Master Thesis

submitted in partial fulfilment of the requirements for the degree M.Sc.

at Technische Universität Berlin

Improving Delivery Time with Predictive Analytics:

A Case Study in E-Commerce

Submitted to

Department of Logistics

Prof. Dr.-Ing. Frank Straube

From

Cand.-Ing. Restria Keptiasari Hertomo

Matriculation number: 327595

Karl-Marx-Strasse 13

12043 Berlin

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne unerlaubte fremde Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Unterschrift

____

(Ort, Datum)

Abstract

In today's challenging competition in e-commerce and the constant increase of online shopping, more and more consumers opt for cross-border purchases every year. However, cross-border deliveries raise the complexity of the supply chain—from the technical logistics solution to the juridical and linguistical-related issues. This research attempts to investigate the application of predictive analytics to predict delivery time in e-commerce and the supporting infrastructure to ensure the compliance of delivery time.

The purpose of this study is to identify predictive analytics techniques for delivery time prediction using historical data as well as to reveal the factors that impact delivery time in the case study of a European-based delivery platform company. These findings are then translated into key operational actions and a suitable analytics infrastructure is analysed to ensure the compliance of delivery time.

This thesis combines a qualitative and quantitative approach—it starts with a systematic literature review and is followed by data analysis using various machine learning models. The literature review serves as a starting point to get an idea about the current studies in this topic. Data is collected from the internal BI platform of the case study subject and contains the shipment data of one of their clients. The results suggest that using clean input data machine learning models are applicable to delivery time prediction and the analytics infrastructure and analytics governance have to be designed in such a way to bring business values from the analytical modelling, such as creating an alerting system.

Im heutigen schwierigen Wettbewerb im E-Commerce und der stetigen Zunahme des Online-Shopping entscheiden sich jedes Jahr mehr und mehr Verbraucher für grenzüberschreitende Einkäufe. Grenzüberschreitende Lieferungen erhöhen jedoch die Komplexität der Lieferkette - von der technischen Logistiklösungen bis hin zu den rechtlichen und sprachlichen Belangen. Diese Forschung versucht, die Anwendung von Predictive Analytics zur Vorhersage der Lieferzeit im E-Commerce und der unterstützenden Infrastruktur zu untersuchen, um die Einhaltung der Lieferzeit sicherzustellen.

Ziel dieser Studie ist es, Techniken in der Predictive Analytics für die Lieferzeitprognose anhand historischer Daten zu identifizieren und die Faktoren, die sich auf die Lieferzeit auswirken, in der Fallstudie einer in Europa ansässigen Liefer-Plattform aufzudecken. Diese Ergebnisse werden dann in operative Schlüsselmaßnahmen umgesetzt und eine geeignete Analyseinfrastruktur analysiert, um die Einhaltung der Lieferzeit sicherzustellen.

Diese Arbeit kombiniert einen qualitativen und quantitativen Ansatz - sie beginnt mit einer systematischen Literaturrecherche und wird von der Datenanalyse mit verschiedenen Modellen des maschinellen Lernens gefolgt. Die Literaturrecherche dient als Ausgangspunkt, um sich ein Bild von den aktuellen Studien zu diesem Thema zu verschaffen. Die Daten werden von der internen BI-Plattform des Fallstudienteilnehmers gesammelt und enthalten die Sendungsdaten eines seiner Kunden. Die Ergebnisse deuten darauf hin, dass die Verwendung von maschinelles Lernen Modelle, vorausgesetzt, dass die Eingangsdaten sauber sind, anwendbar auf die Lieferzeit Vorhersage sind und dass die Analytik-Infrastruktur und Analytik-Governance so konzipiert werden müssen, dass die analytische Modellierung Wertschöpfung bringt, wie z.B. die Schaffung eines Alarmsystems.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANN	Artificial Neural Network
AVL	Automated Vehicle Location
BAT	Bus Arrival time
BI	Business Intelligence
DNN	Deep Neural Network
DT	Decision Tree
EDA	Exploratory Data Analysis
FDA	First Delivery Attempt
FHS	First Hub Scan
GA	Genetic Algorithm
GA-SVM	Genetic Algorithm – Support Vector Machine
ITS	Intelligent Transportation System
KFT	Kalman Filtering Technique
k-NN	*K*-Nearest Neighbour
LMC	Last Mile Carrier
LogReg	Logistic Regression
NN	Neural Network
OFS	Orthogonal Forward Selection
RBF	Radial Basis Function
RBFNN	Radial Basis Function Neural Networks
RFID	Radio-Frequency Identification
RNN	Random Neural Network
SCA	Supply Chain Analytics

SCM	Supply Chain Management
SMOTE	Synthetic Minority Over-sampling Technique
SLA	Service Level Agreement
SVM	Support Vector Machine

1. Introduction

This first chapter aims to present the reader the problem background related to the subject of this research. It starts with the current situation of today’s e-commerce then followed by the introduction of the case study subject. Sub-chapter 1.2 gives an overview of the research objectives, in which the goals of the project and the specific actions to reach these aims are laid down. This is followed by providing the scope and the limitation of this thesis in sub-chapter 1.3. Lastly, the outline of the thesis is discussed in sub-chapter 1.4.

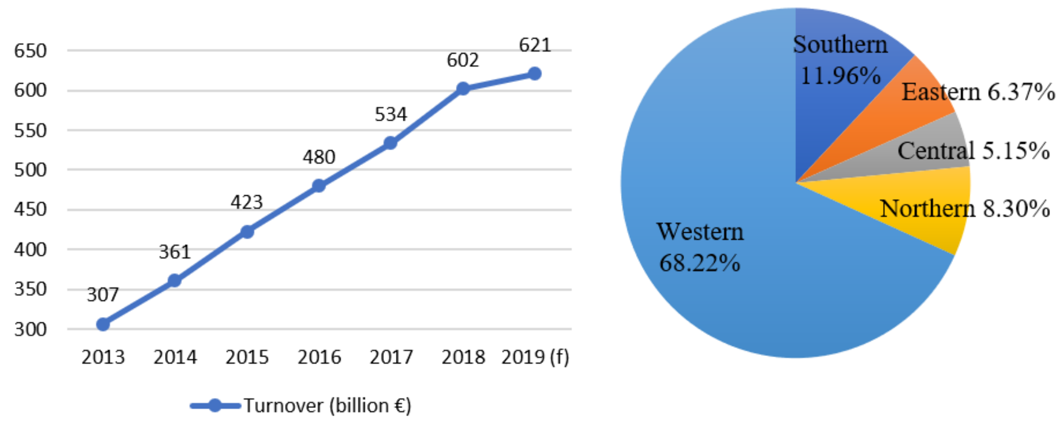
1.1 Background

1.1.1 Cross-border E-Commerce

Shopping habits have changed fast with the rise of online shopping in the early 2000s. In its infancy, e-commerce was hardly smooth sailing. In 2012, the European Commission proposed measures to promote online retail trade and to curb barriers to e-commerce, as it was under its full economic potential with just less than 4% of total European trade (Gomez-Herrera et al. 2013). Issues, such as insufficient number of e-shops willing to sell across the border, inadequate payment and parcel delivery systems and cases of hard to settle abuse and disputes were rampant in the past several years.

As the internet penetration grows, with about 83% of Europeans being connected in 2018, online shopping has expanded in an impressive way. Figure 1 displays an overview of the online shopping situation in Europe. The overall B2C e-commerce turnover is forecasted to grow to €621 billion in 2019, whereas in 2013 it was only €307 billion (E-commerce Europe 2018). Almost 70% of European turnover was generated by Western Europe, including the United Kingdom, Germany, France, Italy and Spain. This makes this part of Europe the largest e-commerce market in the continent.

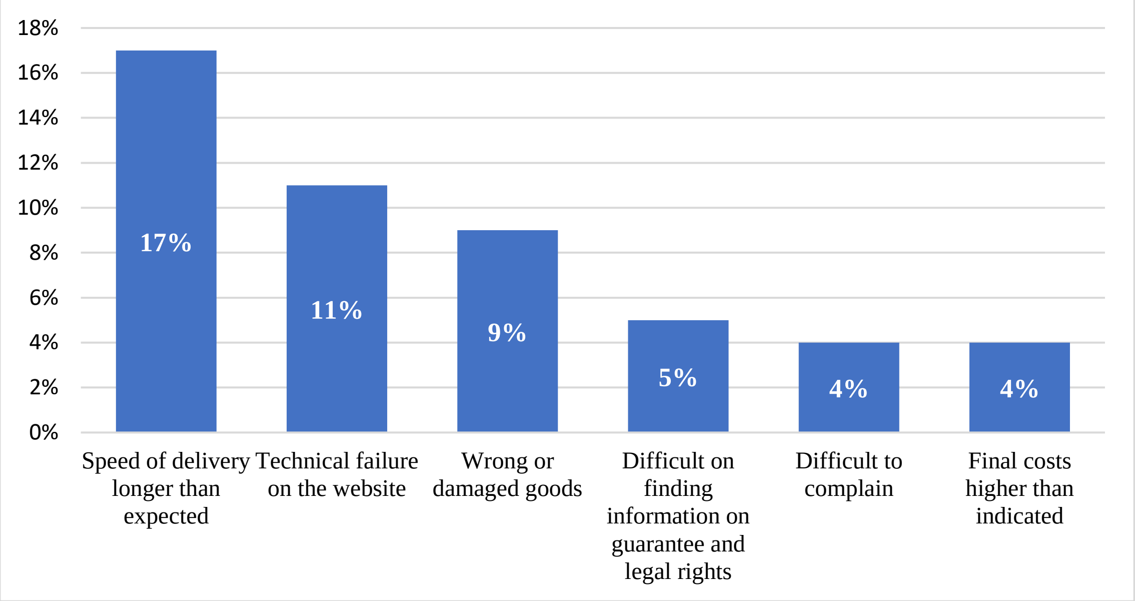
Figure 1 European B2C e-commerce turnover and proportion



The European continent has experienced big changes from internalization—in fact, Europe can be seen as an internal market. Among e-purchases in Europe, 38% of consumers opted for cross-border purchases in 2017. With a rate of 85%, Macedonian and Portuguese shoppers showed the biggest tendency to buy online from merchants located in other countries and jurisdictions. According to the survey held by PostNord (2017), 186 million European consumers ordered from abroad in 2017.

Internalization may represent enormous potential, but it can also unlock a myriad of supply chain challenges to aspiring online shops at the same time. Even during the early years of development of e-commerce businesses, Stumm and Bollo (2004) had already identified that “logistics is not an organisational technique that is adapted to the rapid and unpredictable changes that e-commerce is experiencing”. In their paper, they noted that the type of commercial activity has a direct effect on the type of logistics problems and they are sometimes the result of how these activities were set up. This is still valid in today’s e-commerce situation, as increasing the company’s logistics performance is necessary to meet customer demands (Cerasis 2018). The complexity of the supply chain even rises once a shop decides to deliver cross-border—from the technical logistics solution to the legal and language barriers. In addition, data on the volume and frequency of online purchases for physical goods is scarce, which has made the research in effects of e-commerce regarding city logistics difficult and has limited the understanding of potential developments in national as well as international delivery processes (Morganti et al. 2014). One issue prominent in e-commerce is that of late delivery. This issue in e-commerce is reflected in the 2018 survey results based on 208,618 consumers from the EU-28, as the main problem in online purchases, as shown in Figure 2 (Eurostat 2018).

Figure 2 Major issues with online purchases according to consumers



The increase in customer expectations regarding delivery time and associated costs have augmented the delivery complexity to a critical level that has led to a significant demand for special dedicated delivery services. In addition, the reverse logistics process is yet another parameter in this complex equation. More than half of the survey respondents saw the ability to return as a very important aspect of online purchases. Additionally, good returns management is crucial in creating long-term customer relationships (PostNord 2017). On that basis, online shops are expected to offer delivery flexibility and simple returns procedures. Furthermore, an easy return solution reduces the purchase risk perceived by the customers—and increases the basket value. PostNord's survey also showed that the level of returns is higher in countries with higher average spend per year as well as higher percentage of online consumers. In other words, there is a correlation between the number of returns and the degree of maturity of the e-commerce market. On the other hand, transferring delivery services to third parties generates considerable additional costs. This trade-off might be one of the biggest logistics challenges with which online shops are faced.

A common factor for the cross-border delivery delay is the number of touchpoints within the supply chain that parcels have to pass—from the shop's gate to the end customer's door. According to the study carried out by the research subject, deliveries to France from

Germany take on average 46.5 hours, meanwhile the same process takes only 21.1 hours in Switzerland. Each country has a set of different geographical circumstances which can vary the shipping situations. The reliability of local last mile carriers also plays an important part in delivery times. Thus, it is vital for e-shops to have a real-time overview of their shipments and adequate tools to process these data. The importance of big data and predictive analytics is widely acknowledged by business practitioners in improving business value and performance. Additionally, big data and predictive analytics can enhance visibility, resilience, robustness and organizational performance—thus, improving the supply chain performance (Gunasekaran et al. 2017). Schoenherr and Speier-Pero (2015) further supported the use of predictive analytics as they increase the decision-making as well as demand-planning capabilities and improve supply chain efficiencies and costs.

1.1.2 Research Subject

The subject of this research is a European delivery platform company (the "Delivery Platform") and the data for the analysis contains operational data from one of their clients (the "Seller").

The complexity of the current e-commerce delivery situation has demanded new types of solutions and technologies in the supply chain. The Delivery Platform has provided online shops innovative end-to-end delivery solutions, optimizing the delivery and after-sales process for these shops. By connecting online merchants with local last mile carriers into a supply chain network, the company enables the merchants to perform more effective and cost-sensitive deliveries. As the CEO of the Delivery Platform stated in an interview: "This service gives the e-shops all the components necessary—from labels to shipping and returns management—to connect to more than 100 carriers worldwide with lower costs and a minimum complexity". The Delivery Platform gives recommendations on the best local carriers based on factors such as type of products, weight, delivery country and the desired delivery speed. This selection is not fixed and can be changed by the shops if they prefer so.

The delivery solutions can be connected to web-based tracking and monitoring tools, a solution also offered by the Delivery Platform. It tracks parcels from checkout to delivery and the highly customisable tracking options can be adjusted depending on the shop's preferences. The software for monitoring the parcels is equipped with advanced analytics that allows the user to analyse their shipment data as well to create individual dashboards and reporting. The monitoring also facilitates a proactive contact with sellers in case of delivery problems. The main business idea of the company is to be a single contact point for shipping matters, i.e. the company takes care of all the logistics processes whilst remaining in the background and invisible for the end customers. The entire service provides e-shops with transparency throughout the delivery process and great real-time insights into their shipping performance. The monitoring feature could be further optimized using predictions which allows the Delivery Platform to react proactively in case of potential critical conditions. That way the supply chain performance could be optimized.

1.2 Research Objectives

The importance of data analytics in supply chain management, as mentioned in the previous sub-chapter, leads to a discussion about its usability. Even though predictive analytics have shown to be beneficial for organizational performance, the research into data science and predictive analytics on the operational level has been almost non-existent. In fact, academic research of the analytics field in SCM overall is scarce (Schoenherr and Speier-Pero 2015). Data in SCM is a huge untapped resource that would surely pave the way to revolutionize the field of worldwide e-commerce logistics.

Furthermore, many predictive analytics studies often focus on improving supply and demand, such as demand planning or warehousing. Despite the ever-growing online shopping trends, there seem not to be enough studies on e-commerce deliveries. Predictions can help e-shops and logistics firms to react on diverse logistical problems in a timely manner. The technical and organizational implementation of predictive analytics on e-commerce-related delay prediction is almost non-existent.

This thesis seeks to narrow the gaps in the literature by systematically reviewing existing literature on predictive analytics in delay forecasting and proposing a method to predict parcel delays by performing quantitative data analysis on the Delivery Platform’s historical data. Three research questions were formulated and are to be solved step by step in order to achieve the goal of this thesis. In its conclusion, this thesis will combine the findings from these research questions.

The study begins with finding the adequate techniques to predict delays and is formulated as follows:

RQ1: What predictive analytics techniques, based on historical data as input, can be used to predict delivery time?

After prediction techniques using historical data are found and tested, the determining delay factors of the data analysis are identified, which is expressed by the second research question:

RQ2: What are the factors that have an impact on delivery delays?

The factors, either from existing literature or result from the data analysis, are then assessed. Finally, the third research question deals with determining the technical and organizational implementation of the findings on the prediction techniques and delay features:

RQ3: What courses of actions and restructuring of the analytics or IT system should be undertook in the organization to ensure on time delivery time?

Ultimately, the thesis will present a list of best-scored forecasting techniques assessed from the data analysis as well as the important features on the delay. This is followed by a framework for intra-organizational and inter-organizational measures and IT system to react on potential shipment delays.

1.3 Purpose of the Thesis

Examining the correlation between supply chain variables and shipment delays using the historical data of an online shop, the purpose of this study is to explore how predictive analytics can improve the supply chain process, by predicting potential delayed shipments. Identifying these potential delays would allow the Delivery Platform to act proactively and hopefully reduce delayed shipments and increase customer satisfaction. The prediction results should be connected to a communication system that sends notification to the responsible supply chain agents, so they would be able to immediately do the necessary follow up as well as prioritise the shipments. This feature could be very useful in holiday seasons such as Christmas.

1.4 Scope and Limitations

Time puts constraints on what can be investigated, and therefore in order to keep the study and research within the specific research structure, the author needs to set a clear focus and limitation. The data analysis of this research project will focus on the outbound logistics process of one specific client of the Delivery Platform, an online shop in health/food sector. Furthermore, this thesis only focuses on the supply chain process from said shop to the end customers.

The case study is based on the logistics network of the Delivery Platform that serves shipments for the shop—it may, therefore, reflect mainly processes unique to the Delivery Platform and the shop’s decision-making policies. The logistics network also differs slightly from one shop to another, depending on the location of the shop’s warehouse. This can give different results for the data analysis.

1.5 Outline of the Thesis

The overall structure of the thesis is depicted in Figure 3. It comprises 7 chapters, addressing the three research questions in chapters 3 to 7.

Chapter 1. Introduction – This chapter briefly presents the background relating to the subject of this thesis project, i.e. the e-commerce situation in Europe, the potential of predictive analytics in this business and specifically in the Delivery Platform. Subsequently, the research objectives, purpose, scope and limitations of the thesis are established. Finally, the overall structure of the thesis is outlined.

Chapter 2. Theoretical Framework – In this chapter, the author gives the definition of commonly used terms and concepts as well as theories that are applicable in this thesis.

Chapter 3. Research Methodology and Design – This chapter provides the research approach of this thesis and describes the data collection, method used and research evaluation.

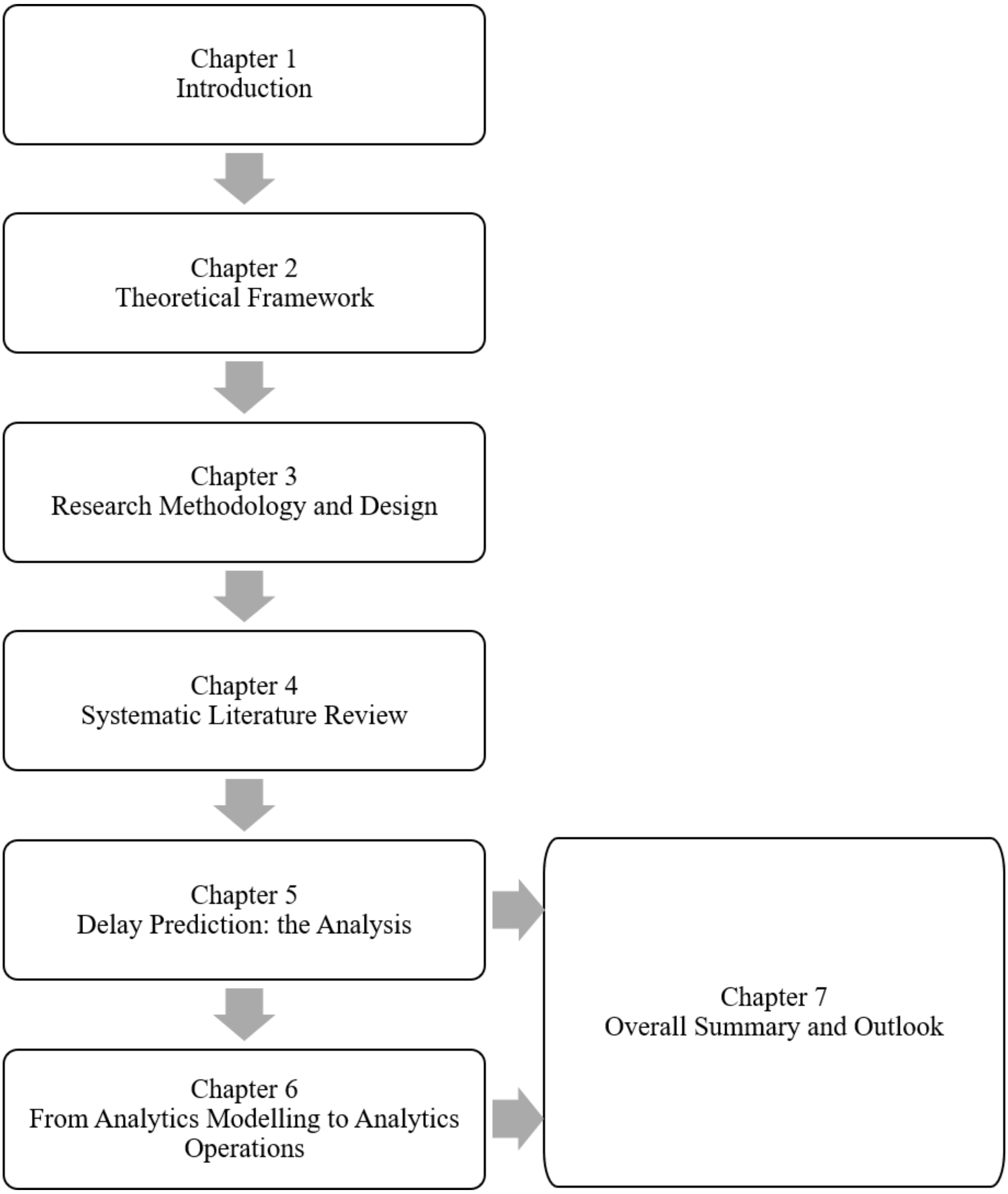
Chapter 4. Systematic Literature Review – This chapter presents the systematic literature review conducted in order to get a better idea on the current and previous studies in the topic of predictive analytics in time prediction.

Chapter 5. Delay Prediction: Analysis – As one of the most important parts of this thesis, this chapter aims to prove quantitatively whether ML is applicable in the delivery time prediction in e-commerce. Here, an extensive data analysis will be explained in a detailed manner and its results will be summarized.

Chapter 6. From Analytics Modelling to Analytics Operations – This chapter provides insights of how to design the analytics infrastructure and governance with the goal of bringing business values from the results of the data analysis from chapter 5. Concrete operational actions derived from the analytics architecture are proposed.

Chapter 7. Overall Summary and Outlook – This last chapter gives summary on the entire research thesis, recommendations for the management as well as limitations of the thesis and opinions on future research.

Figure 3 Thesis structure


Chapter 1
Introduction
Chapter 2
Theoretical Framework
Chapter 3
Research Methodology and Design
Chapter 4
Systematic Literature Review
Chapter 5
Delay Prediction: the Analysis
Chapter 6
From Analytics Modelling to Analytics Operations
Chapter 7
Overall Summary and Outlook

2. Theoretical Framework

This chapter will present theories, terms and concepts relevant for this thesis. The first part of this chapter introduces the concept of predictive analytics and how it differs from machine learning. This sub-chapter also dives into various machine learning types for classification problems. The second part is the concept of predictive analytics specifically in SCM, which is popularly known as SCA.

2.1 Predictive Analytics

In the last decade, the business field has made great strides in applying advanced statistical modelling techniques to support strategic and operational decision-making. This has been the result of the staggering volumes of data businesses collected each day, which paved the way to the term “Big Data”. According to the sixth report edition of the data science platform DOMO:

“Over 2.5 quintillion bytes of data are created every single day, and it’s only going to grow from there. By 2020, it’s estimated that 1.7MB of data will be created every second for every person on earth.” (DOMO 2018)

Predictive analytics is the response to this new trend of data availability, as it is a way to leverage all this information, and get tangible new insights and stay competitive in today’s business world (Ongsulee et al. 2018). Hence, there is a growing number of research in predictive analytics and big data. The following Table 1 shows an overview of definitions of predictive analytics from some researches:

Table 1 Definitions of predictive analytics

Research	Definition
Shmueli and Koppius (2010)	“Predictive analytics include statistical models and other empirical methods that are aimed at creating empirical predictions, as well as methods for assessing the quality of those predictions in practice, i.e. predictive power.”
Waller and Fawcett (2013)	“Data science is the application of quantitative and qualitative methods to solve relevant problems and predict

	outcomes.” … “Predictive analytics is a subset of data science”
Nassif et al. (2016)	“Predictive analytics is concerned with the prediction of future trends and outcomes. The approaches used to conduct predictive analytics can be classified into machine learning techniques and regression techniques.”
Kumar and Garg (2017)	“Predictive analytics is a term which describes a variety of statistical and analytics techniques. It analyses current and historical facts to make predictions about future. It develops such models which can predict future events and behaviours of variables.”
Ongsulee et al. (2018)	“Predictive analytics is an area of statistics that deals with extracting information from data and using it to predict trends and behaviour patterns.”

Researchers, such as Ongsulee et al. (2018) as well as Kumar and Garg (2017), argued that predictive analytics are a compilation of statistical techniques that encompass various statistical tools. These tools, e.g. predictive modelling, machine learning and data mining, have the purpose of making predictions about the future or unknown events by analysing current or historical data. However, the term has a similar meaning to data science, as viewed by Waller and Fawcett (2013). In their paper there is no clear difference between those two terms.

Furthermore, the term predictive analytics distinguishes from the term forecasting due its more detailed level of granularity, e.g. predicting scores for each individual organizational element (Ongsulee et al. 2018). It is also not just about predicting the future, but it also answers the questions regarding what would have happened in the past, given different conditions (Waller and Fawcett 2013).

The view of predictive analytics as a prediction tool is shared by Shmueli and Koppius (2010), who state that predictive analytics also include methods for reviewing the quality of the predictions or commonly known as predictive power. It plays an important role in theory building, theory testing and relevance assessment. Nassif et al. (2016) states that predictive analytics can be classified in two different techniques: machine learning and regression. Even though predictive analytics is deemed as a subset of statistics by the aforementioned researchers due to its relation to quantitative approaches, it also involves qualitative approaches (Waller and Fawcett 2013).

However, all researchers seem to agree on one common thing: predictive analytics is a technique or a process to discover interesting patterns in data and to predict trends or behaviour.

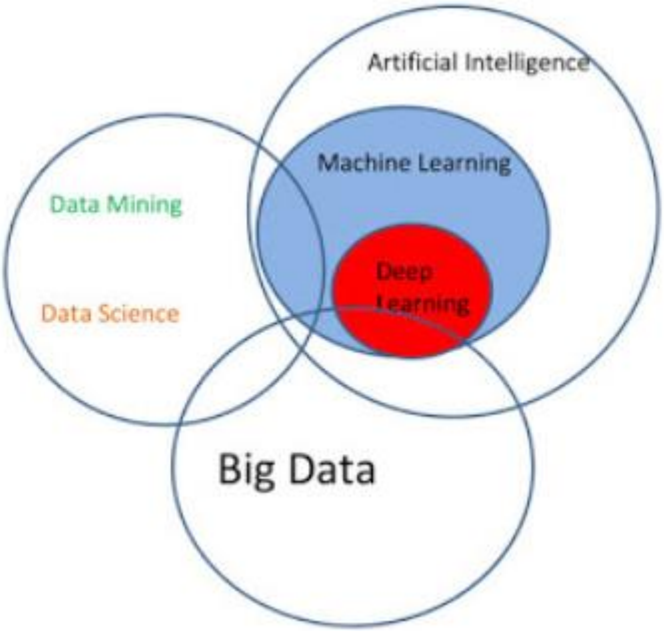
There is a plethora of use cases for predictive analytics spread across different fields, e.g. financial services, insurance, telecommunications, mobility, health care, etc. In the commercial organizations, it is used in tasks from predictive marketing to application of machine learning to improve business processes. Exploiting patterns of historical and transactional data, it is a tool that can identify risks and opportunities for each individual, that could be a customer, employee, or any stakeholder of a company.

The field of logistics and supply chain is no exception to this phenomenon.

2.1.1 Predictive Analytics vs. Machine Learning

Machine learning techniques have become progressively popular in performing predictive analytics due to their capability in handling large scale datasets with uniform characteristics and noisy data. In countless publications, the discussion of predictive analytics is strongly related to machine learning. Even the term *data science* shows a certain overlap with predictive analytics, as seen in the previous sub-chapter. What is the difference between these terms? Finding one common answer is hard, as data science is a rather fuzzily defined field, so it is not a surprise to find that different attempts have been made to define it (Taylor 2016). No visual representation has been found that displays all three terms. Gregory Piatetsky-Shapiro, a co-founder of KDD conferences and the president of the data science website KDnuggets, proposed a Venn diagram as shown in Figure 4.

Figure 4 Data science venn diagram



Machine learning is inherently a multidisciplinary field, drawing on results from fields such as artificial intelligence, probability and statistics, information theory, computational complexity theory, control theory, and philosophy (Mitchell 1997; Ongsulee et al. 2018). As a rapidly growing technical field, it addresses the question of how to build software that improve automatically through experience—lying at the intersection of computer science, statistics, artificial intelligence, and data science (Jordan and Mitchell 2015). It employs statistical techniques to make computers "learn", i.e. progressively improve performance on a specific task with data, without being explicitly programmed. Thus, it is one of the most powerful tools in the field of predictive analytics, thanks to its effective algorithms and frameworks that result in a high predictive accuracy.

In summary, machine learning is a subset of artificial intelligence that is commonly used as a tool in predictive analytics—which itself is a subset of data science.

Machine learning techniques are classified into taxonomy based on the outcome of the algorithm. Common algorithm types are as follows:

1. **Supervised learning**: This algorithm type generates a function that maps inputs to the desired outputs. This set of input and outputs are presented by a "teacher" and the "learner" is required to learn a general rule or a function which maps a vector into one of several "classes" (categories) by looking at input-output

examples of the function the algorithm is trying to emulate. A common supervised learning task is the classification problem.

2. **Unsupervised learning**: This algorithm type models a set of inputs, as no labels are given to the learning algorithm, leaving it on its own to find patterns in the set of inputs. This type of learning can be a goal of itself, e.g. discovering hidden patterns in the data, or a means towards the end, e.g. feature learning.
3. **Semi-supervised learning**: This algorithm type is given both labelled and unlabelled training dataset, i.e. training set with some missing target outputs.
4. **Reinforcement learning**: The algorithm type learns how to act given the feedback to its action in a dynamic environment. The feedback could be a reward or a punishment that would guide the learning algorithm. A classic example of this is playing a game against an opponent.
5. **Active learning**: A variant of semi-supervised learning where the training labels are limited to some set of instances and which the learning algorithm has to optimize its choice of objects to acquire labels for problematic unlabelled instances.

From all the listed learning types, supervised learning is the most commonly used in predictive analytics. As this thesis focuses on classification problem (delayed or not delayed), different types of supervised learning techniques will be further discussed.

2.1.2 Types of Learning for Classification Problem

The selection of adequate algorithms for a certain classification problem depends on many aspects—from the composition of the input data to the purpose of learning. There is a wide range of supervised learning algorithms for us to pick, each with its own strengths and weaknesses. There is, however, no one approach that fits all. Hence, determining the structure of the learned function and its corresponding learning algorithm is vital.

Logistic Regression

Logistic regression, developed for the biological sciences in the early 20th century, is one of the most established methods for binary classification problems. Logistic regression is named after the logit function, which is the inverse of the sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

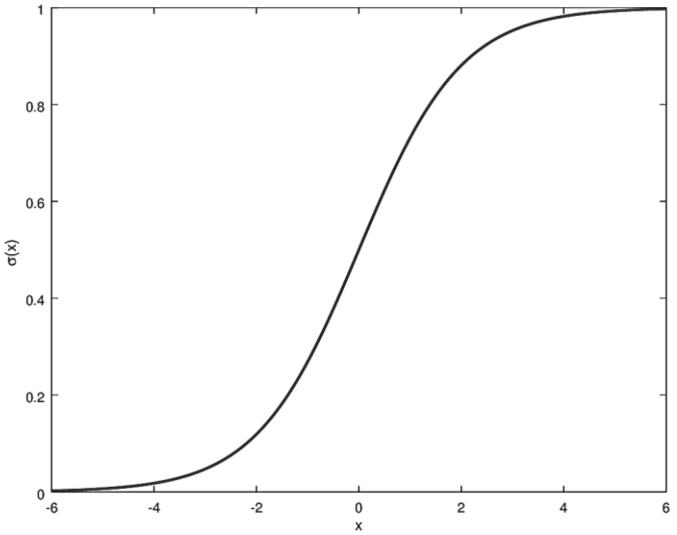
where e is the Euler's number, the base of the natural logarithms. The correlation between the probability of a result and its predictor variables is as follows:

$$\sigma(x) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_k X_{ik}$$

where β_1, …, β_k are coefficients and X_1, …, X_{ik} are predictor variables. Each predictor variable has an associated β, a constant real value, that represents the degree of importance of each predictor (Komarek and Moore 2005). This coefficient is estimated from the training data using the maximum-likelihood estimation. Developed by R.A. Fisher in the 1920s, the maximum-likelihood estimation is a method to seek the desired probability distribution that makes the observed data "most likely", i.e. finding the parameter values that maximize the likelihood of making the observations given the parameters (Myung 2003). To maximum the likelihood, values for the coefficients β are searched that minimize the error in the probabilities predicted by the model.

The S-shaped function takes any real-valued number and map it into a value between 0 and 1, converging at those limits (see Figure 5).

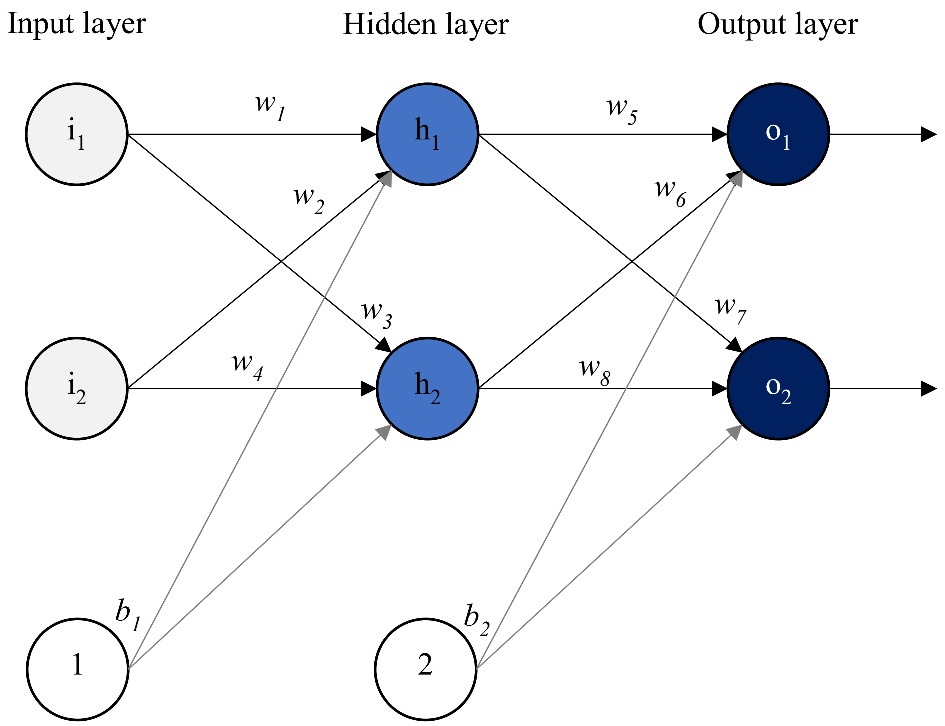
Figure 5 Sigmoid function



Artificial Neural Network (ANN)

ANNs are amongst the most successful learning methods implemented to solve complex real-world problems, such as interpreting sensor data, speech recognition and learning robot strategies. The birth of ANNs was motivated by the biological learning systems—which consist of complex webs of interconnected neurons (Mitchell 1997). In data mining, ANNs are constructed on massively parallel distributed processor made up of simple processing units, replicating the intelligent data processing ability of human brains. Each simple unit takes a number of real-valued inputs, or possibly the outputs of the preceding units, and generates a single real-valued output, which would become the input for the unit next in line. The structure of ANNs is displayed in Figure 6.

Figure 6 Artificial neural network structure



ANNs consist of three layers: input, hidden and output layer. A single unit of connection is called a neuron, which is connected with other neurons through synoptic weights. In Figure 6, there are two neurons in each layer with w_1, …, w_8 as weights of the outputs of the preceding neurons. A perceptron, the most basic form of neural network, has at least one input, a bias, an activation function and a single output. It receives inputs, multiplies them by the according weight and then passes them into an activation function, e.g. logistic, trigonometric and step function, to result an output. Bias (b_1, b_2) is added to the perceptron, which represents a constant weight regardless of the inputs that allows a better fit for the predictive model and, therefore, achieve a higher accuracy. Taking into consideration the desired target value, the ANN will be trained by updating the weight matrix after each iteration of the algorithm. Therefore, the higher the number of iterations, the closer the predicted output matrix is to that of the target value.

Beside perceptrons, another common type of ANNs is the feedforward neural networks, originating in the 50s. Here, all the nodes are connected and the data moves in only one direction—forward from the input nodes until it reaches the output nodes—where the name comes from.

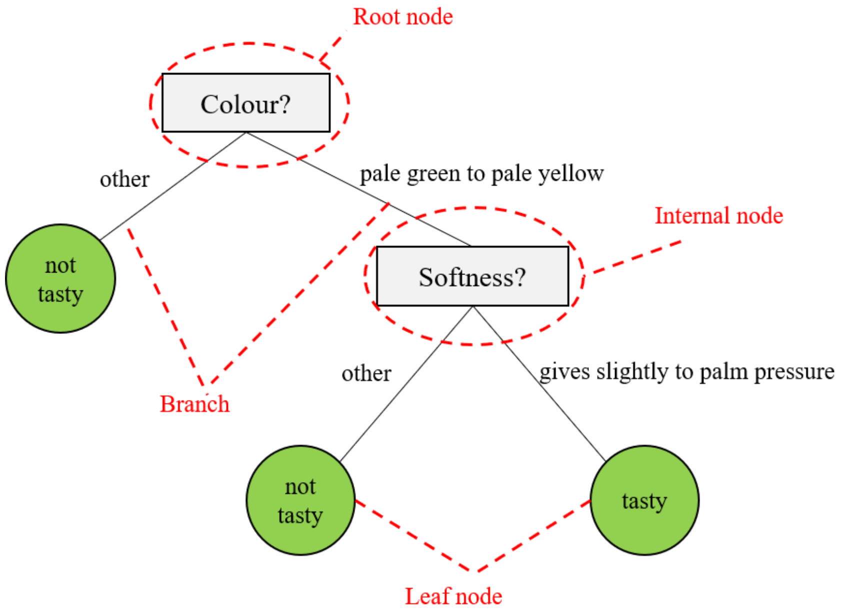
As the size of data grows and demand to find insights into the data rises, the types of ANNs grows. Deep Learning developed exponentially in the 1990s, when GPU started becoming popular and computers started becoming faster at processing data. Deep Neural Network algorithms are a subset of machine learning algorithms that learning multiple levels of representations, using multiple layers that consist of multiple linear and non-linear transformations (Kumar and Garg 2017). It is an ANN with more than one hidden layer between the input and output layer in order to model complex relationships among data (Treethidtaphat et al. 2017). One example is the Multilayer Perceptron neural network, which has at least three layers. Similar to a feedforward neural network, as its neurons are fully connected to the neurons of the next layer, this type fits non-linear functions well and is used extensively in speech and image recognition as well as machine translation technologies.

In general, ANNs are well-suited to train data that contain noise and complex sensor data, such as inputs from cameras and microphones.

Decision Tree

The decision tree is among the most popular inductive inference algorithms. This type of learning approximates discrete-valued target functions, which can be represented as sets of if-then rules to improve human readability (Mitchell 1997). It is robust to errors and missing values in the training data. Figure 7 displays the basic structure of a decision tree. The topmost node in a tree is called the root node, meanwhile the bottommost node is called the leaf node. It predicts the label associated with an instance by sorting it down the tree from the root node to a leaf. At each internal node on the root-leaf path, the succeeding node is selected on the basis of a splitting of the input space, which is based on the features or on a predefined set of splitting rules (Shalev-Shwartz and Ben-David 2014). The splitting continues until leaf nodes, each containing a specific label, are found for all the branches of the tree.

Figure 7 Example of a decision tree structure



CART (Classification and Regression Trees) is one of the most common decision tree-based model. It constructs binary trees using the feature and threshold that yield the largest information gain (gini index) at each node.

Overfitting is an important issue in this type of supervised learning, as the decision tree can create extremely complex trees that do not generalize the data well. To combat this problem, post-pruning of the tree is recommended. Another way to reduce overfitting is by constructing multiple decision trees—a ML method called random forest. It consists of an ensemble of decision trees, in which a prediction is obtained by a majority vote over the predictions of each individual tree.

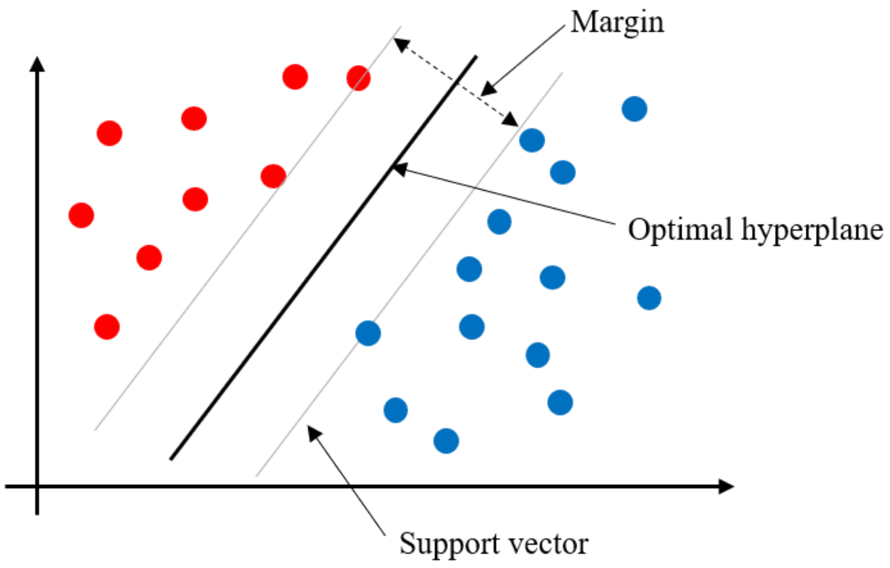
Decision trees have been successfully implemented to solve classification problems, such as classifying patients by their disease, applicants by their likelihood of paying their loans, species of flora and fauna by their characteristics, etc.

Support Vector Machines (SVM)

This supervised learning is very useful tool for learning linear predictors in high dimensional feature spaces (Shalev-Shwartz and Ben-David 2014; Cortes and Vapnik 1995). The high dimensionality of the feature space, which may be caused by the high number of features, a phenomenon called the "curse of dimensionality", increases the sample and computational complexity. The curse of dimensionality causes sparseness in training data, which decreases the performance of a classifier. The main idea behind

SVMs is the construction of a linear decision surface in the feature space and the search for a "large margin" separator, in order to separate the training data without errors (see Figure 8). In other words, a plane separates the training dataset with the largest margin as possible, maintaining the instances on the correct side of the separating hyperplane and keep them away from one another. By restricting the algorithm to produce a large margin separator, the sample complexity can be reduced despite the high dimensionality of the feature space.

Figure 8 SVM visualization

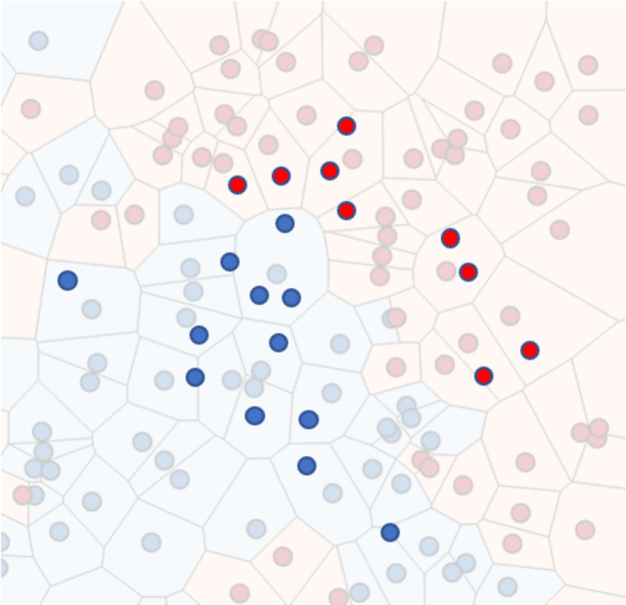


This complexity reduction gives SVM models advantages of strong learning ability in small sample situation, fast learning speed and good generalization ability. The main disadvantages of SVM are: the requirement for a large amount of data, longer running time in both training and testing due to their high algorithmic complexity and considerable memory requirements (Vapnik 1999).

K-Nearest Neighbour (*k*-NN)

K-nearest neighbour is an instance-based learning, where all the work is performed at the time of classifying a new instance rather than when the training data is processed (Witten et al. 2017). Other methods "learn" to produce generalization as soon as the data has been seen, whereas instance-based models waits until they see a new instance. Hence, this type of learning differs from the others in the time at which the "learning" takes place. The name nearest neighbour comes from the general framework of the model itself: each new instance is compared with the existing ones using a distance metric (Euclidean distance, Manhattan distance, etc.), then the label of the new instance is predicted on the basis of the labels of its closest neighbours in the training set. The rationale behind this is that the relevant features used in the labelling of an instance are similar in a way that make close-by points likely to have the same label. In *k*-nearest neighbour learning, more than one nearest neighbour is used, with *k* representing the number of neighbours. The majority class is assigned as the label for the new instance. The value *k* signifies the complexity of the nearest neighbour model—the higher *k* is, the less adaptive the model is (Jiang et al. 2012). Figure 9 visualizes the *k*-nearest neighbour classifier, with the bright points as the new instances that have to be classified, using the distance metrics from the nearest *k* other points surrounding them (the faded points).

Figure 9 K-nearest neighbour visualization. Adapted from Zakka (2016)



This simple type of classifier works well on basic recognition problem and is robust to noise in the training data. It is easily controllable and can be implemented in real-time with enough representative data.

2.2 Predictive Analytics in Supply Chain Management

The advent of big data and the growing combination of resources, tools and applications have deep effects in the field of supply chain management, opening a myriad of opportunities and challenges. Supply chain managers are increasingly relying upon data to gain insights into expenditures, trends in costs and performance, and support processes such as inventory monitoring, production optimization, and shipping. SCM businesses are seeking to gain competitive advantage by capitalizing on data analysis (Hazen et al.

2014). The benefits of predictive analytics have been recognized by researchers and have been the point of interest of many business journals, as the advance of predictive analytics' ability would refine and improve supply chain decision-making. Waller and Fawcett (2013) proposed a definition of supply chain predictive analytics:

"SCM predictive analytics use both quantitative and qualitative methods to improve supply chain design and competitiveness by estimating past and future levels of integration of business processes among functions or companies, as well as the associated costs and service levels."

Even though there is a remarkable amount of attention pouring into supply chain analytics, there is still lack of integration between SCM systems and performance management systems (Stefanovic 2014). These performance measurement models have focused mainly on single organizations or a specific type of performance such as financial. Many scholars emphasize that the application of big data and business analytics within logistics and SCM, defined as supply chain analytics (SCA), is still in its infancy (Wang et al. 2016). Wang and his colleagues added, that the successful adoption of SCA methodologies and techniques depends on two factors:

- Robust data collection and data cleansing
- Major investments in technological infrastructure and human resources

Most SCM big data sources are unfortunately generated in unstructured formats, which makes data normalization and data cleansing harder (Varela Rozados and Tjahjono 2014). This is further exacerbated by the finding that the unstructured formats and high volume or velocity are positively correlated. Thus, an adequate data integration across all SCM data sources is a prerequisite to generate valuable insights to the organisation through SCA.

Wang et al. (2016), upon the results of their literature review, remarked that SCA, on a strategic level, is mostly applied in sourcing, supply chain network design and product design. Among its application in tactical or operational level, SCA is mainly involved in analysing and assessing supply chain performance on demand planning, procurement, inventory and logistics. Not many studies have dealt with SCA applications in the operational level, let alone predictive analytics. Flows of data stemming from RFID tags and mobile devices can be harnessed for logistics planning purposes and transforming these data with the aid of predictive analytics tool would be beneficial to bring supply chain flexibility into operational logistics tasks. One of the most common applications is the vehicle routing problem. The goal of the vehicle routing task is to optimize the sequence of visited nodes in a route, such as for a parcel delivery truck, taking into account the distances between each pair of nodes, traffic volume, left turns and other constraints. Similar practical application is proposed by Varela Rozados and Tjahjono (2014)—real time route optimization using spatial regression models as well as traffic density, weather conditions, transport systems constraints, ITS as types of data.

3. Research Methodology and Design

This thesis follows a specific research strategy to increase confidence in the study's findings. As Nenty (2009) puts it: "…research design involves procedures through which we can explore and analyse the relationship among the variables involved in our problem with minimum error while controlling for sources of extraneous variability." Using machine learning models, which will be a large part of this quantitative work, this author will seek to predict delivery times with company data as input. The step-by-step analysis will be performed based on the chosen research strategy, which will be discussed in this chapter.

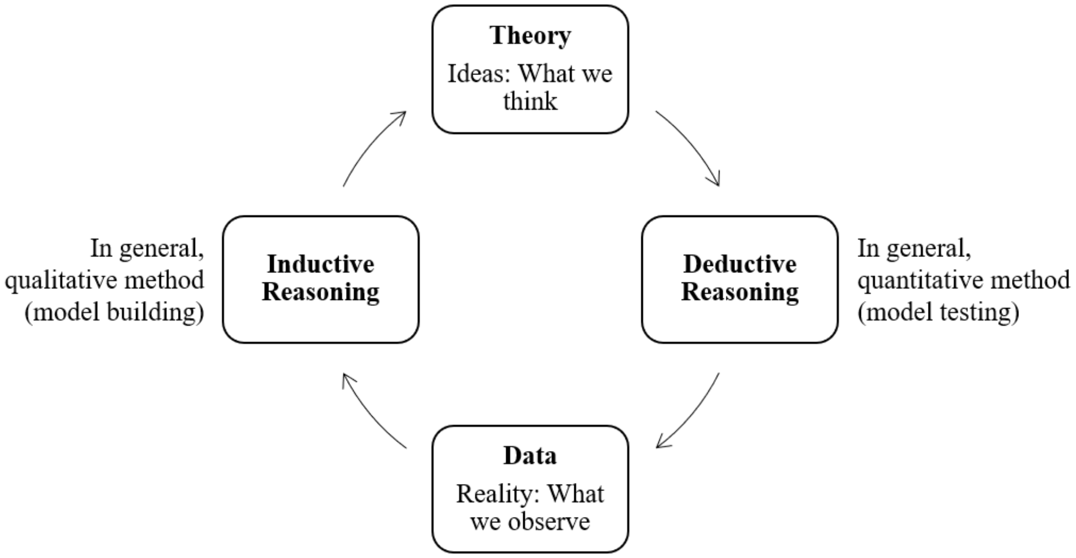
As extracted from the literature review from the previous chapter, there are many different techniques and inputs to predict travel or arrival time as well as factors that cause delay. More importantly, there is a lack of studies that focus on investigating prediction time in commercial transportation. This thesis aims to predict delivery time in e-commerce with the aid of internal company data as inputs and hopefully contribute to this research field.

The remainder of the present chapter is structured as follows: Sub-chapter 3.1 explains the methodology and design used for the data analysis; Sub-chapter 3.2 briefly presents the technologies and tools employed to collect and process the data; Sub-chapter 3.3 reviews the procedure of data gathering. Finally, sub-chapter 3.4 exposes the required data cleaning and preparation approach to transform the data into a format suitable for the ML algorithms.

3.1 Research Approach

The methodology that was followed for the purposes of this research is the deductive approach. This approach usually begins with a hypothesis and is aimed to test theories (see Figure 10). This thesis incorporates a mix of qualitative and quantitative methods with the goal of addressing the overall research purposes: a systematic literature review and a case study through data analysis using predictive analytics techniques.

Figure 10 Link between theory and data. Adapted from Straube (2018)


Theory
Ideas: What we think
Deductive Reasoning
In general, quantitative method (model testing)
Data
Reality: What we observe
Inductive Reasoning
In general, qualitative method (model building)

As a qualitative method, the systematic literature review in chapter 4 aims to gain a greater understanding of studied phenomena and concentrates more on finding opinions and experiences. By the end of the SLR, this author expects to get more ideas of the current situation in delivery time predictions in SCM using predictive analytics.

Afterwards, a case study will be conducted by analysing the data from the delivery platform company in the chapter 5. The quantitative method is used to quantify the problem through generating numerical data. Within the scope of this thesis, the results of the data analysis will show whether delivery time can be predicted using certain defined variables.

3.2 Instrumentation

The data analysis was performed with the following tools and specifications:

- Looker (a browser-based data analysis and data visualization platform)
- Microsoft Excel for Office 365 ProPlus
- Python v3.6.6.
    - Libraries include pandas, numpy, scipy, sklearn, matplotlib, seaborn, and imblearn.
- ThinkPad X240
    - Operating system: Windows 10 Pro
    - Processor: Intel(R) Core(TM) i5-4300U CPU @1.90GHz 2.49 GHz
    - RAM: 8.00 GB

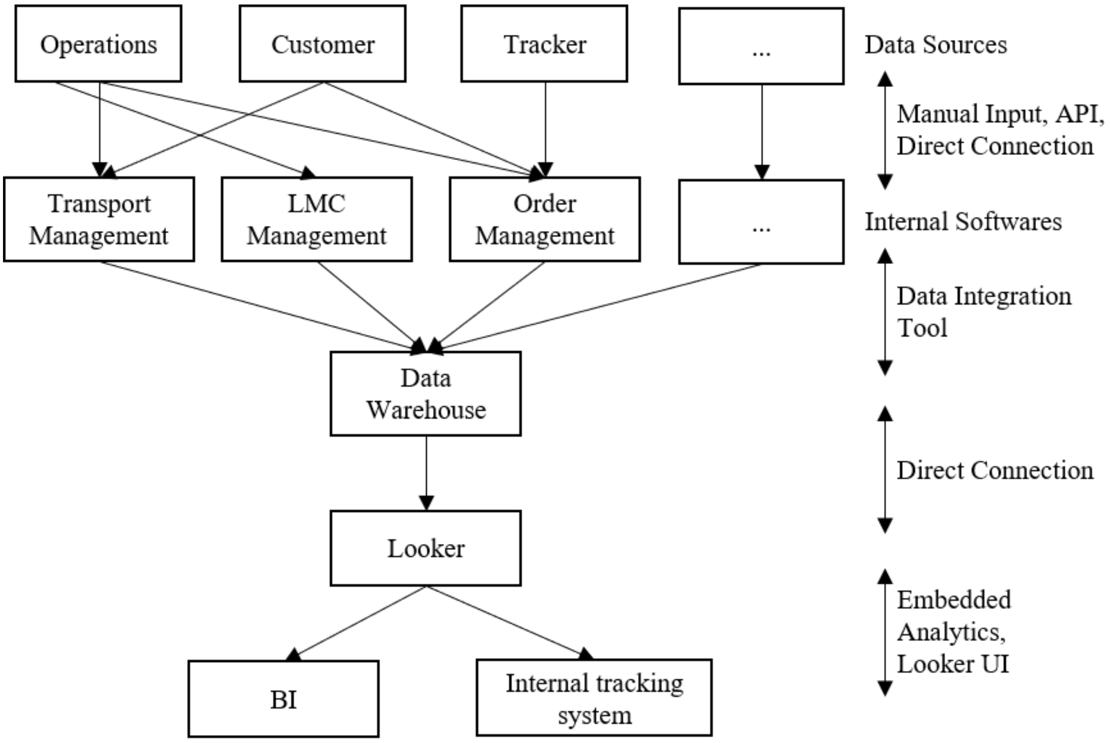
3.3 Method for Data Collection

As mentioned in the first chapter, the data used for this research was provided by the Delivery Platform and contains the shipment data from one of their clients in the food and health industry (the “Seller”). Although explicit authorization to use the data for this research was received, the companies and any identifying information related to the operations of the companies involved were anonymized throughout this paper. Any identification (ID) used in this paper are anonymized IDs assigned during the analysis, and not the actual IDs used in at the Delivery Platform.

Delivery Platform is a logistics start-up specializing in end-to-end delivery solutions for European-based online shops. The philosophy behind their business model is to revolutionize the international cross-border shipments from single touch-points to a seamless customer journey by efficiently managing the physical supply chain from clients’ warehouse to the consumer’s doorstep. Beside physical parcel delivery, they also offer a SaaS-based software solution that provides their clients with a continuous monitoring, tracking solutions and a proactive customer communication.

In a real business application, it is usually necessary to bring data together from different departments (Witten et al. 2017). Different departments have different types of record keeping and different conventions. The data must be assembled, integrated and cleaned up. This is where a standardized company-wide database warehousing plays a big role. The data for this study were extracted from the internal BI software and data analytics platform Looker. From a functional perspective, they contained information across sales, operations, freight forwarder and key account management teams. The quantitative data collection generated numerical data and was suitable if a large dataset could be collected. Figure 11 shows how the data are flown within the company’s internal and external platforms and made available in Looker.

Figure 11 Data architecture of Delivery Platform



Data obtained from different sources are transferred into the internal company's management systems and integrated into a cloud-based data warehouse. The data integration tool consolidates the data from different sources and provides users with a unified view of these data (Lenzerini 2002). The quality of the data viewed in Looker is therefore strongly dependant on the design of the data integration system, as the system is responsible to cleanse, transform and map the data for the users to use. Data in Looker is automatically updated as soon as there is any update or changes in the data warehouse.

Data collection in Looker was quite straightforward—any data on shipments or transports can be easily explored by clicking on the fields, e.g. features of the data that will be depicted as columns, and filters by non-technical personnel. The shipment data contain 27 fields as columns and filtered by the following fields:

- Customer Address Country: European countries (EU as well as non-EU)
- Shipment Created At Date: September 2018
- Shop Name: Seller

The raw data describe the shipments coming from the Seller to their European-addressed customers in September 2018. Irrelevant features, e.g. shop order-related information, are removed. For a better visualization Appendix A shows a sample of the data with changed data points with the purpose of data protection. After fields were chosen and filters were applied, Looker ran the query and the data are saved into a CSV file.

Upon inspecting the CSV file, there are 62902 entries or rows for that month alone. A preliminary data cleaning process is necessary as numerous data points are missing in several columns. To get a better overview of the data, the CSV file was imported into an Excel sheet where a descriptive analysis would be performed.

3.4 Method for Data Preparation

Data preparation has long been a hot topic in the data mining community. As Zhang and colleagues pointed out in their "Data Preparation for Data Mining" paper, "data preparation is a fundamental stage of data analysis" (Zhang et al. 2003). They added that transforming data into cleaned forms can be used for high-profit purposes and to reach that goal, there is an imminent need for data analysis aimed at cleaning raw data. Cleaning and preparing data are so time consuming, that it has been generally found that it takes approximately 80% of the total data engineering effort. Witten et al. (2017) also pointed out that preparing input for data mining "consumes the bulk of the effort invested in the entire data mining process" and one has to understand those complexities.

Real-world data are often regarded as "dirty". Zhang et al. broke down the importance of data preparation in three aspects. First, missing, noisy and inconsistent data have the potential to disguise useful patterns. Secondly, data preparation leads to a smaller but cleaner dataset than the original one, which enhances the efficiency of data mining and reduces its running time, e.g. by selecting relevant data or sampling. Lastly, data preparation promotes high quality data which results in quality patterns.

Therefore, there are enough reasons to prepare the data before analysing them, i.e. before feeding them into the ML algorithms. A two-step data preparation is used for this goal. First, a descriptive analysis of the data will be performed by using Excel to remove missing data points in key features and find correlations. Then, the dataset will be balanced and standardized using Python libraries (scikit learn and imblearn).

4. Systematic Literature Review: the Application of Predictive Analytics on Delivery Time

There are numerous motivations to conduct a systematic literature review. With the ever-increasing demand on scientific publications, academics and scholars cannot be expected to search through every single paper relevant to their work (Erren et al. 2009). Hence it is necessary to synthesize recent literature in a professional way. It is useful to describe the available knowledge as well as to identify projects and techniques of a topic (Fink 2005). Petticrew and Roberts (2006) stated that the systematic review is a scientific tool. One should get an overview of what has been done on their research issue. They even argued that one should first search for any existing systematic reviews before embarking on the task of conducting a SLR.

The previous chapters provided the necessary context and addressed the research objectives of this thesis. This chapter is organized as follows: Sub-chapter 4.1 discusses the research methodology of the systematic literature review, in which the literature locations and inclusion criteria were presented; Sub-chapter 4.2 reviews the results of the literature research and sub-chapter 4.3 concludes with the scientific implications of the literature.

4.1 Research Design

The thesis applies a systematic review approach to identify analytics techniques to predict delivery time in e-commerce. A general framework of this will be discussed intensively in this sub-chapter. An exhaustive assessment of current literature relevant to the application of predictive analytics on delivery time will be conducted according to SLR guidelines in SCM (Durach 2015). Literature is searched in multiple databases using a certain search string and around 26 articles that meet the crafted inclusion criteria are selected.

4.1.1 Literature Search

The literature search is performed through two research platforms: **Web of Science and EBSCO**. All available databases were selected in the former platform, meanwhile the latter is only set for the database **Business Source Complete**, as the other databases do not focus on fields relevant to this study. These platforms were chosen as they offer the most popular, fully indexed, curated databases in the scientific and business world (Li et al. 2018; Okoli and Schabram 2010).

Before determining the final search strings for the SLR, initial search attempts using different search strings were performed. It became apparent that there is no sufficient literature about delivery time prediction specifically for commercial transportation. The search results are mostly within the scope of arrival time predictions in public transportation, such as busses. These articles are also valid for further examinations in this research, as the techniques to predict time travel can be transferable from one transportation to the other. The search string for each research platform is constructed as seen in Table 2.

Table 2 Search strings

	Search strings
Web of Science	(TS = ("arrival time" OR "delivery time") AND TS = (prognos* OR forecas* OR estimat* OR predict*) AND TS = (transport* OR logisti*))
EBSCO	("arrival time" OR "delivery time") AND (prognos* OR forecas* OR estimat* OR predict*) AND (transport* OR logisti*)

Despite the different approaches on the search, the results of both research platforms showed a big overlap. Web of Science produced considerably more results than EBSCO, as it searched more than one database. Business Source Complete covers about 2000 full-text journals and magazines, meanwhile Web of Science covers about 18000 journals in total from all its databases (EBSCO 2017; Clarivate Analytics 2017).

4.1.2 Literature Selection

To protect the objectivity of the SLR, the review should be conducted through a certain protocol, i.e. a plan with explicit descriptions of the steps to be taken (Tranfield et al. 2003). This thesis intends to answer the defined research question about what analytics techniques are employed to predict shipment delays as well as to come up with a conceptual framework for actions and communication system within the organization. Therefore, this SLR can be interpreted as a process of exploration, in which a flexible protocol would be suitable as it can be modified throughout the study.

Due to the above-mentioned lack of papers about delivery time prediction in commercial transports, the inclusion criteria are modified to also include travel time prediction within public transportation, as seen in Table 3. One reason to do so is that the techniques to predict time travel can be transferable from one transportation to the other. The review is conducted solely by the author. After going through the inclusion criteria, 24 papers are selected.

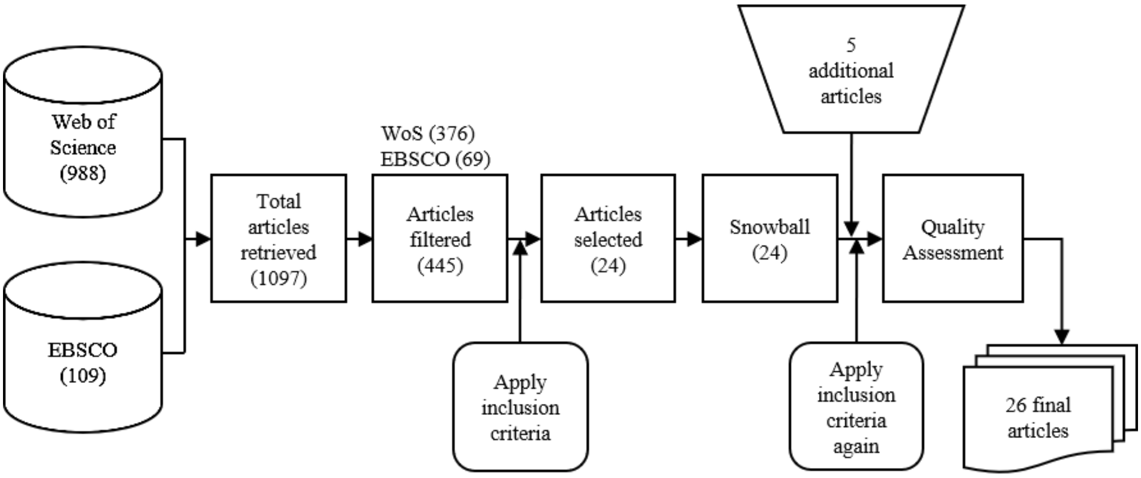
Table 3 Inclusion criteria

	Inclusion Criteria
1	Abstract must contain analytics as the focus of the research
2	Abstract must mention the role of forecasting or prediction on transportation or logistics with historical data
3	Abstract must show indication of impact of analytics on delivery, arrival or travel time
4	Article must be written in English

Using the reference program Citavi, the citations and bibliographical information of each selected papers are imported and managed.

Figure 12 shows the literature selection process. With 988 and 109 search results on Web of Science and EBSCO respectively, these had to be refined by multiple features, such as document types, publication years and research areas. This step helped to filter out old and irrelevant papers.

Figure 12 Literature selection process



As previously mentioned, there is a complete overlap between the search results of both platforms. All the filtered 69 EBSCO articles are to be found in the 376 articles in Web of Science.

A snowballing process is performed to search for possible missing papers. Snowballing refers to the use of the reference list of a paper or the citations to the paper to identify additional sources (Wohlin 2014). In this context, forward snowballing was carried out, where new papers referenced in the initially selected papers are identified. Afterwards, the same selection process is performed, i.e. title, abstracts and keywords of the 5 snowballed papers are analysed using the inclusion criteria. Finally, a total of 26 literature are selected (see Appendix B).

4.1.3 Literature Synthesis

In order to reduce human bias and error, a literature data extraction form is employed (Tranfield et al. 2003). The 24 studies were analysed and entered into a Google spreadsheet containing general features, such as topic, problem, goal, methods, techniques, challenge and conclusion of each paper. Having the summary of all studies collected in one sheet lead on to a clearer overview of the literature results, which simplified the literature synthesis.

From the selected literature, 22 studies addressed the arrival or travel time prediction of public transportation, with 21 studies delving into bus arrival time predictions. Only the two remaining papers dealt with commercial transportation, i.e. truck arrival time prediction in certain distribution centres and delivery time prediction in freight rail networks. Studies on delivery time prediction on commercial transportation are visibly scarce. In addition, the term *predictive analytics* is not commonly used or even mentioned in 23 papers. Instead, all of them use the term *machine learning*. As previously discussed, ML is widely deemed in today’s world as an outstanding predictive analytics technique due to its effective algorithms and frameworks that result in a great predictive accuracy. Simultaneously, predictive analytics are a part of the ML domain which is limited to predicting future outcome from data based on previous patterns. Therefore, there is an overlap in both fields. For the sake of simplicity, this thesis took into account studies that mention one or both terms.

4.2 Results

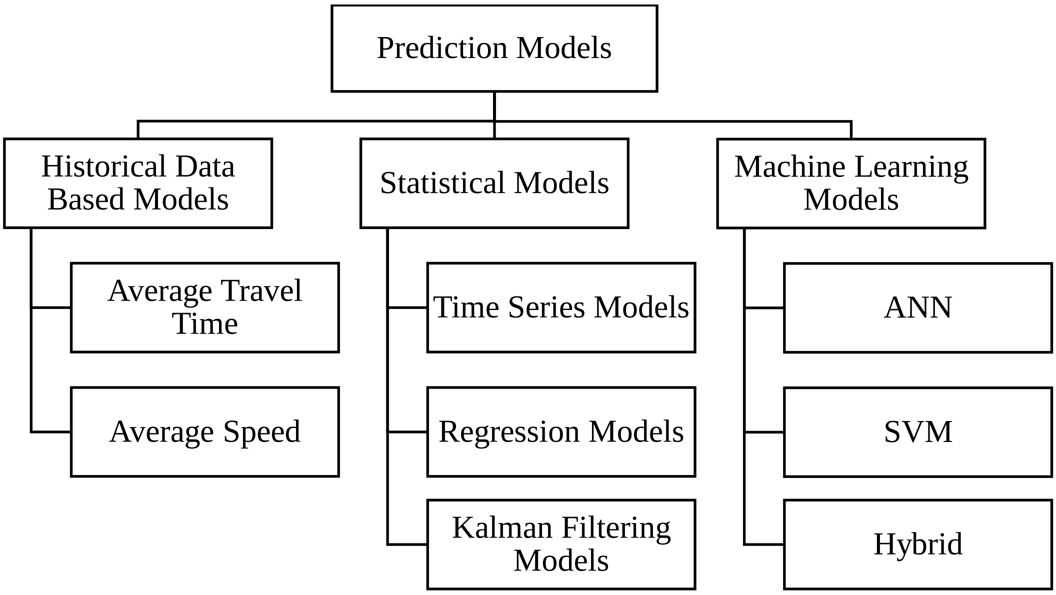
Based on the information synthesized from the SLR, this sub-chapter seeks to review the results of the literature analysis related to the research objectives, previously mentioned in sub-chapter 1.2.

RQ1: What predictive analytics techniques, based on historical data as input, can be used to predict delivery time?

Due to the scarcity of papers about predictions in commercial transports, the topic of forecasting delivery time is rarely to be found in the SLR. Techniques used in predicting arrival or travel time were analysed instead. Before delving into specific prediction methods, several researchers have published various groups of forecasting models and its potential use-cases. Kumar and his fellow researchers divided methods employed to predict arrival or travel time into two categories: data-driven and less data-driven techniques (Kumar et al. 2017b; Kumar et al. 2017a; Kumar et al. 2014). Data-driven techniques are those that have a historical approach, i.e. use historical data, and handle large scale of datasets. These encompass all machine learning techniques. Whereas less data-demanding techniques are model-based approaches suitable for real time implementation, as they require less data. However, the key to good prediction accuracy lies on how good the inputs are, irrespective of type of technique used. Therefore, the identification of which inputs are significant for a prediction model is also a crucial step.

Another classification was proposed, in which forecasting models are clustered into three groups as shown in Figure 13 (Treethidtaphat et al. 2017; Altinkaya and Zontul 2013; Choudhary et al. 2016). Models based on historical data are suitable for areas in which traffic patterns are stable, as predictions are built from historical travel time of precedent journeys (Choudhary et al. 2016). Statistical models incorporate independent and dependent variables that affect travel time and are expressed in mathematical functions (Altinkaya and Zontul 2013). ML models excel in dealing with complex, non-linear relationships between predictors within a huge amount of data and in processing noisy data.

Figure 13 Types of prediction models



From the selected literature, ML models are the most common method used for arrival time prediction. This is probably due to the unpredictable nature of travel and the sheer amount of data that must be processed.

Furthermore, specific prediction models are going to be explored in the next part of the sub-chapter. In Appendix C, prediction techniques discussed in the papers are listed, along with the frequency of appearance in the studies.

Artificial Neural Networks

Across all the selected studies, the ANNs technique is the most commonly written ML technique, i.e. 13 of the studies analysed it. However, only 5 of them are in favour of the technique, while in the other 5 studies the ANN’s predicting performance and accuracy did not perform the best, when compared with other models. Alongside the traditional ANN, different NN-based models were also discussed in the selected literature: Random Neural Networks (RNNs), Deep Neural Networks (DNNs) and Radial Basis Function Neural Networks (RBFNNs).

ANN is a general term that scientists use to refer to predicting models that are based on neural network, a massively parallel distributed processor made up of simple processing units that have a natural propensity for storing experimental knowledge and making it available for later use. In this study, conventional NN-based models are simply called ANN. Kumar et. al (2017a; 2014) and Yaghini et al. (2013) showed that their ANN models outperformed other predicting models, under certain circumstances. In Kumar et. al's experiment with data collected through GPS from a bus route in India, they compared the performance of travel time prediction based on the data-driven technique Artificial Neural Network (ANN) and model-based approach Kalman Filtering Technique (KFT). KFT is a widely used estimation technique to predict traffic parameters such as density, travel time, etc. It was observed that ANNs only performed better if the input data amounted to at least two weeks. They concluded that the performance of the prediction selection depends on the database size and quality. Yaghini and his colleagues compared the results of three data inputs (i.e. normalized real number, binary coding and binary set encoding values) in three different network architectures (i.e. quick, dynamic and multiple method) and with three predicting models: ANN, decision tree and multinomial logistic regression. Quick method referred to a method, where only a single ANN is trained, whereas the dynamic method had a changing topology during training, with neurons adjusted to improve accuracy. Lastly, in the multiple method, multiple networks were trained in parallel, where the model with the highest accuracy won. Their outcomes revealed that the ANN model is a great solution as it had the greatest accuracy and a low training time.

Chen (2018) proposed random neural networks to randomly train several ANN models as an alternative to the traditional ANN to predict the arrival time of motor carriers. RNNs were developed to mimic the spiking behaviour of biological neurons in the brain (Timotheou 2010). Contrary to most artificial NN models, neurons interact with each other via excitatory and inhibitory spikes which modify each neuron's action potential in continuous time (Yin 2018). When a biological neuron is excited, it transmits a set of signals, or spikes, along its axon to either excite or inhibit the receiving neurons. Hence, this combined effect of excitatory and inhibitory inputs modifies the potential level of the next neuron and determine whether it will become excited. In her study, Chen's RNN model outperformed previous statistical approaches and datamining methods, which included the statistical mean value, logistic regression and backpropagation ANNs, with a 94.75% accuracy in highways and 78.22% accuracy in urban roads.

A Deep Neural Network (DNN) model was proved superior by Treethidtaphat et al. (2017) to develop a bus arrival time prediction at any distance along the route. A DNN model is an ANN model with multiple hidden layers between the input and output to deal with the large and complex variable relations. Using a public bus data collected by GPS, they compared the result of this model with the one of ordinary least square (OLS) regression model—it showed that DNN model outperforms OLS prediction for the three measures: MAE, RMSE, and MAPE.

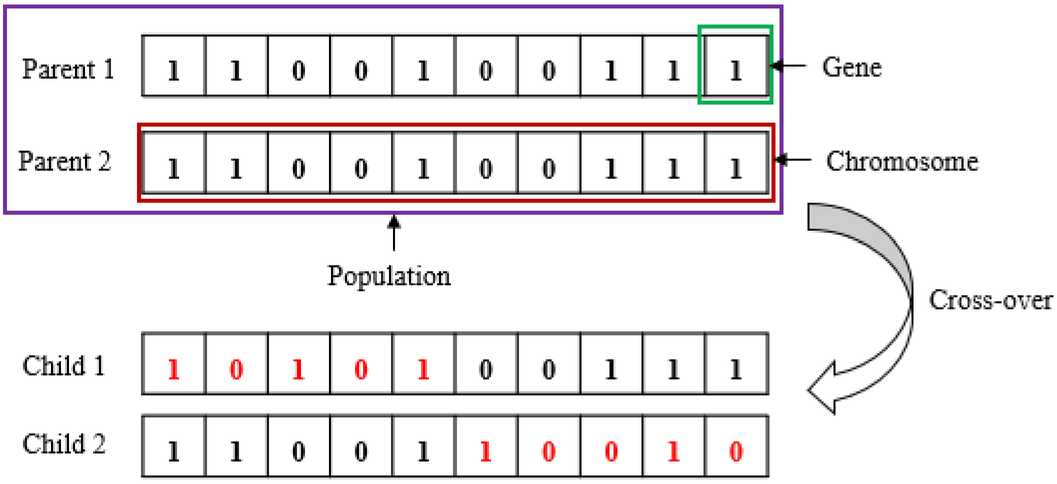
Combining historical and real-time situation data to forecast bus arrival times, Wang et al. (2014) applied the RBFNN model in their two-phase approach. First, RBFNN model is used to approximate the non-linear relationship in historical data then real-time information using KFT is fed to the system to adjust the actual situation, which would modify the predicted result of the first phase. RBFNN is an ANN that uses radial basis function as activation function which can provide a global approximation to the target function and it typically has a structure of only three layers, i.e. one hidden layer. In contrast with multiple linear regression, backpropagation NN and conventional RBFNN, the adjusted RBFNN showed achieved the lowest MAPE.

Support Vector Machines

With the sum of 8 papers, SVMs are the second most common time prediction method. Yang et al. (2016) argued that SVMs partnered with genetic algorithm (GA-SVM) showed a more accurate prediction than other methods, such as the traditional SVM and ANN model. Additionally, Peng et al. (2018) showed that pairing Principal Component Analysis and genetic algorithm with SVM (PCA-GA-SVM) provided even higher prediction accuracy than GA-SVM.

Inspired by Darwin’s natural evolution theory, GA provides a learning method where successor hypotheses (analogous to chromosomes) are generated by repeatedly mutating and recombining parts (analogous to genes) of the fittest hypotheses (Mitchell 1997). A visual explanation can be seen in Figure 14. At each iteration, a set of hypotheses called the population is updated by replacing some parts of the population by offspring, or child, of the best hypotheses.

Figure 14 Genetic algorithm



Hypotheses are represented by bit strings. The hypotheses contain a set of parameters or variables known as the genes. The best hypothesis is defined as the one that optimizes the given problem called fitness function. Hence, the fitness function is the function that the algorithm is trying to optimize. GA is therefore a randomized search method to increase the speed and optimality of the parameter selection (Yang et al. 2016). In their GA-SVM study, Yang and colleagues optimized the penalty parameter *C* and kernel coefficient γ then applied these optimized parameters into the SVM model. The GA-SVM model resulted in the lowest RMSE when it was compared with NN and conventional SVM method.

PCA is a popular multivariate technique to reduce the dimension of characteristic index of the data using inter-correlated quantitative dependent variables to describe features or observations (Abdi and Williams 2010). The extracted important information from the data table is expressed as a new set of new orthogonal variables called principal components. By compressing the size of the dataset, it simplifies the description of the dataset itself and also shortens the time of training. Peng et. al modelled a 3-step prediction process in their PCA-GA-SVM study. First, PCA was applied to compress the dimension of the SVM's training data samples. Secondly, GA was performed to determine parameters, such as the penalty coefficient *C* and RBF kernel function value γ, which would then be fed into their SVM model. Last, a dynamic slippage was applied to adjust the time, so that the model could retrain sample and continue to predict when forecasting the next bus stop. By comparing the prediction results with those of GA-SVM, variable-parameter state-space model, and traditional SVM, PCA-GA-SVM always performed the best measured by effectiveness indicators MAPE, MAE and RMSE.

***K*-NN**

K-NN can be used as a classifying algorithm to select prediction inputs by identifying trips from the historic data that share resemblance to the current trip (Kumar et al. 2018). The output of the algorithm would be historical trips that have a similar pattern as the current trip. In order to do so, a sensitivity analysis is carried out to determine the optimum number of previous trips that happened before the current trip. Analysing the MAPE behaviour, the optimum number was 4 historical trips. That means, 4 previous consecutive trips were taken as input for the current trip estimation. By calculating the Euclidean distance between the current trip and the 4 historical trips, similar trajectory patterns could be identified and used for the estimation step. The algorithm can update itself for each trip as a function of the previous trips in real time. Regarding this particular study, the *k*-NN model was not used as the estimation method, but only as an input selection classifier.

One study focussed on the discussion of a modified *k*-NN model as a bus arrival time predictor. Liu et al. integrated a *k*-NN method with cluster analysis and PCA, using historical GPS data as input (Liu et al. 2012). The data acquired through GPS is incredibly large in order for it to cover all the possible bus arrival time patterns, but at the same time data redundancy would slow the runtime heavily. The clustering method was useful to improve this trade-off. The PCA came in handy to reduce the dimensionality of the state vector.

Other models

Hybrid models have in recent times appeared as a positive prediction approach, but their deployment is hindered by the inherent complexity of these models and no hybrid prediction model has been developed with the goal of real-time bus travel time prediction (Fadaei et al. 2016). Fadei et al. proposed a hybrid model that is composed of a linear combination of schedule, instantaneous and historical predictors. The model required estimating the weight of each predictor and parameters associated with the instantaneous and historical data, which are collected by AVL device.

Similar concept of using different data sources is also developed by Agafonov and colleagues (Agafonov et al. 2015). They used real-time, historical and timetable data for their linear regression model and concluded that calculating separately coefficients for each road network segment and taking delays in real-time data into account achieved the best prediction results. Chen et al. (2013) evaluated their proposed regression-based model by comparing it with a historical-based model and came to a conclusion that the regression-based model achieved a better prediction after buses have passed the 7th bus stop due to the stability of the estimated regression parameters after the 7th bus stop.

RQ2: What are the factors that have an impact on delivery time?

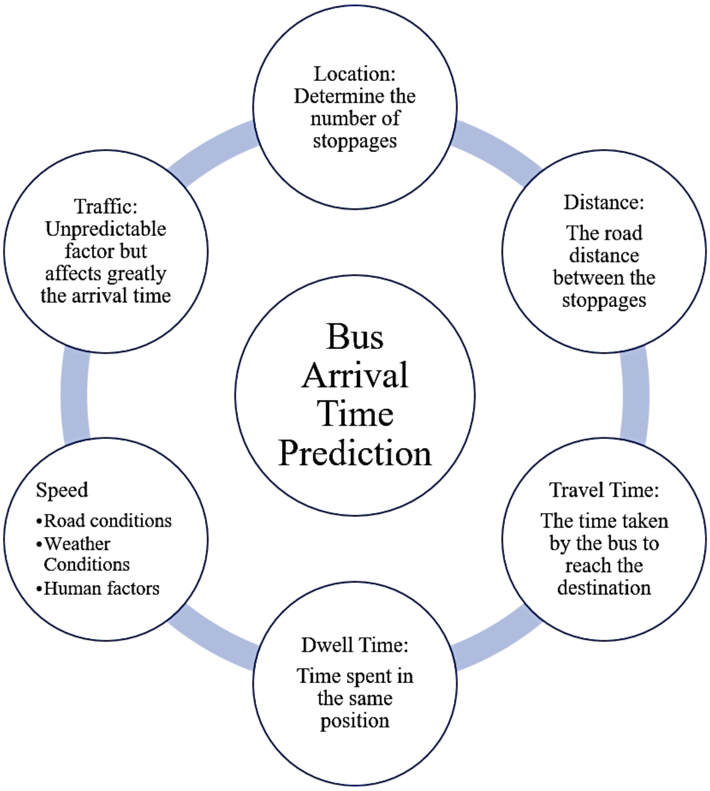
Analogous to the previous research objective, delivery time would be substituted to arrival time or travel time due to the scarcity of literature on commercial transports.

The papers offer different approaches, theories and models, which come with different eminent features for the prediction. Many of the selected literature have based their prediction on historical travel time of prior journeys on the same span of time (Kumar et al. 2014; Sun et al. 2017; Wang et al. 2014; Maiti et al. 2014; Chen 2018; Li et al. 2017). This basis is only reliable in places with traffic congestion is minimum. Unfortunately, that is not the case in real-world traffic. To tackle this problem, real time data is taken into account in the prediction calculation (Maiti et al. 2014; Fadaei et al. 2016; Sun et al. 2017). Real time data include road traffic condition, weather, time of day, etc. In the example of the bus arrival time prediction conducted by Wang et al. (2014), a two-phase method is employed: training an RBFNN model using historical data and then adjusting the baseline data from the RBFNN model by developing an online filter method to import the instant speed in real-time.

Chen et al. (2013) stated that there are four types of factors that contribute to travel time. First, there are the *infrastructure factors*, which include the number of stops, the number of signal-controlled intersections and the length of segment. Second, there are the *external conditions*, such as weather and traffic conditions. Then, *driver behaviour*, e.g. schedule recovery behaviour, and lastly *operation management*, such as the timetable and schedule.

Choudhary et al. (2016) summarized the factors that impact on forecasting the arrival times on busses (see Figure 15).

Figure 15 Factors to bus arrival time prediction

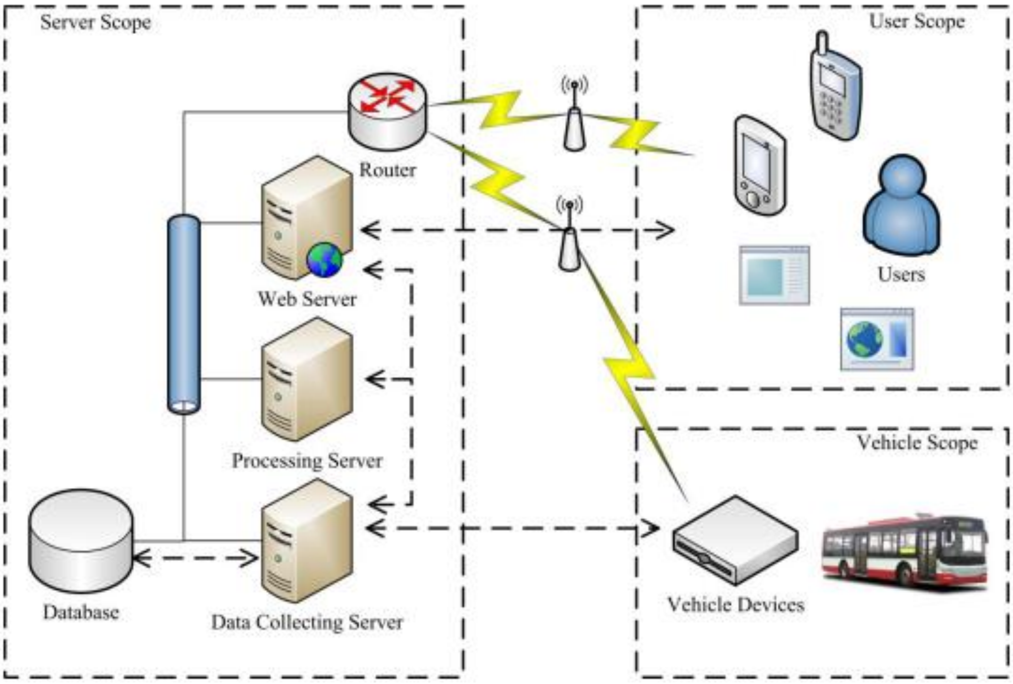

Location: Determine the number of stoppages
Distance: The road distance between the stoppages
Travel Time: The time taken by the bus to reach the destination
Dwell Time: Time spent in the same position
Speed
•Road conditions
•Weather Conditions
•Human factors
Traffic: Unpredictable factor but affects greatly the arrival time
Bus Arrival Time Prediction

RQ3: What courses of actions and restructuring of the analytics or IT system should be undertook in the organization to ensure on time delivery time?

As almost all the literature deals with time prediction in public transportation, there are no concrete courses of actions or any structure of communication system recommended for private businesses to ensure delivery times. However, some of the papers give advice regarding the computer software or communication system architecture, which can potentially be adapted to commercial applications.

To obtain high quality data in transportation, it is key to optimize the equipment routing, e.g. AVL in public transportation. These devices are installed on buses or trains in service and collect archive the stop-level information, such as vehicles arrive at a stop or leave it, in real time (Chen et al. 2013). In their experiment, Wang et al. (2014) designed a computer system architecture to implement their bus arrival forecasting approach (see Figure 16). The design is divided in three main components: *Server Scope* that handles the data from the vehicle and users, uses the data to train the model automatically and does online prediction for any request; *Vehicle Scope* that collects bus vehicle operating data; and *User Scope* which provides the interface for the users in order to satisfy their request for any specific bus arrival time prediction.

Figure 16 Computer system design (Wang et al. 2014)



4.3 Insights from the Literature Reviewed

This chapter analysed a set of literatures on predictive analytics techniques and factors for delay prediction that have been studied and published since 2009. A sample of 26 articles is selected for further review. The SLR has shown that the current studies of predictive analytics in delay prediction are weighted towards the prediction of arrival or travel time in public transportations, mainly busses. There is a big lack of studies in delivery time prediction specifically for commercial transportation. The main points of differences between public and commercial transportation are that the public transportation focuses on:

- Public transportation vehicles, such as busses and trains, instead of commercial delivery vehicles
- Arrival or travel time prediction instead of delivery time prediction

The technical knowledge regarding both predictive analytics methods and factors in delivery time prediction are transferable between these two types of transportation.

For the first research question, it is shown that there is a plenty of ML models as well as other predictive analytics techniques that were successfully employed to predict arrival or travel time, such as *k*-NN, SVM and ANN. However, not all articles unanimously agree on the same algorithm as the best predictor. There are even articles that contradicts each other in their analysis results. These ML models will be further studied in the data analysis chapter, in order to predict delivery time.

For the second research question, the transferability of the literature review results to the commercial transportation domain is a bit problematic. The factors that impact the arrival or travel time are collected real time through AVL devices, with which the public transportation vehicles are equipped. Meanwhile, the commercial trucks are not equipped with this data collection devices. External factors, such as weather and traffic conditions, are also hard to be taken into account in the delivery time prediction within the scope of this thesis. This is due to the large area the e-commerce deliveries cover, as the object of case study is a delivery platform that specializes in European-wide cross-borders deliveries. Therefore, the factors that contribute in the prediction of a city public busses may not be the same as the ones that impact cross-border European commercial deliveries.

Lastly, the lack of management recommendations regarding what courses of action had to be taken and how the communication system should be designed within an organization to ensure on-time deliveries (see RQ.3) is due to the discipline that mostly experiments with this type of predictions. Articles that study the arrival or travel time prediction on public transportation are likely published in ITS-related periodicals, and not in business or SCM journals. Therefore, they are unlikely to offer organizational and business recommendations.

In summary, the selected literature on public transportation has showed that machine learning techniques are suitable to predict arrival time, but the factors for arrival time prediction for short-distance trajectories, which are mostly the case in public transportation, might not be the same for long-distance commercial transportation. Recommendations regarding courses of actions or organizational communication system to ensure on-time deliveries are hardly provided through this SLR.

5. Delay Prediction: The Analysis

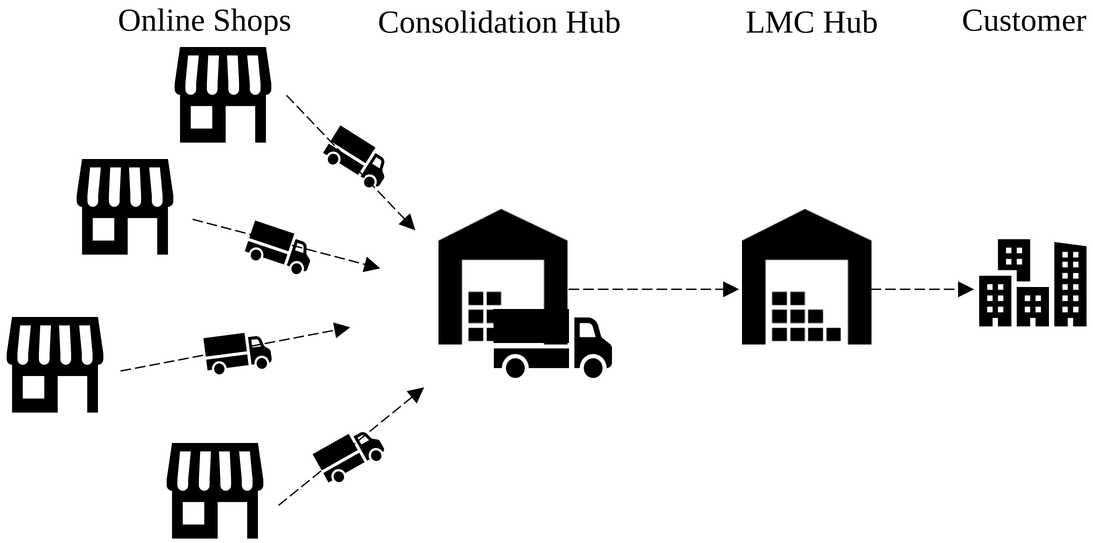
The previous chapter has shown that there is a lack of available bodies of literature that have dealt with delivery time prediction in the domain of e-commerce. However, ML models have been proven to be good techniques to forecast arrival or travel time of busses and trains. From these insights, the chapter aims to show quantitatively whether ML models are applicable to predict delivery time in parcel deliveries, using the internal historical data from the delivery platform company. An extensive data analysis will be performed in a detailed manner and the results will be summarized.

The first sub-chapter introduces the current supply chain within the Delivery Platform. An Excel data pre-processing is carried out in order to see the data better as it is formatted in a tabular form, then followed with the exploratory data analysis to check correlations (sub-chapter 5.2). Sub-chapter 5.3 describes additional steps for data pre-processing, such as variable encoding, data splitting, synthetic resampling and standardizing. After the data is cleaned, the ML models are built, as described in sub-chapter 5.4. Sub-chapter 5.5 explains the application of *K*-fold stratified cross-validation on the built models and compares the results to the ones from the previous sub-chapter. The best performing ML models are selected for hyper-parameter tuning and feature importance identification in sub-chapters 5.6 and 5.7, respectively. Finally, the findings of this chapter's quantitative data analysis is summarized in sub-chapter 5.8.

5.1 Understanding the Current Supply Chain

The main cost-effective aspect of the business is the consolidation of parcels originated from various German-based online shops into one truck, subsequently feeding them into existing local carrier networks. The business idea is based on building their own virtual premium network from existing transport networks, which are specialised regionally, but also limited locally, and extending it all over the world. Hence, the company has partnered with a logistics firm that offers a consolidation hub and with more than 100 carriers worldwide. Figure 17 visualizes the supply chain of the Delivery Platform.

Figure 17 Supply Chain of Delivery Platform



The parcels are scanned in each touch point and recorded in the tracking system, which can be viewed real time through the company's software solutions and BI platform. There are two ways to handle the incoming parcels from the shops: consolidation in a pallet or PiP (Parcel in Parcel). Pallet consolidation, as the name suggests, is the consolidation of parcels either from one or from multiple online shops into one pallet. PiP is the consolidation of parcels into one larger package, large enough to still ship them through courier vans. The selection of packaging type depends on the types of products and the shop's requests. Generally, pallet consolidation enables shops to ship out large amount of their products and lower their shipping costs, especially if they send out heavy or bulky shipments. PiP, on the other hand, is suitable for shops that ship out light and small parcels, as huge quantity of the parcels is needed to fill in a palette to make it worth it; for parcels that have to be delivered quickly; and also for parcels that are shipped to a regions where there are yet few parcels that go there. A good example of shops that usually employ PiP are shops that sell smartphones and laptop cases. Most of them do not have enough orders in a day or two to fill one pallet and they do not have robust packaging that support pallet transportation.

PPU is the planned time a parcel should be picked up from the shop's warehouse. FHS is the point of time at which a parcel is scanned in the consolidation hub, preparing it for the journey to the LMC hub. FDA is the point of time in which the parcel is delivered to the end customer's door for the first time. Delivery time is the point of time, in which the end customer receives the parcel. Therefore, FDA does not always mean delivery time, as the parcel can also be not received in the first delivery attempt for whatever reason, e.g. customer’s absence, refusal of acceptance, etc. There are cases of big difference between FDA and delivery time, reaching up to 42 days. An explanation for this phenomenon is entry error or lost parcel. The shipping time variables could be good features to investigate the causes of shipment delay.

5.2 Initial Data Pre-processing and Exploratory Data Analysis

It is a good practice to understand the data first and try to gain insights from it. EDA is a statistical strategy to that provides conceptual and computational tools for discovering patterns with the goal of discovering patterns in data, which stemmed from the area of psychology in the early 1960s (Behrens 1997). The technique is characterized by multiple aspects, such as that it puts an emphasis on the understanding of data that address the broad question of “what is going on here?” and an emphasis on visual representation of data. EDA relates to detective work, a view that its founder John Tukey regarded. Yu (2017) in his paper “Exploratory Data Analysis” remarked that the role of a researcher is to explore the data in as many as possible until a plausible “story” comes to the surface. It is, therefore, a systematic way to investigate a situation from multiple perspectives.

EDA plays a big role in data mining as it eases the tediousness and arduousness of detecting data patterns in large and wide datasets. Beside discovering patterns, it can help to spot anomalies, to test hypothesis and to check assumptions using descriptive statistics and visual representations. In other words, EDA is about making sense of data in hand, before polluting them with it.

5.2.1 Check Missing Values

From the initial approximately 63000 rows of dataset, there was a huge disparity between on time and delayed shipments, with around 58000 and 4500 shipments, respectively. A shipment is considered delayed if the delivery time surpasses that of the SLA.

After removing duplicates, the next step would be processing missing data, which became apparent as soon as the CSV file is imported into Excel (see Table 4). All the 942 rows without shipment tracking codes also did not have shipment delivery, FDA and FHS time stamps. These rows would not be any use for the prediction because of the grave lack of information.

As the study focuses on predicting delivery time as well as finding causes of shipment delays, it is crucial to choose solely rows that don’t have missing data points in any of the time stamps columns. Table 4 shows the amount of missing data points before and after removing rows with missing data.

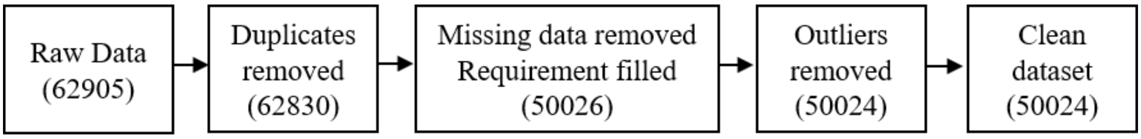
Table 4 Overview of missing data points

	Number of rows with missing data points in…	
Columns	Raw dataset	Cleaned dataset
Shipment Tracking Code	942	0
Customer ZIP Code	0	0
Customer Address City	2	2
Customer Address Country	116	81
Carrier Company and Country	943	0
PPU Time	5	0
FHS Time	3095	0
FDA Time	3423	0
Delivery Time	3564	0
Shipment Created Date	5	0
Shipment Status	5	0
Shipment delayed?	5	0

The dataset contained rows with empty Customer Address Country. In practice, carriers only perform business within their own country as stated in the column Carrier Company & Country, e.g. LMC1 – DE: LMC1 only does deliveries in Germany. Missing address countries can therefore be filled with values from Carrier Company and Country using a simple Excel string formula. However, there are exceptions: deliveries addressed to small countries, e.g. Liechtenstein or Monaco, are performed by the LMC operated in the closest country the “Delivery Platform” partners with. Fortunately, all the missing address countries are the ones that have their own operating LMCs.

Several row removal attempts were made in Excel but it did not take long to find out that the author’s CPU did not support large data processing. This data cleaning was performed using Python’s pandas library instead.

Figure 18 Data cleaning process


Raw Data (62905)
Duplicates removed (62830)
Missing data removed Requirement filled (50026)
Outliers removed (50024)
Clean dataset (50024)

5.2.2 Feature engineering

Upon searching literature on feature engineering, the author realized that there are almost no papers or chapters in books that are dedicated in this topic. Scott Locklin, a Machine Learning and Blockchain Engineer, wrote in his blogpost "Neglecting machine learning ideas": "feature engineering is another topic which doesn't seem to merit any review papers or books, or even chapters in books, but it is absolutely vital to ML success." (Locklin 2014). Despite of the scarcity of literature, the performance of machine learning methods is heavily dependent on the choice of data features on which they are applied, as other researchers acknowledged. It is common practice to engineer new features from the existing ones, which might be ratios, differences or other mathematical transformations (Heaton 2016). Guyon and Elisseeff (2003) argued that feature engineering facilitates data visualization and data understanding, reduce measurement and storage requirements as well as training and utilization times and challenge "the curse of dimensionality" to refine prediction performance. Feature engineering is more than just feature selection. It is the process of extracting features from a raw dataset, transforming these features into ones which are easier to interpret and that better represent the underlying problem to the predictive models (Brownlee 2014).

The purpose of the study is to examine what variables play in the delay of shipments. The time stamps variables, such as PPU, FHS and FDA, do not strongly correlate to delay, but perhaps the difference between these features do. The extraction of new features representing various shipping time duration variables would make the dataset easier to interpret.

The time stamps columns are compared with each other by inserting new columns in the data. The specific variables are explained in Table 5 as follows.

Table 5 Shipping duration variables

Column Name	Description	Formula	Requirement
PPU - FHS	Elapsed time between pick up time and injection in the consolidation hub	PPU – FHS	> 0
FHS - FDA	Elapsed time between injection in the consolidation hub and first delivery attempt at the customer address	FDA – FHS	> 0
FDA-Delivery	Elapsed time between first delivery attempt at the customer address and parcel received	Delivery Time – FDA	> 0

Beside obtaining new features, another benefit of extracting shipping duration variables is to check whether the data are correct. Logically, injections in any hub can impossibly occur after the parcel arrives at the end customer’s door, or vice versa. Unfortunately, the data contained these impossibilities, but this is obviously not the truth. In practical, logistics operations contain human errors. In a lot of these cases, the actors of the supply chain, whether they are packing, hub or LMC staff, did not record the parcel data correctly in the database or even forgot to scan the parcels, among other reasons. Rows that did not fill the requirement were removed from the dataset, reducing the dataset to 50024 rows, as seen in Figure 18.

Some massive outliers were found in column PPU-FHS with 1046 days. These were removed.

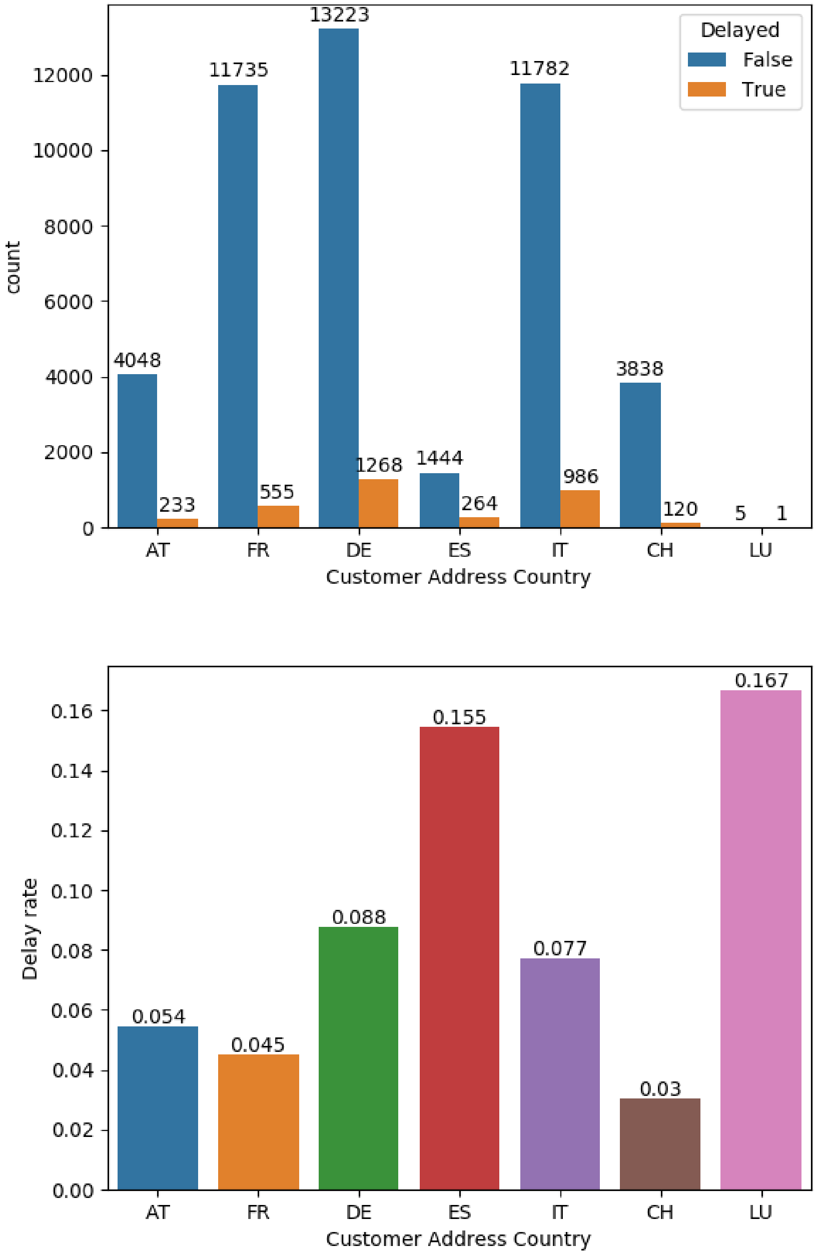
5.2.3 Correlation of Delay with Other Variables

In this sub-chapter, the correlation of various features with the delay is examined.

Customer Address Country vs. Delay

The cleaned data contains 15 countries in total, but delayed shipments only occurred in 7 of them. Figure 19 displays the delivery timeliness in the 7 countries. The first 5 countries (Austria, France, Germany, Spain, Italy and Swiss) are the ones with the most shipping volume. One out of 6 parcels shipped to Luxembourg was delayed, which resulted in a high delay rate of 0.167. Delay rate of 10% in a month is much, as there are 100 parcels delayed for every 1000 parcels shipped. Furthermore, Spain shows to be the second highest in the ranking, with more than 15% of shipments to Spain to be delayed. Germany, despite being in the third place, has the highest on time deliveries with 13,223 shipments.

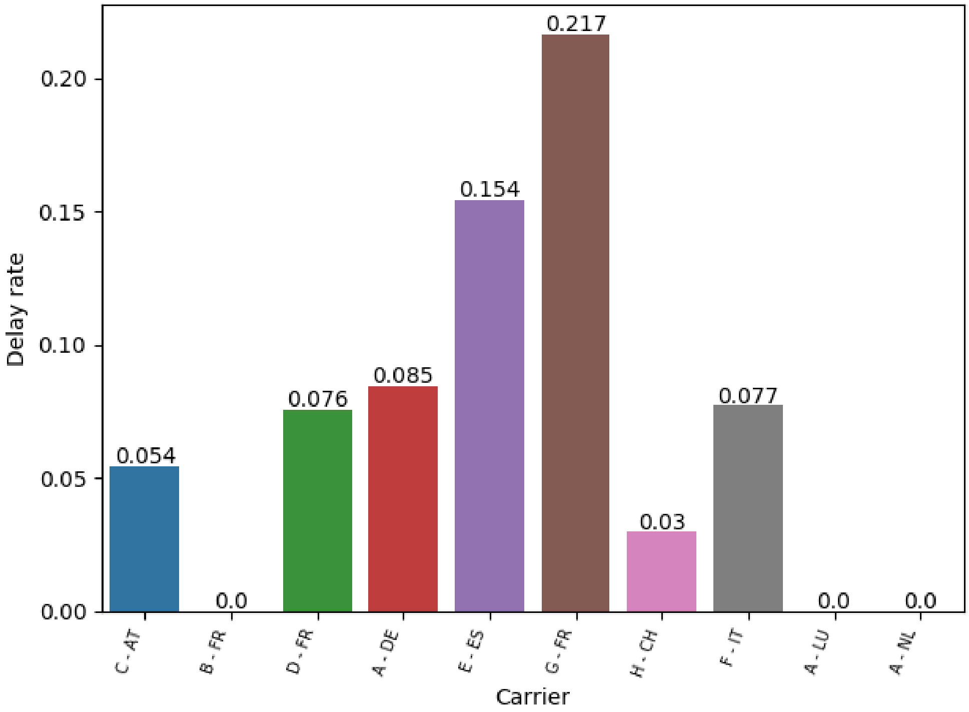
Figure 19 Delay by countries



Carrier vs. Delay

In Figure 20, there is a strong correlation between the selection of LMC and delay rate. It is shown that the parcels shipped with the LMC G - FR are more likely to be delayed than the ones shipped with any other LMCs (around 21% of all G - FR shipments). This is an interesting revelation due to France having one of the lowest delay rate seem in Figure 19.

Figure 20 Delay by carrier companies

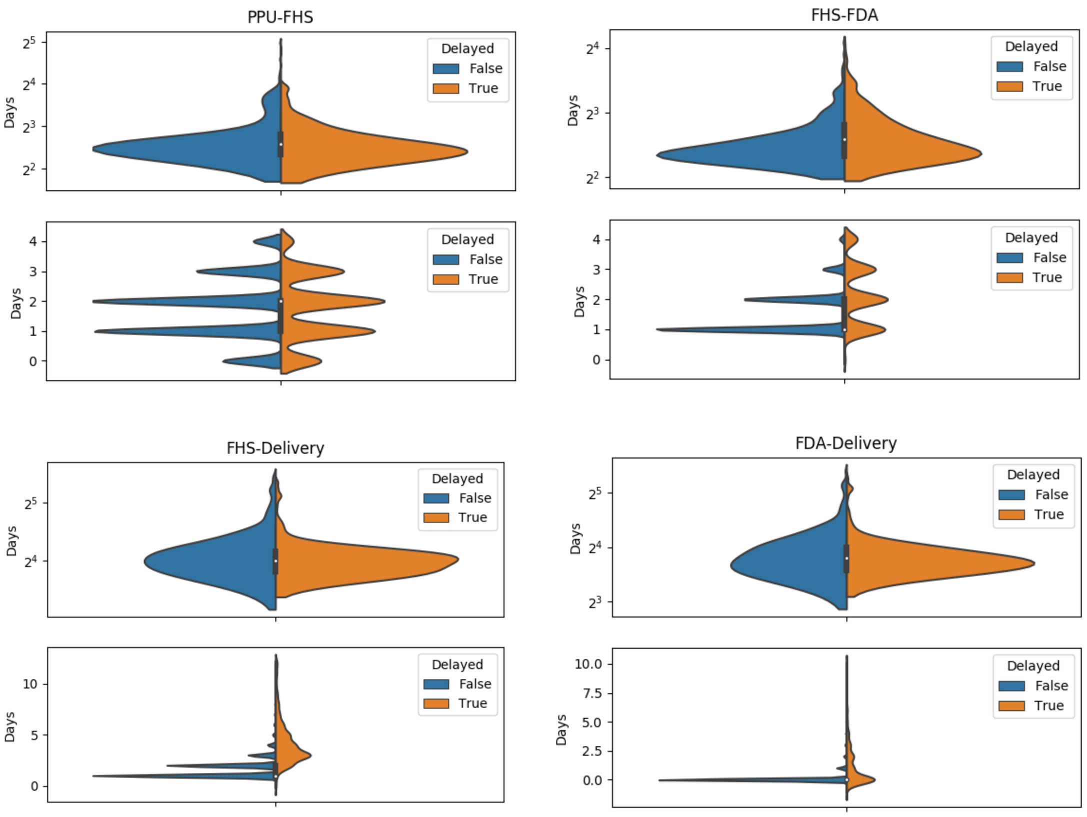


Shipping Duration Variables vs. Delay

Correlations between the shipping time features and delays were also examined and represented in violin plots, as seen in Figure 21. PPU-FHS varied from 0 to 32 days, where most of the shipments (99% quantile) had a duration of 0 to 4 days. For this reason, each shipping duration plot is divided into two: the top one being of the distribution where q≥0.99, whereas the bottom one is of the distribution where q<0.99. The violin plot shows that on time shipments had a relatively more concentration in the bottom plot and slightly less in the top one. There is only a mild difference between PPU-FHS$_{delay}$ and PPU-FHS$_{on-time}$.

This right-skewed distribution is caused by outliers. Processing outliers might be a wise idea to do before feeding the data into the ML algorithms.

Figure 21 Shipping duration variables on delays



It was found that there was a substantially longer elapsed time on average between FHS and FDA and between FHS and delivery time (see Table 6).

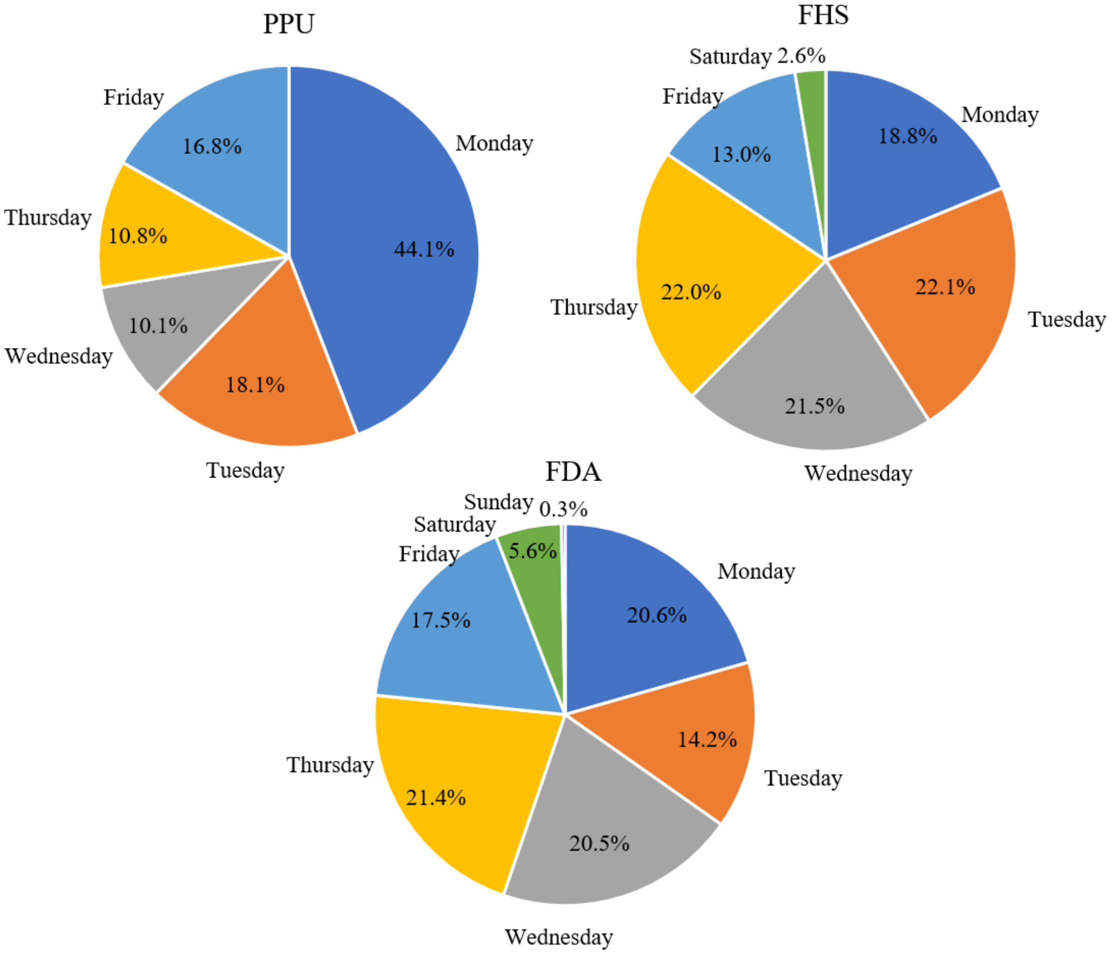
Table 6 Comparison of shipping duration

	On time	**Delayed**
Avg. FDA-Delivery	0.21	1.49
Avg. FHS-FDA	1.53	2.84
Avg. PPU-FHS	1.77	1.84

Shipping Days vs. Delay

Figure 22 displays the share of delayed shipments from the total amount of delays in September, ordered by day in which the parcels were processed. 44% of all delayed shipments were picked up from the shops on Monday. This high number is not a surprise, as orders that are placed over weekends are usually abundant, waiting in line to be processed. Meanwhile, parcels that were scanned on Mondays in the consolidation hub only resulted in around 19% of the total delays. One cause of this is that there are not as many parcels scanned in the hub as parcels delivered from the shops to the hub. It is common that the parcels picked up on Monday arrive until much later in the consolidation hub and are scanned the next day before they are further shipped to the next stop in the supply chain.

Figure 22 Delay shares by day of the week



Overall, there are no striking differences one can take from the FHS and FDA delay share, as the delays are spread consistently across all days, except on Saturdays and Sundays due to them not being workings days.

5.3 Further Data Pre-processing

After the cleaned data are explored and correlations are discovered, certain measures were taken to maximize further the prediction results and minimize the running time. Data pre-processing is a part of data preparation and a critical step toward building a working machine learning model. Pyle (1999) argued that preparing the data correctly prepare both the miner and the data. He also added, "Preparing the data means the model is built right. Preparing the miner means the right model is built."

Pre-processing packages provide various common utility functions to change raw feature vectors into a representation that is more suitable for the predictive models. Python's library scikit learn offers standardization, normalization, encoding categorical features, discretization, imputation of missing values, etc.

However, data pre-processing approaches have rarely been studied directly, not to mention the orders these approaches should be performed (Locklin 2014). This decision is fully up to the engineers and scientists and are often completely experience-based. Within the scope of this thesis, features that would affect greatly the target variable are selected. Some of these features would then be encoded, which will be further explained in sub-chapter 5.3.1. The data were subsequently split into training and test datasets.

The data reveals significant count difference between the delayed and on time shipments. In order to rectify the problem of class imbalance, the oversampling technique SMOTE was applied. This step, as well as data scaling, will be explained in sub-chapter 5.3.2.

5.3.1 Encoding and Splitting Data

After EDA was performed, the next data preparation proceeded in two stages: encoding categorical variables and splitting the dataset into training and test datasets.

Categorical data are variables that contain label values rather than numeric values. Categorical data are also called nominal data, a type of data that is used to label variables without providing any quantitative value (Corporate Finance Institute 2018). It is the simplest form of a scale of measure. Unlike ordinal data, nominal data cannot be ordered and cannot be measured. Many machine learning algorithms require that their input is numerical and therefore categorical features must be transformed into numerical features before it can be used in any of these algorithms (Ferreira 2018).

There are many different encoding techniques and their application depends on the characteristics of the data. Some important encoding techniques are as follows:

- Ordinal Encoding: encodes each unique value into an integer, starting from 1.

- Label encoding: encodes labels with value between 0 and the number of classes minus 1. Similar to ordinal encoding but this technique does not return a data frame on scikit learn.
- One-hot Encoding: encodes categorical features as a one-hot numeric way— which means that it creates one column for each unique value. The new column gets either a 0 or 1 if the row contains that column's value or if it does not, respectively.

One Hot Encoding, which is deemed as the classic approach by researchers and practitioners, is not suitable for high cardinality columns and decision-tree based algorithms (Hale 2018). This is logical because the higher variety of values a column has, the higher the dimension of the data frame will be, a phenomenon called the curse of dimensionality. Thus, the adequate selection of encoding techniques can mean a better model performance. As the data for this analysis does not contain many columns of categorical variables, the one-hot encoding method is chosen for this research. Categorical variables, such as Customer Address Country, Carrier, PPU day, FHS day, FDA day and Delivery day, were transformed into numerical values, widening the data into 56 columns in total.

The encoding should be, in this case, performed before splitting, as the same encoding system is needed both in train and test datasets.

After all variables were transformed into numerical, the data frame is divided into two: one that contained only the target variable *y* and one of the predictors *x*. Target variable, the variable whose values are to be modelled and predicted by other variables, is within the scope of this research would be the delay shipment, expressed in boolean data type with 1 is equal to delayed in column Delayed. Meanwhile, predictor variables are the ones whose values will be used to predict the value of the target variable. In this case, they would be the remaining 55 columns.

Testing is an important part of building machine learning models. When the models are tested over the same data used to train them, the issue of overfitting might likely to occur. Hawkins (2004) stated in his journal article that overfitting is "the use of models that include more terms than are necessary or use more complicated approaches than necessary". The models fit the training data too well, as they learn the noise in the training data to a degree that it negatively impacts the performance of the new data.

Using the `train_test_split()` function from scikit learn Python package, the entire dataset is split into training and test dataset, i.e. 66,66% and 33,33%, respectively. There are predictor variables and their associated target variable in training (*x_train*, *y_train*) as well as in the test dataset (*x_test*, *y_test*). The composition of the split is shown in Table 7.

Table 7 Shapes of split dataframes

	x_	**y_**
train	33516 rows x 55 columns	33516 rows x 1 column
test	16508 rows x 55 columns	16508 rows x 1 column
train with delay	2334 rows x 55 columns	2334 rows x 1 column
train without delay	31182 rows x 55 columns	31182 rows x 1 column

5.3.2 Synthetic Resampling and Standardizing

From a total of 50024 shipments, 93% arrived on time and only 7% came late to the customers' door. This is a huge imbalanced dataset with a ratio of delay to on-time instances of 93:7. Imbalanced class distribution refers to the issue with classification problems where the classes are not represented equally. In many cases, there is a large amount of observations for one class, or *majority class*, and much fewer observations for one or more other classes, or *minority classes*.

With imbalanced dataset, classification rules that predict the minority classes tend to be fewer and weaker than those that predict the majority classes - subsequently, test samples belonging to the minority classes are misclassified more often than those belonging to the majority classes (Sun et al. 2006). Standard classifiers usually perform poorly on imbalanced data sets, as they are designed to generalize from training data and to return the simplest algorithm that best fits the data. Therefore, the simplest algorithm ignores the rare cases, as most of predictive models are built on the assumption that, maximizing the accuracy is the goal (Provost 2000). In other words, models look at the data and decide that the best thing to do is to always predict the majority class and achieve high accuracy. This is what scholars and data scientists call accuracy paradox, i.e. phenomenon where accuracy is not always a good metric to measure the quality of predictive models. In predictive analytics, accuracy paradox states that predictive models with a lower level of accuracy may have greater predictive power than models with higher accuracy.

Imbalanced datasets are in practice a common occurrence and in some cases, identifying rare instances is crucial. Few examples are fraud detection, where the vast majority of the transactions are not fraud; customer churn, where the majority of the customers stay with a certain service, and also few cases in medical fields such as cancer detection, where the majority of patients do not have cancer. As Rocca (2019) pointed out in his blog post "Handling imbalanced datasets in machine learning", it is important to detect "naïve behaviour", as the resulted high accuracy is due to the imbalanced class dataset. Additional performance metrics, such as confusion matrix, precision and recall, should be taken into consideration, alongside accuracy. Even though it sounds trivial, collecting more data could balance the dataset. A larger data shape might uncover a different perspective on the classes.

The state-of-the-art research methodologies to deal with imbalanced class issues can be categorized into the following:

1. Sampling strategies. One common practice is to artificially rebalance the training datasets, either up-sampling (adding copies of instances from the under-represented class) or down-sampling (removing instances from the over-represented class) (Provost 2000). These two techniques are commonly known as over-sampling and under-sampling, respectively. Various studies in imbalanced datasets have experimented different types of over- and under-sampling and have presented conflicting viewpoints on usefulness of over-sampling versus under-sampling (Chawla 2010). Random under- and oversampling methods have both their short-comings. Random under-sampling has the potential to remove certain important features useful for the prediction, and random oversampling can lead to overfitting.
2. Synthetic data generation. This technique aims to tackle the imbalance by artificially generating data samples (He et al. 2008). A well-known synthetic data generation technique is SMOTE (Synthetic Minority Oversampling Technique), which will be applied for the quantitative data analysis of this research. In SMOTE, over-sampling by replicating synthetic minority instances can lead to similar but more specific regions in the feature space as the decision region for the minority class and shifting therefore the classifier learning bias toward the minority class. Another technique, ADASYN (Adaptive Synthetic Sampling Approach for Imbalanced Learning) generates more synthetic data for minority

class instances that are harder to learn compared to those minority instances that are easier to learn, and thus adaptively shift the decision boundary to focus on those difficult to learn instances.

3. Cost-sensitive learning. Different from the previous strategies, this approach does not modify the imbalanced data distribution directly. Using a set of cost-matrix for different type of errors, it facilitates learning from imbalanced datasets. One example of this approach is the application of misclassification costs on any particular data sample. Instance-weighting can also be used to induce cost-sensitive trees that results in better performance (He et al. 2008).
4. Active learning. This strategy uses the model-based instance selection in an incremental setting and is conventionally used in unlabelled training data. This method can choose informative instances from a random set of training populations in an effective way.
5. Kernel-based methods. Kernel-based methods, such as KBA (Kernel Boundary Alignment) uses the kernel matrix modification to according to the imbalanced data distribution and OFS-based algorithm to optimize the model generalization for learning two-class imbalanced data sets.

From the short review above, the synthetic data generation approach is selected due to two major reasons. First, synthetic data generation methods do not show drawbacks shown by random sampling strategies. Second, this approach is easy to apply and easier to understand than other above-mentioned alternatives. With a few lines of code, synthetic data can be easily generated. One synthetic over-sampling and one synthetic under-sampling technique were employed, i.e. SMOTE and NearMiss method, respectively. SMOTE, as previously mentioned, is an approach in which the minority class is over-sampled by creating synthetic' examples rather than by over-sampling with replacement. Each under-represented class sample is taken and synthetic examples are introduced along the line segments joining any/all of the *k* minority class nearest neighbours. Neighbours from the k-nearest neighbours are randomly chosen, depending on the amount of over-sampling.

NearMiss algorithm is a synthetic under-sampling method that puts into practice some heuristic rules to select samples. The Python library *imbalanced-learn* offers three versions of this under-sampling method. According the library’s documentation, “NearMiss-1 selects samples from the majority class for which the average distance of the k-nearest samples of the minority class is the smallest". On the other hand, the majority class samples in NearMiss-2 have the smallest average distance to the furthest samples of the negative class. NearMiss-3 involves a two-step algorithm—m-nearest neighbours of each minority sample are kept then the majority samples, with the largest average distance to the k-nearest neighbours, are selected. For this thesis, the default NearMiss version, NearMiss-1, was selected.

Between the two synthetic sampling methods there is a substantial difference in resulted amount of balanced class data, as shown in Table 8. For the SMOTE method, the sampling strategy 'not majority' is chosen. This is the default sampling strategy of the *imbalanced learn* package, which resamples all classes but the majority class. The under-represented data were oversampled to the amount of the over-represented data, i.e. 31182 rows of data. On the other hand, the sampling strategy selected for the NearMiss is 'not minority', which, in the opposite, resamples all classes but the minority class. Analogous to the oversampling method, the over-represented data were under-sampled to the amount of the under-represented data, i.e. 2334 rows. The proportion of on-time and delayed shipments after each synthetic sampling method is 0.5

Table 8 Number of rows of training data before and after synthetic replication

	Majority class	Minority class	Total
Pre-processed	46597	3427	50024
SMOTE	31182	31182	62364
NearMiss-1	2334	2334	4668

Another parameter manually defined for the sampling methods was random_state. This is a pseudo-random number that controls the randomization of the algorithm. By specifying the random_state with an integer, the same output will be obtained every time the code is run. This is useful to create reproducible results during testing period.

After both of the sampling methods are defined, they were fitted into the training (*x_train, y_train*) dataset, resulting in two different new training datasets: a training dataset SMOTE (*sm_x_train, sm_y_train*) and a training dataset NearMiss (*nm_x_train, nm_y_train*). Resampling techniques are not fitted into the test dataset, as these generate more data synthetically. These are evidently not real world data.

Once the classes are balanced, scaling is performed. This is due to the fact that most of the ML algorithms use Eucledian distance between two data points in their computation—thus, the high variation in magnitudes and range would become a problem. If a feature has a higher magnitude than other features, that particular feature would weigh in a lot more in the distance calculations, resulting in its domination over the other features.

Scaling, standardization and normalization often times are used interchangeably. There is still considerable ambiguity with regard to the definitions of these three terms. According to Pyle (1999), normalization is taking values that span one range and representing them in another range, thus requiring to remap values from an input range to an output range. Patro and Sahu (2015) claimed that normalization is a scaling technique where a new range is created from an existing range. In some literature, normalization is one step above scaling as it changes the observations to fit into a normal distribution. Some of important scaling, standardizing and normalizing techniques are as follows:

- MinMaxScaler: a scaling technique that rescales the dataset into the range of 0 to 1.
- StandardScaler: a standardizing technique that removes the mean and scales the data into a unit variance.
- Normalizer: a normalizing technique that rescales each sample observation (row) into a unit norm, regardless of the distribution of samples.

Within the scope if the thesis, the predictor variables are scaled using the `StandardScaler()` function from scikit learn, as deep learning algorithms would benefit from zero mean and unit variance as well as regression-based algorithms would benefit usually from normally distributed data. This function standardizes features by removing the mean (mean = 0) and scaling to unit variance (Pedregosa et al. 2011). In essence, each value of the dataset will have the mean value subtracted and divided by the standard deviation of the whole dataset. This way it will transform the data in such way that the distribution will have a mean value of 0 and a standard deviation of 1.

With the completion of the pre-processing step, the research will be continued by building various machine learning models to assess how much the features affect the predictability of the delay and how accurate the prediction of each model is.

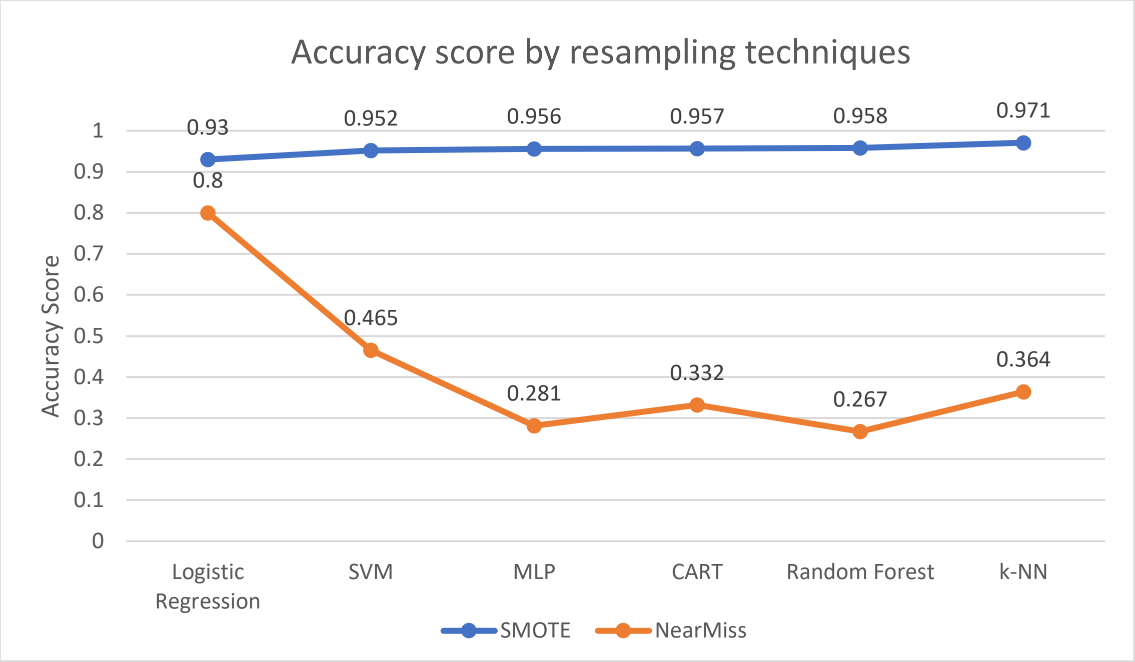
5.4 Building Machine Learning Models

Once the data are cleaned and pre-processed, the next step would be building the actual prediction models. There is a myriad of packages and libraries available one can choose from to perform this, depending on the prediction goal and the data format. Thus, it is more sensible to know what each machine learning model does and select them rather than blindly running the data through all of the models, as training models could take a lot of time, especially on low-capacity CPUs. Selected machine learning models can be easily built using the Python's libraries. The chosen models are as follows:

- Logistic regression
- Random Forest
- MLP Classifier
- CART
- K-Nearest Neighbour
- SVM

First, the instance of each prediction model was created then the models are fitted into the two resampled training datasets. This step returned the prediction results of the target variable. These predicted target variable values were compared to the true target variable values to calculate the accuracy of the models, using the function `accuracy_score()` from scikit learn, as shown in Figure 23. It is revealed that the over-sampling method SMOTE yielded a much higher accuracy score across all selected prediction models. The accuracy difference is striking—predictions using SMOTE achieved more than 60% more accuracy than employing NearMiss in some models, e.g. random forest, CART, *k*-NN and MLP. In addition, the accuracy scores of all models, except logistic regression, with NearMiss method are lower than the even random guessing. Interestingly, the huge accuracy difference is not shown in logistic regression. The model only counted on 0.13 accuracy improvement from the NearMiss technique. The logistic regression results in both graphs are worth mentioning, as in the over-sampled training data the model came out as the worst model, meanwhile in the under-sampled training dataset it was the best model, far in comparison with the rest of the algorithms.

Figure 23 Accuracy scores by resampling techniques



A few highlights could be drawn from the accuracy results. The strong contrast in the accuracy scores between both resampling methods could be attributed to the fact that the models were trained only with 4668 data rows in NearMiss method, whereas 62364 data rows were fed into the models in the SMOTE method. This underlines just how important the amount of data is in training machine learning models and these results offered compelling evidence for the effectivity of over-sampling methods, in relative with under-sampling methods. Logistic regression, on the other hand, can be used relatively smaller sample size and is also robust against outliers, which explains the weaker contrast between the two resampling techniques. However, these findings needed to be interpreted with caution as the extreme contrast between the two resampling results could be associated with issues in the input data at the time of running the NearMiss method, such caused by poor data sampling or data scaling.

As previously mentioned, accuracy is not the only metric one should rely on. To obtain more details on the findings, a confusion matrix, or error matrix, was used in order to visualize the performance of a classification model. The two rows of the matrix represent the instances of the actual labels and the two columns represent instances of the predicted labels. As the name suggests, it identifies the confusion between classes, i.e. mislabelling of instances to another class, as it summarises the number of correct and incorrect predictions with count values and broken down by each class. Appendix D and E show the confusion matrix of each prediction algorithm in SMOTE and in NearMiss, respectively.

There are 6 confusion matrixes corresponding to 6 different machine learning models, for each resampling technique. The scale that goes from blue to yellow on the right side gives the boxes in the confusion matric colour depending on the number of instances. There are 16508 instances in the test dataset in total, which serve for the models to predict. From those instances, 1093 shipments came with delay and the majority, 15415 shipments came in a timely fashion at door of the end customers. Therefore, there are more than 14 times as many on-time shipments as there are delayed shipments. From the whole dataset, training plus test datasets, the on-time shipments count for more than 13 times of the delayed ones. This indicates that the proportion of delayed and on-time shipments in the entire as well as in the test dataset is similar. This similar proportion is quite interesting, as the `train_test_split()` function from scikit learn split a certain dataset into random train and test datasets.

First, let's discuss the overall accuracy results from the over-sampled training dataset. The best performing algorithm from the experiment, K Nearest Neighbour, succeeded in predicting 15209 *true positives*, i.e. outcome where the classifying model correctly predicts the positive class (on-time), as well as 819 *true negatives*, i.e. outcome where the classifying model correctly predicts the negative class (delayed). The model predicted similar amount of incorrect predictions in both classes, with 206 *false negatives*, i.e. outcome where the classifying model incorrectly predicts the negative class, and 274 *false positives*, i.e. outcome where the classifying model incorrectly predicts the positive class. On the other hand, the model with the worst accuracy score, logistic regression, could predict the most instances labelled as delayed in comparison with KNN, or even with the rest of the models. With a whopping 1029 correct predictions of the negative class, the logistic regression model leads in the race of *true negatives*. However, the logistic regression-based model resulted also in the highest number of *false negatives* with 1099 instances. Meanwhile, the model categorized only 64 instances incorrectly as on-time. This is one of the most conspicuous observations to emerge from the confusion matrix comparison. The logistic regression model from the experiment may have the tendency to label the dataset as delayed.

The above-mentioned overall accuracy scores are to be interpreted as the number of correct predictions of both classes divided by all instances of the test dataset. Thus, they do not express the models' prediction bias. The overall accuracy is in the experiment quite high but it does not reflect the actual quality of the model. This is where the average accuracy come in handy. The average accuracy is the average of the overall accuracy score of each class (see Table 9). The average accuracy score is worse than the overall accuracy shown in the previous Figure 23 across all models. Surprisingly, the logistic regression model showed a higher accuracy score at predicting the class delayed than the class on time. Meanwhile, the rest of the models showed the opposite. The logistic regression model exhibited the lowest accuracy score of the class on time—a strong evidence that the model predicted incorrectly data as delayed than the other way around. It is demonstrated that the logistic regression model from the experiment had the tendency to label the dataset as delayed.

Table 9 Average accuracy per class (SMOTE)

	Accuracy class On time	Accuracy class Delayed	Average Accuracy
Logistic regression	0.929	0.941	0.935
Random Forest	0.962	0.898	0.93
MLP Classifier	0.959	0.898	0.928
CART	0.962	0.885	0.923
KNN	0.987	0.749	0.868
SVM	0.953	0.933	0.943

Second, the classifying models fitted into the under-sampled training data using NearMiss showed a very different outcome. The experiment was unsuccessful in providing an overall acceptable accuracy score. Contrary to expectations, all classifiers had an immense tendency to label the data as the minority class—a finding that is strongly highlighted by the average accuracy per class in Table 10. In fact, all of them except logistic regression predicted more instances as delayed shipments than as on-time shipments, which is reflected in the extreme low accuracy scores. There is therefore a strong bias towards the minority class. This finding is contradictory to previous results reported extensively in the literature, where rebalancing dataset was evident in reducing bias and classification error (Iosifidis and Ntoutsi 2018; He et al. 2008).

Table 10 Average accuracy per class (NearMiss)

	Accuracy class On time	Accuracy class Delayed	Average Accuracy
Logistic regression	0.794	0.881	0.838
Random Forest	0.221	0.912	0.567
MLP Classifier	0.271	0.942	0.607
CART	0.291	0.903	0.597
KNN	0.336	0.76	0.548
SVM	0.433	0.911	0.672

Further investigations were needed to validate these prediction results.

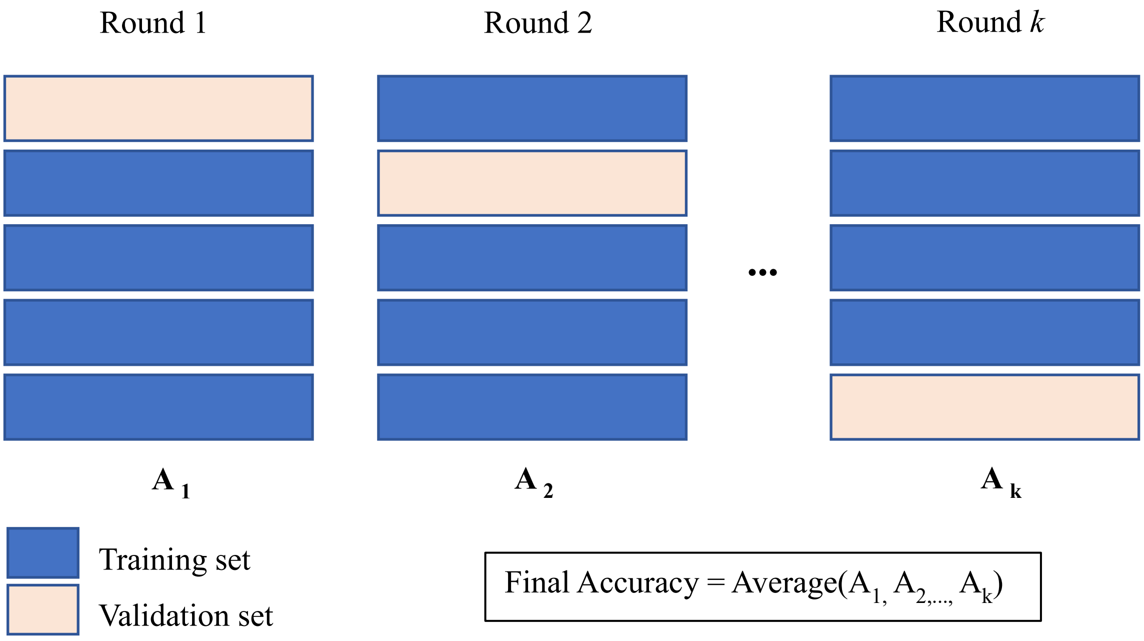
5.5 Cross-Validation

As demonstrated from the previous accuracy results, fitting the models into the over-sampled dataset yielded a very good accuracy. However, by fitting it to one training data sample, the prediction results only give information on how the models perform to that one training data sample after once tested with the test dataset. In business practice, it is vital to prove how these classifying models perform on new sets of data despite the limited data sample. In addition, cross validation reduces overfitting as the training sample is independent from the validation sample (Arlot and Celisse 2010).

Cross-validation is a popular strategy to assess the predictive performance of the models and how they perform outside the sample new datasets. As mentioned by Arlot and Celisse (2010), the main idea behind cross-validation is to split data, once or several times, with the aim of estimating the risk of each algorithm. The training dataset is split into training sample, i.e. sample used for training each algorithm, and validation sample, i.e. sample used for estimating the risk of the algorithm. Finally, the algorithm with the smallest estimated risk is then selected. There are several cross-validation techniques used in machine learning. In summary, cross-validation uses a portion of the data sample to estimate the prediction power of the model by executing predictions on unseen data during the model training.

The method selected for this experiment was the stratified *k*-fold cross-validation (see Figure 24). In general, k-fold cross-validation is a prevalent method because it is simple to execute and results in a less biased and less optimistic prediction results, comparing it with the simple train/test split, due to the independency between training and validation datasets. The parameter *k* describes the number of subsets that the data sample is to be split into. The procedure consists in taking one of the *k* subsets as a hold out, which serves as the test set or validation set, and putting the remaining *k*-1 subsets together as the training dataset. This process is repeated *k* times, so that each one of the *k* subsets is held out once as the validation set. Stratification ensures that each k subset contains a good representative of the whole dataset. In a binary classification study, each one of *k* subsets should therefore hold around half of the instances of each class.

Figure 24 k-fold cross-validation

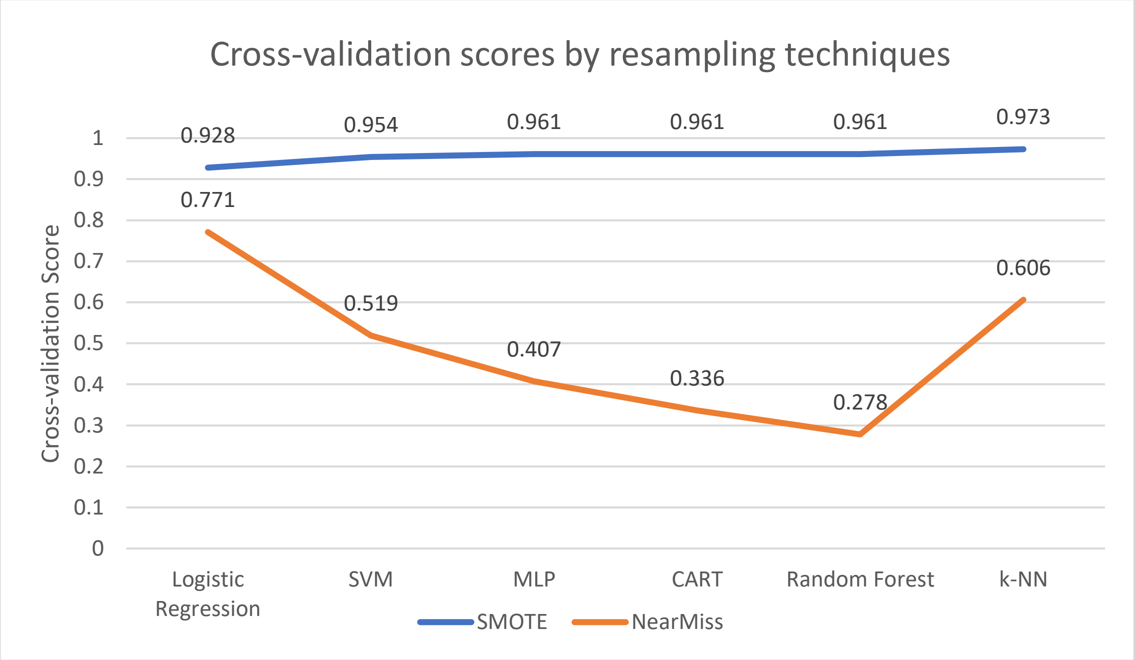


With all the stated benefits of cross-validation, the technique could improve the under-sampled accuracy results. The suspicious low accuracy scores and the models' high bias towards the minority class could be further reviewed as cross-validation verifies the quality of the patterns made by the algorithms.

Due to the computer capacity and running time, the selected value for *k* is 5. Thus, the validation technique is also called 5-fold cross-validation. The concept of training and validation is simple, but the practical execution is a little bit more complicated as the validation set should be the real data, whereas the training dataset should go through class rebalancing first, either through over-sampling or under-sampling. Hence, a separation of these two datasets is necessary before feeding the data into the cross-validation helper function, in this case is the *cross_val _score* from scikit learn. For this purpose, a pipeline function is used to combine the resampling process with the classifying algorithms. This combined function is then applied into the cross-validation function. Therefore, the resampling process is not only performed once before the cross validation, but it is performed for each *k* iteration. Naturally, this process takes more computer capacity and running time. In addition, the training dataset of each resampling method is split using the `StratifiedKFold` function from scikit learn. The classifiers are fitted into a cross validation function that evaluates the accuracy score of each round by cross-validation. Once the score evaluation of a prediction model is finished, the function will evaluate the next prediction model in line.

Using plotting libraries seaborn and matplotlib, the average and the standard deviation of the k-fold accuracy scores of each prediction model are calculated and plotted in Figure 25, where the standard deviation is shown by the black line at the end of each bar.

Figure 25 Cross-validation accuracy scores



Both resampling methods yielded some interesting results. All algorithms used in SMOTE method showed an increase in accuracy, even though insignificantly so. The performance did not change much and once again, *k*-NN proved to be the best performing algorithm. This finding is in contradictions with the finding that testing the models repeatedly against unseen data leads to more pessimistic prediction results, in comparison with the common train/test split. In addition, the standard deviation of each classifier is fairly small, in which *k*-NN has the highest standard deviation with 0.002. This shows that the variation of accuracy score across the 5 folds is small, as they are close to the mean. Thus, the models did not pick up a high variance, which reflects the low noise of the training data.

The accuracy results in the NearMiss method displayed a more surprising outcome than the SMOTE version. A significant rise in accuracy is registered across some of the models, i.e. *k*-NN and MLP with 0.242 and 0.126, respectively. Meanwhile, the accuracy scores of the rest of the models only increased insignificantly. However, there is much higher standard deviation in the NearMiss results than the SMOTE-sampled data, i.e. accuracy scores resulted from the under-sampled dataset are more dispersed from the mean than the scores from the over-sampled data. For further examinations, the breakdown of accuracy score of each *k*-repetition is shown in Table 11.

Table 11 Accuracy score of each fold (NearMiss)

	Fold 1	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Std**
LogReg	0.76654008	0.79482311	0.77728908	0.75989604	0.75629748	0.0139
RF	0.26144313	0.26284229	0.28528589	0.29768093	0.28208717	0.0138
MLP	0.51169298	0.34529282	0.42882847	0.3832467	0.36615354	0.0591
CART	0.31211273	0.32680392	0.37594962	0.34386246	0.32017193	0.0226
KNN	0.6063362	0.62572457	0.6307477	0.54658137	0.61835266	0.0306
SVM	0.51009394	0.54067559	0.51669332	0.48910436	0.53978409	0.0193

Within one model, there is a substantial difference in the accuracy scores across the folds. The high variation of accuracy scores suggests that the training data contains a high variance in each fold, which leads to a high variance in the whole dataset. This could be caused by a considerable amount of noise.

On the basis of these cross-validation results, the best performing models are then selected for further adjustment. Due to limited running time and computer capacity, three of the SMOTE models are selected: *k*-NN, CART and Random Forest.

5.6 Adjust Hyper-parameters

In the field of machine learning, hyper-parameters are higher-level properties of the algorithm model, which can impact substantially its complexity, its learning speed and also its application results (Chicco 2017). Hyper-parameters should be set before the training step, the algorithms cannot learn them from the training phase. Thus, the process of identifying the best value for hyper-parameters is called hyper-parameter optimization, which goal is to minimize the generalization error of the model (Bergstra and Bengio 2012). On the other side of the coin, hyper-parameter tuning has its challenges, mainly because determining values for hyper-parameters is complex. Claesen and Moor (2015) stated that, each evaluation may take a good amount of time depending on the available computational resources, the nature of the learning algorithm and size of the problem. This statement is proved by the practical experiment in the course of this study, as this step is the one that took the longest computing time. Training times in the order of minutes is considered as fast, meanwhile the order of hours is quite common. For the optimization of the random forest model using grid search, it took 66.1 minutes to finish. This challenge is encountered frequently in grid search, the most widely used strategy for hyper-parameter tuning. This strategy suffers from the curse of dimensionality as the number of trials grows exponentially with the number of hyper-parameters. It requires that a set of values is pre-specified for each parameter, where the set of trials is formed by assembling every possible combination of these values, so the number of elements in a grid search is $S = \prod_{k=1}^{K} |L^{(k)}|$, where K is the configuration variable, $L^{1}...L^{(K)}$ are the set of value for each variable. Taking the previously mentioned random forest model as an example, there are four parameters to be configured with at least 2 values for each parameter, resulting in total elements of S = 3 x 3 x 3 x 2 = 54 candidates. On top of that, a stratified 5-fold cross-validation was applied on each of these candidates, which brought to in total of 270 fits that had to be performed by the computer. In addition, the required time to train and test models depend on the choice of hyper-parameters, as they can influence the architecture of the model, e.g. number of hidden layers in a neural network model (Claesen and Moor 2015).

Another drawback is the complex search spaces that entail from this tuning process. Previous research has demonstrated empirically that only a handful of hyper-parameters considerably impact the performance of machine learning models, though identifying the relevant ones is difficult (Claesen and Moor 2015; Bergstra and Bengio 2012). Some researchers argue that the random search strategy is superior for this exact reason. Random search is able to find estimators that as good or even better than the common grid search within a small fraction of the computation time by selecting the variables randomly (Bergstra and Bengio 2012). Baseline for this strategy is as follows: to examine the effect of a particular hyper-parameter, random search is performed with the goal of optimizing other hyper-parameters. Then, choose a set of random values for these hyper-parameters and use it for each value of the hyper-parameter of interest.

Beside grid and random search, hyper-parameters can be tuned with manual search, where the human identify promising space where the good hyper-parameters are by incorporating knowledge about how those adjustments would influence the behaviour of the model and by developing the intuition necessary to choose the set of values for the variables ($L^{1}...L^{(K)}$). In other words, the researcher tunes the hyper-parameters through trial and errors, so it is difficult to reproduce the results. Due to its familiarity, the grid search method is employed for this study.

Table 12 shows the adjusted parameters and their set of values for the selected models, e.g. *k*-NN, CART and random forest. These parameters and values are chosen intuitively, taking into consideration their possible effect in the learning of the algorithm.

Table 12 Values and parameters for hyper-parameter tuning

Model	Parameters	Values	Best value	Best accuracy score
***k*-NN**	n_neighbors	[3, 5, 11, 19]	11	0.974
	weights	['uniform', 'distance']	'distance'	
CART	min_samples_split	[2, 3, 10]	10	0.961
	max_depth	[1, 20, 2]	20	
	min_samples_leaf	[1, 3, 10]	1	
Random forest	max_features	[1, 3, 10]	10	0.963
	min_samples_split	[2, 3, 10]	2	
	min_samples_leaf	[1, 3, 10]	3	
	n_estimators	[100, 300]	100	

Using `GridSearchCV` function from scikit learn, the grid search is performed with stratified 5-fold cross-validation and over-sampled training data. The experiment started with random forest with 4 parameters, which is performed in the author’s local computer. As previously mentioned, it took 66.1 minutes to finish all the 270 possible fits. The issue began with the next model, *k*-NN, as the computing process did not end even after 3 hours elapsed. A quicker way to tune had to be found for the remaining models. For this purpose, the number of parameters was reduced and the virtual machine from Google Cloud was chosen as the computational power. The hyper-parameter tuning of the *k*-NN model with 40 fits was carried out it only 19.3 minutes. Subsequently, the CART model only took a mere 22.2 seconds for 45 fits. The accuracy score of each best set of values for the parameters is shown in Table 12.

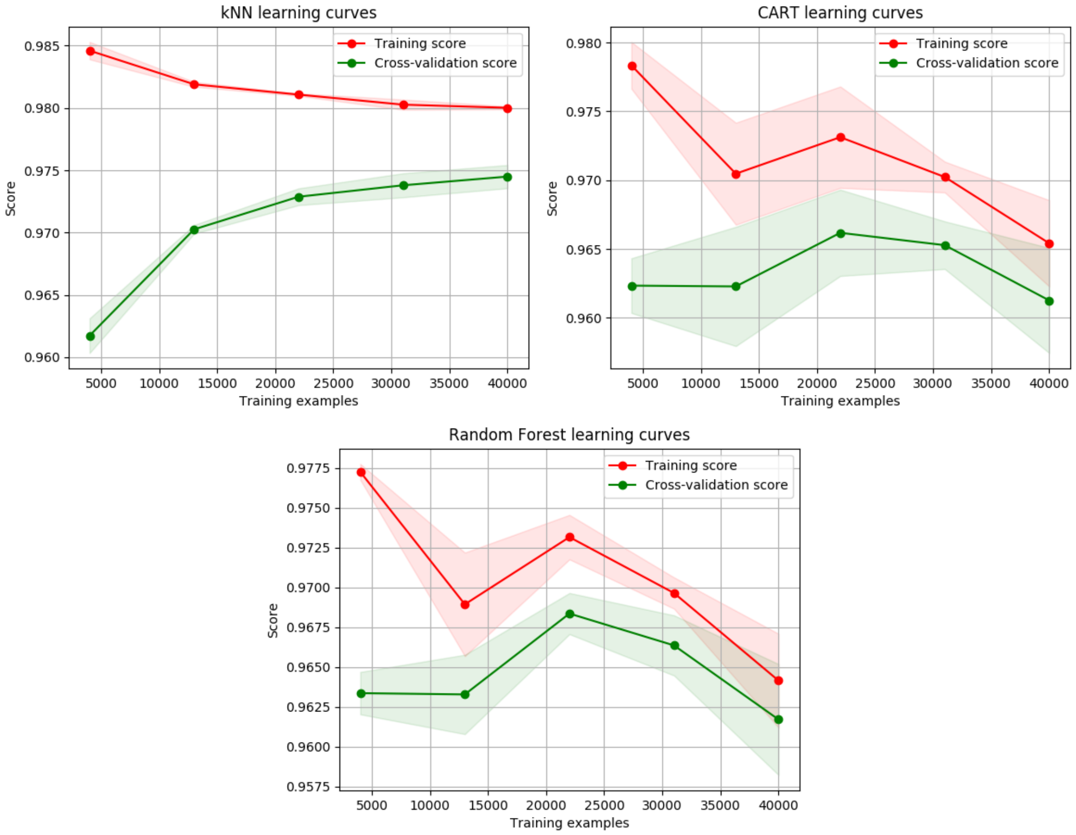
Appendix F displays more details on the hyper-parameter tuning code and results. The adjustment only increased *k*-NN's accuracy score by 0.002, which is minuscule considering how long it took to compute all the possible fits. The same fate happened to the other two models, i.e. CART with only 0.001 and random forest with 0.002 hike. In conclusion, the grid search did not yield very satisfying results. In future research the author would perform this step employing random search and compare the results with grid search.

To review the learning process of the models over experience, learning curves are employed (see Figure 26). Learning curves display changes in learning performance over time in terms of experience., i.e. they show how better do the models get at predicting the target as the number of training instances increased. The concept is simple: the models are evaluated on the training and the validation dataset after each update during training and this performance is measured and plotted. Hence, learning curves show the relation between experience or number of instances (x-axis) and the evaluation metric used (y-axis), which is the classification accuracy in this case. Dual learning curves are commonly created for each machine learning model:

- **Training learning curve**. The model is evaluated on the training data, which can be interpreted as how well the model is "learning" (shown as red lines in Figure 26),
- **Validation learning curve**. The model is evaluated on the hold-out validation dataset, which can be interpreted as how well the model is "generalizing" (shown as green lines in Figure 26).

It is a very useful tool to find out how much beneficial adding more training data would be and to diagnose variance and bias-related problems in machine learning algorithms, such as underfitting and overfitting.

Figure 26 Learning curves of models



The `learning_curve()` function from scikit learn used in this study runs a 5-fold cross-validation in the whole dataset, as depicted in the 5 points in all of the graphs. For each subset, the accuracy scores of the training and validation dataset are averaged over all 5 runs and the standard deviation is calculated. The coloured areas along the learning curves are the difference between the mean of the cross-validated accuracy scores and the standard deviation.

- *k*-NN learning curves. The training score is very high in the beginning and gradually decreases. The validation score is very low but increases towards the end. Adding more training instances could possibly be more beneficial for this model, as the validation learning curve could converge toward the training learning. The kind of shape of learning curves are to found often in complex datasets (Pedregosa et al. 2011). The large gap between the curves show that there is a high variance and the high accuracy training learning curve shows a low bias. In general, the model “learned” well, but there is a possibility that there might be overfitting.

- CART learning curves. The model generalizes better than the k-NN model by the end of the iteration, as both of the learning curves are closer to each other. The training learning curve decreases, experiencing the biggest decrease on the second round of cross-validation. The classifier found its highest point at around training iteration 22 000, which is the same case with the validation curve. Furthermore, the validation learning curve shows a stagnation in the first two rounds of cross-validation. There could be problem with the dataset, such as noise that was captured by the model. More data or features could be useful for the model.
- Random Forest learning curve. The model yields similar learning curves as the previous model as both of them are decision tree-based models.

In conclusion, the *k*-NN model has the best accuracy score, followed by random forest and CART. Therefore, it is subjected to further investigation. After the results are optimized, the next step is to discover the important features for the predictions.

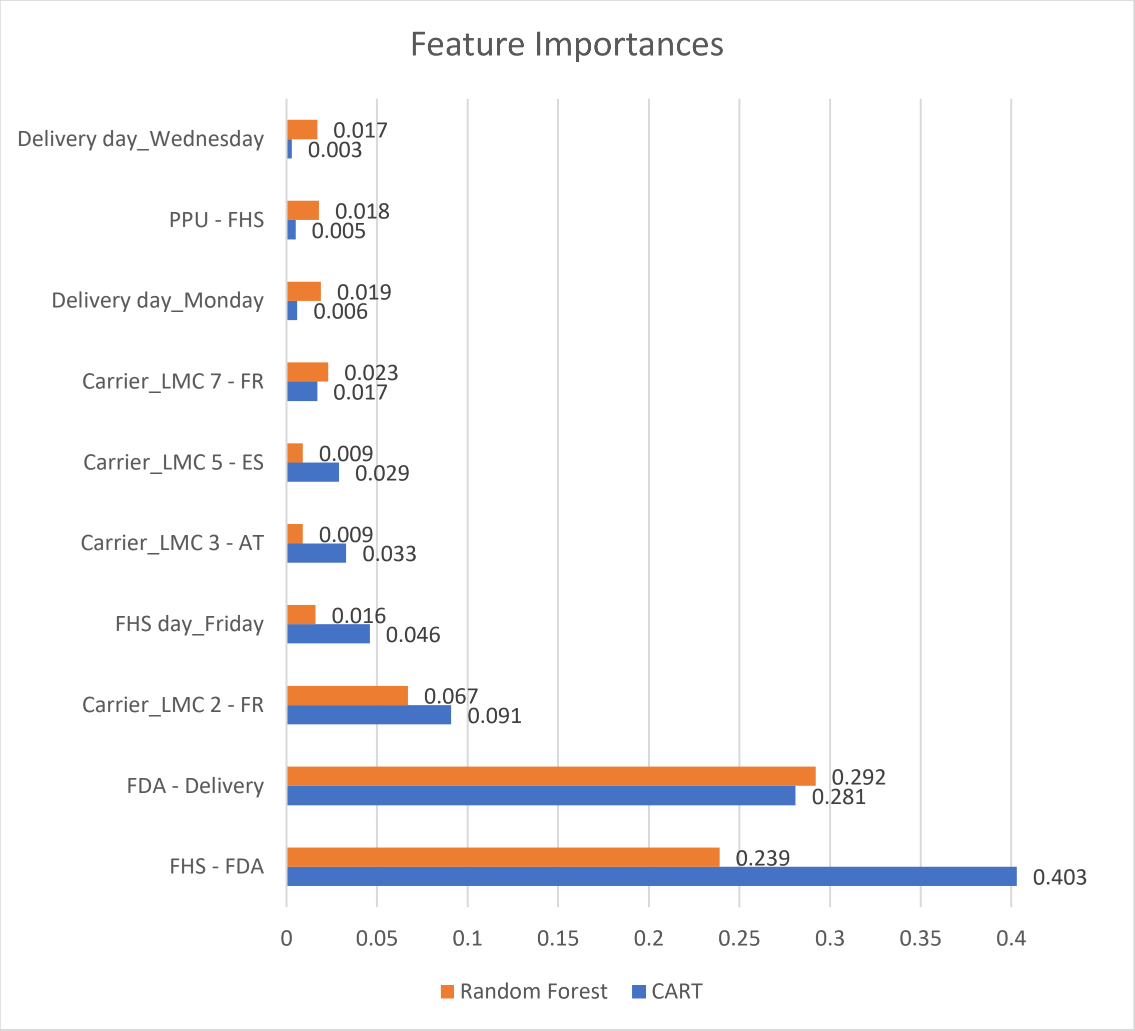
5.7 Feature Importance

A great model performance is just a good start in the real world. Successful code and excellent prediction result do not necessarily mean that the data is clean. As mentioned in the cross-validation chapter, training data may contain biases that may be picked up by the models. That being said, the test dataset could be biased too, which would bias the model evaluation. It is important to understand how the machine learning models work and how the features in the models contribute to the prediction.

Decision trees make splits that maximize the decrease in impurity. Therefore, in decision tree-based models, such as CART and random forest, the feature importance is calculated as the decrease in node impurity weighted by the probability of reaching that node. The higher the decrease value is, the more important the feature is. A benefit of using ensembles of decision tree methods they can automatically provide estimates of feature importance from a trained predictive model. For a single decision tree, the importance is calculated by the number that feature split point improves the performance measure (Gini index) and weighted by the number of observations the node is responsible for. The feature importances are then averaged across all the decision trees within the model.

The impact of the features in the tree-based models (CART and random forest) can be measured using the method `feature_importances_` from scikit learn (see Figure 27).

Figure 27 Feature importance - CART and random forest



In both models, the features FHS-FDA and FDA-Delivery show a high relative importance in the prediction, i.e. most used attributes to make key decisions. However, the important features ranking differs after the first two features in the models. Appendix G and H display the 25 most important features for CART and random forest, respectively.

The selection of LMCs impacts also in the delivery time prediction. There are big challenges in last mile deliveries, so this part of the supply chain usually takes the longest time. First, parcel receivers are widely spread across a certain city and each LMC has its own network of hubs to fulfil customers' demands. It is possible that a parcel has to go through more than 2 LMC hubs until it reaches the parcel receiver. Therefore, the LMC's performance and network is considerably correlated to the delivery time.

5.8 Summary of Results

This sub-chapter will provide the discussion regarding the results on the prediction of delivery times.

It is demonstrated that ML models can be used to predict delivery time using the Delivery Platform's internal data. All the selected ML models, e.g. logistic regression, SVM, *k*-NN, MLP, random forest and CART, show great accuracy scores using the *k*-fold cross-validation. Due to time and computational power constraints, three of the best models are selected for hyper-parameter adjustment using grid search. With a maximum accuracy increase of 0.002 in random forest and *k*-NN, grid search did not provide satisfactory results considering how long it took to compute. The feature importance of the decision tree-based models is examined and ranked, which results in the features FHS-FDA and FDA-Delivery as the most relatively important attributes. The results are not surprising, as the delivery time is directly correlated to shipping time variables. Table 6 also showed that there is a difference in average time between the two classes.

Many factors can lead to long FHS-FDA and FDA-Delivery, e.g. wrong customer address, defective barcodes, incomplete customer information, etc. In practice, parcels that contain these issues have to stay in the consolidation hub until the issues are resolved, extending that way the transit time in the supply chain instance.

6. From Analytics Modelling to Analytics Operations

The previous chapters have dealt extensively with a quantitative experiment to expose the application of analytics models to predict delivery times on shipments. However, no insights derived from any statistical modelling would be beneficial if the organization is not able to extract business values and competitive advantage out of it. A set of courses of actions and a supply chain-wide IT system has to be placed in order to implement the outputs of the models to make decisions and to take actions that bring business values. An analytics governance should be appropriately structured to enable a business organization to reach these goals.

This chapter aims to provide an analytics infrastructure and a set of suggestions for specific actions on how to conceptualize the results of the analytics modelling from the previous chapter in such a way that they add value to the actual business. Hence, the chapter focuses on incorporating the analytics modelling into the daily operational activities. The first sub-chapter delves into the theoretical perspective of the structure of analytics organization and its influence in the life cycle of analytics models. The proposed analytics infrastructure and the suggested course of actions will be discussed in the second and third sub-chapters, respectively. Lastly, the implications of the findings will be discussed in the fourth sub-chapter.

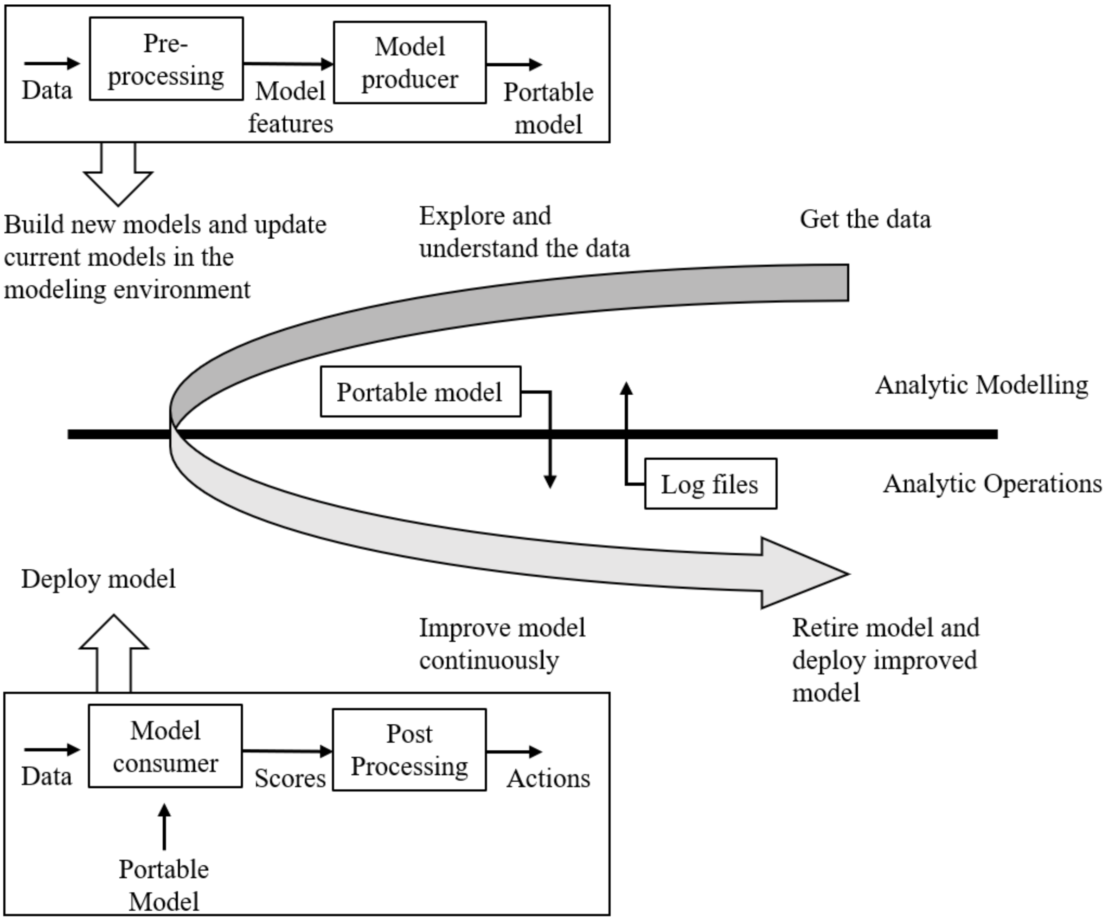
6.1 Analytics Organizational Structure

Generally speaking, the analytics function in an organization is composed of three aspects: **analytics models**, **analytics infrastructure** and **analytics operations** (Grossman and Siegel 2014). The first aspect, analytics models, is the combination of statistical, predictive, or data mining models that are empirically derived from data, e.g. building ML models. This task was discussed in the previous sections of the thesis. Secondly, analytics infrastructure encompasses the software components, software services, applications and platforms for data management, data processing, model building and using these models to generate alerts, make decisions and take adequate actions. The main concern of analytics infrastructure is the smooth data management and the deployment of models and other analytics tools that are integrated in the organization’s products, services and operations. Lastly, analytics operations refers to the processes that come after gaining insights from the analytics modelling, which involve making decisions and taking decisions relevant to the goals of the enterprise, e.g.

increasing revenues, decreasing costs or improving operational workflow. The main concern of analytics operations is the fruitful application of the results of analytics models into the organization's products, services and operations.

The symbiosis of these three aspects are managed through an analytics governance structure. The structure does not only provide a hierarchy of individuals that have authority for certain analytics-related decisions, but also provides mechanisms for identifying, communicating and resolving issues encountered in analytics-related projects as well as mechanisms to ensure the availability of analytics resources within the organization. The interconnection between analytics modelling and analytics operations can be better visualized from the life cycle of analytics models, displayed in Figure 28. Usually models do not generate value for an organization until they are deployed. It is the analytics infrastructure task to ensure that the analytics models reach to the operations.

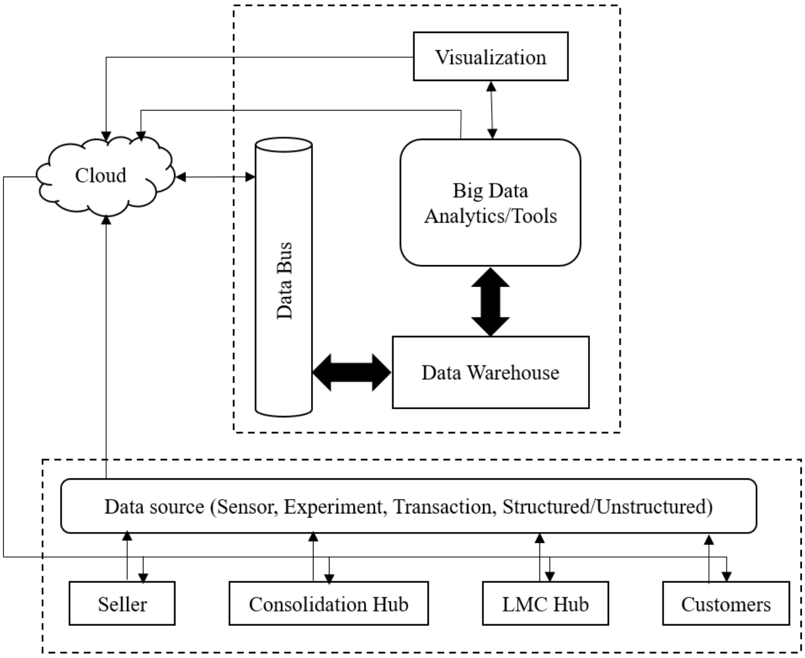
Figure 28 Life cycle of analytics models. Adapted from (Grossman 2018)



6.2 Analytics Infrastructure for Supply Chain

A recommended analytics infrastructure for SCM was proposed by Biswas and Sen (2016) (see Figure 29). The framework is adapted to the supply chain of the object of the case study, the "Delivery Platform".

Figure 29 Big data driven supply chain structure



The proposed framework consists of 6 components:

- **Data acquisition devices**. This system is responsible for collecting raw data at each stage of supply chain and serves as data source. It includes sensors, actuators, RFID, barcode scans, tag enabled objects, camera connected objects, among others. While it tracks the status of objects in each supply chain instance, the handling of the large volume of data generated from this process presents challenges.
- **Cloud infrastructure**. The data acquired from the devices are passed over to the cloud, a collection of computing and storage facilities over the internet that offers dynamic storage and processing requirement.
- **Data bus**. The data from the cloud is processed in the system data bus depending on the data requirements, i.e. data that requires real time processing and analysis are routed in such a way that there is minimum delay.
- **Data warehouse**. The data routed through the data bus is passed to this component, where it will be stored, pre-processed and cleaned. It includes a big

database management system, is responsible for converting the raw data into a form that can be efficiently processed by the analytics engine.

- **Data analytics engine**. This is where the goal of analytics architecture lies on. It includes processing algorithms to extract meaningful insights from the raw data.
- **Data visualization**. The goal of this system is to present a visual depiction of analysis results as a support to the data analysis, so that it eases the business decision-making process.

To ensure the effectivity of the framework, a high level of availability, robustness and interoperability of the sub-systems is essential. However, there is a trade-off between these factors. A high level of availability means a high level of replication of the system components, which leads to high costs. Furthermore, robustness and interoperability increase the usage of the hardware. These trade-offs have to be balanced.

6.3 Deriving Actions for Supply Chain Analytics Operations

In the previous sub-chapter, the general architecture was explored to design the analytics infrastructure for a data-driven supply chain. This sub-chapter will handle the concrete courses of actions to ensure the promised delivery times in this case study. This set of actions is derived from the literature and is focused on the actual problems encountered by the "Delivery Platform".

The main reason of delayed shipments is that the time taken for them to arrive at the door of the end costumers from the seller is longer than what the "Delivery Platform" and the "Seller" agreed upon. This type of standards of shipping processes is detailed in what is called service-level agreement or SLA, and as a contract, the service provider is obliged to meet. In practice, the delivery operations managers lack insights whether shipments are delayed until the sellers submit a complaint. A constant monitoring tool and a support system for process owners are imperative to quickly obtain information of any potential delays, such as an alerting system. Herden and Bunzel (2018) explored the archetypes of SCA initiatives and recognized that **alerting SCM process owner automatically on indicators of predefined critical conditions or events** as one. Alerting SCM process owners, as an archetype, covers all the initiatives characteristics in a way that "*it improves various objectives, by providing means of automatic evaluation, triggering decision support on alert, with mixed teams of experts, who create purpose-built tools, for high velocity data analysis, using predictive and descriptive techniques.*" In this case study, the alerting system would allow process owners passively getting alerted on actions required depending on the critical situations, e.g. parcels have not left the consolidation hub for more than two days, which demands informing the responsible hub team and possibly escalating the problem to the manager. Therefore, defining the rules to critical situations is crucial to design an alerting system. The rules may represent numerical thresholds or conditions for a situation to be deemed as critical. Possible rules are as follows:

- Specify the maximum duration in which parcels are shipped from one instance to another within the supply chain. As shown in the data analysis, the duration variables (FHS-FDA and FDA-Delivery) have a big correlation to the shipment delays. Alerts trigger can be set to a certain threshold. Table 6 can be used as initial values for the threshold, e.g. by specifying the maximum FHS-FDA duration to two days, SCM process owner would be alerted when the parcels take more than that duration to be delivered.
- Specify the maximum duration in which parcels can stay from in each instance of the supply chain. The thresholds can be calculated from the duration variable between two instances minus the average travel time (the amount of time the parcels actually spend “on the road”).
- Adjust durations according to geographical conditions. The farthest the delivery country is located from the consolidation hub or seller, the longer it takes for the parcels to arrive. Furthermore, mountainous areas or islands usually have longer shipping duration.

Based on the proposed rules, the alerting system tools should monitor shipping processes constantly and recommend specific actions for improvements in case the alert is triggered.

The alerting system should be supplemented with the suitable analytics structure governance by **establishing a clear communication escalation process**. An action plan should be developed in such a way that issues can be handled in an appropriate and efficient manner. It includes procedures for communicating and escalation issues through the hierarchy of staff. Potential delay alerts can be implemented in the following manner: the delivery operations manager and the employee responsible in the affected supply chain instance are alerted in case the rules are not fulfilled or parameters go below a certain threshold in the form of an app or email. The feature allows a direct communication between the two parties in order to clarify the issue. Escalation to the staff next in the hierarchy happens if there is communication problem, e.g. lack of response from employee of the supply chain instance.

Another action that can be taken to ensure delivery time is **increasing the number or types of data acquisition devices** throughout the supply chain. This is intended to diagnose potential delays earlier. One good use case specifically for this “Delivery Platform” supply chain is the usage of RFID tags that capture the incoming pallets or parcels automatically in the consolidation hub. From experience, there are times where the consolidation hub employees unsuccessfully or even forgot to scan the barcodes of the incoming goods—thus, the FHS time data were not gathered correctly. Most of the time this data had to be retrospectively entered into the data storage system. Acquiring an RFID on each parcel can reduce human error in this work process and optimize the workflow. However, there is a trade-off between data availability and costs and it should be taken into consideration in the data analytics infrastructure.

6.4 Implications

This chapter discussed a combination of specific courses of action and analytics/IT infrastructure to support the compliance of delivery time, as stated in the SLA. The general goal of this chapter is how to gain valuable insights for business decision-making from the application of analytics in predicting delivery time.

To get a better idea of how the analytics/IT infrastructure should be designed, the baseline for an analytics organization structure found in literature is analysed. Beside software infrastructure, a clear analytics governance is vital for analytics modelling to meet its goal from the business perspective. The analytics infrastructure for the specific “Delivery Platform” supply chain is developed. The proposed analytics infrastructure differs in some points from the current data architecture in the “Delivery Platform” (see Figure 11). The current data architecture mainly has one-way communication—from the data sources to the internal BI platform and tracking system. There is a lack of supply chain collaboration, i.e. enabling the sharing of information and knowledge within the supply chain players and its exploitation for key activities. Collaborative SCA secures the information sharing between an organization with external business partners to carry out supply chain operations and helps the organization achieve a more integrated supply chain (Wang et al. 2016). Pushing for a higher integration of the supply chain instances into the communication system, such as consolidation hub and LMC hub, would be a great example of this.

Finally, recommended actions are proposed - deriving from the results of the previous data analysis chapter, common supply chain issues encountered in Delivery Platform and studies from the literature in the previous sub-chapters. Potential delays can be better handled by alerting SCM process owners and can be better diagnosed by increasing data acquisition devices. However, data integration is not an easy and cheap task and the tools to execute these measures possibly lead to a considerable investment cost.

7. Overall Summary and Outlook

The aim of this chapter is to draw conclusions of this study toward the defined main problem. This is based on the whole process of solving the three research questions step by step, as shown in the first sub-chapter. The last two sub-chapters will deal with the recommendations and limitations of the thesis.

7.1 Conclusion

RQ1: What predictive analytics techniques, based on historical data as input, can be used to predict delivery time?

An initial SLR was conducted in chapter 4 to get an overview of the status quo studies in predictive analytics techniques, in which a total of 26 articles are selected for further review. It showed that most of the publications employed ML models to predict arrival or travel time in public transports, with ANN, SVM and *k*-NN as the most used algorithms. Motivated by the findings, this study continued with an extensive data analysis of various ML models to predict the delivery time using the historical data from "Delivery Platform". The data contains more than 60 000 rows of shipments labelled as either delayed or not delayed. The aim of the analysis is to build an ML model that can predict accurately the label of the shipments. From six ML models, three models that achieved the highest cross-validation accuracy scores were selected for further review, e.g. *k*-NN, Random Forest, CART. Despite of the suboptimal learning curves, the author believes that ML models, especially the three mentioned previously, are good predictive analytics techniques to predict delivery time in commercial setting. The data correctness and the appropriate scaling method have to be ensured to result in good predicting models. Missing values and outliers would affect negatively in the models.

RQ2: What are the factors that have an impact on delivery delays?

Various factors that impact travel or arrival time were revealed from the SLR. Some of the literature use historical data (travel or arrival time of prior journeys on the same span of time) and real time data (road traffic condition, weather, time of day, etc.). In the domain of public transportation, there are generally four types of factors that contribute travel time: infrastructure, external, driver behaviour and operation management factors.

However, in this case study, there is a lack of real time data. The data used for the analysis are historical data obtained from the internal BI platform. From the results of the decision tree-based models, the shipping time variables FHS-FDA and FDA-Delivery unsurprisingly represent the most influence in the prediction results (see Figure 27). In this study, FHS – FDA is a feature added to describe the elapsed time between the incoming parcel in the consolidation hub and the first delivery attempt by the LMC at the end customer address. Meanwhile, FDA-Delivery describes the elapsed time between the first delivery attempt and the actual time when the parcel is received by the end customer. The longer the shipping time variables are, the more likely is a parcel to be delayed. Cases of unclear addresses, defective barcodes, lack of shipment, or even human error can be the reasons for the long FHS-FDA.

Additionally, the local LMC selection, the day of first hub scan (FHS day) and the day of delivery (Delivery day) have influence in the delivery time, even though these features are far unimportant than the two previously mentioned shipping time variables. In practice, the data integration between the LMCs and "Delivery Platform" is not sufficient –each LMC has its own parcel tracking portal and usually issues have to be resolved through emails and telephone calls.

The author believes that these features are factors that impact greatly in the overall delivery time in e-commerce setting.

RQ3: What courses of actions and restructuring of the analytics or IT system should be undertook in the organization to ensure on time delivery time?

To bring business values, the analytics modelling performed in chapter 5 should be translated into key operational actions and their supporting infrastructure, which was discussed in chapter 6. Based on the theoretical framework of analytics organizational structure, a proper data analytics infrastructure that puts importance on collaborative SCA is necessary to ensure the smooth communication between the supply chain organization and its external business partners. The proposed analytics infrastructure, shown in Figure 29, represents a high level of data integration between the supply chain players.

The suggested actions are derived from the analytics infrastructure, the common issues in "Delivery Platform" and the results from the data analysis. One action is alerting SCM process owner automatically on indicators of predefined critical conditions or events. For this purpose, a set of exemplary rules for critical conditions were proposed. To execute this alerting system, the analytics governance of the organization needs to be reviewed and a clear communication escalation process needs to be established. Furthermore, increasing the number or types of data acquisition devices throughout the supply chain, e.g. RFID tags in the incoming pallets/parcels at the consolidation hub.

7.2 Management Recommendations

In conclusion, the recommendations are divided based on the level of decision-making process: strategic and operational.

Strategic:

- Establish a clear analytics governance.
- Invest in data integration towards a collaborative SCA
- Develop an alerting system for delivery time
- Train delivery operations managers and external business partners

Operations:

- Perform periodical evaluation on the alerting system throughout all partners
- Specify rules for critical conditions
- Document the escalation process
- Enter the data, on which the rules for the alerting system is based, correctly

7.3 Limitations and Future Research

This study, as any other researches, is a subject to limitations due to time or data constraints. In this sub-chapter, the author will provide suggestions for future research that could be valuable for delivery platforms in order to continue optimizing their supply chain.

The findings of this thesis are focused on the application of predictive analytics in B2C e-commerce for predicting delivery times for goods on European-wide cross-border shipments. Based on the focus, there are few points that have to be acknowledged. First, the results are restricted for physical goods with small volume and low weight. Hence, it is uncertain whether the factors of delivery time, the proposed framework for analytics infrastructure and the courses of actions would be suitable for service supply chains.

Second, the findings are restricted to European B2C shipments. This is influenced by the supply chain of the organization and the LMC performance in Europe. Therefore, the factors that impact delivery time prediction might be different in other parts of the world, depending on their geographical conditions. A place with high uncertain traffic condition, for instance, would use other features or maybe techniques to predict delivery times. Not just traffic conditions, different locations might have a completely different baseline of supply chain architecture. Predicting delivery times in another continent would be a possible future research.

Third, the consolidation hub is in the same area as the seller's warehouse. The transportation between these two points is scheduled on a regular basis and has a fixed route. Thus, the elapsed time between pick up at the seller's warehouse and injection in the consolidation hub does not show a big impact in the end delivery time.

Fourth, the features considered for the research are historical data from the internal BI platform. Therefore, no real-time or environmental data was taken into consideration in this study due to the large geographical location and time constraint. Future research could be performed by putting more features in the analytics models, such as weather and traffic conditions.

8. Bibliography

Abdi, Hervé; Williams, Lynne J. (2010): Principal component analysis. In *WIREs Computational Statistics (Wiley Interdisciplinary Reviews: Computational Statistics)* 2 (4), pp. 433–459. DOI: 10.1002/wics.101.

Agafonov, A. A.; Chernov, A. V.; Sergeev, A. V. (2015): Using satellite monitoring and statistical data to predict arrival time of city public transport. In *Pattern Recognition and Image Analysis* 25 (3), pp. 385–388. DOI: 10.1134/S1054661815030013.

Altinkaya, Mehmet; Zontul, Metin (2013): Urban bus arrival time prediction: a review of computational models. In *International Journal of Recent Technology and Engineering (IJRTE)* 2 (4), pp. 164–169, checked on 6/19/2019.

Arlot, Sylvain; Celisse, Alain (2010): A survey of cross-validation procedures for model selection. In *Statistics Surveys* 4 (0), pp. 40–79. DOI: 10.1214/09-SS054.

Behrens, John T. (1997): Principles and procedures of exploratory data analysis. In *Psychological Methods* 2 (2), pp. 131–160. DOI: 10.1037/1082-989X.2.2.131.

Bergstra, James; Bengio, Yoshua (2012): Random search for hyper-parameter optimization. In *The Journal of Machine Learning Research* 13 (1), pp. 281–305. Available online at http://dl.acm.org/ft_gateway.cfm?id=2188395&type=pdf.

Biswas, Sanjib; Sen, Jaydip (Eds.) (2016): A proposed framework of next generation supply chain management using big data analytics. National Conference on Emerging Trends in Business and Management. Kolkata, India (Proceeding of the National Conference on Emerging Trends in Business and Management).

Brownlee, Jason (2014): Discover feature engineering, how to engineer features and how to get good at it. Available online at https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/, updated on 9/26/2014, checked on 7/8/2019.

Čelan, Marko; Lep, Marjan (2017): Bus arrival time prediction based on network model. In *Procedia Computer Science* 113, pp. 138–145. DOI: 10.1016/j.procs.2017.08.331.

Cerasis (2018): Three key trends in logistics e-commerce. Available online at https://cerasis.com/2018/01/04/logistics-e-commerce/, checked on 12/27/2018.

Chawla, Nitesh V. (2010): Data mining for imbalanced datasets: An overview. In Oded Z. Maimon, Lior Rokach (Eds.): Data mining and knowledge discovery handbook, vol. 30. 2nd ed. New York, London: Springer, pp. 875–886.

Chen, Chi-Hua (2018): An arrival time prediction method for bus system. In *IEEE Internet Things J.* 5 (5), pp. 4231–4232. DOI: 10.1109/JIOT.2018.2863555.

Chen, Guojun; Yang, Xiaoguang; Liu, Haode; Liu, Xianglong (2013): Regression-based approach for bus trajectory estimation. In : 16th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2013. 6-9 Oct. 2013, Kurhaus, The Hague, The Netherlands. 2013 16th International IEEE Conference on Intelligent Transportation Systems - (ITSC 2013). The Hague, Netherlands, 10/6/2013 - 10/9/2013. IEEE. Piscataway, NJ: IEEE, pp. 1876–1881.

Chicco, Davide (2017): Ten quick tips for machine learning in computational biology. In *BioData Mining* 10. DOI: 10.1186/s13040-017-0155-3.

Choudhary, Rubina; Khamparia, Aditya; Gahier, Amandeep Kaur (2016): Real time prediction of bus arrival time: A review. In Amit Agarwal (Ed.): Proceedings on 2016 2nd International Conference on Next Generation Computing Technologies (NGCT). October 14th-16th, 2016, Center for Information Technology, University of Petroleum and Energy Studies, Dehradun. 2016 2nd International Conference on Next Generation Computing Technologies (NGCT). Dehradun, India, 10/14/2016 - 10/16/2016. Piscataway, NJ: IEEE, pp. 25–29.

Claesen, Marc; Moor, Bart De (2015): Hyperparameter search in machine learning. Available online at http://arxiv.org/pdf/1502.02127v2.

Clarivate Analytics (2017): Web of Science. Available online at https://clarivate.com/products/web-of-science/, checked on 6/6/2019.

Corporate Finance Institute (2018): Nominal data. Corporate Finance Institute. Available online at https://corporatefinanceinstitute.com/resources/knowledge/other/nominal-data/, updated on 12/3/2018.

Cortes, Corinna; Vapnik, Vladimir (1995): Support-vector networks. In *Machine Learning* 20 (3), pp. 273–297. DOI: 10.1007/BF00994018.

DOMO, Inc. (2018): Data never sleeps 6.0.

Dong, Jian; Zou, Lu; Zhang, Yan (2013): Mixed model for prediction of bus arrival times. In IEEE Staff (Ed.): 2013 IEEE Congress on Evolutionary Computation (CEC). 2013 IEEE Congress on Evolutionary Computation. Cancun, 6/20/2013 - 6/23/2013. Piscataway: IEEE, pp. 2918–2923, checked on 1/24/2019.

Durach, Christian F. (2015): A theoretical and practical contribution to supply chain robustness. Dissertation. Technische Universität Berlin. Available online at http://nbn-resolving.de/urn:nbn:de:101:1-201804177659, checked on 1/10/2019.

EBSCO (2017): Business source complete. Available online at https://www.ebsco.com/products/research-databases/business-source-complete, checked on 6/6/2019.

E-commerce Europe (2018): Press kit for the European B2C e-commerce report 2018, checked on 1/3/2019.

Erren, Thomas C.; Cullen, Paul; Erren, Michael (2009): How to surf today's information tsunami: on the craft of effective reading. In *Medical hypotheses* 73 (3), pp. 278–279. DOI: 10.1016/j.mehy.2009.05.002.

Eurostat (2018): E-commerce Statistics for Individuals. Available online at https://ec.europa.eu/eurostat/statistics-explained/pdfscache/46776.pdf.

Fadaei, Masoud; Cats, Oded; Bhaskar, Ashish (2016): A hybrid scheme for real-time prediction of bus trajectories. In *J. Adv. Transp.* 50 (8), pp. 2130–2149. DOI: 10.1002/atr.1450.

Ferreira, Hugo (2018): Dealing with categorical features in machine learning. Available online at https://medium.com/hugo-ferreiras-blog/dealing-with-categorical-features-in-machine-learning-1bb70f07262d, updated on 6/25/2018.

Fink, Arlene (2005): Conducting research literature reviews. From the Internet to paper. Arlene Fink. 2nd ed. Thousand Oaks, Calif.,, London: SAGE.

Godwin, T.; Gopalan, Ram; Narendran, T. T. (2015): Estimating order delivery times and fleet capacity in freight rail networks: part II - analytic approximation. In *IJOR* 24 (4), p. 369. DOI: 10.1504/IJOR.2015.072722.

Gomez-Herrera, Estrella; Martens, Bertin; Turlea, Geomina (2013): The drivers and impediments for cross-border e-Commerce in the EU. Joint Reseach Centre.

Grossman, Robert L. (2018): A framework for evaluating the analytic maturity of an organization. In *International Journal of Information Management* 38 (1), pp. 45–51. DOI: 10.1016/j.ijinfomgt.2017.08.005.

Grossman, Robert L.; Siegel, Kevin P. (2014): Organizational models for big data and analytics. In *JOD* 3 (1), p. 20. DOI: 10.7146/jod.9799.

Gunasekaran, Anggapa; Papadopoulos, Thanos; Dubey, Rameshwar; Wamba, Samuel Fosso; Childe, Stephen J.; Hazen, Benjamin; Akter, Shahrier (2017): Big data and predictive analytics for supply chain and organizational performance. In *Journal of Business Reserach* 70, 308-317. Available online at https://doi.org/10.1016/j.jbusres.2016.08.004, checked on 12/28/2018.

Guyon, Isabelle; Elisseeff, André (2003): An introduction to variable and feature selection. In *Journal of machine learning research* 3 (Mar), pp. 1157–1182.

Hale, Jeff (2018): Smarter ways to encode categorical data for machine learning. Available online at https://towardsdatascience.com/smarter-ways-to-encode-categorical-data-for-machine-learning-part-1-of-3-6dca2f71b159, updated on 9/11/2018.

Hawkins, Douglas M. (2004): The problem of overfitting. In *Journal of chemical information and computer sciences* 44 (1), pp. 1–12. DOI: 10.1021/ci0342472.

Hazen, Benjamin T.; Boone, Christopher A.; Ezell, Jeremy D.; Jones-Farmer, L. Allison (2014): Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. In *International Journal of Production Economics* 154, pp. 72–80. DOI: 10.1016/j.ijpe.2014.04.018.

He, Haibo; Bai, Yang; Garcia, Edwardo A.; Li, Shutao (2008): ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In : IEEE International Joint Conference on Neural Networks, 2008. IJCNN 2008 (IEEE World Congress on Computational Intelligence) : 1-8 June, 2008, [Hong Kong, China. 2008 IEEE International Joint Conference on Neural Networks (IJCNN 2008 - Hong Kong). Hong

Kong, China, 6/1/2008 - 6/8/2008. IEEE World Congress on Computational Intelligence. Piscataway, N.J.: IEEE, pp. 1322–1328.

He, Zhiying; Yu, Haitao; Du Yong; Wang, Jingjing (2013): SVM based multi-index evaluation for bus arrival time prediction. In : International Conference on ICT Convergence (ICTC), 2013. 14-16 Oct. 2013, Ramada Plaza Jeju Hotel, Jeju Island, Korea. 2013 International Conference on ICT Convergence (ICTC). JEJU ISLAND, Korea (South), 10/14/2013 - 10/16/2013. International Conference on ICT Convergence; International Conference on Information and Communication Technology Convergence; ICTC. Piscataway, NJ: IEEE, pp. 86–90, checked on 1/26/2019.

Heaton, Jeff (2016): An empirical analysis of feature engineering for predictive modeling. In SoutheastCon (Ed.): SoutheastCon 2016. Marriott Norfolk Waterside Hotel, 30 Mar 2016-03 Apr 2016. SoutheastCon 2016. Norfolk, VA, USA, 3/30/2016 - 4/3/2016. SoutheastCon; IEEE SoutheastCon. [Piscataway, NJ]: IEEE, pp. 1–6.

Herden, Tino T.; Bunzel, Steffen (2018): Archetypes of supply chain analytics initiatives—An exploratory study 2 (2). DOI: 10.14279/DEPOSITONCE-8923.

Iosifidis, Vasileios; Ntoutsi, Eirini (2018): Dealing with bias via data augmentation in supervised learning scenarios. Available online at https://pdfs.semanticscholar.org/6b44/b4e0b9cf14674ddaed8cea65d55b7ac9ad3d.pdf.

Jiang, Shengyi; Pang, Guansong; Wu, Meiling; Kuang, Limin (2012): An improved K-nearest-neighbor algorithm for text categorization. In *Expert Systems with Applications* 39 (1), pp. 1503–1509. DOI: 10.1016/j.eswa.2011.08.040.

Jordan, M. I.; Mitchell, T. M. (2015): Machine learning: Trends, perspectives, and prospects. In *Science (New York, N.Y.)* 349 (6245), pp. 255–260. DOI: 10.1126/science.aaa8415.

Jorge, Romero; Felix, Vega; Cesar, Pedraza (2016): Design and implementation of a system to estimate arrival times of public transport vehicles with active RFID technology. In Yuli Andrea Rodríguez, Euro American Conference on Telematics and Information Systems (Eds.): 2016 8th Euro American Conference on Telematics and Information Systems (EATIS). Conference proceedings, 27-29 April 2016, Cartagena, Colombia. 2016 8th Euro American Conference on Telematics and Information Systems (EATIS). Cartagena, Colombia, 4/28/2016 - 4/29/2016. Euro American Conference on Telematics and Information Systems; EATIS; IEEE EATIS. [Piscataway, NJ]: IEEE, pp. 1–4, checked on 1/24/2019.

Komarek, Paul; Moore, Andrew W. (2005): Making logistic regression a core data mining tool: A practical investigation of accuracy, speed, and simplicity. Carnegie Mellon University. Pittsburgh, PA (CMU-RI-TR-05-27).

Kumar, B. Anil; Kumar, Vivek; Vanajakshi, Lelitha; Subramanian, Shankar (2017a): Performance comparison of data driven and less data demanding techniques for bus travel time Prediction. In *European Transport* 65 (9), checked on 5/9/2019.

Kumar, B. Anil; Vanajakshi, Lelitha; Subramanian, Shankar C. (2017b): Pattern-based time-discretized method for bus travel time prediction. In *Journal of Transportation Engineering, Part A: Systems* 143 (6), p. 4017012. DOI: 10.1061/JTEPBS.0000029.

Kumar, B. Anil; Vanajakshi, Lelitha; Subramanian, Shankar C. (2018): A hybrid model based method for bus travel time estimation. In *Journal of Intelligent Transportation Systems* 22 (5), pp. 390–406. DOI: 10.1080/15472450.2017.1378102.

Kumar, Vaibhav; Garg, M. L. (2017): Deep learning in predictive analytics: A survey. In International Conference on Emerging Trends Computing and Communication in Technologies (Ed.): 2017 International Conference on Emerging Trends in Computing and Communication Technologies, ICETCCT-2017. November 17-18, 2017 : venue: Graphic Era Hill University, Society Area Clement Town, Dehradun, Uttarakhand, India. 2017 International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT). Dehradun, 11/17/2017 - 11/18/2017. International Conference on Emerging Trends in Computing and Communication Technologies; ICETCCT. [Piscataway, NJ]: IEEE, pp. 1–6.

Kumar, Vivek; Kumar, B. Anil; Vajanakshi, Lelitha; Subramanian, Shankar (2014): Comparison of model based and machine learning approaches for bus arrival time prediction. In : 93rd Annual Meeting of Transportation 2014. Washington, D. C., checked on 1/24/2019.

Lenzerini, Maurizio (2002): Data integration. In Serge Abiteboul, Phokion G. Kolaitis, Lucian Popa (Eds.): PODS 2002. the twenty-first ACM SIGMOD-SIGACT-SIGART symposium. Madison, Wisconsin, 6/3/2002 - 6/5/2002. Association for Computing

Machinery. SIGMOD; Association for Computer Machinery. SIGART; Association for Computing Machinery. SIGACT. New York, New York, USA: ACM Press, p. 233.

Li, Jinglin; Gao, Jie; Yang, Yu; Wei, Heran (2017): Bus arrival time prediction based on mixed model. In *China Commun.* 14 (5), pp. 38–47. DOI: 10.1109/CC.2017.7942193.

Li, Kai; Rollins, Jason; Yan, Erjia (2018): Web of Science use in published research and review papers 1997-2017: a selective, dynamic, cross-domain, content-based analysis. In *Scientometrics* 115 (1), pp. 1–20. DOI: 10.1007/s11192-017-2622-5.

Liu, Tao; Ma, Jihui; Guan, Wei; Song, Yue; Niu, Hu (2012): Bus arrival time prediction based on the k-nearest neighbor method. In IEEE (Ed.): 2012 Fifth International Joint Conference on Computational Sciences and Optimization. 2012 Fifth International Joint Conference on Computational Sciences and Optimization (CSO). Harbin, Heilongjiang, China, 6/23/2012 - 6/26/2012. Piscataway: IEEE, pp. 480–483, checked on 1/24/2019.

Locklin, Scott (2014): Neglecting machine learning ideas. Available online at https://scottlocklin.wordpress.com/2014/07/22/neglected-machine-learning-ideas/, updated on 7/22/2014, checked on 7/8/2019.

Maiti, Santa; Pal, Arpan; Pal, Arindam; Chattopadhyay, T.; Mukherjee, Arijit (2014): Historical data based real time prediction of vehicle arrival time. In : Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on. Date 8-11 Oct. 2014. 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC). Qingdao, China, 10/8/2014 - 10/11/2014. [Piscataway, N.J.]: IEEE, pp. 1837–1842.

Mitchell, Tom M. (1997): Machine learning. New York, London: McGraw-Hill (McGraw-Hill series in computer science).

Morganti, Eleonora; Seidel, Saskia; Blanquart, Corinne; Dablanc, Laetitia; Lenz, Barbara (2014): The impact of e-commerce on final deliveries: Alternative parcel delivery services in France and Germany. In *Transportation Research Procedia* 4, pp. 178–190. DOI: 10.1016/j.trpro.2014.11.014.

Myung, In Jae (2003): Tutorial on maximum likelihood estimation. In *Journal of Mathematical Psychology* 47 (1), pp. 90–100. DOI: 10.1016/S0022-2496(02)00028-7.

Nassif, Ali Bou; Azzeh, Mohammad; Banitaan, Shadi; Neagu, Daniel (2016): Guest editorial: special issue on predictive analytics using machine learning. In *Neural Computing and Applications* 27 (8), pp. 2153–2155. DOI: 10.1007/s00521-016-2327-3.

Nenty, H. Johnson (2009): Writing a quantitative research thesis. In *International Journal of Educational Sciences* 1 (1), pp. 19–32. DOI: 10.1080/09751122.2009.11889972.

Okoli, Chitu; Schabram, Kira (2010): A guide to conducting a systematic literature review of information systems research. In *SSRN Journal*. DOI: 10.2139/ssrn.1954824.

Ongsulee, Pariwat; Chotchaung, Veena; Bamrungsi, Eak; Rodcheewit, Thanaporn (Eds.) (2018): Big data, predictive analytics and machine learning. Sixteenth International Conference on ICT and Knowledge Engineering: IEEE.

Patro, S. Gopal Krishna; Sahu, Kishore Kumar (2015): Normalization: A preprocessing stage. Available online at http://arxiv.org/pdf/1503.06462v1.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O. et al. (2011): Scikit-learn: Machine Learning in Python. In *Journal of machine learning research* 12, pp. 2825–2830.

Peng, Zixuan; Jiang, Yonglei; Yang, Xiaoli; Zhao, Zhigang; Zhang, Liu; Wang, Yitian (2018): Bus arrival time prediction based on PCA-GA-SVM. In *NNW* 28 (1), pp. 87–104. DOI: 10.14311/NNW.2018.28.005.

Petticrew, Mark; Roberts, Helen (2006): Systematic reviews in the social sciences. Oxford, UK: Blackwell Publishing Ltd.

PostNord (2017): E-commerce in Europe 2017. Andersson, Arne. Available online at https://www.postnord.fi/siteassets/raportit/e-commerce-in-europe-2017_en_low.pdf, checked on 1/5/2019.

Provost, Foster (2000): Machine learning from imbalanced data sets 101. In *AAAI Technical Report WS-00-05* Workshop on Imbalanced Data Sets.

Pyle, Dorian (1999): Data preparation for data mining. San Francisco, Calif.: Morgan Kaufmann; London : Taylor & Francis [distributors].

Rocca, Baptiste (2019): Handling imbalanced datasets in machine learning. What should and should not be done when facing an imbalanced classes problem? Available online at https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28.

Schoenherr, Tobias; Speier-Pero, Cheri (2015): Data science, predictive analytics, and big data in supply chain management: Current state and future potential. In *Journal of Business Logistics* 36 (1), pp. 120–132. DOI: 10.1111/jbl.12082.

Shalev-Shwartz, Shai; Ben-David, Shai (2014): Understanding machine learning. From theory to algorithms. Cambridge: Cambridge University Press.

Shmueli, Galit; Koppius, O. (2010): Predictive Analytics in information systems research. In *SSRN Journal*. DOI: 10.2139/ssrn.1606674.

Stefanovic, Nenad (2014): Proactive supply chain performance management with predictive analytics. In *The Scientific World Journal* 2014, p. 528917. DOI: 10.1155/2014/528917.

Straube, Frank (2018): Einführungsveranstaltung: Grundlagen des wissenschaftlichen Arbeitens. Berlin.

Stumm, Marielle; Bollo, Daniel (2004): E-commerce and end delivery issues. In *Logistics Systems for Sustainable Cities*, pp. 405–419. Available online at https://doi.org/10.1108/9780080473222-029, checked on 12/27/2018.

Sun, Yanmin; Kamel, Mohamed; Wang, Yang (2006): Boosting for learning multiple classes with imbalanced class distribution. In Christopher Wade Clifton (Ed.): Sixth International Conference on Data Mining. ICDM 2006 : proceedings : 18-22 December, 2006, Hong Kong. Sixth International Conference on Data Mining (ICDM'06). Hong Kong, China, 12/18/2006 - 12/22/2006. Los Alamitos, Calif.: IEEE Computer Society, pp. 592–602.

Sun, Yao; Yan, Qianqian; Jiang, Yonglei; Zhu, X. F. (2017): Reliability prediction model of further bus service based on random forest. In *Journal of Algorithms & Computational Technology* 11 (4), pp. 327–335. DOI: 10.1177/1748301817725306.

Taylor, David (2016): Battle of the data science venn diagrams. In *KDNuggets News*.

Timotheou, S. (2010): The random neural network: A survey. In *The Computer Journal* 53 (3), pp. 251–267. DOI: 10.1093/comjnl/bxp032.

Tranfield, David; Denyer, David; Smart, Palminder (2003): Towards a methodology for developing evidence-informed management knowledge by means of systematic review. In *British Journal of Management* 14 (3), pp. 207–222. DOI: 10.1111/1467-8551.00375.

Treethidtaphat, Wichai; Pattara-Atikom, Wasan; Khaimook, Sippakorn (2017): Bus arrival time prediction at any distance of bus route using deep neural network model. In IEEE Intelligent Transportation Systems Conference (Ed.): IEEE ITSC 2017. 20th International Conference on Intelligent Transportation Systems : Mielparque Yokohama in Yokohama, Kanagawa, Japan, October 16-19, 2017. 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). Yokohama, 10/16/2017 - 10/19/2017. IEEE Intelligent Transportation Systems Conference; Intelligent Transportation Systems Conference; IEEE ITSC; ITSC. Piscataway, NJ: IEEE, pp. 988–992, checked on 1/24/2019.

van der Spoel, Sjoerd; Amrit, Chintan; van Hillegersberg, Jos (2017): Predictive analytics for truck arrival time estimation: a field study at a European distribution centre. In *International Journal of Production Research* 55 (17), pp. 5062–5078. DOI: 10.1080/00207543.2015.1064183.

Vapnik, V. N. (1999): An overview of statistical learning theory. In *IEEE transactions on neural networks* 10 (5), pp. 988–999. DOI: 10.1109/72.788640.

Varela Rozados, Ivan; Tjahjono, Benny (2014): Big data analytics in supply chain management: Trends and related research. In : Proceedings of the 6th International Conference on Operations and Supply Chain Management. 6th International Conference on Operations and Supply Chain Management. Bali, Indonesia. Bali.

Waller, Matthew A.; Fawcett, Stanley E. (2013): Data science, predictive analytics, and big data: A revolution that will transform supply chain design and management. In *Journal of Business Logistics* 34 (2), pp. 77–84. DOI: 10.1111/jbl.12010.

Wang, Gang; Gunasekaran, Angappa; Ngai, Eric W.T.; Papadopoulos, Thanos (2016): Big data analytics in logistics and supply chain management: Certain investigations for research and applications. In *International Journal of Production Economics* 176, pp. 98–110. DOI: 10.1016/j.ijpe.2016.03.014.

Wang, Lei; Zuo, Zhongyi; Fu, Junhao (2014): Bus arrival time prediction using RBF neural networks adjusted by online data. In *Procedia - Social and Behavioral Sciences* 138, pp. 67–75. DOI: 10.1016/j.sbspro.2014.07.182.

Witten, I. H.; Pal, Christopher J.; Frank, Eibe; Hall, Mark A. (2017): Data mining. Practical machine learning tools and techniques. Fourth edition. Cambridge, MA: Morgan Kaufmann. Available online at http://proquest.tech.safaribooksonline.de/9780128043578.

Wohlin, Claes (2014): Guidelines for snowballing in systematic literature studies and a replication in software engineering. In Martin Shepperd, Tracy Hall, Ingunn Myrtveit (Eds.): EASE 2014. The 18th International Conference on Evaluation and Assessment in Software Engineering : London, May 12th-14th, 2014. the 18th International Conference. London, England, United Kingdom, 5/13/2014 - 5/14/2014. New York, New York: Association for Computing Machinery (ICPS), pp. 1–10.

Yaghini, Masoud; Khoshraftar, Mohammad M.; Seyedabadi, Masoud (2013): Railway passenger train delay prediction via neural network model. In *Journal of Advanced Transportation* 47 (3), pp. 355–368. DOI: 10.1002/atr.193.

Yang, Ming; Chen, Chao; Wang, Lu; Yan, Xinxin; Zhou, Liping (2016): Bus arrival time prediction using support vector machine with genetic algorithm. In *NNW* 26 (3), pp. 205–217. DOI: 10.14311/NNW.2016.26.011.

Yin, Yonghua (2018): Deep learning with the random neural network and its applications, 10/8/2018. Available online at https://arxiv.org/pdf/1810.08653.

Yu, Chong Ho (2017): Oxford bibliographies online datasets.

Zakka, Kevin (2016): A complete guide to K-nearest-neighbors with applications in Python and R. Available online at https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/.

Zhang, Shichao; Zhang, Chengqi; Yang, Qiang (2003): Data preparation for data mining. In *Applied Artificial Intelligence* 17 (5-6), pp. 375–381. DOI: 10.1080/713827180.

9. Appendix

Appendix A Dataset

Tracking Code	Customer Address Zip Code	Customer Address City	Customer Address Country Code	Shop Name	Carrier Company & Country	Shipment Planned Pickup (PPU) Time	First Hub Scan (FHS) Time	First Delivery Attempt (FDA) Time	Shipment Delivery Time	Shipment # Delayed (*)
0001	47xx	AT_City1	AT	Seller	LMC1-AT	03/09/2018 16:00	04/09/2018 12:11	05/09/2018 10:29	05/09/2018 10:29	0
0002	74xxx	DE_City1	DE	Seller	LMC2-DE	12/09/2018 19:00	12/09/2018 15:58	13/09/2018 13:29	13/09/2018 13:29	0
0003	40xxx	FR_City1	FR	Seller	LMC1-FR	10/09/2018 18:00	11/09/2018 18:26	13/09/2018 12:49	13/09/2018 12:49	0
0004	94xxx	FR_City2	FR	Seller	LMC3-FR	10/09/2018 18:00	11/09/2018 02:00	13/09/2018 02:00	13/09/2018 02:00	0
0005	70xxx	DE_City2	DE	Seller	LMC3-DE	12/09/2018 19:00	12/09/2018 16:07	13/09/2018 12:03	13/09/2018 12:03	0
0006	93xxx	FR_City3	FR	Seller	LMC1-FR	03/09/2018 18:00	05/09/2018 16:46	07/09/2018 11:18	09/09/2018 19:29	0

(*) Boolean value (0 = on-time, 1 = delayed)

Appendix B Reviewed Literature

	Title	Author	Content
1	Predictive analytics for truck arrival time estimation: a field study at a European distribution centre	van der Spoel et al. (2017)	Causing factor of arrival time estimation
2	Performance Comparison of Data Driven and Data Demanding Techniques for Bus Travel Time Prediction	Kumar et al. (2017a)	Inputs and methods for accurate travel time prediction
3	Comparison of Model Based and Machine Learning Approaches for Bus Arrival Time Prediction	Kumar et al. (2014)	Bus stops as variable for time travel estimation
4	Estimating order delivery times and fleet capacity in freight rail networks: part II - analytic approximation	Godwin et al. (2015)	Delivery time quotation using analytic approximation to in rail freight
5	Design and implementation of a system to estimate arrival times of public transport vehicles with active RFID technology	Jorge et al. (2016)	Application of RFID, system architecture and implementation
6	Bus arrival time prediction based on Random Forest	Li et al. (2017)	Application of random forest and Space rectangular Coordinate System
7	Reliability prediction model of further bus service based on random forest	Sun et al. (2017)	Comparison of reliability evaluation with random forest and other techniques

	Title	Author	Content
8	An Arrival Time Prediction Method for Bus System	Chen (2018)	Application of RNN to randomly train NN models
9	Bus arrival time prediction at any distance of bus route using deep neural network model	Treethidtaphat et al. (2017)	Application of DNN using GPS data from a bus line
10	Railway passenger train delay prediction via neural network model	Yaghini et al. (2013)	NN model with high accuracy to predict delay of passenger trains
11	Bus Arrival Time Prediction Using RBF Neural Networks Adjusted by Online Data	Wang et al. (2014)	Prediction using historical data (mined by RBFNN model) and real-time situation
12	Historical Data based Real Time Prediction of Vehicle Arrival Time	Maiti et al. (2014)	Application of ANN and SVM regression models using real bus data
13	Bus Arrival Time Prediction Based on Mixed Model	Li et al. (2017)	Three-staged mixed model: Pattern training (mined by k-NN and K-means), single step prediction and multi-step prediction
14	Mixed model for prediction of bus arrival times	Dong et al. (2013)	Prediction using real-time traffic condition for short distance BAT and KNN for long distance BAT
15	Using satellite monitoring and statistical data to predict arrival time of city public transport	Agafonov et al. (2015)	Prediction using linear regression and using real-time GPS data, history data and timetable data

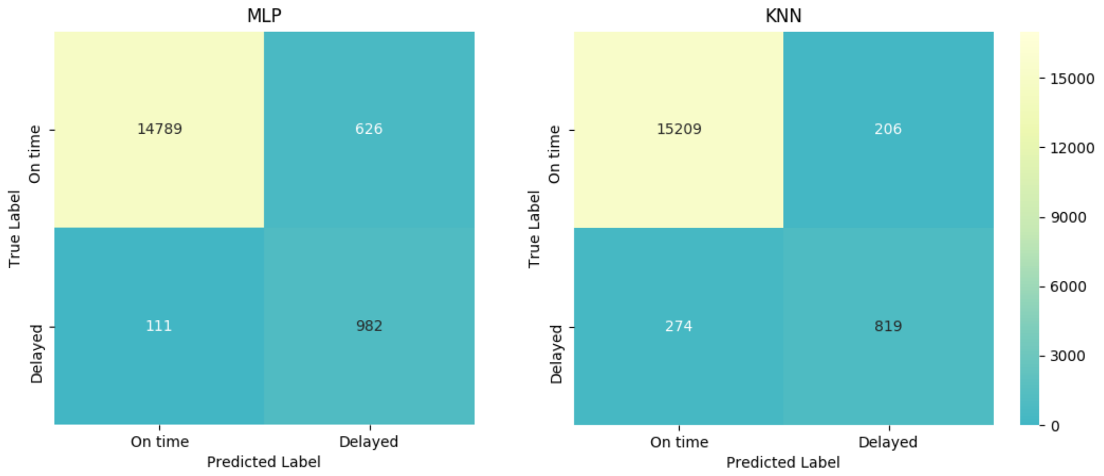
	Title	Author	Content
16	Regression-based approach for bus trajectory estimation	Chen et al. (2013)	Comparison of regression-based with historical-based model
17	A hybrid scheme for real-time prediction of bus trajectories	Fadaei et al. (2016)	Using inputs from schedule, instantaneous and historical data and linear regression heuristic to minimize prediction error
18	Bus arrival time prediction based on network model	Čelan and Lep (2017)	Real-time prediction of arrival times at bus stops using 4 types of data models
19	Bus Arrival Time Prediction Based on the k-Nearest Neighbour Method	Liu et al. (2012)	Modified k-NN method using historical bus GPS data
20	A hybrid model based method for bus travel time estimation	Kumar et al. (2018)	KNN-mined inputs and model using model combining exponential smoothing technique based on KF technique
21	Pattern-Based Time-Discretized Method for Bus Travel Time Prediction	Kumar et al. (2017b)	Travel time pattern analysis to find patterns in inputs and prediction using model-based Kalman filtering algorithm
22	Bus Arrival Time Prediction using Support Vector Machine with Genetic Algorithm	Yang et al. (2016)	Input vectors are adopted in SVM and GA search algorithm is combined to find the best parameters

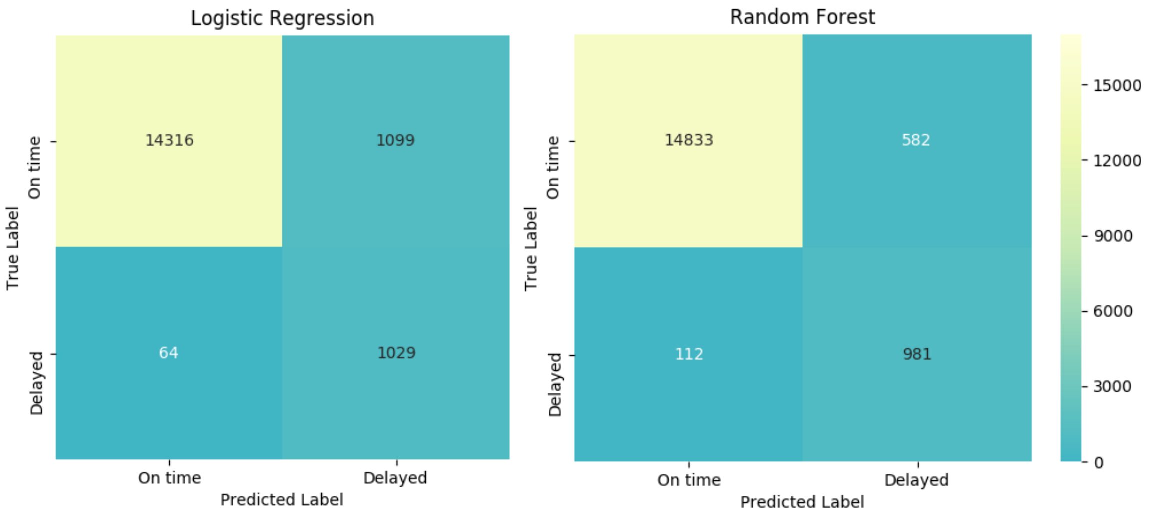
	Title	**Author**	**Content**
23	Bus arrival time prediction based on PCA-GA-SVM	Peng et al. (2018)	Comparative analysis between different types of SVM models
24	SVM based multi-index evaluation for bus arrival time prediction	He et al. (2013)	Model with indexes including GPS coverage, release rate, accuracy rate trained with SVM
25	Urban Bus Arrival Time Prediction: A Review of Computational Models	Altinkaya and Zontul (2013)	A review of prediction models
26	Real Time Prediction of Bus Arrival Time A Review	Choudhary et al. (2016)	A review of different prediction models

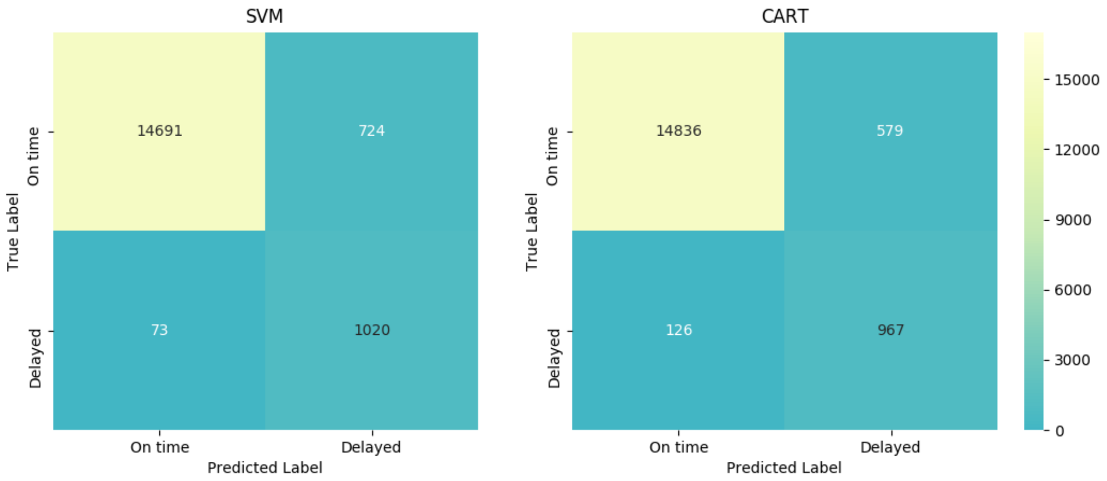
Appendix C Count of Predictive Techniques in the Literature

	ANN	k-NN	SVM	RF	Lin.Reg	LR	DT	Adaboost	Hybrid	KFT	Other non-ML models
S1		•	•	•			•	•			
S2	•									•	
S3	•									•	
S4											•
S5											•
S6				•							
S7	•		•	•							
S8	•					•					
S9	•										
S10	•					•	•				
S11	•				•						
S12	•		•								
S13		•								•	
S14	•	•									
S15					•						
S16					•						
S17					•						
S18											•
S19	•	•									
S20		•									
S21										•	
S22	•		•								
S23			•							•	
S24			•								
S25	•		•						•	•	
S26	•	•	•						•	•	
∑	13	6	8	3	4	2	2	1	2	7	3

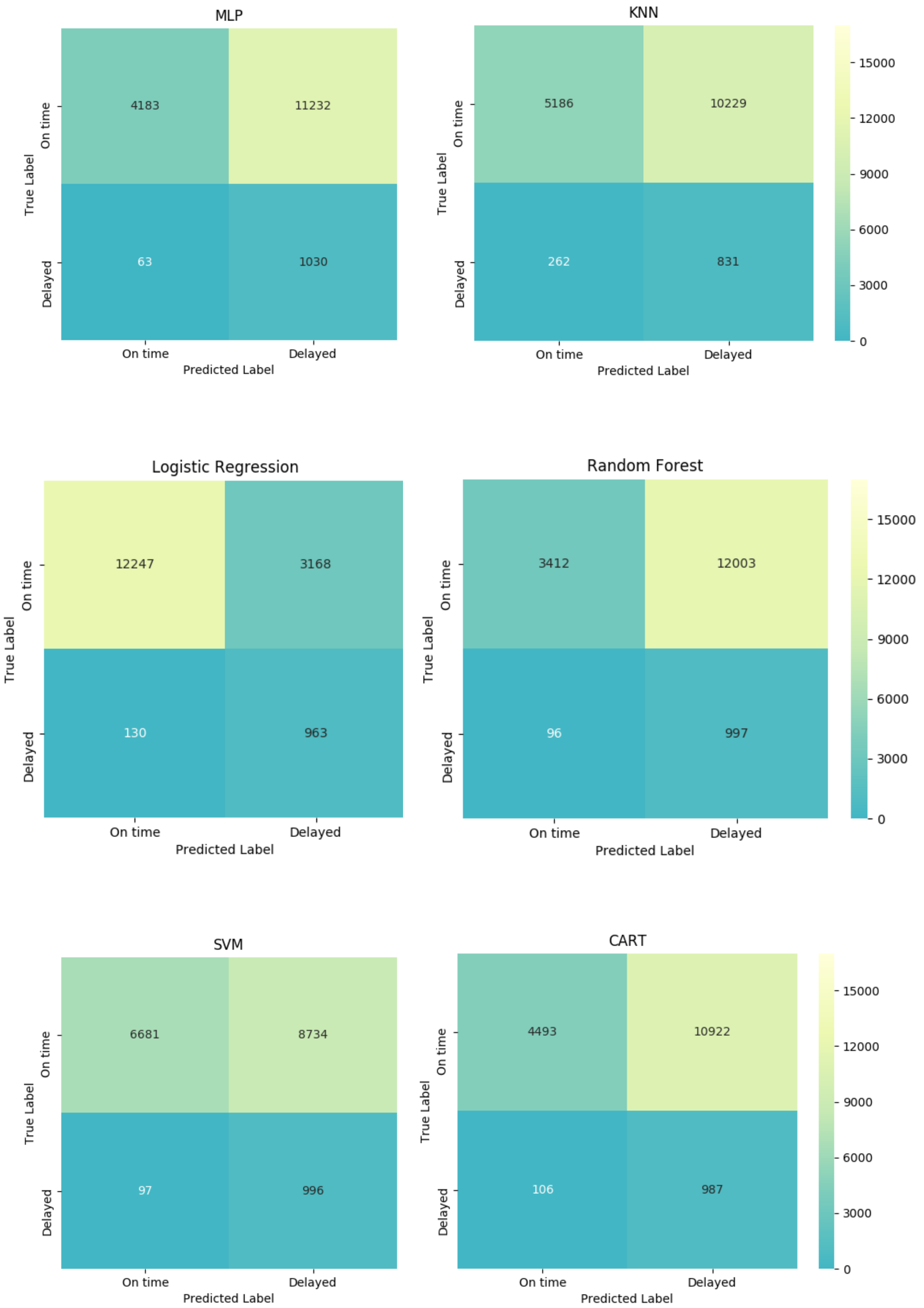
Appendix D Confusion Matrixes of Predictive Models (SMOTE)







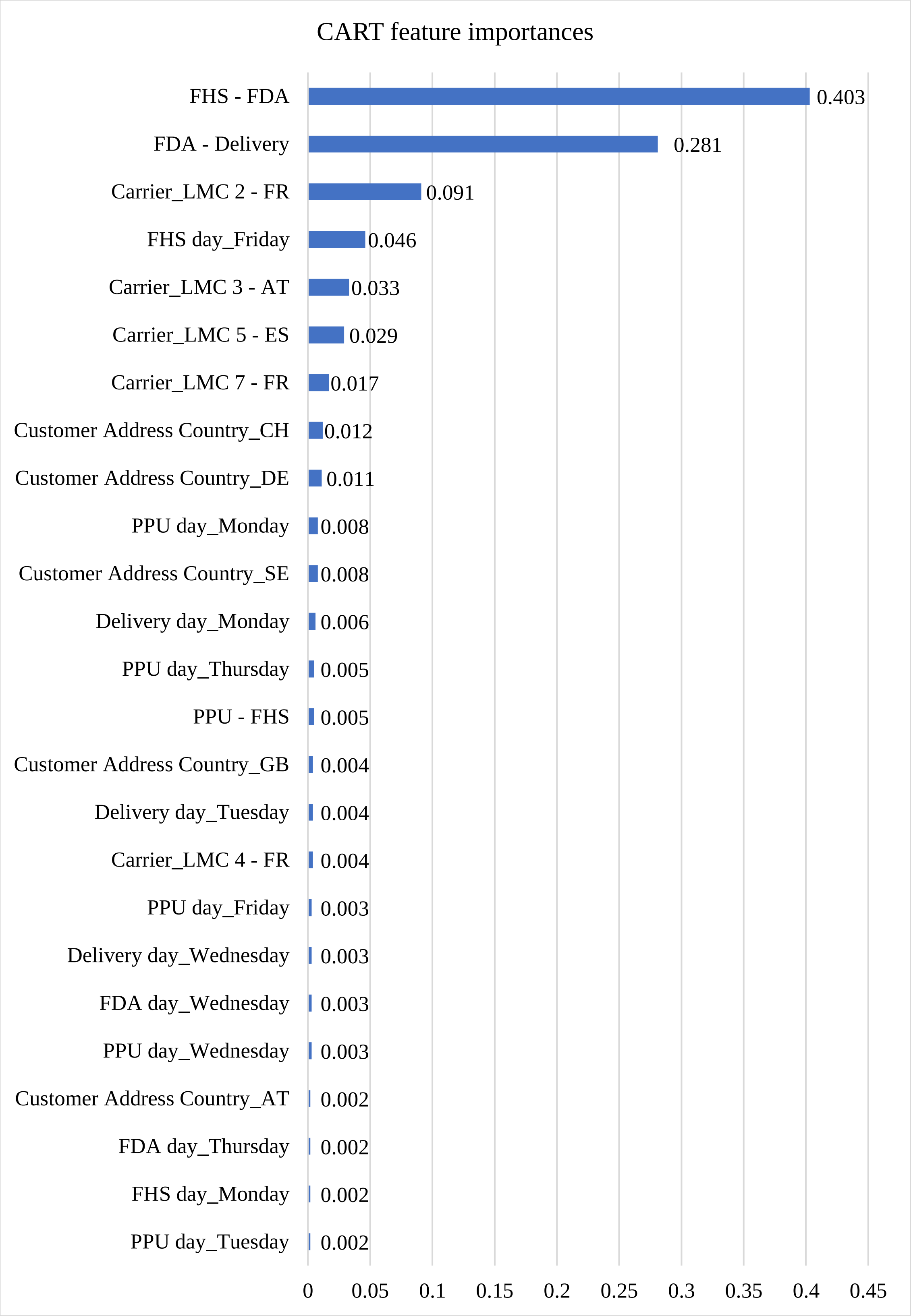
Appendix E Confusion Matrixes of Predictive Models (NearMiss)



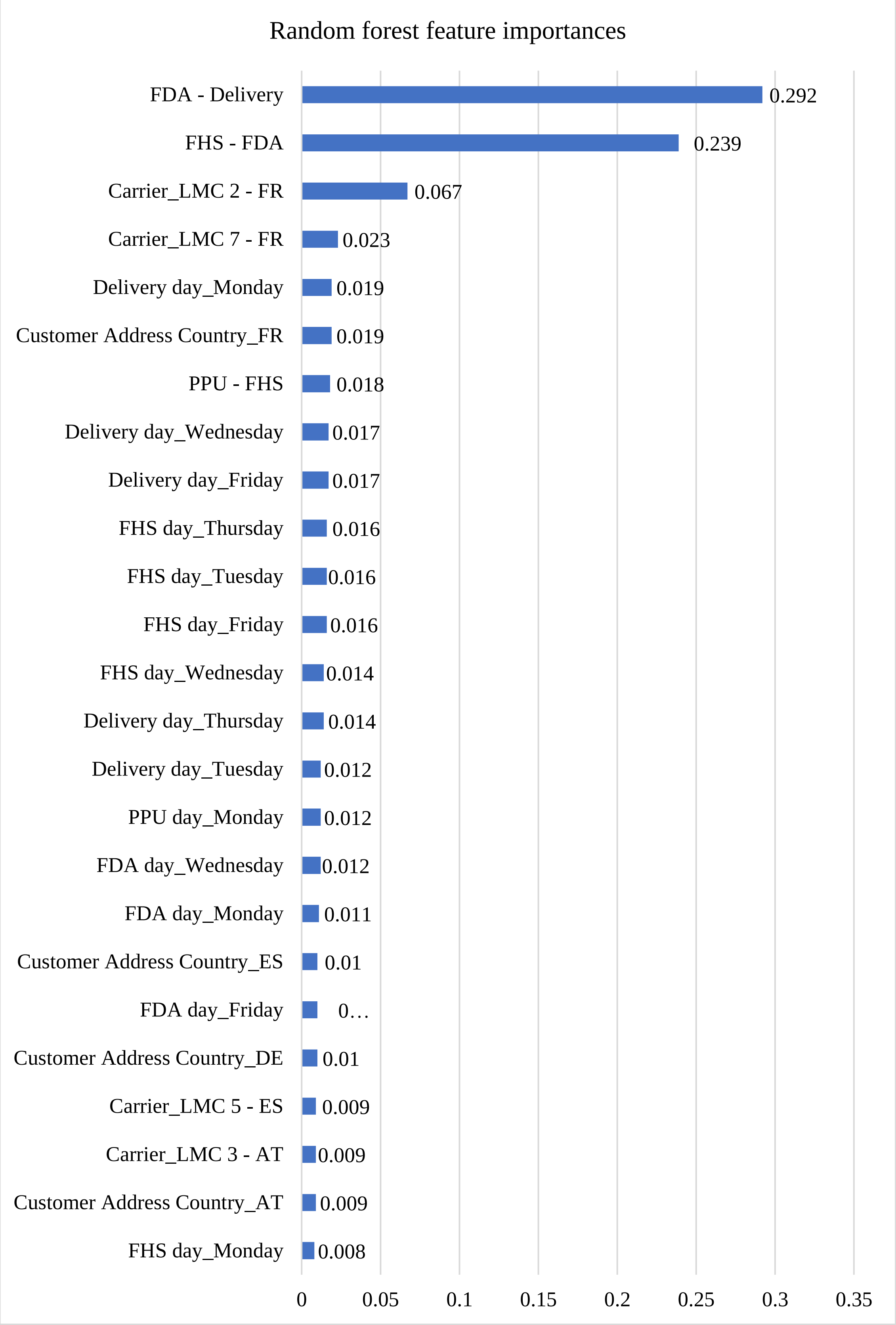
Appendix F Hyper-parameter Tuning Results

<table>
<tr><th colspan="2">Random Forest</th></tr>
<tr><td>Elapsed time</td><td>Fitting 5 folds for each of 54 candidates, totalling 270 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done 42 tasks | elapsed: 2.2min
[Parallel(n_jobs=-1)]: Done 192 tasks | elapsed: 10.3min
[Parallel(n_jobs=-1)]: Done 270 out of 270 | elapsed: 18.5min finished</td></tr>
<tr><td>Best parameters</td><td>{'rf__max_features': 10, 'rf__min_samples_leaf': 3, 'rf__min_samples_split': 2, 'rf__n_estimators': 300}</td></tr>
<tr><td>Best estimator</td><td>RandomForestClassifier(bootstrap=True, class_weight=None,
criterion='gini, max_depth=None, max_features=10,
max_leaf_nodes=None, min_impurity_decrease=0.0,
min_impurity_split=None, min_samples_leaf=3,
min_samples_split=2, min_weight_fraction_leaf=0.0,
n_estimators=300, n_jobs=None, oob_score=False,
random_state=1, verbose=0,
warm_start=False)</td></tr>
<tr><td>Best score</td><td>0.9633176075483768</td></tr>
<tr><td colspan="2"><b><i>k</i>-NN</b></td></tr>
<tr><td>Elapsed time</td><td>Fitting 5 folds for each of 8 candidates, totalling 40 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done 40 out of 40 | elapsed: 18.2min finished</td></tr>
<tr><td>Best parameters</td><td>{'knn__n_neighbors': 11, 'knn__weights': 'distance'}</td></tr>
<tr><td>Best estimator</td><td>KNeighborsClassifier(algorithm='auto', leaf_size=30,
metric='minkowski', metric_params=None,
n_jobs=None, n_neighbors=11, p=2,
weights='distance')</td></tr>
<tr><td>Best score</td><td>0.9744922437230129</td></tr>
<tr><td colspan="2"><b>CART</b></td></tr>
<tr><td>Elapsed time</td><td>Fitting 5 folds for each of 27 candidates, totalling 135 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done 42 tasks | elapsed: 15.8s
[Parallel(n_jobs=-1)]: Done 135 out of 135 | elapsed: 55.6s finished</td></tr>
<tr><td>Best parameters</td><td>{'cart__max_depth': 20, 'cart__min_samples_leaf': 1, 'cart__min_samples_split': 10}</td></tr>
<tr><td>Best estimator</td><td>DecisionTreeClassifier(class_weight=None,
criterion='gini', max_depth=20, max_features=None,
max_leaf_nodes=None, min_impurity_decrease=0.0,
min_impurity_split=None, min_samples_leaf=1,
min_samples_split=10,
min_weight_fraction_leaf=0.0, presort=False,
random_state=1, splitter='best')</td></tr>
<tr><td>Best score</td><td>0.9613585478970095</td></tr>
</table>

Appendix G Feature Importance – CART



Appendix H Feature Importance – Random Forest


Random forest feature importances
FDA - Delivery
FHS - FDA
Carrier_LMC 2 - FR
Carrier_LMC 7 - FR
Delivery day_Monday
Customer Address Country_FR
PPU - FHS
Delivery day_Wednesday
Delivery day_Friday
FHS day_Thursday
FHS day_Tuesday
FHS day_Friday
FHS day_Wednesday
Delivery day_Thursday
Delivery day_Tuesday
PPU day_Monday
FDA day_Wednesday
FDA day_Monday
Customer Address Country_ES
FDA day_Friday
0…
Customer Address Country_DE
Carrier_LMC 5 - ES
Carrier_LMC 3 - AT
Customer Address Country_AT
FHS day_Monday