identifying which opportunities throughout the order cycle have the most potential. Complementary to its identification, the Lead Time models developed in Chapter 4 enables the quantification of these scenarios.

Looking at the predictive features found in the previous chapter, it is not clear how the most influencing ones could be manipulated in order to achieve more efficient processes. For example, in *Speed of Sending*, *Order Creation Weekday* and *Order Creation Hour* appear as the most relevant variables. However, one cannot explicitly control the demand generation and distribution, which makes any recommendation regarding these variables not actionable and of little interest. This way, after evaluating the preliminary analysis of Chapter 3 and the feature importances of Chapter 4.2, three main actions to refine Lead Time performance are proposed and displayed in Table 5.1. These are the opportunities that are influenced by variables that can be manipulated and present a significant impact.

Table 5.1: Main Improvement Opportunities

Opportunity	Main Stakeholder
Approval Automation	Farfetch
Pickup Alignment	Carrier/Partner
Domestic Routing	Farfetch

Regarding *Speed of Sending*, the most straightforward influencing factor of efficiency is *Approval Type*. Despite not showing as a preponderant predictive feature in the models, it is perceived as an impactful indirect factor in partner's operations, as it is an external requirement that partners have in order to continue their fulfillment. It is of overall interest that *Fraud Validation* occurs smoothly and is as automated as possible. Additionally, the automation level is an internally controlled parameter that can be adjusted within certain thresholds.

Another opportunity arises when studying pickups. This is a process that requires alignment between both partners and the carriers, and it was proven to be a friction point in the order cycle flow. *Pickup Alignment* is guaranteeing that partners are synchronizing their processing cut-off with the arrival of the carrier, maximizing the orders that are expedited in each day. Lastly, it is proposed an evaluation of the impact of having more domestic order allocation. Having supply closer to customers enables shorter *Transit Time* by diminishing distance between partner and customer and avoiding the need for customs clearance, the two critical factors in Step 6 performance.

5.1 Approval Automation

The order *Approval Type* is *Farfetch*'s main touch-point in the order cycle. The inherent premise is that a fast and automated fraud validation enables partners to proceed more smoothly with order fulfillment, reducing SoS. To support this claim, a simple methodology using the *Speed of Sending* models was done. The goal is to feed the model with different testing data-sets, that only differed in the amount of orders that were automatically approved. This way, any variation in the output is generated by the proposed manipulation. The *SoSNoLogs* model was used, in spite of its inferior accuracy in predictions. *SoSWithLogs* was discarded because it considers the dates of each step. Thus, when changing the *Approval Type*, one would have to adjust the following step dates to reflect this manipulation. However, this procedure is not straightforward to replicate.

This way, 4 testing data-sets were created, each with a different *Approval Automation* percentage, and were fed to the model. Table 5.2 reflects the outputs of the predictions. It is clear that incrementing the automation levels has a positive impact on reducing partner fulfillment time. This means that minimizing the time *Farfetch* intervenes in order processing causes a more continuous process on the partner side, maximizing efficiency. Currently, the process stands around 85% automation mark. This means that there is still an estimate of at least 8% worth of SoS improvement that can be achieved by increasing the approval automation to 95%.

Table 5.2: Variation of Speed of Sending (days) with Approval Automation (%)

Approval Automation (%)	SoS predicted (days)
80	1.29
85	1.24
90	1.19
95	1.13

In practical terms, improving the automation depends on the quality and integration of fraud providers and the internally developed tools. As this is close control of *Farfetch*, the goals set in the previous scenarios are attainable. However, it is necessary to take into account that the increase in automation must be achieved sustainably, minimizing the risks of false positive evaluations - rejecting a legitimate order or accepting a fraudulent one.

5.2 Pickup Alignment

The model results in chapter 4 appoint the *Order Pickup Date* as a feature with high relevancy in predicting the outcome of both *Speed of Sending* and *Transit Time*. Special importance is given to the *Weekday* and *Hour* of pickups. For most partners, the pickups are performed regularly on a daily basis, which invalidates trying to directly adjust pickup *Weekday* to obtain a better performance. However, pickup *Hour* is a specific characteristic that can have evidence different values according to each stock-point. The intrinsic hypothesis is that partners can maximize the orders they process and ship if the pickup is done later in the day. This would increase the number of orders that are dispatched within a daily processing cycle, reducing *Speed of Sending*. Additionally, since most orders are processed early in the week, this can have a waterfall consequence of anticipating the *Order Pickup Weekday*, decreasing *Transit Time*. To test this theory's impact, one has to alter the pickup data. A new testing data-set was built on the assumption that every time Step 4 was performed before a pre-determined cut-off hour, the pickup was performed in the same day, to simulate the ideal processing scenario. The *SoSWithLogs* model was used.

Results for a 5pm cut-off are given in Table 5.3. As expected, with the reduction of failed pickups, more orders were dispatched within the daily order fulfillment cycle, leading to a reduction of 23% in *Speed of Sending*. This way, the average would be around one day, closing the gap to same day fulfillment - dispatching an order in the same day it is placed. Regarding *Transit Time*, the expected impact was not reproduced. Although the more favourable pickup weekdays had more orders, its *Transit Time* impact was balanced by adjacent drivers, such as increase in less beneficial routes and weekdays.

Table 5.3: LT impact of Pickup Alignment

Pickup Alignment	SoS predicted (days)	TT predicted (days)
No	1.3	2.6
Yes	1.0	2.6
Variation	-23%	0%

To implement these procedures, *Farfetch* has to be in close relationship with both the carrier and the partner. On one hand, partners need to be convinced to restructure and standardize their fulfillment timings. On the other hand, carriers might need to adjust their own picking routes, and guarantee that postponing the pickup hour does not interfere with subsequent legs of transportation (e.g. catching an international flight in the same day). As several stakeholders are in play, executing this hypothesis requires precise coordination among every party involved.

5.3 Domestic Routing

The results regarding the *Transit Time* model reveal that the most relevant driver in this *Lead Time* section are the route characteristics. This includes *Customer Country*, *Store Country* and *Service*

Type. As analyzed in Chapter 3, *Express* services lead to a shorter *Lead Time*. Additionally, the relative geography of partners and customers influences not only the distance travelled by the parcel, but also the necessity of customs clearance. In absolute measures, these two conditions are the ones with the most *Lead Time* impact. In an ideal scenario, the customer would be served from his own location, minimizing the distance *In Transit* and eliminating the need for customs clearance. This way, the goal is to evaluate the impact of having a closer supplying partner in each of the top three *Customer Countries* in terms of order volume. These were the chosen countries as they are the common destination of the top five routes, making up 41% of total orders. A test data-set was generated, that attributed the same value to *Partner Country* and *Customer Country* (if that *Customer Country* belongs to the top three). The assumption behind this scenario is that the *Speed of Sending* for the orders would remain constant with the new allocation.

Running the *Transit Time* model delivered the results presented in Table 5.4. It is clear that most of the orders are fulfilled from border crossing routes, that suffer significant improvements with the domestic routing strategy. Managing to shorten the gap between supply and demand is a driver of *Transit Time* efficiency, specially in routes that have *Customer Countries* requiring thorough customs clearance (Routes 1 and 3). Avoiding this bureaucratic process can reproduce improvements of up to 30% in *Transit Time*.

Table 5.4: Domestic Routing impact

		Before	After	Variation
Route 1	TT (days)	1.85	1.29	-30%
	%Domestic Orders	0.94%	100%	
Route 2	TT (days)	2.77	2.59	-6%
	% Domestic Orders	10.62%	100%	
Route 3	TT (days)	3.45	2.53	-27%
	% Domestic Orders	0.03%	100%	

Guaranteeing this type of stock availability requires the expansion of the partner network to numbers and locations that are not in line with what is the present situation. Although this scenario is not attainable in the short term, it could be an indication of a future growth strategy, if domestic partners guarantee strategic routes (e.g. are not located in remote locations).

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This project investigated the *Lead Time* repercussions of a multi-step and multi-participant order cycle, set in a e-commerce environment. The main goals were to provide an unified overview of the process that enhanced its monitoring and to develop tools that support strategic decisions. This unprecedented control over the order cycle allows for more proactive actions towards inefficiencies, instead of the reactive approach previously in place.

The first phase was to consolidate all the segregated knowledge regarding all the order processing steps. The subsequent exploratory analysis showed each fulfillment step's relevancy when assessing *Lead Time*. This evaluation reflects their specific operational characteristics, meaning it is possible to quantify different activities and their interactions. Furthermore, all activities' impact were attributed to pre-identified variables.

It is inferred that *Speed of Sending* is mainly influenced by the *Order Creation Weekday* and *Hour*, *Approval Type* and the partner's processing configuration, proxied by variables *Site ID* and *Step Processing Hour*. Regarding *Transit Time*, the main impacts arise from route length, customs verification, *Pickup Weekday* and *Service Type*. Furthermore, it was observed that 35% of orders are fulfilled from the same nine routes. This supply and demand dependency leads to the conclusion that one has to be have a strategical focus when approaching any improvement decisions. This way, one of the proposed scenarios consisted in manipulating the order allocation to increase the number of domestic routes. This strategy can reduce the *Transit Time* by 30%, in routes where customs clearance was previously needed.

Furthermore, it is inferred that delays can be caused by different process stakeholders. Partners are supported and encouraged to have an optimized processing configuration and efficient backlog control. For example, managing an efficient pickup coordination with the carrier is of great importance as to maximize the number of orders dispatched in a daily processing cycle, which has a predictive reduction of 23% in overall *Speed of Sending*. From *Farfetch*'s side, the main objective is to minimize disturbance in partner order processing, achieved by increasing the automation in the only internal order touch-point - fraud validation. A 10% increment in the number of orders automatically processed has a predictive reduction of 8% in overall *Speed of Sending*. To complement the previous results, it is important to refer that the processing weekday seasonality plays an important and difficult to exploit role in determining the *Lead Time* outcome, so it is crucial to evaluate the indirect impacts of every proposed opportunity in these patterns.

All these insights were gathered from the analytical tools developed in the course of this thesis - a monitoring dashboard, and a compilation of predictive models based on gradient boosting. A key factor in their continuous scalability is the constant update of both tools to better define requirements proposed by stakeholders, to enable ever increasing order cycle understanding.

In conclusion, the *Lead Time* patterns identified in this analysis can bring a significant upside to the operational performance of the order cycle. Additionally, it is important to assess their implications on eventual implementations, as it is crucial to align all the main involved stakeholders to iteratively define the new execution procedures.

6.2 Future Work

Along the development of this project, some limitations of the proposed approach were identified, and are interesting to be tackled in further investigations.

Firstly, the gathered information will be enhanced by integrating data from returned orders, that were not considered due to time implications. Additionally, all the analysis were only centered around *Lead Time*, but it should also be relevant to assess the impact of the improvement opportunities in other metrics, such as cost (impact on shipping costs and duties) or customer experience (amount of stock-outs and contacts to customer service). An obvious next step is joining this research with the quantification of *Lead Time* gains in terms of revenue. Establishing a faster order cycle may have short-term and direct costs, but it also drives customer retention, and therefore increasing the customer lifetime value and generating long-term benefits.

With the continuous maintenance of the proposed tools, there are further opportunities to explore other variable implications on *Lead Time*, leading to the development of new projects based on the work presented in this thesis. Some examples might be the evaluation of new routes or of new carriers on *Transit Time* performance, or even the assessment of new partner processing configurations on *Speed of Sending* execution.

Bibliography

Akter, S. and Wamba, S. F. (2016). Big data analytics in e-commerce: a systematic review and agenda for future research. *Electronic Markets*, 26(2):173–194.

Anscombe, F. J. (1960). Rejection of outliers. *Technometrics*, 2(2):123–146.

Arienti, P. (2019). Global powers of luxury goods. https://www2.deloitte.com/content/dam/Deloitte/ar/Documents/Consumer_and_Industrial_Products/Global-Powers-of-Luxury-Goods-abril-2019.pdf. (Accessed in 27/02/2019).

Berg, A. and Amed, I. (2019). The state of fashion. www.mckinsey.com/~/media/McKinsey/Industries/Retail/Our%20Insights/The%20State%20of%20Fashion%202019%20A%20year%20of%20awakening/The-State-of-Fashion-2019-final.ashx. (Accessed in 27/02/2019).

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Bergstra, J., Yamins, D., and Cox, D. D. (2013). Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in science conference*, pages 13–20. Citeseer.

Carvalho, I. (2016). Estimating delivery dates in fashion e-tail. Master's thesis, Faculdade de Engenharia da Universidade do Porto.

Chen, H., Chiang, R. H., and Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS quarterly*, 36(4).

Chepenko, D. (2013). Introduction to gradient boosting on decision trees with catboost. https://towardsdatascience.com/introduction-to-gradient-boosting-on-decision-trees-with-catboost-d511a9ccbd14. (Accessed in 04/04/2019).

Djelic, M.-L. and Ainamo, A. (1999). The coevolution of new organizational forms in the fashion industry: A historical and comparative study of france, italy, and the united states. *Organization science*, 10(5):622–637.

Dorogush, A. V., Ershov, V., and Gulin, A. (2018). Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*.

Ershov, V. (2018). Catboost enables fast gradient boosting on decision trees using GPUs. https://devblogs.nvidia.com/catboost-fast-gradient-boosting-decision-trees/. (Accessed in 04/04/2019).

Friedman, J. H. (2006). Recent advances in predictive (machine) learning. *Journal of classification*, 23(2):175–197.

Galarnyk, M. (2018). Understanding boxplots. `https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51`. (Accessed in 25/03/2019).

Gantz, J. and Reinsel, D. (2011). Extracting value from chaos. *IDC iview*, 1142(2011):1–12.

Goel, E., Abhilasha, E., Goel, E., and Abhilasha, E. (2017). Random forest: A review. *International Journal of Advanced Research in Computer Science and Software Engineering*, 7(1).

Hawkins, D. M. (1980). *Identification of outliers*, volume 11. Springer.

Heim, G. R. and Sinha, K. K. (2001). Operational drivers of customer loyalty in electronic retailing: An empirical analysis of electronic food retailers. *Manufacturing & Service Operations Management*, 3(3):264–271.

Khan, M. and Khan, S. S. (2011). Data and information visualization methods, and interactive mechanisms: A survey. *International Journal of Computer Applications*, 34(1):1–14.

Leys, C., Ley, C., Klein, O., Bernard, P., and Licata, L. (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764–766.

Li, G., Li, G., and Kambele, Z. (2012). Luxury fashion brand consumers in china: Perceived value, fashion lifestyle, and willingness to pay. *Journal of Business Research*, 65(10):1516–1522.

Marchessou, S., Achille, A., and Remy, N. (2018). Enhance the customer experience and transform your business to survive and prosper in the luxury digital era. `https://www.mckinsey.com/~/media/mckinsey/industries/retail/our%20insights/luxury%20in%20the%20age%20of%20digital%20darwinism/the-age-of-digital-darwinism.ashx`. (Accessed in 01/03/2019).

McKinney, W. (2011). Pandas: a foundational Python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, 14.

Okonkwo, U. (2010). *Luxury Online: Style, Strategies, Systems*. Palgrave Macmillan.

Osborne, J. W. and Overbay, A. (2004). The power of outliers (and why researchers should always check for them). *Practical assessment, research & evaluation*, 9(6):1–12.

Otaris (2011). Data analysis, modeling and reporting. `https://www.otaris.de/gb/datenanalysen-modellbildung-reporting/`. (Accessed in 25/03/2019).

Patel, S. (2018). Data science essentials: Why train-validation-test data? `https://medium.com/datadriveninvestor/data-science-essentials-why-train-validation-test-data-b7f7d472dc1f`. (Accessed in 25/03/2019).

Probst, P., Wright, M. N., and Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3).

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). Catboost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, pages 6638–6648.

Qiu, J., Wu, Q., Ding, G., Xu, Y., and Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 2016(1):67.

Sheng, T. and Liu, C. (2010). An empirical study on the effect of e-service quality on online customer satisfaction and loyalty. *Nankai business review international*, 1(3):273–283.

Singh, H. (2018a). Understanding gradient boosting machines. `https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab`. (Accessed in 04/04/2019).

Singh, S. (2018b). Understanding the bias-variance tradeoff. `https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229`. (Accessed in 25/03/2019).

Socol, S. (2018). How the luxury industry is overcoming its eCommerce problem. `https://www.yotpo.com/blog/how-the-luxury-industry-is-overcoming-its-ecommerce-problem/`. (Accessed in 01/03/2019).

Tauriello, F., Abbafati, L., and Festa, A. (2016). Disruptive innovation in luxury e-commerce: The case of farfetch. Master's thesis, Department of Business and Management, Libera Università Internazionale degli Studi Sociali.

Tranmer, M. and Elliot, M. (2008). Multiple linear regression. *The Cathie Marsh Centre for Census and Survey Research (CCSR)*, 5:30–35.

Walfish, S. (2006). A review of statistical outlier methods. *Pharmaceutical technology*, 30(11):82.

Wang, L., Wang, G., and Alexander, C. A. (2015). Big data and visualization: Methods, challenges and technology progress. *Digital Technologies*, 1(1):33–38.

Wild, C. M. (2018). One feature attribution method to (supposedly) rule them all: Shapley values. `https://towardsdatascience.com/one-feature-attribution-method-to-supposedly-rule-them-all-shapley-values-f3e04534983d`. (Accessed in 15/04/2019).

Wirth, R. and Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, pages 29–39. Citeseer.

Appendix A

Outlier Removal code

```
def interquartilrange_outlier():
    global data

    outlier_fields = ['LeadTime_hours', 'TransitTime_hours', 'SpeedofSending_hours']

    for i in outlier_fields:
        Q1 = data[i].quantile(0.25)
        Q3 = data[i].quantile(0.75)
        IQR = Q3 - Q1

    for i in outlier_fields:
        data = data[~((data[i] < (Q1 - 1.5 * IQR))
                    | (data[i] > (Q3 + 1.5 * IQR))).any(axis=1)]

    print('Outliers removed successfully')
```

Figure A.1: Outlier Removal Python code

Appendix B

Speed of Sending and Transit Time frequency distribution (days)

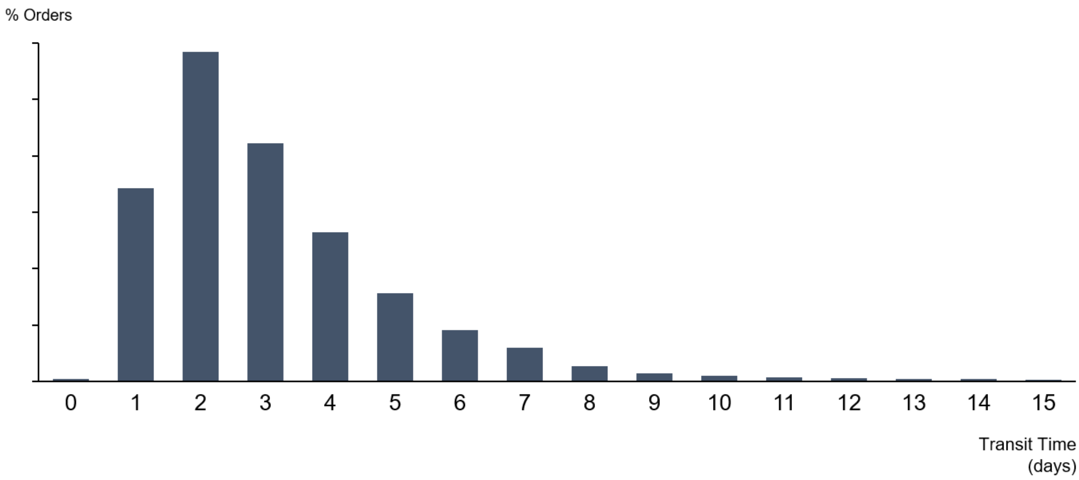

Figure B.1: Transit Time frequency distribution (days)

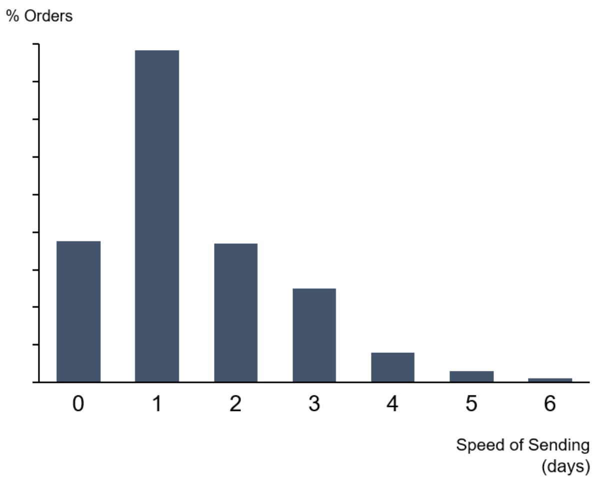

Figure B.2: Speed of Sending frequency distribution (days)

Appendix C

Lead Time seasonality

Figure C.1: Lead Time seasonality in data-set time range

Appendix D

Step Time Until Fulfillment

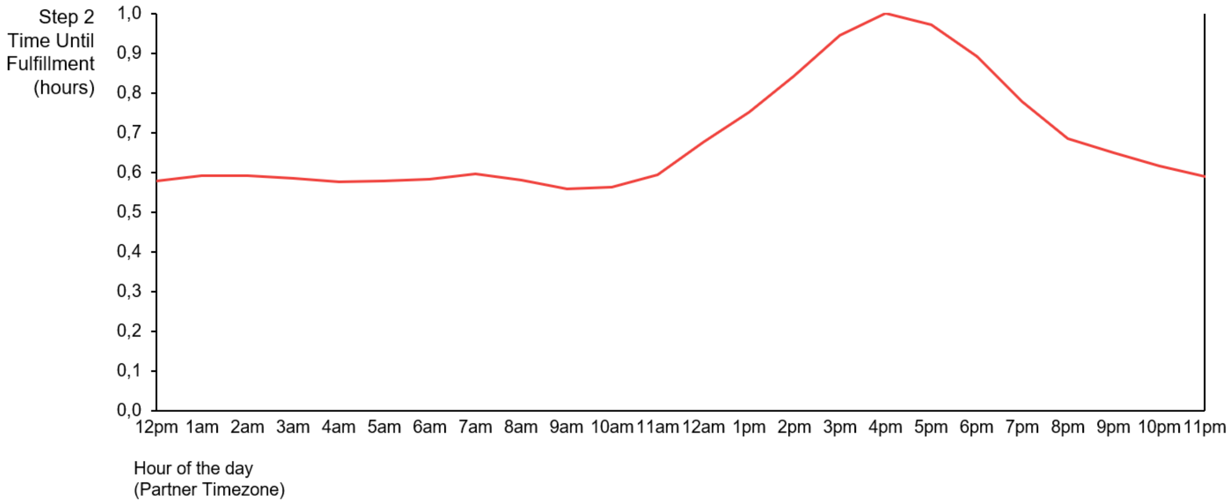

Figure D.1: Step 2 Time Until Fulfillment (hours)

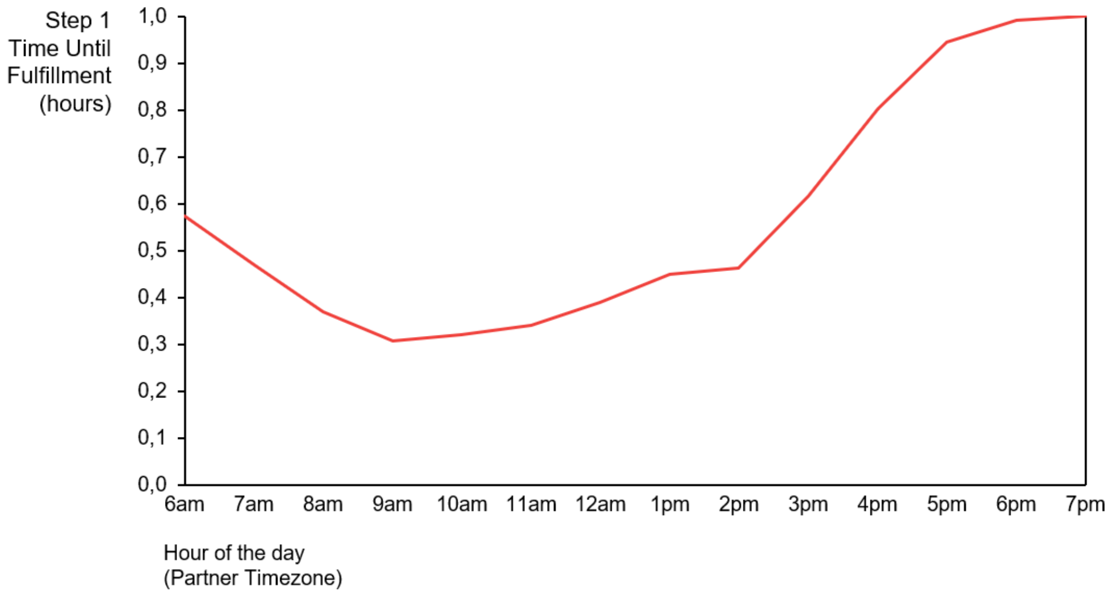

Figure D.2: Step 1 Time Until Fulfillment (hours)

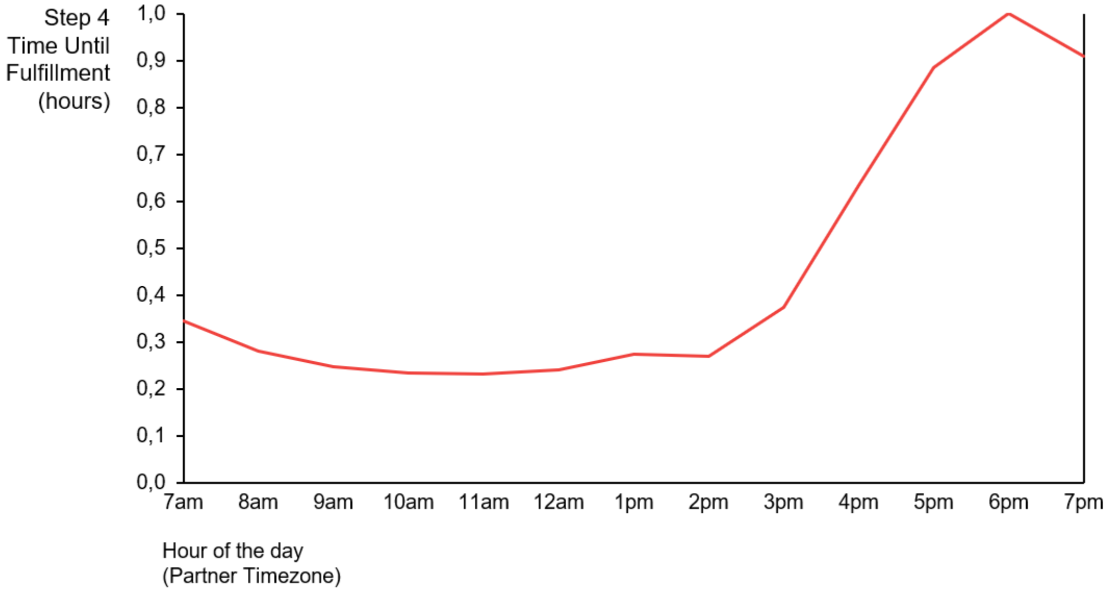

Figure D.3: Step 4 Time Until Fulfillment (hours)

Appendix E

Express vs *Standard Transit Time* comparison

Table E.1: *Express* vs *Standard* TT comparison - Random Routes

Partner Country	Customer Country	*Express* vs *Standard* TT uplift (%)
Australia	Australia	67%
France	Portugal	229%
Italy	United Kingdom	265%
United Kingdom	Italy	175%
United States	United States	76%

Appendix F

Pipeline variable creation code and SHAP Values visualization

```
juke = 0
def pipeline(row):
    global juke
    row['pipeline'] = (data[(row['SiteID']==data['SiteID'])
                        & (row['Step0_GMT']>=data['Step0_GMT'])
                        & (row['Step0_GMT']<data['Step5_GMT'])].shape[0])
    clear_output()
    juke = juke + 1
    print(juke)
    return row
```

Figure F.1: *Pipeline* variable creation Python code

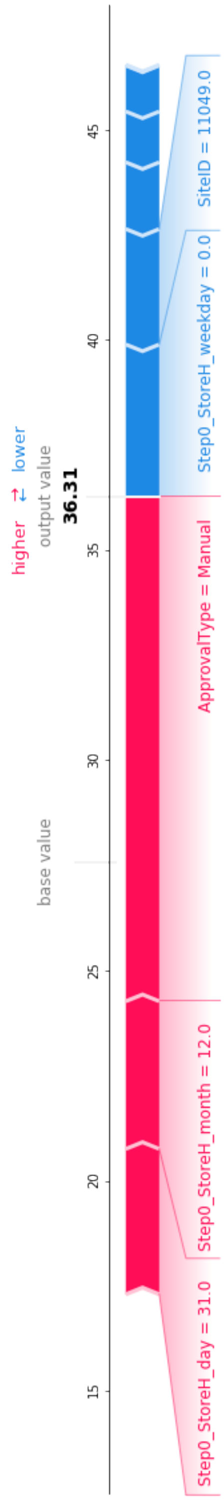

Figure F.2: SHAP Value example in model *SoSNoLogs*

Appendix G

Correlation analysis

G.1 Pearson Correlation

The Pearson correlation coefficient quantifies the linear dependencies between two continuous variables. It indicates how the collected data fits in a modeled line of their association. Equation (G.1) presents the Pearson correlation coefficient between variables X and Y, in a sample of size m.

$$r_{XY} = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})))}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}} \tag{G.1}$$

This test is based on the assumption that the observations are independent. The normality of data distribution assumption can be skipped, due to the sample size. The results are presented in figures G.1 and G.2.

G.2 Chi-squared Test

The Chi-squared test evaluates if there is a significant relationship between categorical variables. To do that two hypothesis are established:

- H0: The two categorical variables are independent
- H1: H0 is false

For each pair of categorical variables, the p-value is calculated. Table G.1 presents the final results. Once a variable was identified as dependent, no further tests were done with it.

Table G.1: P-values for the Chi-squared test regarding categorical variables

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Partner ID (1)	-	0.00	0.02	0.55	0.63	0.00	0.51	0.34	0.33
Partner Tier (2)	-	-	-	-	-	-	-	-	-
Partner Country (3)	-	-	-	-	-	-	-	-	-
Customer Country (4)	-	-	0.47	-	0.00	-	0.43	0.21	0.03
Customer Region (5)	-	-	-	-	-	-	-	-	-
Daily Pick Up (6)	-	-	-	-	-	-	-	-	-
Approval Type (7)	-	-	-	-	-	-	-	0.28	0.47
ServiceType (8)	-	-	-	-	-	-	-	-	0.01
Crossborder (9)	-	-	-	-	-	-	-	-	-

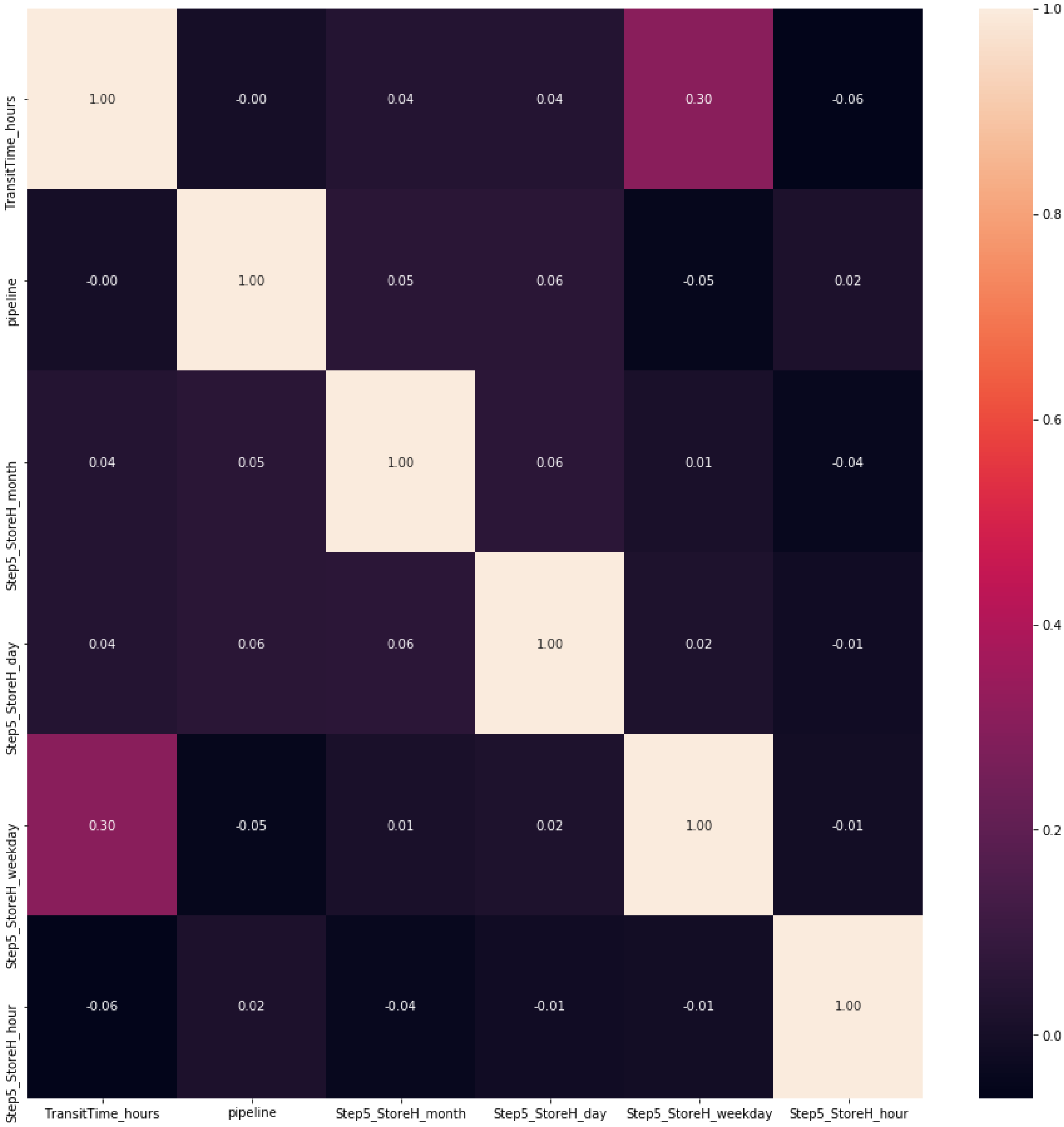

Figure G.1: Pearson Correlation Analysis results - TT

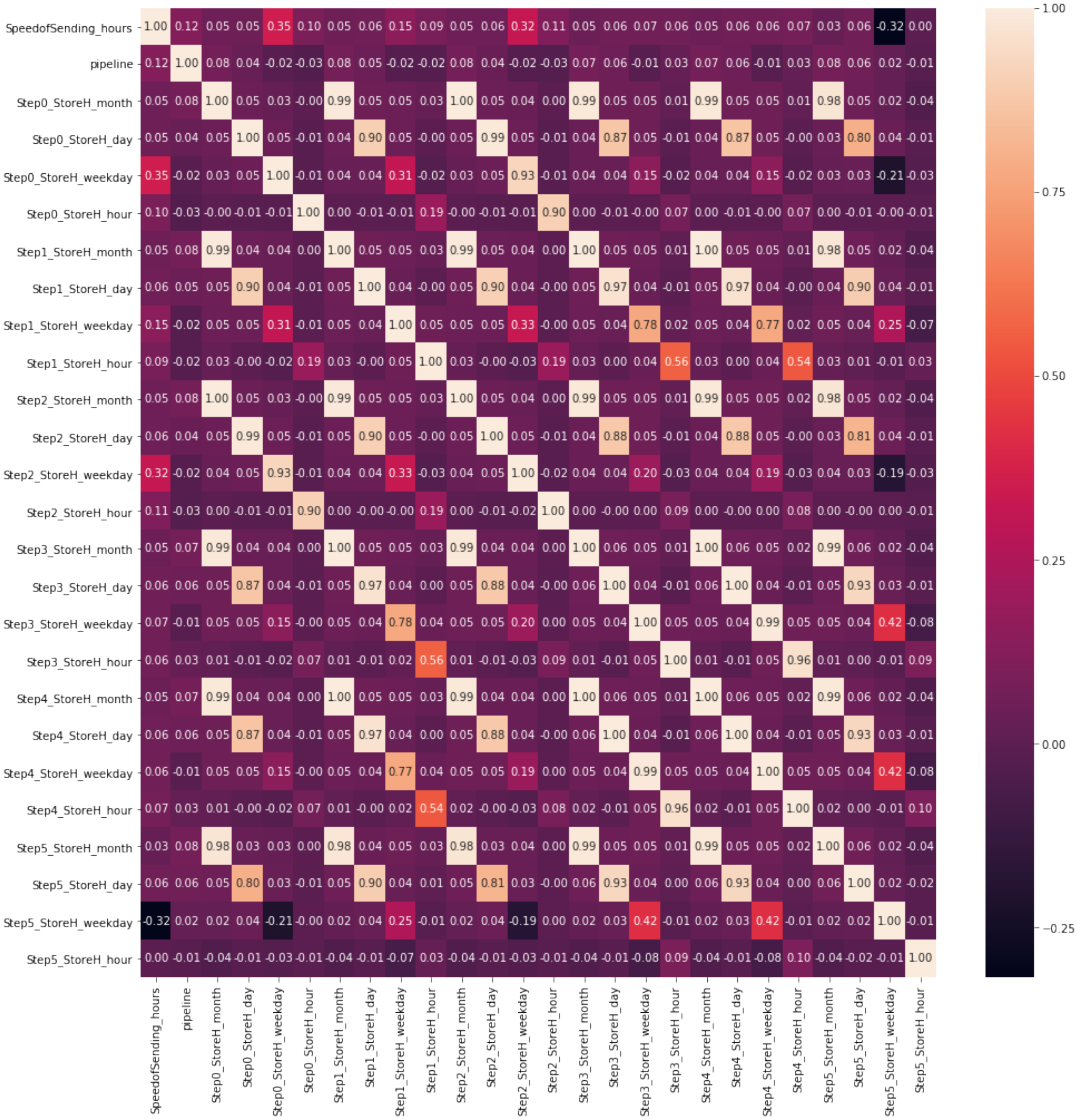

Figure G.2: Pearson Correlation Analysis results - SoS

Appendix H

Order Cycle Lead Time Dashboard

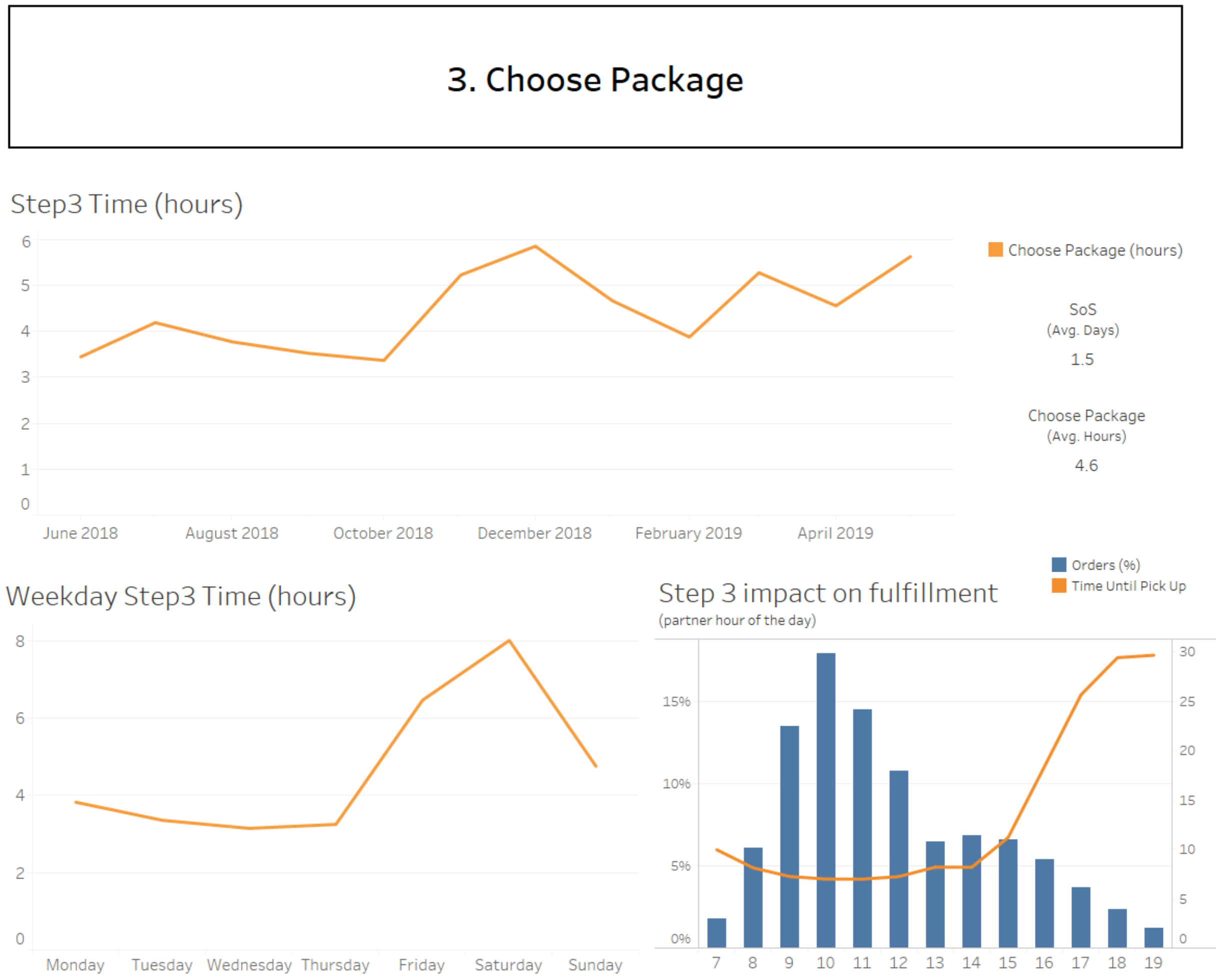

Figure H.1: Step 3 overview dashboard page

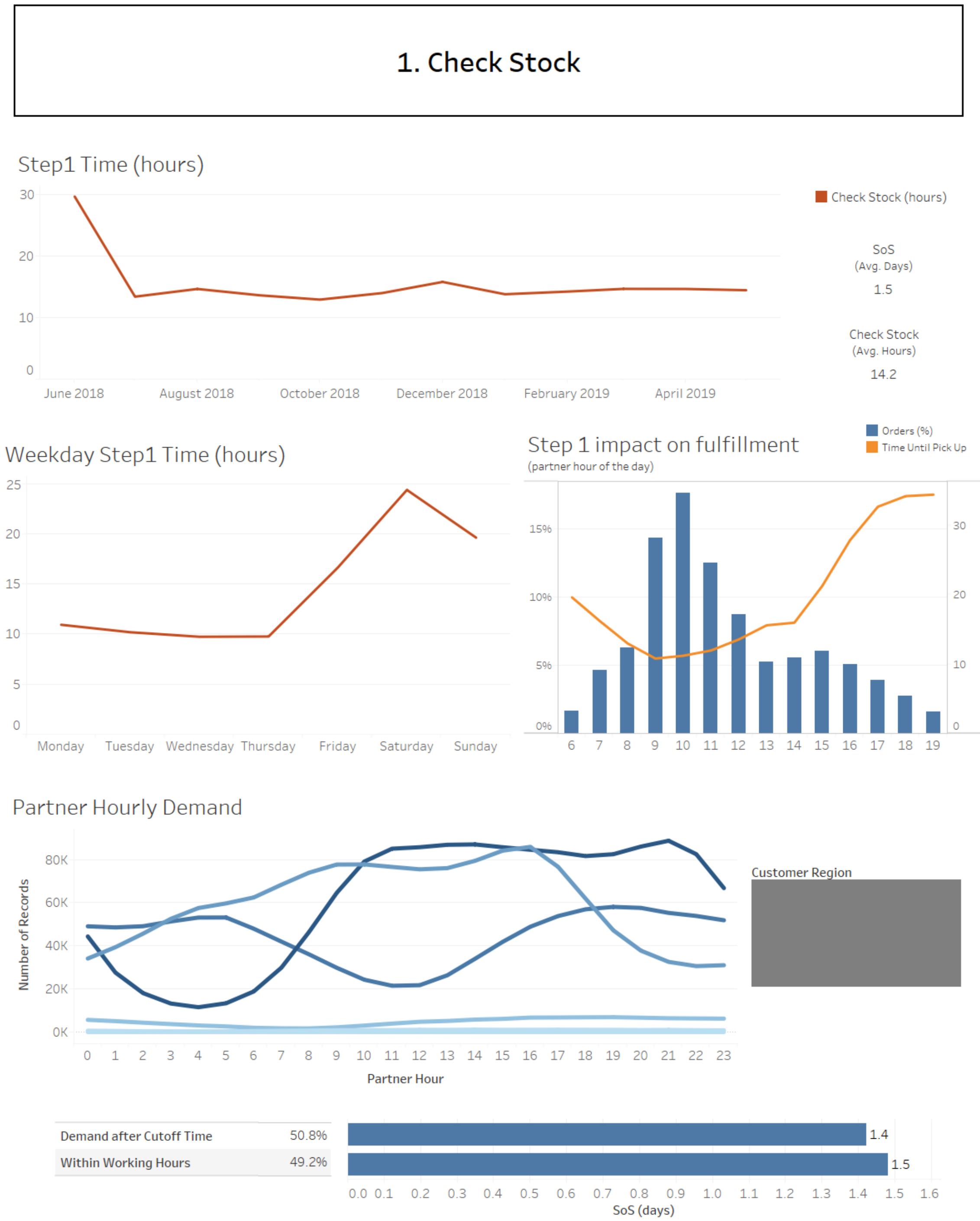

Figure H.2: Step 1 overview dashboard page

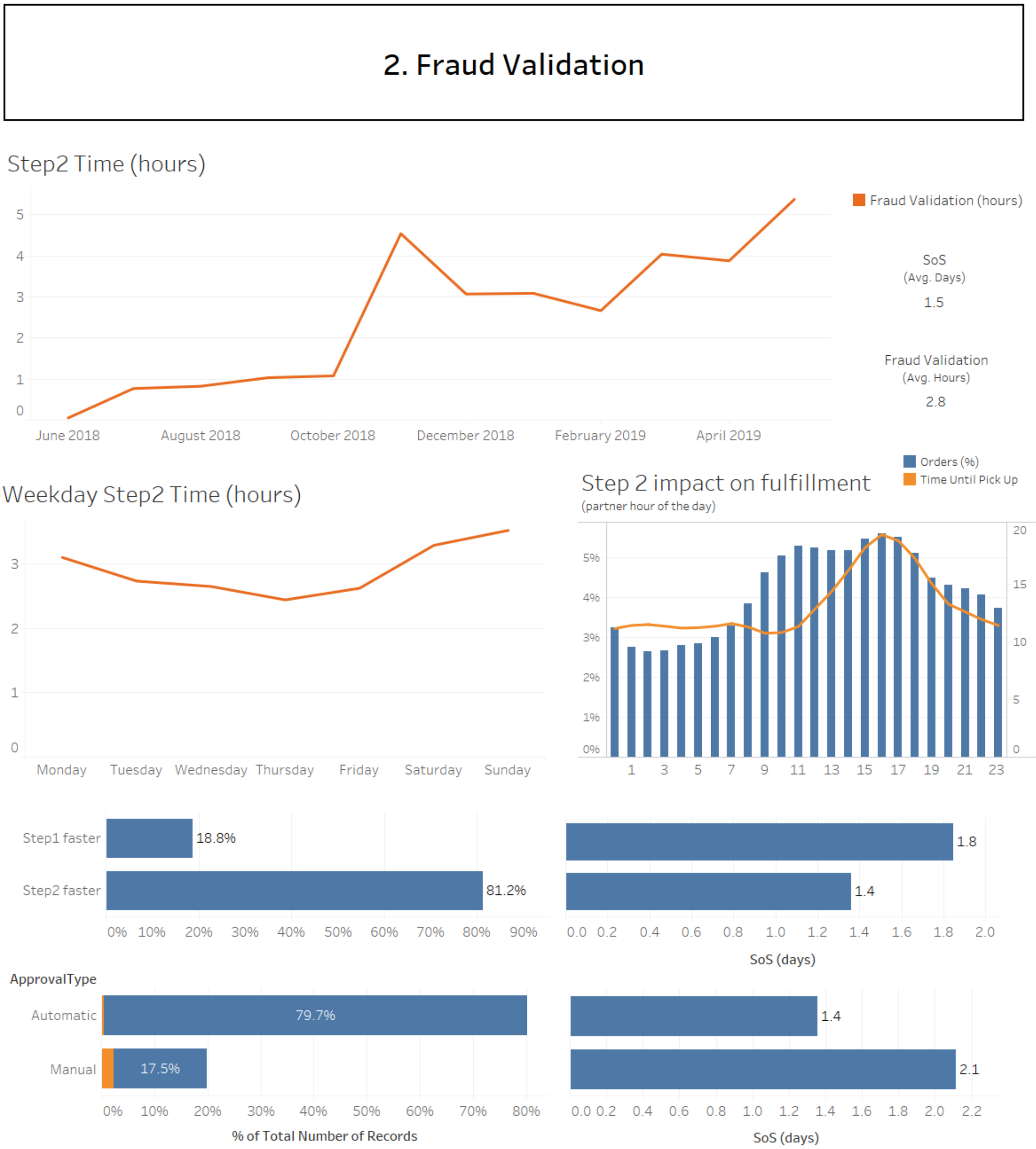

Figure H.3: Step 2 overview dashboard page

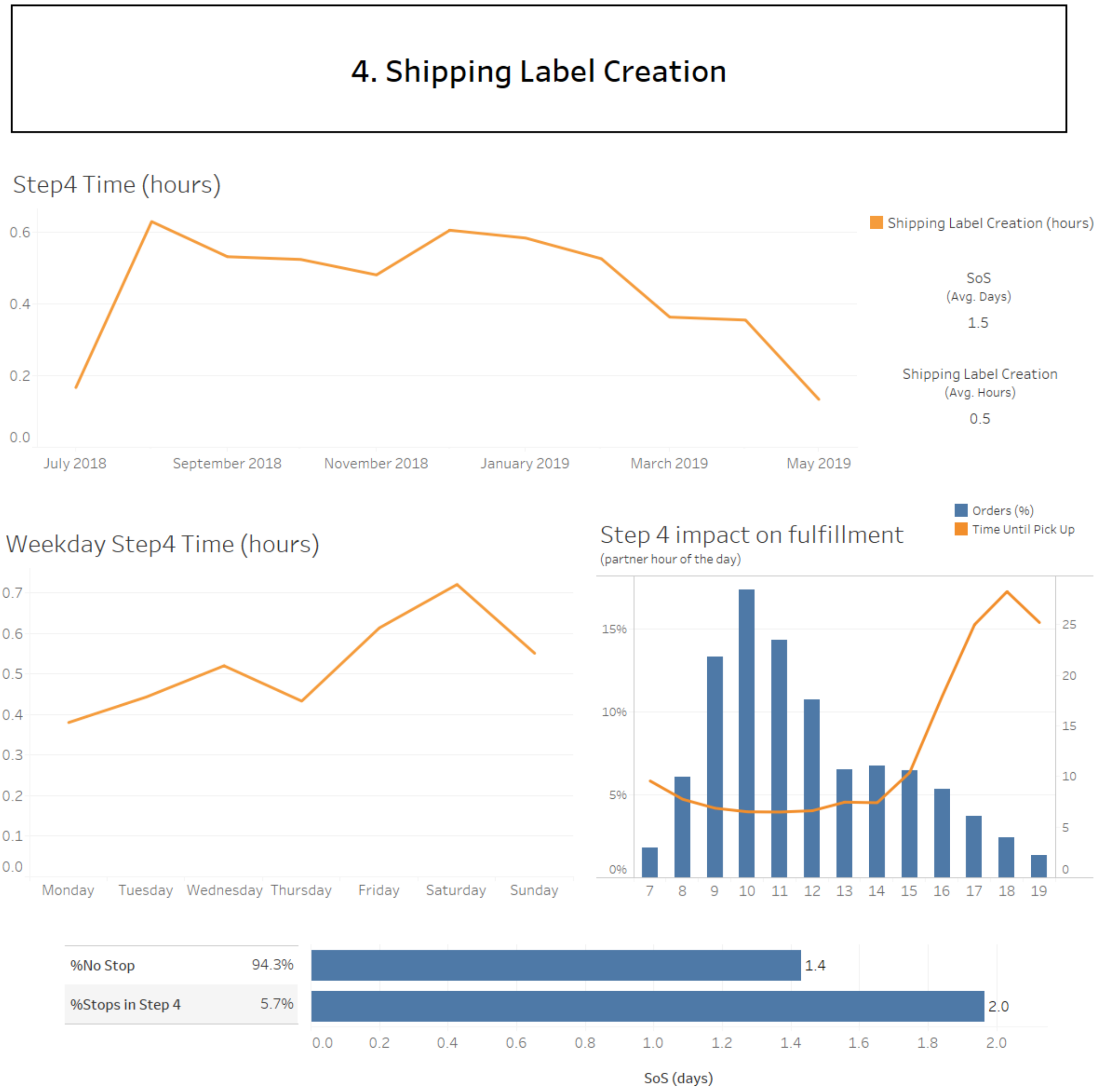

Figure H.4: Step 4 overview dashboard page

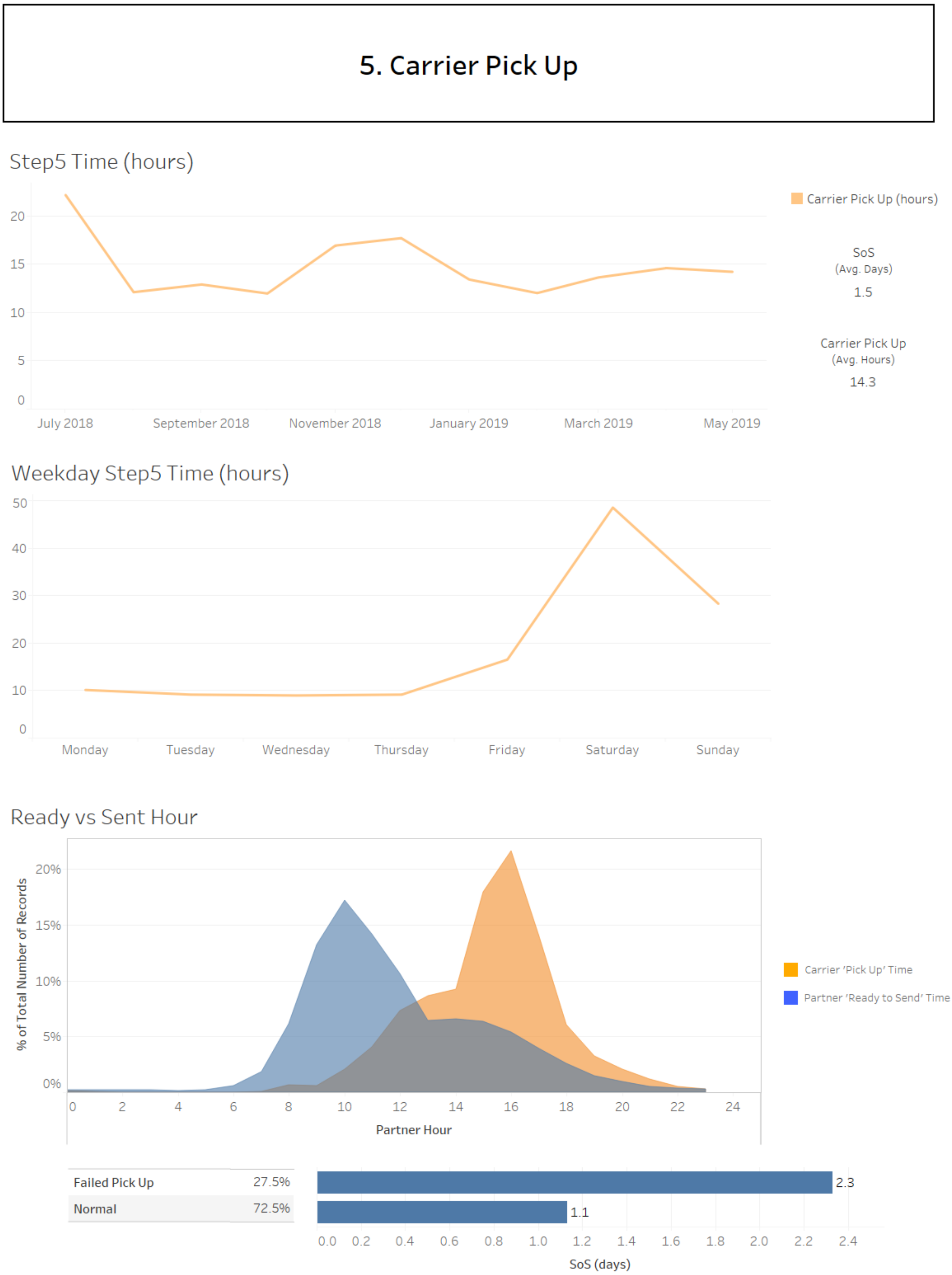

Figure H.5: Step 5 overview dashboard page

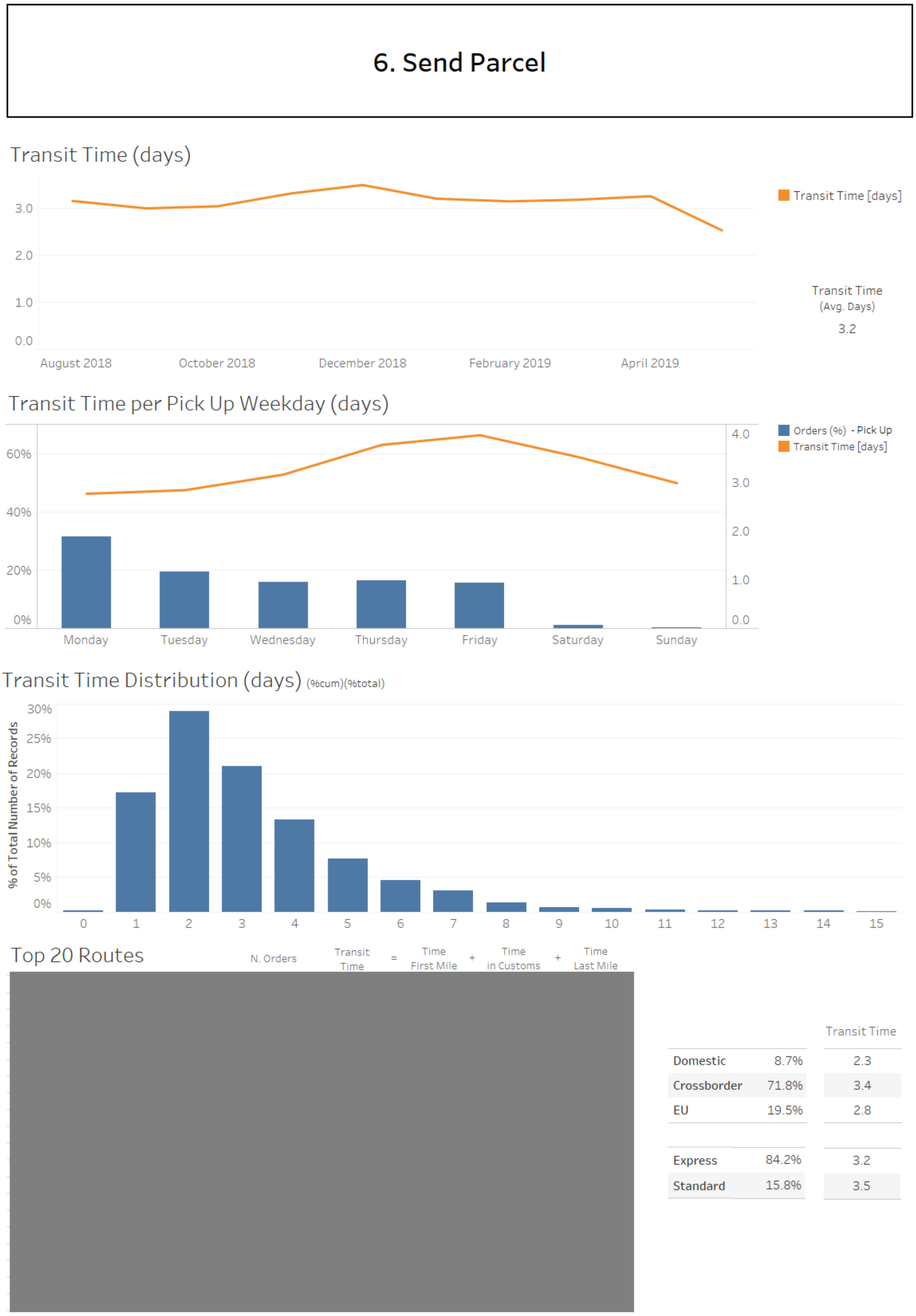

Figure H.6: Step 6 overview dashboard page