19th of July, 2019

Master thesis

Predicting arrival times of container vessels

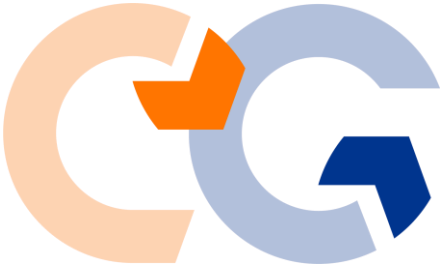
A machine learning application

PUBLIC VERSION

N. H. Bussmann (Nina)

MSc Industrial Engineering and Management

UNIVERSITY OF TWENTE.



EXAMINATION COMMITTEE

dr. ir. M. R. K. Mes, University of Twente

M. Koot, PhD Candidate, University of Twente

J.P.S. Piest, PDEng, University of Twente

M. Wesselink, Consultant, Cape groep

COLOPHON

In partial fulfilment of the requirements for the degree of

Master of Science in Industrial Engineering and Management

Document	Master Thesis
Title	Predicting arrival times of container vessels: a machine learning application
Keywords	Container Shipping, Freight Transport, Arrival Time Prediction, Machine Learning, Predictive Analytics
Author	N.H. Bussmann (Nina)
Educational Institution	University of Twente Faculty of Behavioural Management and Social Sciences Department of Industrial Engineering and Business Information Systems
Educational program	Industrial Engineering and Management Specialisation: Production and Logistics Management
Graduation committee	**University of Twente** dr. ir. M. R. K. Mes M. Koot, PhD Candidate J.P.S. Piest, PDEng **Cape Group** M. Wesselink
Date	Oldenzaal, 19th of July, 2019

This page is intentionally left blank

ABSTRACT

Freight transport is one of today's most important activities due to its influence on all economic sectors. A Dutch Logistic Service Provider (LSP) currently applies a reactive attitude towards arrival time information that is solely based on the carrier's sailing schedule. However, this sailing schedule historically appears to be unreliable: 20% of the orders that the LSP executed last 2.5 years, did not arrive on time. Note that this on time performance is based on a threshold of at least six days deviation from the scheduled arrival time before an order is classified as 'not on time'. When only zero deviation in the scheduled arrival time is allowed, the on time performance becomes even worse: 74% of the orders did not arrive on time, and had a deviation of at least one day. Since LSPs remain dependent on carriers from the container shipping industry, a platform capable of delivering and processing accurate information is essential for increasing efficiency, visibility and customer service. Not being able to exactly know when an order will arrive, negatively affects the businesses of both the LSP and the customer in terms of decreased efficiency and increased costs. We therefore propose a more proactive attitude towards arrival times by means of a predictive model based on historical order data. We applied the Random Forest technique to this end. The model is able to predict the deviation in the arrival time that is provided by the carrier in their sailing schedule in advance of actual shipment. After training and testing the Random Forest, the model is able of accurately predict the deviation in arrival time. Finally, deployment of the actual prediction algorithm is expected to lead to improved business processes in terms of increased efficiency and decreased costs for both the LSP and the customer. The LSP is expected to increase their efficiency by 84% for having less customer contact in case of a deviated shipment. This in turn positively affects the LSPs reputation as the customer's need for more proactively and accurate arrival time information is granted. From the customer's perspective, the customer is expected to save costs directly relating to a deviation in arrival time, because the prediction algorithm is better capable of predicting an accurate arrival time which leads to less deviation. It is expected that customers together can save an average of €771,025 euros on a yearly basis.

EXECUTIVE SUMMARY

This report describes the research of developing a prediction model for a Dutch Logistic Service Provider (LSP) that is responsible for the outbound global logistics of frozen potato products. The LSP solely bases the arrival time of an order, which is also communicated to the customer, on the sailing schedule of the executive carrier. This source of arrival time information however historically appears to be unreliable: 20% of the orders that the LSP executed last 2.5 years, did not arrive on time. Note that this on time performance is based on a threshold of at least six days deviation from the scheduled arrival time before an order is classified as 'not on time'. When only zero deviation in the scheduled arrival time is allowed, the on time performance becomes even worse: 74% of the orders did not arrive on time, and had a deviation of at least one day. Customers are aware of the LSPs bad performance with respect to arrival time information and from a customer survey the need became visible for proactive provision of more accurate arrival time information. Not being able to exactly know when an order will arrive, negatively affects the businesses of both the LSP and the customer in terms of decreased efficiency and increased costs. In case of a deviation in the Estimated Time of Arrival (ETA), the LSP is busy having increased customer contact to inform the customer with the deviation, that would have been unnecessary otherwise. In the worst case, the LSP fears potential loss of customers. The customer is particularly financially affected by a deviating ETA. Rescheduling costs are incurred when the order appears to arrive at another day than the customer had accounted for. Or when the customer is not able to pick up the goods on an ad-hoc basis, the customer risks being charged for demurrage fees. It is for that reason that customers indicate that they do not care that much about an order arriving too early or too late, but they do want to know exactly when the order will arrive. The LSP collects order data since October 2016, and we use this historical order data to develop a prediction model that is able to predict the deviation in the arrival time in advance of actual shipment. If the LSP then communicates this predicted arrival time to the customer instead of the arrival time that is solely based on carrier's sailing schedule, we aim to comply to customer's needs of proactively communicating a more accurate ETA.

For developing the prediction model, we use historical order data of the LSP. The target variable that we aim to predict is called the Delta and is the difference in actual and scheduled arrival time. This actual arrival time is based on the moment in time the container is unloaded from the container vessel in the destination port. First, we clean the data and their quality is addressed on the presence of ambiguity and missing values. We then transform the data by deriving attributes that are intuitively of predictive power for our target variable. We already do some hypotheses here about which variables are possibly good predictors of the target variable. This results in a list of 12 attributes readily available to predict the target. The next step is to apply feature selection. We use the wrapper approach with a bi-directional search method and find the following optimal subset of features: Departure Week (of the year), Departure Day (of the week), Arrival Week (of the year), Arrival Day (of the week), Carrier, the Port of Delivery, and the Transit time. After performing an additional experiment, Transit time appears not to be of good predictive power and we choose to exclude this variable. With the remaining 6 predictor variables, we build our prediction model that aims to predict the Delta, which is our target variable.

As a result of extensive literature research and some experimental tests, we decide to apply random forest as machine learning algorithm to train and test our model. Random forests have some advantages over other machine learning techniques as they can handle correlated predictor variables, which is the case with some of the variables in our model. Besides, random forests are robust to overfitting. After training the model, we can indeed conclude that the model has a good fit. The results also indicate that the model is able to accurately predict the target variable, which is the Delta. The validation of the model on the test set confirms our findings from the training the model. The capability of accurately predicting the target variable is an indication for a good model fit. There are some outliers present from which the model is not able to predict the output. Unfortunately, an additional analysis on this outlier set does not reveal a pattern to a predictor variable that can enables us to explain the outliers.

In the deployment phase, we actually implemented the prediction model into the already existing LMS. The predicted ETA is now displayed at the general order overview, from which customers are being informed about the arrival time of their order. We also displayed the predicted ETA at the page where the transport planner chooses the most appropriate sailing for a new order.

Now we have a prediction model that is capable of predicting the deviation of the communicated arrival time, we make the translation to improved business processes for both the LSP and the customer. We choose to address a cost savings' model from the perspective of the customer, as they are financially most affected by the events directly resulting from a deviation in the arrival time. In the cost savings' model, three cost parameters are included: demurrage fees, rescheduling costs and costs for running out of stock. Because all costs remained unknown when executing this research, we are forced to estimate them and we decide to include two extreme values for each cost parameter to this end. This results in an experimental design with $2^3 = 8$ combinations of parameter settings. In the cost savings' model, we compare the costs incurred in the current situation with the costs incurred in the new situation. In the current situation, we base the costs on the target variable Delta. Because currently, this is the deviation an order has when the communicated arrival time is solely based on the carrier's sailing schedule. In the new situation, we assume that the predicted arrival time is communicated to the customer in advance of actual shipment (rather than the arrival time from the carrier). Then, the deviation over which we must calculate the related costs, is the actual deviation Delta minus the predicted deviation. This is also referred to as the residual. The savings are then the difference between the costs in the current situation minus the costs in the new situation. The cost savings' model reveals it is expected that all customers together can save an average of €771,025 euros on a yearly basis when the LSP communicates the predicted ETA to the customer instead of the arrival time solely based on the carrier's sailing schedule. However, the LSP has more to gain than just a satisfied customer who can save costs by getting a more accurate ETA. We therefore also address the improved business processes from the LSPs perspective. We quantify their increased efficiency by counting the times that the LSP is required to have customer contact in the current situation and in the new situation (in which the ETA is based on our prediction model). Customer contact is required from a deviation of 4 days or more and is meant to inform the customer with the delay. When we then compare the current situation with the new situation, in 84% of the orders there is no customer contact required anymore since the ETA did not deviate that much. This would positively affect the LSPs reputation as the customer's need for more proactively and accurate arrival time information is granted. The LSPs concern of potential loss of customers would be eliminated to this end.

PREFACE

"It's not where are you from, it's where are you going" – Ella Fitzgerald

Dear reader,

With proud I present to you my master thesis, which is a result of eight months of learning new things, every day again. Even though my time as a student comes to an end here, I hope that the learning part never stops (or not yet).

I will never forget how I started this journey in the premaster's group, totally unsure if I was educated enough to follow a technical master's after I finished my bachelor in communication sciences. Well, here is the prove that I apparently was.

I am very grateful for the opportunity that CAPE Groep offered me. In particular, I want to thank Maik Wesselink, who provided me with all the information I needed to execute my research. I also want to thank Sebastian Piest, who has multiple years of working experience at CAPE Groep but recently started his PDEng at the University of Twente, and therefore had the 'best of both worlds' perspective.

My sincere acknowledgements go to my two supervisors of the University of Twente, Martijn and Martijn (after a meeting, in my notes denoted by Martijn[2]). My first supervisor, Martijn Mes, really helped me with the bigger picture of placing my research in the context of an academic business case and prevented me from losing myself in the abstraction of 'not even knowing what I am actually doing'. My second supervisor, Martijn Koot, has set aside a lot of time for me, even though he was busy enough meeting his own deadlines as a PhD candidate. For that I am really grateful as with the most basic questions, he gave me the most valuable answers.

Moreover, I want to thank the 'UT squad' as without you my time at the University of Twente wouldn't have been the same. In particular, I want to thank the 'maiden', who became good friends after we've spent 5 months in Taiwan. Marleen and Suzan, thank you for making the study abroad semester an unforgettable one.

Furthermore, I would like to thank Max, for the endless phone calls in which we discussed all the ins and outs of my research. Thank you for your critical view and listening ear.

Last, but certainly not least, I would like to thank my family for their unconditional support, I really hope I made you proud.

All that remains for me is to wish you an enjoyable read.

Nina Bussmann

Oldenzaal, July 2019

This page is intentionally left blank

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ArrDay	Arrival Day
ArrMonth	Arrival Month
ArrWeek	Arrival Week
ATA	Actual Time of Arrival
ATD	Actual time of Departure
AWS	Amazon Web Service
BCO	Booking Confirmation
CSC	Customer Service Center
DepDay	Departure Day
DepMonth	Departure Month
DepWeek	Departure Week
ETA	Estimated Time of Arrival
ETD	Estimated Time of Departure
ETL	Extract, Transform, Load
IT	Information Technology
kNN	k-Nearest Neighbours
LMS	Logistics Management System
LSO	Linger Shipping Operator
LSP	Logistic Service Provider
MSE	Mean Squared Error
NN	Neural Network
PoD	Port of Delivery
PoL	Port of Loading
RF	Random Forest
RMSE	Root Mean Squared Error
STA	Scheduled Time of Arrival
STD	Scheduled Time of Departure
SVM	Support Vector Machine
VGM	Verified Gross Mass
XML	Extensible Mark-up Language

Arrival time definitions used in this report

ATA	Actual Time of Arrival, determined from the moment the container is unloaded from the container ship.
STA	Scheduled Time of Arrival, determined by the carrier and published in their sailing schedule
ETA	Arrival time that is communicated to the customer, where ETA = STA (current situation)
ETA_{pred}	Predicted arrival time which is the result of our prediction model (new situation)

NOTES TO THE READER

- The following words are used interchangeably but cover the same semantic space:
  - Attribute, variable, feature (these are the columns in the dataset)
  - Order, shipment, record, data point (these are the rows in the dataset)
  - Response variable, target variable, dependent variable, output variable
  - Explanatory variables, predictor variables, independent variables, input variables

- In the current situation, the LSP communicates an ETA to the customer in advance of actual shipment which is the STA (the scheduled arrival time that is published in the carrier's sailing schedule).

  Current situation $ETA = STA$

- In the new situation, we aim to predict the ETA better, so with the use of the proposed prediction model we get a new arrival time in the form of the STA plus a possible deviation ε.

  New situation $ETA_{pred} = STA + \varepsilon$

1 INTRODUCTION

In this chapter, we treat the background of this research. We give a brief context of the problem space in Section 1.1, followed by a description of the companies that are involved in this research in Section 1.2. Section 1.3 continues with the problem identification and research objectives. Following from this section, the research question and sub questions are formulated in Section 1.4. In the two following sections, we describe the practical and scientific relevance of executing this research. Section 1.7 describes the methodology that we applied in this report. We end this first chapter with a planning of the research.

1.1 CONTEXT

Transportation is an important domain of human activity as it supports other social and economic exchanges. Especially freight transport is one of today's most important activities due to its influence on all economic sectors. However, transportation is also a complex domain in which adaptation to the rapidly changing political, social and economic trends is essential. This especially counts for sea transportation: intercontinental trade and the in- end export of food and manufactured goods would not be possible without it. Not to be surprised that at present, the international shipping industry is responsible for the carriage of around 90% of global trade volume (ICS, 2018). But the overall container shipping industry is a dynamic and complex one (Salleh, Riahi, Yang, & Wang, 2017) where on average only an on-time performance of 73% is achieved (Drewry Shipping Consultants, 2015 in Salleh et al., 2017). Following Drewry Shipping Consultants (2012), a vessel is considered as being 'on-time' if the divergence between actual and scheduled arrival time is within one day or less. Since logistic service providers remain dependent on carriers from the container shipping industry, a platform capable of delivering and processing accurate information is essential for increasing efficiency, visibility and customer service (Dobrkovic et al., 2016).

1.2 COMPANIES INVOLVED

CAPE Groep is a company that works with model driven platforms to realize the integration of Information Technology (IT) solutions. One of the partners of CAPE Groep is a specific Logistic Service Provider that we denote by LSP in the remainder of this report. The LSP is responsible for the transport of frozen food products of company X, both by road (European distribution) and by sea (global forwarding). Company X has customers in more than 100 countries and is one of the world's biggest producers of potato products. With a prediction of 7500 containers but an actual export amount of 9000 containers in 2018, company X is experiencing a rapid growth.

1.3 PROBLEM IDENTIFICATION

The LSP documents and saves order data since October 2016. The historical order data of the LSP from the last two and a half years reveal that **20.3% of all shipments did not arrive on time** according to the Estimated Time of Arrival (ETA) that the LSP communicated to the customer. This on-time performance is based on a threshold of at least six days deviation from the ETA before an order is classified as 'not on time'. Since this deviation can either be six days too early or six days too late, the resulting bandwidth for calculating the on-time performance is twelve days. When only zero deviation is

allowed, thus a bandwidth of one day, this percentage of shipments that are not on time increases to 74%.

For identifying the core problem, we have spoken to and had meetings with three members that are highly involved in the domain, which are the manager of the global forwarding department of the LSP, a transport planner at the LSP and the IT consultant from CAPE that is responsible for the project at the LSP. The IT consultant at CAPE gave us all the right documents to do our research with, for example all historical order data. We also spent a day with the transport planner to experience how he executes all daily activities including booking an order and communicating order information to customers. The manager of the global forwarding department of the LSP mainly provided us with strategic insights, like their mission and vision and to what extent they are busy trying to achieve those. As a result of these meetings and interviews, we were able to identify the core problem together with its causes and effects, which we visually represent in Figure 1-1.

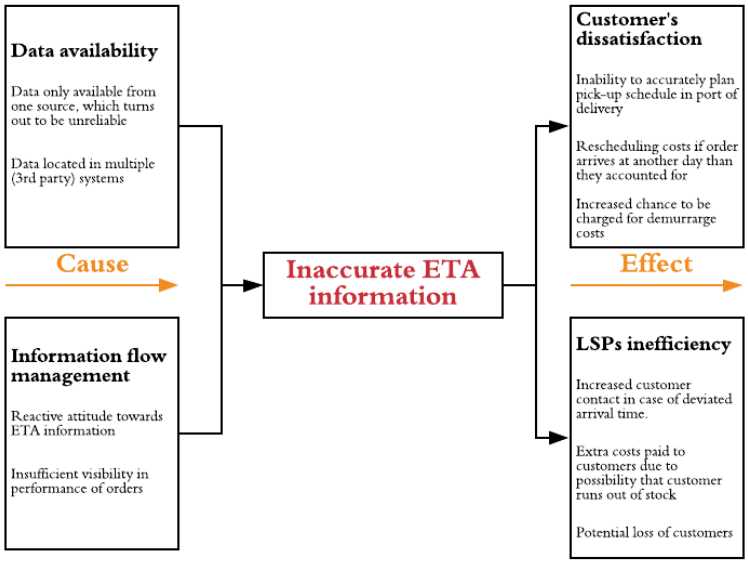


Figure 1-1: Core problem with its causes and effects

Currently, the LSP bases the ETA of orders on the sailing schedules published by the executive carrier. During the order's trip, the LSP keeps track of updates from that same executive carrier. Up till now, the published schedules and tracking updates from the carrier remain their only source of arrival time information.

If these schedules and updates of ocean freight carriers were reliable, there would be no problem only counting on this source of arrival time estimation. However, historical order data show that more than 20% of the orders cannot meet an arrival time falling within a deviation of twelve days from the ETA. To put this in perspective: the maximum permitted deviation of twelve days for an order to be 'on-time' is a relatively large permitted deviation taking into account an overall average transit time of 33 days. Despite this, the assumption is still that it is the responsibility of the carriers to provide accurate and timely arrival time information. It is also not likely that carriers will be able to improve their arrival time estimations in the new future, simply because it is not in the carriers'

interest to provide accurate and timely information about their own poor on-time performance.

Not being able to exactly know when an order will arrive, negatively affects the information flow, and customers of the LSP are aware of this. A survey among 203 customers conducted by an external agency, revealed that the LSP scores poorly on *proactive* and *accurate* provision of information towards their customers regarding arrival times (Cape Groep, 2018).

Proactive and accurate arrival time information are very important for customers of the LSP, as deviated deliveries affect their business directly: stock shortages can cause loss of clients when demands cannot be met. From the incoterms, that are further elaborated in Section 2.3.1, it also becomes clear that it is in the customers' interest of the LSP proactively providing an accurate ETA as the customer becomes responsible for all the operations and associated costs from the moment the order arrives at the port of delivery. Rescheduling costs are incurred when the order appears to arrive at another day than the customer had accounted for. It becomes impossible for the customer to accurately plan the pick-up schedule in the port of delivery, when the ETA where the schedule is based on, is not that accurate. It is for that reason that customers indicate that they do not care that much about an order arriving too early or too late, but they do want to know when the order will arrive.

The aforementioned problems also affect the LSP. When their customers run out of stock in case of a late delivery, additional costs for the urgent seek for products from other parties are for the LSP. Or in the worst case, they can lose their customers by delivering too late too often. Besides, inaccuracy in provided ETAs results in increased customer contact between the LSP and the customer what would have been unnecessary otherwise.

Now we have identified the core problem, we will formulate this as an action problem such that the core problem is correctly interpreted by all stakeholders. Following Heerkens and Winden (2017), an action problem is defined as the discrepancy between the *norm* and *reality*, as perceived by the *problem owner.*

The problem that both the LSP and their customers (*problem owners*) face, is the interdependency of only one source for arrival time estimation, and that source turns out to be unreliable. This leads to inaccurate arrival time information (*reality*). However, a better source for arrival time information would help the LSP providing more accurate arrival time information to the customer (*norm)*, and this directly affects their businesses in terms of decreased efficiency and increased costs across the supply chain. We formulate the action problem as follows:

> **Action problem:** *The interdependency of only one source for arrival time information that turns out to be unreliable obstructs the LSP to communicate an accurate arrival time, which decreases efficiency and increases costs across the supply chain for both the LSP and the customer.*

1.4 OBJECTIVES

The problem identification revealed the need of more accurate information regarding arrival times. One way to achieve this is to use data beyond what the carrier is reporting. In this research, we are going to use historical order data to develop a model that aims to predict the deviation in the STA provided by the carrier in advance of actual shipment. We then strive to predict an ETA that deviates less from the Actual Time of Arrival (denoted by ATA) compared to the STA provided by the carrier. In the objective, we can say that we want our prediction model to minimizes the difference between actual and predicted arrival time.

Theoretical objective: $\min |ATA - ETA_{predicted}|$

The theoretical objective of this project is thus to translate historical order data into predictive insights to let it function as additional source of arrival time estimation at time of booking an order, i.e., in advance of the actual shipment.

Being able to make predictions in advance of the shipment about to what extent the STA of the carrier will deviate, can be translated into improved business situations in terms of increased efficiency and decreased costs for both the LSP and its customers. Because the costs are mainly on the customer's side (referring to the incoterms), we chose to address a cost saving's model from the perspective of the customer. The second objective is therefore business related and is as follows:

Business objective: *When minimizing the deviation between actual and predicted arrival time, the average yearly costs of the customer, directly associated with that deviation, decrease.*

As a consequence of an improved customer's situation, we will address the improved business processes for the LSP. Several operational activities and costs are directly related to the customer's dissatisfaction about arrival time information: increased customer contact in case of a deviation in delivery and potential loss of customers are relevant here. We conclude that with an improved situation for the customer, the efficiency at the LSP increases.

1.5 RESEARCH QUESTIONS

Now we discussed the problem identification, we outline the research questions formed in order to improve the current situation. The main research question is:

> **Research question:** *To what extent can the operational efficiency at the global forwarding department of an LSP be increased by predicting the accuracy of the ETA provided by the carrier at the moment of booking?*

In order to extensively answer this question, the following sub questions are determined:

I. *What is the current situation at the global forwarding department of the LSP?*
   a. *How does the Logistics Management System (LMS) work?*
   b. *What is the process flow of booking and tracking a container?*
   c. *What is the process flow of communicating the ETA to the customer?*
   d. *What does a general global export analysis of the current situation show us?*

II. *What does literature tell us about arrival time prediction?*
   a. *Which techniques are frequently used in arrival time prediction models?*
   b. *Which technique is most suitable for developing a model that is able to more accurately predict the ETA?*
   c. *Which evaluation metrics are available to determine the performance of a prediction model?*

III. *What data is available at the LSP that are relevant for predicting the accuracy of an ETA?*
   a. *Which attributes does the historical order data contain and which attributes can we derive from it?*
   b. *Which attributes can possibly predict an anomalous ETA?*
   c. *Which attributes are included in the final model?*

IV. *What are the characteristics of the prediction model?*
   a. *Which modelling technique are we going to apply?*
   b. *What are the relevant and optimal parameter values?*
   c. *What is the predictive performance of the proposed model?*

V. *How can we use a better prediction to improve business processes?*
   a. *What costs can be saved when arrival time is based on the predicted ETA rather than the arrival time provided by the carrier?*
   b. *How can efficiency be improved arrival time is based on the predicted ETA rather than the arrival time provided by the carrier?*

VI. *How can the proposed model be implemented in the LMS?*
   a. *What is the expected impact of implementing the model in predicting the ETA?*
   b. *What does the prototype look like?*

1.6 METHODOLOGY

As the focus of this thesis will be on data mining methods, we chose to apply the Cross-Industry Standard Process for Data Mining (CRISP-DM) method (Chapman et al., 2000), which is referred to as most frequently used in practice (Larose, 2005; Piatetsky, 2014; Rogalewicz & Sika, 2016). The methodology is developed in 1996 by a consortium formed by the companies Daimler Chrysler AG, SPSS Inc., and NCR Systems Engineering (Chapman et al., 2000). Because of its high frequency in practice, the CRISP-DM methodology is followed in this research. Besides, since the nature of this research is data mining oriented, the methodology will be easily applicable. It provides a structured approach to planning a data mining project, consisting of six phases schematically shown in figure 1-2.

Figure 1-2: The CRISP cycle

The sequence of the phases is not rigid: the outcome of each phase determines the input of the next phase but moving back and forth between different phases is always required. Each phase in the CRISP cycle roughly represents one chapter in this report. Only chapter 3 is not part of the CRISP cycle, but is a theoretical chapter containing the literature review. In Table 1-1 we will outline which research question is answered in which phase of the CRISP cycle.

Table 1-1: Research questions answered in which chapter, based on the phase in the CRISP cycle

Phase in CRISP cycle	*Research question*	*Treated in*
Business understanding	*What is the current situation at the global forwarding department of the LSP?*	Chapter 2
Literature review	*What does literature tell us about arrival time prediction?*	Chapter 3
Data understanding	*What data is available at the LSP that are relevant for predicting the accuracy of an ETA?*	Chapter 4
Data preparation	*Which attributes can we derive from the original dataset and which ones are included in the final prediction model?*	Chapter 5
Modelling	*What does the prediction model look like?*	Chapter 6
Evaluation	*How can we translate a better prediction to improved business processes?*	Chapter 7
Deployment	*How can the proposed model be implemented in the LMS?*	Chapter 8

1.7 SCIENTIFIC RELEVANCE

This master thesis is part of the 'Autonomous Logistics Miners for Small-Medium Businesses' project of the University of Twente. The goal of this project is to increase the competitive power of the Dutch logistics sector, by providing small-medium sized businesses with intelligent data mining agents that can perform the most common data mining functions and require minimal supervision and domain knowledge from the human employee (University of Twente, 2018). With this research, we aim to attain knowledge about data mining tools and machine learning techniques that are most suitable for predicting the accuracy of container vessels' arrival times with historical order data as input. Furthermore, we will conduct a literature research about travel time prediction models, so we can investigate the most suitable ones to find out which technique results in a predictive model with the highest possible accuracy.

1.8 PRACTICAL RELEVANCE

CAPE Groep participates in the 'Autonomous Logistics Miners for Small-Medium Businesses' project of the university because of its potential. With this research, we aim to form the bridge between human operators and smart use of data mining. We focus on increasing the competitive power of the LSP in question by implementing an easy-to-use and understandable data mining tool that processes available data in such a way that it becomes useful information. With this information, the transport planner at the global forwarding department of the LSP is then able to provide a more accurate ETA prediction with minimal supervision.

1.9 CONTRIBUTION OF THIS THESIS

The contribution of this master thesis will be fourfold: (i) to collect historical order data and to discover how knowledge can be obtained from available data; (ii) to identify techniques that are useful for making predictions based on historical order data; (iii) to test and compare the accuracy of the developed model; and (iv) to propose a prototype that can be implemented in the already existing Logistic Management System (LMS) that provide transport planners the information that they cannot acquire by themselves essential for process optimization.

1.10 OUTLINE

The remainder of this thesis is structured as follows. Each phase of the CRISP-DM cycle represents a chapter in this report. There is one exception, which is the literature review covered in Chapter 3 that does not belong the a phase of the CRISP-DM cycle. In the second chapter, we explore the current situation at the global forwarding department of the LSP. This chapter is referred to the *business understanding* phase. The *data understanding* phase is covered in Chapter 4 in which we analyse all available data. Chapter 5 is all about the preparation of the dataset to be used for our prediction model, which represents the *data preparation* phase. Here, the main focus is on finding the best subset of variables to base our prediction model on. We perform feature selection to this end, in which we execute an analysis of variance, a correlation test and a multicollinearity check. In the 6th chapter, we go through the *modelling* phase in which we care about building the actual prediction model based on our selected features. We also evaluate its performance. In the *evaluation* phase, which forms Chapter 7, we perform a cost savings' model to translate a better prediction into improved business processes. In Chapter 8 we will actually implement the proposed algorithm in the Logistic Management System of the LSP and this all represents the *deployment* phase. In the last chapter, we discuss the conclusions, recommendations and limitations of our research.

2 BUSINESS UNDERSTANDING

In this chapter, we will answer the first sub question: *What is the current situation at the global forwarding department of the LSP?* We answer this question by first listing the involved stakeholders and their interrelationships in Section 2.1. Section 2.2 is about the operational proceedings, containing (i) a description of the Logistics Management System, (ii) the process of booking a sailing, (iii) the process of obtaining track & trace information and (iv) the process of ETA communication. We end this chapter by a global export analysis of the LSP in Section 2.3. Here we outline the trade volumes, the allocation of incoterms and the distribution of carriers and ports.

2.1 INVOLVED STAKEHOLDERS

Company X is a business-to-business company and produces a wide range of different types of potato products, changing in shape, size and preparation method. The biggest product category is fries, available in a lot of different sizes. The products are stored and shipped deeply frozen. Company X has set up a separate company especially for the transport of this frozen food products; that company is denoted by LSP in this report. Note that at the LSP, only outbound logistics occur. The LSP handles the road transport for European distribution and ocean sea transport for the global forwarding. This research only focuses on the global forwarding part of the export. For the ocean sea transport, the LSP is in direct contact with the carrier who executes the shipment. The overview in Figure 2-1 shows the communication flows between the different stakeholders together with their role. The arrows indicate the direction of communication. In the following sections, a detailed description is given about which information flow between which stakeholders. A visual representation of this information exchange is also depicted in Appendix A Appendix B .

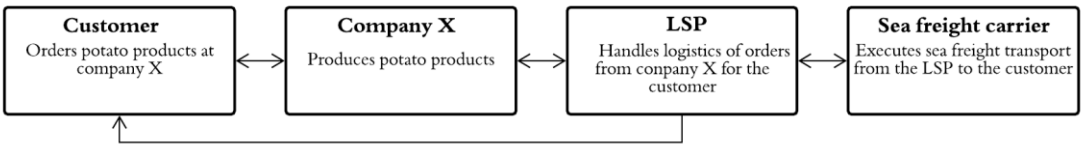


Figure 2-1: Communication flows between different stakeholders

2.2 OPERATIONAL PROCEEDINGS

At the global forwarding department of the LSP, five people are responsible for the daily planning and controlling of container ships. For these activities, the LSP uses one platform that integrates, processes and delivers relevant data for planning, executing and optimizing transportation fleets. This platform is designed and developed by CAPE Groep, called the Logistics Management System (LMS).

2.2.1 The LMS

The global forwarding department of the LSP uses this platform daily for planning and controlling all shipments by sea. Figure 2-2 shows the home page of the LMS.

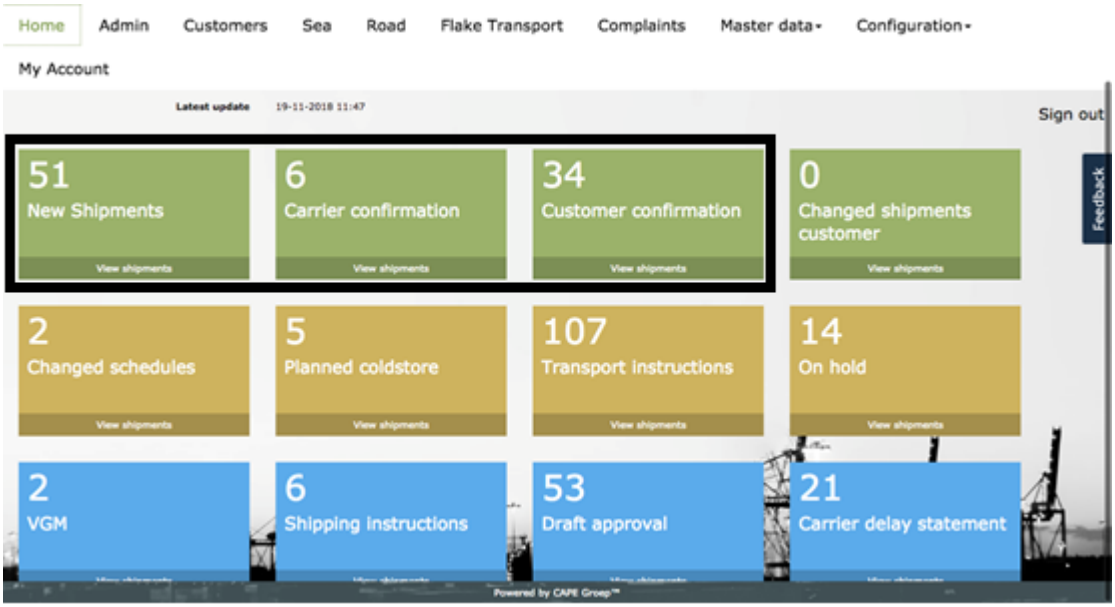


Figure 2-2: Overview page of the LMS

The two tabs 'road' and 'flake transport' (which is an overarching name for all the residual products of the production) belong to another department of the LSP and falls beyond the scope of this research. We will thus not further discuss these tabs. From the 'home' tab, the transport planner sees an overview of all actions that need to be performed regarding the container shipments by sea – each box representing another action. The number in the box represents the number of orders that needs attention in some way. The three boxes that are marked with a black rectangle are relevant in the process of booking a sailing for an incoming order. In section 2.2.2, we will further explain these actions in detail.

An important part of the order details in the shipment overview of the LMS are the different types of departure- and arrival times. In Figure 2-3, one sees an example of how these times are displayed at an order in the LMS.

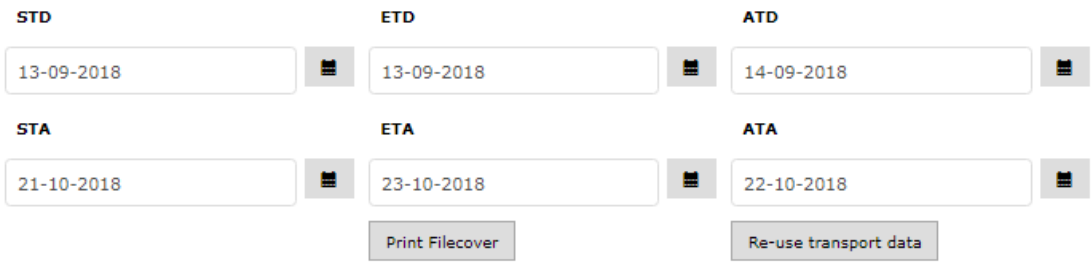


Figure 2-3: Example of Departure- and Arrival Times in the LMS

Table 2-1 shows a list of abbreviations used for the departure- and arrival times.

Table 2-1: Abbreviations for departure- and arrival times

Abbreviation	*Description*
STD	Scheduled Time of Departure
ETD	Estimated Time of Departure
ATD	Actual Time of Departure
STA	Scheduled Time of Arrival
ETA	Estimated Time of Arrival
ATA	Actual Time of Arrival

The STA is the scheduled arrival time, determined by the carrier based on their sailing schedule. After a booking request is placed at the carrier, the transport planner fills in the STA as set by the carrier. When the carrier confirms the booking, the ETA is filled in and gets the same date as the STA. From the moment the container is shipped, the ETA is subject to change due to delays or other disruptions during shipment. In Section 2.2.2, we will describe how and when the ETA of an order changes. The ATA is determined afterwards and is only used by the LSP for own documentation. The times of departure are less relevant in this case and become more important when focusing on the loading of the containers *before* shipment. Since this part is beyond the scope of this research, we will not further explain the detailed definition of the different types of departure times.

2.2.2 Process of booking a sailing

This section describes the detailed process of booking a sailing. The shipment planning process starts when an order comes in from company X. The order is now available in the first box 'new shipments' (see Figure 2 and 4). The incoming order is processed in threefold: the order is automatically sent to an invoice company; the new order will also be printed, from which a physical dossier is made. Last, a departure is scheduled based on the following factors:

- The arrival time requested by the customer.
- The contracted carrier to the destination.
- The port of loading (Rotterdam or Antwerp).
- The port of delivery.

When the factors mentioned above are determined, the transport planner will first look in the database. Here, all previously scheduled sailings are saved. If there is a sailing in the database with an arrival time corresponding with the requested arrival time at the destination, the planner chooses the sailing in the database. If not, the planner searches on the website of the carrier for the availability of a sailing with an STA that corresponds with the customer's requested arrival time. If an appropriate sailing is found, the new

sailing is added to the database and the booking is placed by sending it to INTTRA. INTTRA is an ocean trade platform where almost all carriers are connected to and that handles the communication between the LSP and the carrier. If the carrier is not connected to INTTRA, the LSP does a manual booking via email directly at the carrier. However, the order of procedure for both ways of the booking process is the same.

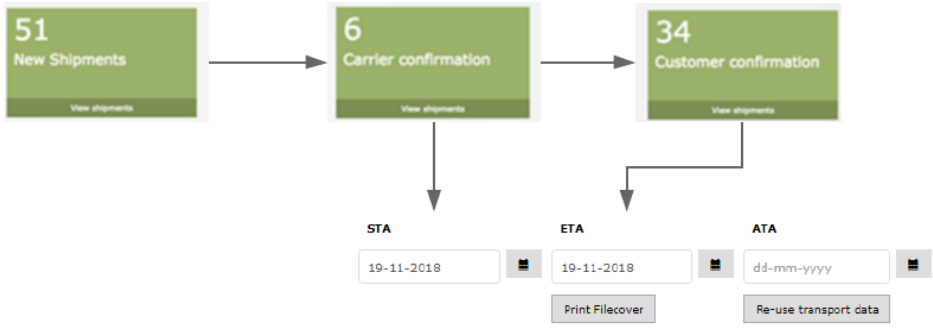


Figure 2-4: Phases of the booking process

In this phase of the booking process, the order is moved to the box 'carrier confirmation' (Figure 4) and the transport planner fills in the STA: it is the arrival time of the requested booking according to the sailing schedule of the carrier. Via INTTRA (or by email), the carrier will confirm the booking if the container can be shipped. If this is not the case, the carrier will refuse the booking and the planner must search for another appropriate sailing in their database or on the website of the carrier. If the carrier confirms the booking, an email is sent to the LSP and the confirmation of the booked sailing is added to the LMS. The order is now moved to the box 'customer confirmation'. Here, the dossier is updated with booking confirmation details and the conducted information is verified. Now, the ETA is filled in; it is again the arrival time of the requested booking according to the sailing schedule of the carrier, and thus the same as the previously filled in STA. It happens rarely that the carrier made a change in the sailing schedule in the meantime as sailing schedules are often determined three months in advance.

The last step consists of verifying the booking: if the information in the dossier does not correspond with the initial requested order, a corrective message is sent to the carrier via INTTRA (or by email). The carrier changes the booking and sends a new confirmation via INTTRA (or by email) back to the LSP. A schematic representation of the booking process can be found in Appendix A .

2.2.3 Obtaining track & trace information

During execution of the shipment, the LSP is dependent on the carrier for the determination and communication of the container vessel's arrival time: the LSP must keep track of the container's tracking status. From the moment the order is shipped, the STA can change due to a wide variety of events happening along the way. Although we have no insight in how the carrier determined the STA in their sailing schedule (to what extent did they already account for uncertainties), what we do know is that in practice this arrival time is subject to change due to uncertainties and disruptive events (Dobrkovic et al., 2015), meaning that it is realistic to assume that the ETA is in practice the initial STA plus or minus a possible deviation due to uncertainties during transport.

Currently, the LSP only acts reactive to these changes by receiving the track & trace updates at the moment the deviation has already take place.

The tracking status of active shipments that is available in the LMS is obtained in three different ways: via INTTRA, via a cloud scraper and manually. In the following three sections we will further explain how track & trace information is obtained in the three different ways respectively.

2.2.3.1 INTTRA

INTTRA is an ocean trade platform where almost all sea freight carriers are connected to. From its roots, INTTRA created a standard electronic booking system for the ocean freight industry but evolved in enabling electronic booking and digitalizing the exchange of shipping instructions and container tracking. After shipping, the sea freight carriers transmit the container status to the INTTRA platform. INTTRA will in her turn send the track & trace information to the LSP in the form of an XML message. Such a message contains a specific event code together with the location where the event happened. An event code refers to a specific event that has happened along the way at that location (e.g., VD is a code standing for Vessel Departure). All updates are collected in a specific tab of the order overview. Only if the message concerns a change in the estimated arrival time, the ETA is adapted on the frontpage of the order overview. See the figure below for an example of a change in the ETA. Note that from the moment the ETA changes along the way, the initial STA is not the same as the ETA anymore.

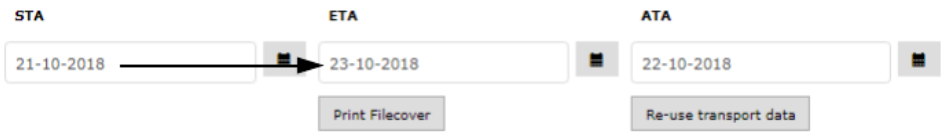


Figure 2-5: Change in the arrival time

2.2.3.2 The cloud scraper

Sea freight carriers typically have their own website on which they also publish tracking information of active container vessels. When a carrier is not connected to INTTRA, the track & trace information is not automatically sent to the LMS in case of a tracking update. The LSP must then search himself on the website of the concerning carrier for tracking information for each of the booking numbers. Since this is very time-consuming, the recently developed cloud scraper is implemented in the LMS. The cloud scraper uses web scraping techniques to automatically check carrier websites for the current ETA information of each booking number. The cloud scraper is currently only programmed to look at ETA updates: other tracking updates are ignored. The logic of the cloud scraper must be implemented in the LMS for each carrier website separately as every website has a unique data architecture. Since the cloud scraper is a relatively new technique, not all carrier websites are included yet.

2.2.3.3 Manually

In case of a carrier that is not connected to INTTRA and their website is not implemented in the cloud scraper technique yet, employees at the LSP must manually search for tracking information at carrier websites and subsequently add the obtained track & trace information to the LMS. As it occurs rarely that a carrier is neither connected to INTTRA,

nor implemented in the cloud scraper technology, manually adding track & trace information in the LMS only happens exceptionally.

2.2.4 Process of ETA communication

In this section, we describe the process of estimated arrival time communication between all involved stakeholders; the customer, company X, the LSP and the carrier. Look back at Figure 2-1 for a stakeholder overview. Since we formulated the problem from the LSPs perspective, we will focus on their role in the communication flow.

As soon as the carrier confirmed the booking and the LSP verified the correctness, company X gets a booking confirmation from the LSP with an associated arrival time that they in their turn communicate to the customer. From this moment, the LSP monitors tracking updates in the three previously mentioned ways. The tracking updates are collected in a special 'tracking history' tab accessible from each order. Table 2-2 gives a brief overview of which technique collects what kinds of tracking updates:

Table 2-2: Tracking techniques and what they collect

Tracking technique	*Collects*
INTTRA	All tracking updates
Cloud scraper	Only ETA updates
Manually	Only ETA updates

As can be seen in Figure 2-5, if the tracking update contains a deviation in the ETA, this field is adapted in the order overview. The ETA now takes a different value than the initial STA, and one can speak of a deviation in arrival time. The deviation is the difference between the ETA and the STA of order *s*, expressed in days:

$$Deviation_s = ETA_s - STA_s \tag{2.1}$$

Depending on the degree of the deviation, certain actions are taken at the global forwarding department.

If the *deviation < 4 days*, no action is taken. For this reason, the orders with a relatively small deviation in arrival time, will often remain unnoticed as the concerning orders are not highlighted in a certain overview.

If *the deviation has risen to ≥ 4 days*, an email is sent automatically to inform the customer with the delay. But again, for the transport planner the deviation will remain unnoticed as the concerning orders will not be highlighted in or transferred to a certain overview.

Only after the *deviation becomes ≥ 6 days*, the order will be visible in a separate tab in the LMS called 'delayed'. This is based on the fact that the LSP decided a deviation of 6 days to be the threshold for an order to be 'delayed'. Only from a deviation this large, the concerning orders are visible in an overview page. The transport planner uses this overview to navigate to the deviated orders to perform the actions associated with the delay. The tasks belonging to a 'delayed' order are asking the carrier for a statement request that is used by the Customer Service Center (CSC) of company X to subsequently call the customer with an explanation of the deviation in the order's arrival time.

2.2.5 Global process description

Now we have a detailed description of all the different operational proceedings that are relevant for the purpose of our research, it is meaningful to provide the reader with a

more high-level process description to get a better idea of how all activities relate to each other. In this process description, we combine the processes of booking a sailing, obtaining track and trace information and communicating the ETA to the customer, that all take place in the LMS (described in Section 2.2.1 until 2.2.4).

When an order comes in from company X, the LSP books a suitable sailing at the carrier that is contracted on the port where the order has to be shipped to. The carrier confirms the booking and gives back an STA to the LSP, that they derive from their own published sailing schedule. The LSP communicates an ETA to the customer in advance of actual shipment, for which holds that ETA= STA. After the order is shipped, the LSP keeps track of ETA updates on the carrier's website. If the ETA deviates 4 days or more from the initial STA, the LSP informs the customer with the anomalous ETA. Figure 2-6 shows a visual representation of this global process description.

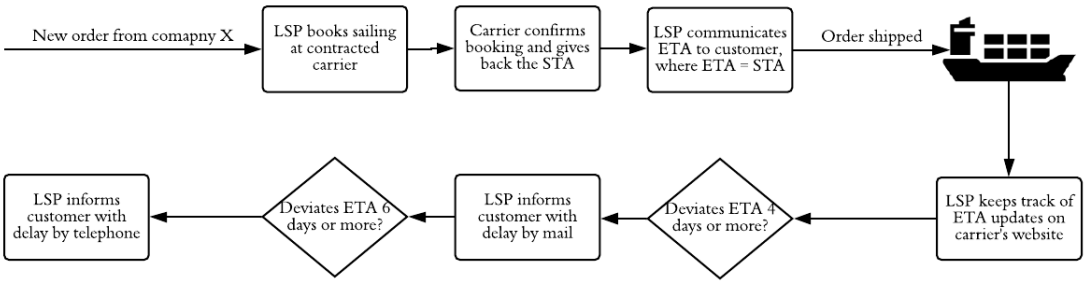


Figure 2-6: Global process description

2.3 GLOBAL EXPORT ANALYSIS

The historical order data where we aim to make our prediction model from, contain orders from October 2016 until January 2019. This initial dataset contains 4932 records. A detailed description of data exploration is given in Chapter 4. There is no data available of before October 2016 since the LSP did not collect order data before. Figure 2-6 shows the ports where the LSP ships to. The size of the dot indicates the trade volume, i.e., the amount of orders that is shipped to that port from October 2016 until January 2019.

Figure 2-7: Visualization of global export volumes

The LSP has divided their customers into six different trade regions: North America, South America, Africa, Middle East/India, Asia and Oceania. In Appendix E for each region the individual ports that fall within this region are listed. From that list the differences in trade volumes in Figure 2-7 can be explained, as for example the regions Asia and South America contain more ports than Africa. This is also reflected in the world map in Figure 2-7.

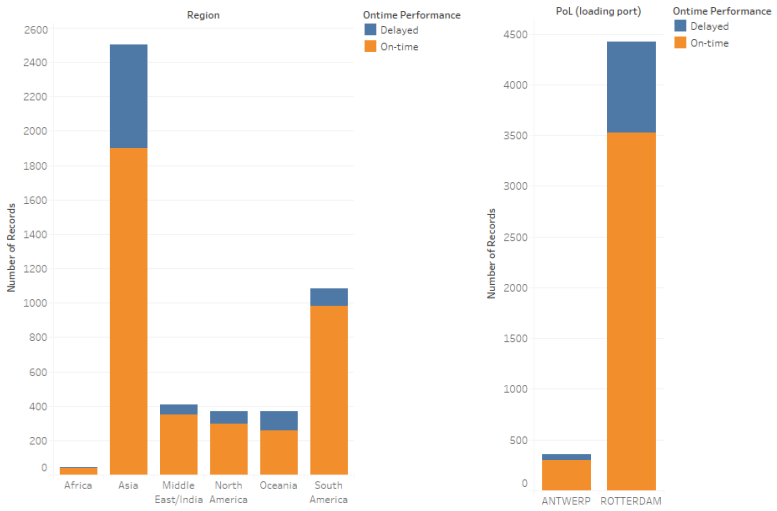


Figure 2-8: Trade volumes per Region and per Port of Loading, broken down by their on-time performance: notice that the total delayed orders are 20.3% of total trade volume

The LSP ships from two ports: Rotterdam and Antwerp. Most of the orders are shipped from the port of Rotterdam, as is shown in Figure 2-7. When comparing the trade volumes of 2017 with 2018, it is clearly visible that the LSP is experiencing a growth where the average monthly volume in 2018 compared to 2017 is increased with 222.92%.

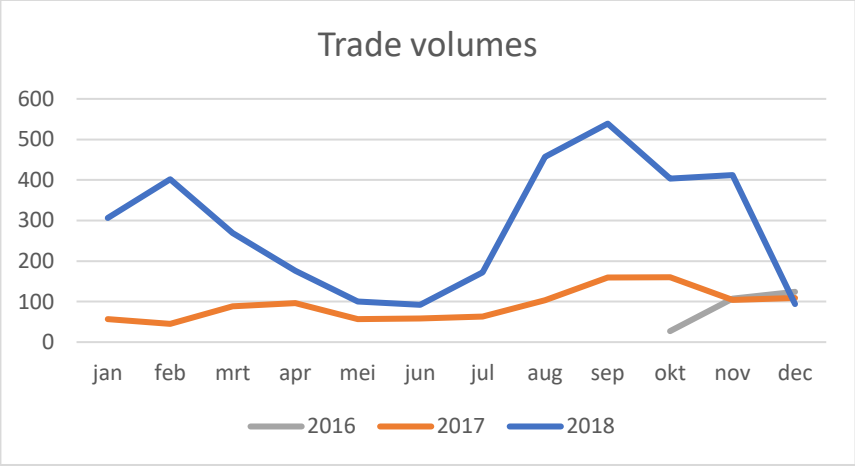


Figure 2-9: Trade volumes per year

Note that the records in the table are calculated from the data of the *arrival* times of the orders, as these orders are actually delivered already. For this reason, trade volumes for 2019 are not visible yet since these orders are *departed* in 2019 but not arrived at the destination yet.

2.3.1 Incoterms

An important part of the export of goods for both the LSP and the customer, are the incoterms. These trade terms are an internationally recognised standard and are a fundamental part of international contracts of sale, as they tell the parties who is responsible for what part of carrying the goods from seller to buyer, including import- and export clearance (ICC, 2011). Roughly all orders (98.2%) are shipped with the incoterm CIF (Cost, Insurance and Freight). The other 1.8% of orders have the incoterm CFR (Cost and Freight). Both incoterms do not differ much from each other, with only one

difference in insurance agreements. In Table 2-3, the allocation of costs to buyer/seller is displayed.

Table 2-3: The allocation of responsibilities for two types of incoterms: CIF and CFR

Incoterm	*CIF*	*CFR*
Loading at origin	Seller	Seller
Export customs declaration	Seller	Seller
Carriage to port of export	Seller	Seller
Unloading of truck in port of export	Seller	Seller
Loading on vessel in port of export	Seller	Seller
Carriage to port of import	Seller	Seller
Insurance	Seller	Buyer
Unloading in port of import	Buyer	Buyer
Loading on truck in port of import	Buyer	Buyer
Carriage to place of destination	Buyer	Buyer
Import customs clearance	Buyer	Buyer
Import duties and taxes	Buyer	Buyer
Unloading at destination	Buyer	Buyer

The incoterms are a good argument for the customer's concerns of inaccurate ETA information: the table above shows that from the moment the goods arrive at the destination port, the customer becomes responsible for unloading the goods from the container and all the activities following further in the supply chain. For these operations, a material and requirements planning is needed to efficiently allocate human and mechanical resources. However, when it is unsure when the order will arrive at the port of destination, it becomes difficult for the customer to keep such operations cost-efficient.

2.3.2 Carriers

The LSP works with carrier contracts per quarter per destination port. Company X makes a forecast of how many containers must be shipped to a specific port in a specific quarter and the LSP sets up a contract with the most beneficial carrier, with as consequence that within a quarter – exceptions excluded – only one carrier ships to a specific port. The data contains 26 different carriers that have executed one or more shipments in the period from October 2016 until January 2019. As figure 2-9 shows, there is much difference between carriers' trade volumes.

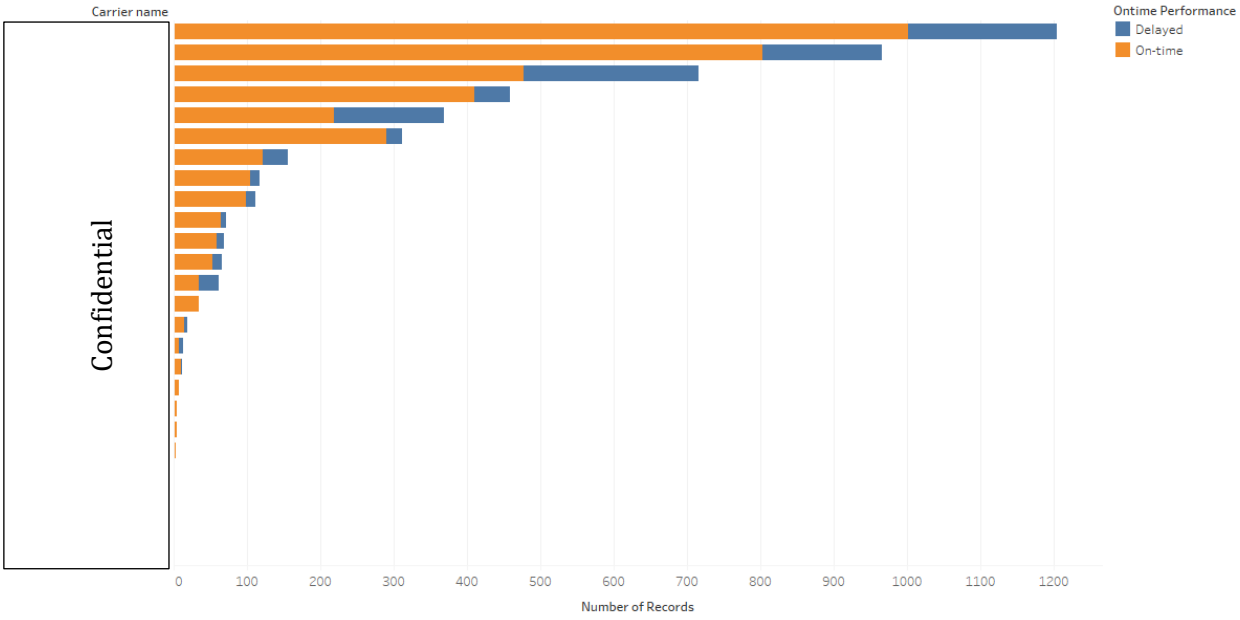


Figure 2-10: The trade volume per carrier, broken down to their on-time performance

2.3.3 Ports

As stated above, each port where the LSP must deliver to is represented by a carrier who executes the shipment to that port. Each quarter of the year, the carrier per port can change when new carrier contracts are negotiated. In the period from October 2016 until January 2019, the LSP shipped to a total number of 99 ports. See the figure in Appendix H for an overview of all the ports where the orders are shipped to. There is a large difference between the port's trade volumes: the three busiest ports together represent more than 25% of total trade volume versus ports where only one record is available (meaning that an order is shipped to that port only once).

2.4 CONCLUSION

In this chapter, we performed a high-level stakeholder analysis of parties that are involved in the research domain. We explored the current situation at the global forwarding department of the LSP, where we described the operational proceedings very extensively, containing of the description of the LMS, the process of booking a sailing, the process of the process of obtaining track and trace information and the process of communicating the ETA to the customer. To clarify how these separate activities together function as a whole, we ended with a global process description of the daily activities at the global forwarding department of the LSP. Last, we performed a global export analysis in which we described the allocation of responsibilities that are included in the incoterms, the carriers where the LSP ships with and the ports where the LSP ships to.

3 LITERATURE REVIEW

In the literature review, we are going to answer the second sub question: *What does literature tell us about arrival time prediction?* We first go through several arrival time prediction models in Section 3.1, which ends with a conclusion about our findings. These findings lead to a focus on one technique. In Section 3.2, a general description of the field of data mining that is found in literature is given, including our own interpretation of the definition that is used in this report. This includes our most important finding related to the distinction between statistics and machine learning. Hereafter, we continue with specific machine learning models that are relevant in this research in Section 3.3. Next, we study feature selection methods and performance measures frequently used in predictive models in Sections 3.4 and 3.5 respectively. We end this chapter with a conclusion in Section 3.6.

3.1 ARRIVAL TIME PREDICTION MODELS

In this section, we survey the most relevant prediction models and the extent to which they are applicable to the domain of this report. However limited research is found about predicting arrival times of ocean container vessels, but there are some articles about travel time prediction models focusing on other types of transport.

3.1.1 Historical data based models

These models base their prediction on travel time of historical journeys in the same period, whereas traffic conditions are assumed to remain stationary from period to period (Williams and Hoel, 2003). This being said, outputs are only reliable if indeed traffic patterns within a period (e.g., time of the day, day of the week, week of the year) are relatively stable (Shalaby & Farhan, 2004).

3.1.2 Time series models

Time series models make use of historical data as input for mathematical functions that calculate patterns that happen occasionally over time. This model also assumes that traffic patterns remain stationary. Its success highly depends on a function that calculates the degree of correspondence between historical and real-time data (Altinkaya & Zontul, 2013), as variation in one of the data sources can significantly decrease its predictive power (Smith & Demetsky, 1995).

3.1.3 Regression models

By using a set of independent variables as input, a certain dependent variable is intended to be explained and predicted. Unlike the historical data based model and time series model, regression models are capable of working under non-stationary traffic patterns. These models measure the effects of various factors simultaneously affecting the dependent variable, where the factors need to be independent of each other. Several studies (Jeong, 2004; Patnaik et al., 2004; Ramakrishna et al., 2006) using regression models to predict bus arrival times indicated that these models outperformed other models with as greatest advantage that it reveals which affecting factors more or less contributes to the models' predictive power. For example, a revealing finding of Patnaik (2004) was that, counter-intuitively, weather data is not an important input for the prediction model. The biggest drawback of regression models is that the applicability is

limited because variables in transportation systems are often highly correlated (Chien et al., 2002).

3.1.4 Machine learning models

As a branch of data mining, machine learning constructs models that can learn from patterns in data. The two stages machine learning roughly contains, are first choosing a candidate model and then predicting parameters through learning on existing data (Jin & Sendhof, 2008). Machine learning has some benefits compared to the other prediction models, as it is capable of dealing with complex relations between predictors by processing complicated data in such a way it becomes useful information (Recknagel, 2001). In literature, the confusion exists of machine learning and regression models being the same. Because the same methods can be used in both machine learning and regression models, this assumption has arisen (Kutner, Nachtsheim, Neter, & Li, 2004). However understandable, this is a misconception that we will therefore discuss in the next section.

3.1.5 Regression versus machine learning

The main difference between machine learning and regression models lies in the purpose: whereas machine learning models have the goal to make the most accurate predictions possible from the available data, regression models are designed for finding inference about relationships between variables in that data including their significance (Marsland, 2015). But there is a gray area in this definition of the difference between these two concepts, because regression models *can* make predictions (but not very accurate), and machine learning models *do* provide some insight in inferences (but not very interpretable) (Kutner et al., 2004). This similarity in model characteristics makes that people often assume that they are the same.

By this reasoning, the same model can be evaluated in two different ways, from the machine learning and from the regression perspective. We give a practical example to discuss this, which is the most basic linear regression. We can train a linear regressor to obtain the same outcome as the regression model with the aim to minimize the error between the data points. We are doing machine learning here, as evaluation involves *training* the model on a subset of the data, and until we *test* the model on the remaining data we do not know the predictive performance of the model. For the statistical linear regression, we evaluate the same linear regression model. But now with the aim to characterize the relationship between the data and some output variable, which can be achieved by finding a line that minimizes the error across all data points. However this model can still be used to make predictions from, it is more suitable for finding (the significance of) relationships between variables in the data.

To conclude, machine learning models produce predictions, as accurate as possible, and are all about the results. However regression models are also used to generate predictions, it is more designed to estimate certain properties of a larger population based on a smaller set of observations. In the next section, we will further discuss this split together with the role of data mining, but we can already refer the reader to Figure 3-1 in which the distinction between statistics and machine learning is made.

3.1.6 Conclusion

Most researches that we assessed about travel time prediction models, were about bus travel time. Furthermore, the researches about predicting arrivals at container terminals only concerns over a forecasting horizon of one day, thus both implying a short-term prediction model. This especially was the case in researches were historical data and times series prediction models were used. Thereby, we cannot guarantee that traffic conditions remain stable which makes these two models inappropriate. We have also discussed the difference between regression and machine learning. As we want to make predictions, as accurate as possible, machine learning models remain as most suitable in the context of our research. It is for that reason that we will focus on machine learning models later on. First, as introduction to the field of machine learning, we will describe the more general terminology 'data mining' as there is no clear line that distinguishes between the meaning of the two concepts.

3.2 DATA MINING

Data mining is used in multiple areas of interest and its appearance in literature is often accompanied by concepts as 'statistics', 'machine learning' and 'artificial intelligence' (Tan, Steinbach, & Kumar, 2005; Johansson, 2007; Kantardzic, 2011). However, these concepts are often incorrectly used interchangeably. This implies that there is no consensus on these definitions or even what constitutes these definitions. To avoid such misconceptions in this report, the following section contains an overview of what is known in literature followed by our own interpretation of the interrelationships of these concepts.

3.2.1 Origins

Data mining has its origin in diverse series of disciplines, but all types of data mining are about the search for new, valuable information in big data (Kantardzic, 2011). It is a symbiotic relationship between humans and computers. Data mining is a method used in handling big data in order to discover patterns that might otherwise remain unknown; it turns raw data into useful information (Tan e al., 2005). According to Kantardzic (2011), data mining has its roots in statistics and machine learning. Statistics has its origins in mathematics and has therefore the desire to test something on theoretical grounds before bringing it into practice. The machine learning community has its roots in computer science and has therefore, in contrast to mathematics, a practically oriented view in which something is tested out to see the performances without necessarily requiring a formal proof of effectiveness. Moreover, statistics have an emphasis on models, whereas machine learning emphasizes algorithms, which is not surprisingly as the word 'learning' implies a process and processes are implicit algorithms. This is similar to, but less extended, as the definition of Tan et al. (2005), who state that data mining can be seen as a confluence of many disciplines and draws upon ideas such as sampling, estimation and hypothesis testing from statistics on one hand and search algorithms, modelling techniques and learning theories from artificial intelligence and machine learning on the other hand.

3.2.2 Purpose

The purpose of data mining is either to *describe* certain patterns or to *predict* certain values (Tan et al., 2005; Johansson, 2007). The descriptive purpose is to gain an understanding of the analysed data by uncovering patterns and relationships; the predictive purpose is to create a model that can be used to perform classification, prediction, estimation or other similar tasks (Kantardzic, 2011). In the following section, we will further elaborate on descriptive and predictive mining. However, the two purposes could (and should) be used in conjunction, as in data mining projects it is often helpful to first search for patterns in the data (descriptive) to then use this information as input for the predictive model. When combining the definitions of Kantardzic (2011), Tan et al. (2005), and Johansson (2007) about the roots and purposes of data mining, we see a dichotomy of the descriptive purpose from the field of statistics on one hand, and the predictive purpose form the field of machine learning on the other hand. Figure 3-1 contains a visual representation of how we use the term 'data mining' during this report. It is a perspective based on the previously mentioned definitions of Kantardzic (2011), Tan et al. (2005), and Johansson (2007).

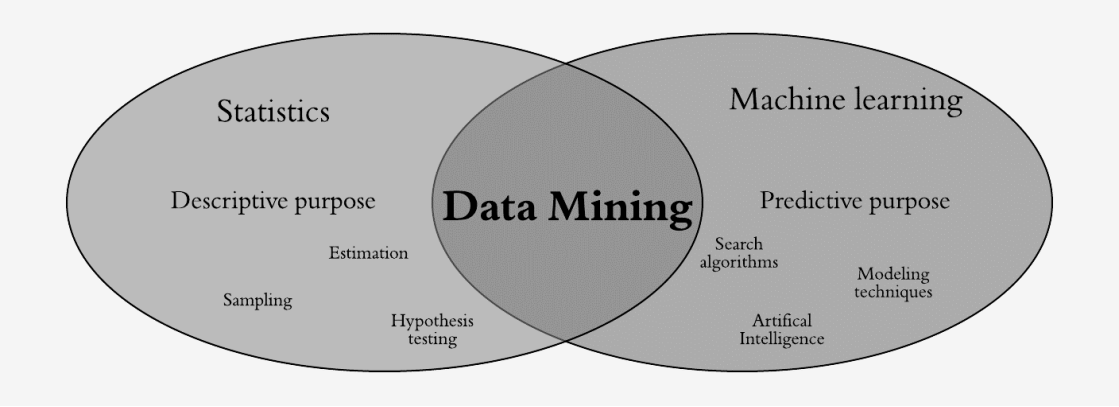


Figure 3-1: Data mining in perspective (figure developed by author)

During this data mining oriented research, we will access and discover the data through statistics. Later on, we will make a model with a predictive character based on machine learning.

3.2.3 Tasks of data mining

Just like there is no uniform origin or meaning of data mining, there are also different tasks that data mining can fulfil. Examples are exploratory, reductive and predictive mining, anomaly detection, association analysis, estimation, clustering, classification, and prediction (Freitas, 2002; Larose, 2005; Tan et al., 2005), all with the purpose to either *describe* patterns or to *predict* values. Description and prediction are elaborated to this end. We follow by discussing the main tasks of data mining: classification and regression, clustering and association.

3.2.3.1 Description

The first step in mining through data, is describing patterns to potentially reveal trends. Data mining models generally need to be as transparent as possible (Larose, 2005), meaning that the models created from the data are easily understandable, reproducible and are based on intuitive interpretation and explanation. Especially the former is important when addressing the human interaction that rises from the objective of the 'Autonomous Logistics Miners for Small-Medium Businesses' project of the University of Twente, that aims for an easy-to-use data mining tool that every employee can use autonomously (i.e. with minimal supervision). However, not all data mining methods are evenly suitable for an intuitive and human-friendly interpretation of results (see Section 3.1.3). The less understandable the method is, the more important it becomes to have a high-quality description (Larose, 2005). Chapter 5 contains such description of data that is accomplished by an exploratory data analysis.

3.2.3.2 Prediction

Predictive data mining is used to predict an unknown, often future value of a specific target variable (Johansson, 2007). Following Freitas (2002), it is about predicting the value that a target variable will take on in the future, based on already observed data. With that being said, prediction is similar to classification except that for prediction the results lie in the future (Larose, 2005). If that target variable is a real number and thus continuous, the data mining task is called *regression*; if the target is one of a predefined number of discrete class labels, the data mining task belongs to *classification.* Both have the function to create a model that minimizes the error between the predicted value and true value (Tan et al., 2005). It is an estimation of the function $y = f(x; q)$ which is able

to predict *y*, given an input vector of measured values *x* and a set of estimated parameters *q* for the model *f* (Johansson, 2007 as cited in Riveiro, 2011). Two general predictive data mining techniques are decision trees and neural networks (Riveiro, 2011). Although neural networks in general produce more accurate models (Shavlik, Mooney, Towell, & Quinlan, 1991), the algorithm is also characterized by a lack of comprehensibility and transparency, which discourages human understanding (Riveiro, 2011). In addition, neural networks only use point estimates which means that one cannot say something about the reliability of a prediction, while other algorithms, like decision trees, have the ability to extend the point prediction to an interval.

3.2.3.3 Classification and regression

Classification and regression are about the task of learning a target function that maps each attribute set to one of the predefined labels in order to classify unseen instances. Classification and regression can be used for both descriptive and predictive purposes. Classification is mostly used for data sets with binary or nominal categories, and are less effective for ordinal categories as classification algorithms do not consider the implicit order found in ordinal attributes; regression is used when input data is numerical (Tan et al., 2005). An efficient classification or regression model is characterized by four major features namely simplicity, comprehensibility, accuracy and interestingness (Pathak & Vashistha, 2015). A trade-off among all these features is desired while generating a model. Both classification an regression apply a systematic approach for building models from an input data set. Examples of techniques that can do both classification and regression are neural networks, decision trees, naïve Bayes and genetic algorithms. The generated model should both fit the input data as well as correctly predict the class labels of new records that the model has never seen before, also called 'generalization': the capability of accurately predicting the class labels of unknown records (Tan et al., 2005). Evaluation of a classification model is based on the counts of test records correctly and incorrectly predicted; evaluation of a regression model is based on the average distance the predicted value is away from the actual value.

3.2.3.4 Clustering

Cluster analysis groups data objects based on information that describes the objects and their relationships. Samples for clusters are represented as a point in a multidimensional space and samples within a certain space are more similar to each other than they are to samples belonging to another cluster (Kantardzic, 2011). However, in many disciplines the definition of what constitutes a cluster is not well defined. The reason for this lies in the fact that data is often in dividable in meaningful clusters in more than one way: data can discover clusters with different shapes and sizes. X shows how data can be clustered in more than one way. The figure shows that the definition of a cluster depends on the nature of data combined with the desired results (Tan et al., 2005). Clustering is strongly related to classification, in the sense that it creates a labelling of objects with class (or cluster) labels (ibid.). Where classical classification is also known as *supervised classification* (i.e. new labelled objects are assigned a predefined class label using a model already developed from objects with known class labels), clustering can be seen as *unsupervised classification* (where class labels of objects are not known on forehand).

3.2.3.5 Association

One methodology in data mining is known as association analysis, in which uncovered relationships in data sets can be presented in the form of sets of frequent items. A common example in association analysis is *market basket transactions*, which collects customer purchase data to subsequently identify which items frequently occur together (Tan et al., 2005). In this approach, one is interested in purchasing behaviour of customers to optimize their businesses. However, association analysis is also applicable to other domains like web mining and scientific data analysis.

3.3 Machine Learning

As stated in the previous section, machine learning finds its benefits in the fact that it can make a prediction of travel time without actually addressing the traffic processes that happen in between. It is for that reason that we will further elaborate the different machine learning techniques used to forecast arrival times. Jeong (2004) forecasted arrival times for bus routes in Houston using multiple linear regression models and Artificial Neural Networks (ANN). Also Francello et al. (2011) used Neural Networks (NN) to successfully reduce the uncertainty of predicting arrivals in a container ship terminal up to 6 hours. Parolas (2016) used both NN and Support Vector Machines (SVM) to find similar results, though found that SVMs consistently outperformed NN. Also Dashko (2017) investigates several techniques appropriate for arrival time prediction, including SVM and NN, together with k-Nearest Neighbours (kNN), Random Forests (RF) and simple linear regression. In the following sections, we will discuss the cited machine learning algorithms, both mathematically as well as functionally.

3.3.1 Multiple linear regression

Multiple linear regression models [1] the relationship between multiple explanatory variables and a response (target) variable by fitting a linear equation to the observed data. Every value of the explanatory variable x is associated with a value of the response variable y. The regression line for p explanatory variables $x_1, x_2, \dots, x_p$ is defined to be $\mu_y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$. The line shows how the mean response μ_y changes with the explanatory variables. The observed values for y vary about their means μ_y and are assumed to have the same standard deviation σ. The fitted values $b_0, b_1, \dots, b_p$ estimate the parameters $\beta_0, \beta_1, \dots, \beta_p$ of the regression line. Because the observed values for y vary around their means μ_y, the multiple regression model includes a term for this variation called *residual* (denoted with ε), which represents the deviations of the observed values y from their means μ_y. The final equation then needs to be extended to:

$$\mu_y = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \varepsilon_i \qquad \forall\, i = 1,2,..,n \tag{3.1}$$

The model is then trying to fit the best possible line to predict the target variable y by minimizing the sum of squares of the vertical deviations from each data point to the line. By first squaring and subsequently summing the deviations, there are no cancellations between positive and negative values. The fitted values $b_0, b_1, \dots, b_p$ are denoted by $\hat{y}_i$ and the residuals ε_i are calculated by $[y_i - \hat{y}_i]$.

3.3.2 k-Nearest Neighbours

Using nearest neighbour methods, the input consists of the k closest training examples in the feature space to x from $\hat{Y}$:

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \tag{3.2}$$

[1] Description of this model is based on the work of Freeman (2005).

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \quad \forall\, i = 1,2,..,n \tag{3.3}$$

Where $N_k(x)$ is the neighbourhood of x defined by the k closest points x_i in the training sample (Larose, 2005). Here, closeness implies a metric where the Euclidean distance is used as default (Hastie et al., 2009). Other metrics are Manhattan, Chebychev and Minkowski distance (Singh, Yadav, & Rana, 2013). For its fit, one parameter is available to tune, namely the number of neighbours k. Nearest neighbour techniques assume that locally the class probabilities are approximately constant and the same weights are used for each of the k selected neighbours. In some situations however, it might be better to change the weight depending on the distance to the sample, such that the weight will become inverse to the distance. Because this algorithm is a so-called lazy algorithm, since it only stores instances of the training data without explicitly making a model of it, high memory is required and the algorithm is computationally expensive (Larose, 2005).

3.3.3 Support Vector Machine

An SVM is a set of supervised learning algorithms used for both classification and regression. SVM has especially demonstrated its success in time-series analysis and statistical learning (Chun-Hsin et al., 2003). Following Haykin (2008), the main idea behind the algorithm is as follows:

> *"Given a training sample, the support vector machine constructs a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized."*

The plane is used as a separation border for classifying new samples. In the hyperplane, the dimension is always one less than its ambient space, meaning that points in a three dimensional space are separated by a two dimensional plane. An SVM for learning a linear classifier as primal problem is of the form:

$$f(x) = \mathbf{w}^T \mathbf{x} + b \tag{3.4}$$

where $\mathbf{x}$ is an input vector, $\mathbf{w}$ is an adjustable weight vector and b is a bias and the SVM can be solved by an optimization problem over $\mathbf{w}$:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \; ||w||^2 + C \sum_{i}^{N} \max\left(0,1 - y_i f(\boldsymbol{x}_i)\right) \tag{3.5}$$

This quadratic optimization is called the primal problem with N number of training points with d dimension of feature vector $\mathbf{x}$. However, instead of optimizing over $\mathbf{w}$, the SVM can be formulated for learning a linear classifier:

$$f(\mathbf{x}) = \sum_{i}^{N} \alpha_i y_i \left(\mathbf{x}_i{}^{\mathrm{T}} \mathbf{x}\right) + b \tag{3.6}$$

but now solving an optimization over $\boldsymbol{\alpha_i}$ with $\boldsymbol{\alpha} \in \mathbb{R}^N$:

$$\max_{\alpha_i \geq 0} \sum_{i} \alpha_i - \frac{1}{2} \sum_{jk} \alpha_j \alpha_k y_j\, y_k \left(\mathbf{x}_j{}^T \mathbf{x}_k\right) \tag{3.7}$$

Subject to constraints:

$$0 \leq \alpha_i \leq C \quad \forall\, i \tag{3.8}$$

$$\sum_i \alpha_i y_i = 0 \tag{3.9}$$

Which is called the dual problem. Both the primal and dual problem of SVM assume linearity between x and y. To also deal with nonlinearly separable patterns, one can apply the so-called 'Kernel trick', which maps the samples to a higher dimensional space such that it becomes possible to separate the two classes with a line. See figure 3-2 for a visual representation of this transformation.

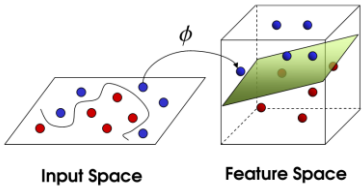


Figure 3-2: The Kernel trick in an SVM enables the algorithm to deal with nonlinearly separable patterns.

Note that this Kernel function, which is of the form $k(\mathbf{x_j},\mathbf{x_k}) = \phi(\mathbf{x_j})^{\mathrm{T}}\phi(\mathbf{x_k})$, is best applicable to the dual problem because of its structure. There are several possible Kernel functions used to this end, like polynomial functions, radial basis functions and sigmoidal functions (Wiering et al., 2013). The SVM can be applied to solve both pattern-classification as well as nonlinear-regression problems. However, SVMs have proven to make the most significant impact in solving difficult pattern-classification problems. Its performance depends on good parameter settings for both the classifier and the Kernel function. Because multiple parameters are important in the SVMs performance, optimal parameter selection becomes a complex task. It is for that reason that Bin et al. (2006) discovered that for large problems, large computation time will be involved.

3.3.4 Neural Networks

The description of neural networks is based on Haykin (2008), who compares the network with a limited version of how the human brains function. A neural network is organized in so-called 'layers', consisting of neurons. Figure 3-3 shows a visual representation of such network with, in this example, three layers.

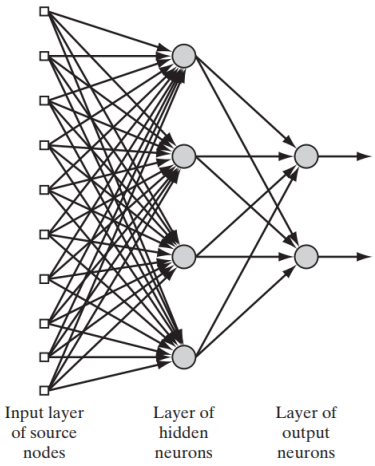


Figure 3-3: Typical neural network with its layers

This example consist of an input layer and an output layer (requirements for a neural network to exist) and one optional hidden layer. The input layer contains some input vector, for example exploratory attributes. The output layer then contains the prediction. The term hidden refers to the fact that this part of the network is not seen directly from either the input or output of the network and is present in the network to somehow intervene between the in- and output. The more hidden layers there are, the more the network is able to extract higher-order statistics from its input. This network is also characterized by the fact that it is fully connected, meaning that every node in each layer is connected to every node in the adjacent layer (the black lines). The connected layers are characterized by respective weights. These weights can be represented by a simplistic form:

$$[x_0, x_1, \dots, x_n]^T \rightarrow [\ ] \rightarrow Z_\theta(x) \tag{3.10}$$

Training of the neural network encompasses selecting the optimal weights for each connection between the neurons from each layer. Given the input vector as exploratory attributes and the output vector as actually occurred observations, the neural network is trained to minimize the error between the prediction and actual occurred observations. Following Haykin (2008), the algorithm that is used for this training phase is called backpropagation algorithm. Drawbacks of the neural network are that training the algorithm is computationally very expensive and the hidden layers in the neural network function as a black box in the sense that even though they can approximate any objective function, studying its structure will not expose any insights on the structure of the objective begin approximated.

3.3.5 Random Forests

The basic principle of the Random Forest (Breiman, 2001) is training the data on multiple trees where each tree in the forest learns from a random sample of the data points. The random sample of input variables in the training phase are drawn *with* replacement, meaning that in one tree, some samples can be used multiple times. This is also called bootstrapping. Then in the test phase, predictions are made by averaging the predictions of each tree. This procedure of training the data on different bootstrapped subsets and then averaging the resulting prediction, is also called bagging (bootstrapped aggregating). By randomly selecting input variables through the tree-growing process, the algorithm can achieve variance reduction without the compromise of increasing the bias. The

algorithm's purpose is thus to improve variance reduction of bagging by reducing the correlation between the trees, without increasing the variance too much. Trees are then good candidates for this purpose as they can capture complex interaction structures in the data and, if grown sufficiently deep, have relatively low bias (Hastie, Tibshirani & Friedman, 2009). In case of regression, we just fit the same regression tree many times to bootstrap-sampled versions of the training data and average the result. That being said, the Random Forest acts like training multiple Decision Trees. Random Forests take advantage of trees being notoriously noisy by averaging the results (James, Witten, Hastie, & Tibshirani, 2013).

As can be derived from the algorithm in Figure 3-4, Random Forests can be applied in both classification and regression problems. For classification, a random forest obtains a class vote from each tree and then classifies using majority vote. For regression, the predictions from each tree at a target point *x* are simply averaged.

Algorithm *Random Forest for Regression or Classification.*

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

Figure 3-4: Pseudo code of the Random Forest algorithm (source: James et al., 2013).

In (b) in the algorithm from Figure 3-4, before each split, $m \leq p$ of the input variables are selected at random as candidates for splitting. Typical values for m are $\lfloor\sqrt{p}\rfloor$ for classification and $\lfloor p/3 \rfloor$for regression. Intuitively, reducing m will reduce the correlation between any pair of trees and thus reduce the variance of the average, which is a good thing, but it also decreases the strength of each individual tree, which one must avoid. However, the best values in practice highly depend on the problem and should therefore be treated as tuneable parameters. Another tuneable parameter is the number of trees T_b to grow from the training data. A default setting of the number of trees to build, is 500 (used in Breiman, 2001; Hastie et al., 2009; the R package `randomForest 4.6-14`. Here also counts that the optimal value for T_b is highly case-dependent, ranging from 200 to 1000 trees.

3.3.6 Conclusion

Now we have described five machine learning algorithms, we will summarize the characteristics of each technique that become relevant when applying it to our data, which are the accuracy, data needed, training time, transparency, non-parametric and frequently used in arrival time prediction models. With this overview, it becomes easier to choose the algorithm that is most applicable to our research.

The *accuracy* depends on how well the algorithm performs in terms of correctly predicting a future variable. The higher the accuracy, the better. The *data needed* refers to the input an algorithm needs to produce a reliable output. Some algorithms, like Neural Networks, need a huge amount of data before it can perform a prediction. The less data is needed, the better. The *training time* refers to the computational complexity of the algorithm: it is the time it takes to run an algorithm (Sipser, 2006). Computational complexity is estimated by counting the number of elementary operations the algorithm must execute. The less training time is needed, the better. Algorithmic *transparency* states that the inputs and the working of the algorithm itself must be known. Variables and parameters that influence the decisions the algorithms make, should be clearly visible and interpretable (Diakopoulos & Koliska, 2017).The more transparent an algorithm is, the better. *Non-parametric* refers to the fact that the algorithm is not based on probability distributions of the input data.

Table 3-1 shows the outcome of summarizing the algorithms on these characteristics. The value (--, -, +-, + or ++) in each cell describes to what extent the algorithm's characteristic is realized. The value says nothing about preferences: for example, when at *data needed* the value is +, it means that much data is needed (and not that the amount of data that is needed, is a good thing).

Table 3-1: Summary of algorithm characteristics

	Linear regression	*kNN*	*SVM*	*NN*	*RF*
Accuracy	-	+	+	+	+
Data needed	-	-	+	++	-
Training time	-	+-	+-	++	+-
Transparency	+	-	-	--	+-
Non-parametric	No	Yes	Yes	No	Yes
Frequently used in arrival time prediction models	+-	-	+-	+-	-

From this table, the random forest algorithm seems to be the most suitable algorithm since it has the most promising characteristics. There is not much data needed and the accuracy is high. However not completely a white box, random forests still perform better than kNN, SVM and NN in terms of transparency. Since we have no insight in the behaviour of our data yet, it is preferable to use an algorithm that does not make any assumption to the distribution of the data because we are not sure if we are able to achieve this with the data available to us. Besides, especially SVMs and NNs are frequently used in already existing arrival time prediction models. The random forest is not a very common algorithm in arrival time prediction models, which makes it an interesting option to discover.

3.4 MACHINE LEARNING FUNDAMENTALS

Since the results of the literature research so far, argue for applying a Random Forest machine learning algorithm to our arrival time prediction model, a more extensive description of the algorithm would be appropriate. Therefore we will introduce two concepts that are fundamental in the field of supervised machine learning, namely over- and underfitting and the bias-variance trade-off. The sections discussing these two concepts are roughly based on Marsland (2015).

3.4.1 Over- and underfitting

When performing a predictive machine learning algorithm, the prediction errors (which one typically wants to minimize) due to bias and variance are important to address. Bias is the difference between the actual value and the average prediction of the model. High bias can cause a model to be too general and to miss essential relations between the predictor variables and the response (target) variable. This occurrence is also referred to as *underfitting*. For models with high bias, the variance is typically low, which is the variability of the model's prediction for a given value which reflects the spread of the data. On the contrary, when a model has high variance it is not only trained to learn the actual relationships, but also any noise that is inevitably present in the dataset. This means that the model is fitted too well on training data: it is perfectly capable of predicting the training data, but will then perform bad on (never seen before) test data as these data is different. The model is said to have high variance because the parameters it has learned from the training data must vary considerably to be able to predict every single record perfectly. This occurrence is also referred to as *overfitting* and for these models it applies that the aforementioned bias is low: for the training data that is (too) perfectly fitted, the difference between the actual and the prediction value will be low.

3.4.2 Bias-variance trade-off

Ideally, we want to build a model that has learned the training data well, but is then also capable of generalizing by translating it to test data it has never seen before. This leads to a model in which one want to minimize the prediction errors due to bias and due to variance. However, it is typically impossible to achieve this combination simultaneously and is therefore referred to as the bias-variance trade-off: it is the optimal balance between creating a model that is fitted well enough on the training data to learn the underlying relationships, but not too well because then the model is not able to generalize it to new unseen data.

3.4.3 Random forest algorithm

Using a basic decision tree, the model is usually suffering from low bias and high variance which we referred to as overfitting. When the depth of the tree is not limited, the tree can keep growing until it has exactly one leaf node for every single observation in the train data. It is for that reason that Breiman (2001) introduced the random forest algorithm, which is constructed out of multiple single decision trees. The 'forest' part in the algorithm's name refers to simply averaging the predictions of each individual tree in the forest. The 'random' part in the algorithm's name is more interesting as it refers to two key characteristics of the model that makes the algorithm robust to both overfitting and underfitting. The first characteristic is the random sampling of training observations where, during training, in each tree a random sample of the training data is presented to make a prediction from. This sampling is done with replacement (also known as bootstrapping), meaning that the same record can be used multiple times in one single tree. Because each tree is trained on different samples of the training data, the variance

within one tree might be high with respect to that particular set of the training data (the sample). However, when doing this often enough for multiple trees the overall variance of the forest is low but not at the cost of increasing the bias. The second characteristic is the random subsets of features for splitting nodes within a decision tree. So for splitting each node in each decision tree, only a random subset (for regression the default is the number of predictor variables divided by two) of features is presented to the model. This means that each tree only has a limited number of features available to split on. Again, when training enough trees in the forest, eventually all features (prediction variables) that the model consists of will have been used in some of the trees. It is this random subset of features that prevents the model from an increased bias and variance. If the features were not chosen randomly, the same features (with most predictive power) would be chosen much more often in growing a single tree. Individual trees in a forest would become highly correlated, which is something that does not contribute to lowering the variance without the cost of the increasing the bias. Moreover, the random subset of features that is available for splitting, alleviates the multicollinearity problem since a random subset is chosen for each tree. This makes a random forest robust to multicollinearity.

To conclude, random forests seem to be a promising learning technique when trying to achieve the perfect balance between overfitting and underfitting, being much more robust than a single decision tree. Also the random forests' robustness against multicollinearity is desirable in our case, since the causal model consists some correlated predictor variables. We therefore decide to apply the Random Forest technique to our prediction model.

3.5 FEATURE SELECTION

Feature selection, variable selection, attribute selection or variable subset selection are all terms referring to the same task of selecting those features that contribute most to the prediction variable you are interested in and removes irrelevant/redundant attributes (Guyon & Elisseeff, 2013). Feature selection is important especially in the field of supervised learning, for example classification. The more features (attributes), the more the dimensionality of data increases which makes testing and training methods difficult. A reduced set of attributes reduces computation time. Less dimensions also counts for a less complex model, which makes the revealing patterns easier to understand (Karegowda, Jayaram, & Manjunath, 2010). Not only execution time reduces to this end, also accuracy increases as irrelevant features can include noise affecting the accuracy negatively (Karabulut, Ozel, & Ibrikci, 2012). Last, overfitting is reduced as less redundant data means less opportunity to make decisions based on noise. A distinction is made between selecting features one by one and selecting subsets of features.

Examples of selecting features one by one are variable ranking and correlation methods, where the presence of each individual feature is evaluated independent of the context of others. One concern with scoring features individually and independently of each other, is potentially losing or neglecting valuable features throughout the filtering process. One attribute can seem to be redundant on its own, but together with another attribute in the original feature set the attributes can be of huge predictive power.

We will therefore now describe the feature subset selection methods that have been developed to tackle this problem. Selecting the best subset of features aims for a systematic approach, generally consisting of four steps:

i) Generate candidate subset
If the original feature set contains n features, the total competing candidate subsets to evaluate is 2^n, which is a huge number even for relatively small feature sets. Generation of subsets is therefore based on a certain search strategy, for example *complete search* (branch & bound, beam search, best first), *heuristic* (forward selection, backward selection, bi-directional selection) and *random search* (genetic algorithm (GA), random generation plus sequential selection (RGSS), simulated annealing (SA)).

ii) Subset evaluation
In this phase, the generated subset from i) is evaluated by using either a filter or a wrapper approach. We will further describe these two approaches in Sections 3.4.1 and 3.4.2.

iii) Stopping condition
As mentioned above, the number of subsets to evaluate can become enormous. It is for that reason to include some stopping criterion, that can be based on the generation procedure (whether a predefined number of iterations is reached) as well as on the evaluation procedure (whether a subset exceeds some predefined evaluation function).

iv) Validation procedure
The generated subset of features is validated by comparing the result with the original feature set according to some algorithm using real-world or artificial data.

Figure 3-4 shows a schematic representation of the feature selection process.

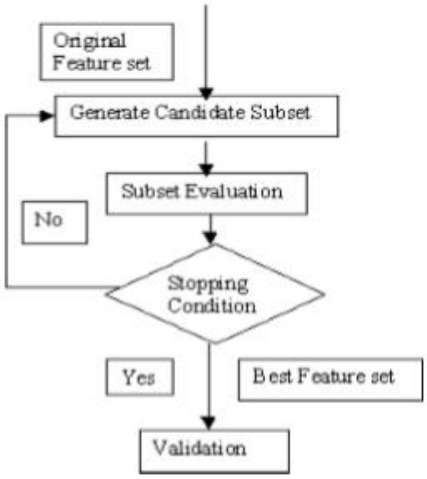


Figure 3-5: Steps of feature selection process (source: Karegowda et al., 2010)

3.5.1 The filter approach

The filter approach precedes the actual classification process and benefits from its computationally fast and scalable performance (Jantawan & Tsai, 2014). Using this approach, feature selection is done once and is then provided as input to different classifiers. The filter approach provides a generic selection of variables that is not tuned by a given learning machine; they are independent of the chosen predictor. It is often used as a pre-processing step to reduce space dimensionality and overcome overfitting.

3.5.2 The wrapper approach

Wrapper approaches use the prediction performance of a certain learning machine to assess the merit (relative usefulness) of a feature subset. They do this by using the method of classification itself to measure the importance of a certain set of features. Better performances are achieved using wrapper methods instead of filter methods because the feature selection procedure is optimized for the classification algorithm to be used (Karegowda et al., 2010). However, the search can become computationally intractable and is known to be NP-hard (Amaldi & Kann, 1998). But several search strategies are developed to tackle the problem of NP-hardness, including branch & bound, genetic algorithms and simulated annealing. It is for that reason that the wrapper approach is preferred at the cost of the filter approach.

3.6 PERFORMANCE MEASURES

There are multiple metrics used in evaluating how well a model is able to predict. In this section, we will discuss the performance measures that we will apply later on to evaluate the performance of our prediction model. We chose this selection out of the wide variety of measures, as these are one of the most frequently used performance metrics that are appropriate for Random Forest regression: the Mean Squared Error (MSE) and the Root Mean Squared Error (RMSE), the R Squared (R^2) and the Adjusted R Squared (R^2_{adj}).

3.6.1 (R)MSE

The MSE is the most common and simplest metric for regression evaluation and it measures the average squared error over the predictors. For each record, it calculates the squared difference between the actual value and the predictions, after all these values are summed and averaged over the total input space. Since the difference between actuals and predictions is squared before summing them, the MSE can never become negative, but would be zero in case of a perfect model. The MSE's formula is as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_{i-}\hat{y}_i)^2 \tag{3.11}$$

Where y_i is the actual output and $\hat{y}_i$ is the model's prediction. To make the performance of a model more interpretative, one can take the square root of the MSE, which results in the RMSE:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_{i-}\hat{y}_i)^2} \tag{3.12}$$

By taking the square root, the measure is in the same units as the target variable, which makes the error better interpretable.

3.6.2 R^2

The coefficient of determination, also called R^2, assesses goodness-of-fit of a regression model on a scale from 0 to 1. It measures the proportion of variance in the response variable that is explained by the explanatory variables: it calculates the percentual reduction in prediction error.

$$R^2 = 1 - \frac{\frac{1}{N}\sum_{i=1}^{N}(y_{i-}\hat{y}_i)^2}{\frac{1}{N}\sum_{i=1}^{N}(y_{i-}\bar{y})^2} \tag{3.13}$$

Where $\bar{y}$ is the mean of the observed y_i. In general, the higher the R^2, the better the model fits the data. However, there are some remarks valuable to mention. For example, the R^2 will always increase by putting more explanatory variables in the model, regardless if these variables are actually contributing to a better prediction. A regression model can thus have a higher R^2 not because it predicts better, but because it contains more explanatory variables. It is for that reason that the adjusted R^2 is invented, that includes the number of variables that your model consists of:

$$R^2_{adj} = 1 - \frac{(1 - R^2)(n - 1)}{(n - p - 1)} \tag{3.14}$$

So, now the value only increases if a new explanatory variables actually improves the model's predictive performance. Intuitively, one wants the adjusted R^2 to be as close as possible to the R^2 because this would mean that there are no 'noise' variables in your model.

3.7 CONCLUSION

During the literature review, we discussed several arrival time prediction models available in literature, under which time series models, regression models and machine learning models. We concluded that machine learning models would be most appropriate to apply in our research. Before we went further into the field of machine learning, a section was dedicated to data mining and we described how this domain relates to machine learning. We ended this chapter with an overview of relevant machine learning models that can be used in predicting arrival times and concluded that random forest would probably be the most promising one. We also discussed feature selection methods from which we will use the wrapper approach in the preparation phase for building our prediction model. Finally, we introduced the (R)MSE, the R^2 and the adjusted R^2 as performance measures, which we will use when evaluating the performance of our model later on.

4 DATA UNDERSTANDING

Since the quality of the data makes or breaks the success of a data mining project, it is important to assess the data, as well as available tools and techniques, as early as possible since this selection may influence the entire project (Chapman et al., 2000). In this chapter, we start with initial data collection with the goal to check the data for missing values and outliers, and to discover first insights to potentially formulate hypotheses from. We will partly give answer to the first sub question of question II: *Which attributes does the historical order data contain?*

4.1 COLLECTION OF INITIAL DATA

The LMS is, as mentioned earlier, the platform that is used daily by the global forwarding department of the LSP for planning and controlling all shipments. This platform also has a storage function as all order data are collected in a data warehouse which is a specific page in the LMS. Orders are automatically added to this file when the status of the order is changed to 'closed'. The data can easily be downloaded in a CSV file format.

4.2 DESCRIPTION OF DATA

The CSV file contains shipments from October 2016 until January 2019 (which is the present at the moment of executing data collection). The initial file consists of 227.192 rows (shipments) and 40 columns (attributes). However, also shipments by road, which are processed by another department, are included in the CSV file. After filtering on shipments by sea, a total of 17.198 shipments remain. However, from these shipments, almost two third neither have a departure time, nor have an arrival time. They have no time identity at all. Because no single reference exists to the time when the shipment was executed (if it was actually executed at all), it is hard to estimate this period in time. Moreover, because this applies to more than two third of all shipments, it would not reflect reality anymore if more values are estimated than that were initially available and from which we are sure about the correctness. We therefore choose to exclude this shipments, which result in a remaining dataset of 4932 shipments.

Intuitively, not all attributes in the dataset are relevant in the context of predicting the accuracy of the ETA. In the table that can be found in Appendix C , we go through all attributes and make a selection of attributes to in- and exclude based on the capability of predicting the anomalous ETA. The attributes that remain form a deliberate collection of possible explanatory attributes for the anomalies in the estimated and actual arrival times of shipments. Table 4-1 contains the selected attributes.

Table 4-1: Selected attributes

Attribute	*Description*	*Data type*
Booking number	The number of the booking (unique identifier)	CHARACTER*
Carrier	The carrier with which the order is shipped	CHARACTER
Ship	Name of the ship with which the order is sent	CHARACTER
Port of Loading (POL)	The port from where the order is loaded	CHARACTER
STD	Scheduled Time of Departure	DATE
ETD	Estimated Time of Departure	DATE
ATD	Actual Time of Departure	DATE
Port of Delivery (POD)	The port where the order is delivered to	CHARACTER
STA	Scheduled Time of Arrival	DATE
ETA	Estimated Time of Arrival	DATE
ATA	Actual Time of Arrival	DATE

* Character is the data type for categorical data in R statistical software.

4.3 EXPLORATION OF DATA

When focusing on the structure of the data, we notice that each record in the dataset represents an *order* of a customer, also denoted as a *shipment.* It represents a container that is transported from one port to another, on a specific container ship and executed by a specific carrier. But a delay happens on the level of a ship; not on the level of a container on that ship. To do justice to this aspect of the domain, we introduced a *sailing* as a dimension of a shipment which is a transit of a container ship between two ports.

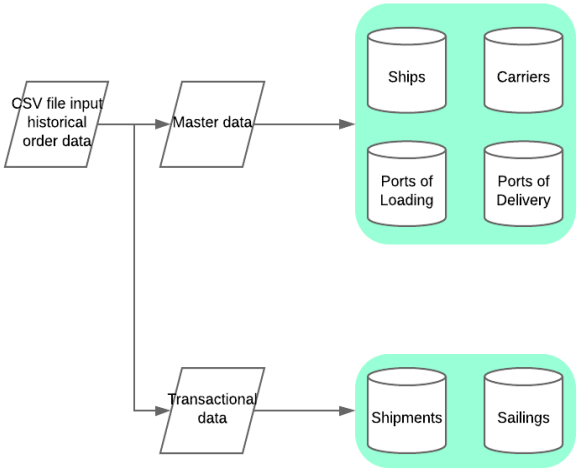


Figure 4-1: Data architecture

An important distinction to be made here is that of Master data compared to Transactional data. We can categorize ships, carriers and ports as master data; transactional data are dependent on the *time* dimension and consist of the departure- and arrival times. A transaction is a shipment or a sailing. As stated before: A shipment is one order by a customer; a sailing is a transit of a ship between two ports. One sailing can contain multiple shipments. A unique combination of master data in a specific point in time (transaction), forms a shipment; shipments that have both master- and transactional data in common, together form a sailing.

In order to systematically process the data, we used `MySQL Server` as database management system (DBMS). In `MySQL Workbench`, we designed the structure of the database using a star scheme, which can be found in Figure 4-2. Note that the data architecture is based on the master- and transactional data as described earlier: each data category corresponds with an individual table in the star scheme. We designed this star scheme to easily make visualizations in Tableau, which is an interactive data visualization software. Tableau can easily connect with MySQL to load the data form the star scheme into Tableau.

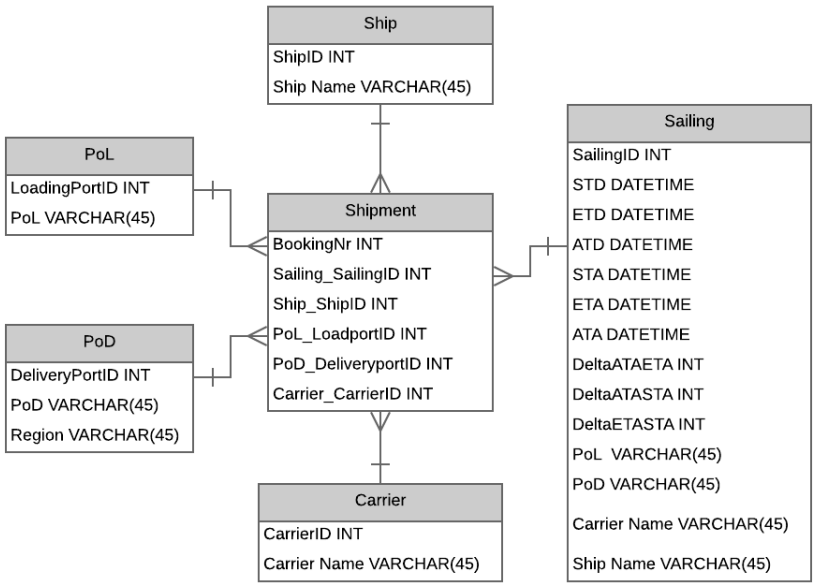


Figure 4-2: Structuring the database using a star scheme (made by the author in MySQL Workbench)

The data warehouse is filled using an ETL (Extract, Transform, Load) flow that we built in the tool called `Kettle`. The ETL flow roughly consists of two parts: the first part is constructing the tables of the Master Data i.e. the ships, carriers, and the ports of loading and delivery. These are simple flows that create a basic table consisting of all unique records in that dimension and adding an ID (unique identifier) to it. Second, we computed the Transactional data tables (consisting of sailings and shipments) out of the four Master Data tables. As the initial CSV file of the LSP contained shipments, this becomes our fact table. Note that a sailing is a dimension of a shipment, as multiple shipments can 'have' the same sailing. The shipment table was not that hard to construct as it processes the data on the same level as the input file: the flow consists of only combining the dimension tables. The sailing table was a little bit more complicated as a sailing is a unique combination of the attributes Port of Loading, Port of Delivery, Ship and STD; when multiple shipments have the same loading- and delivery port, are transported on the same ship at the same point in time, we assume that the shipments were part of the same sailing. The complete ETL flow can be found in Appendix D

4.3.1 Shipment versus sailing

During the first explorations of the data, we have noticed that we can analyse the data from both a shipment and a sailing perspective. The ETL flow gives two transaction tables as output, where the sailing table is nothing more than the shipment table where shipments from the same sailing are merged to one record. The constructed shipment table consists of 4932 records. When merging the shipments to a unique sailing, less than half of the initial records remain (2060). A sailing would be the most appropriate perspective as a delay happens on this level. However, the global forwarding department of the LSP uses a shipment as operational and analytical perspective, which argues for an analysis on this level. Results are easier comparable to the current situation at the LSP when using shipments and, moreover, we have a bigger dataset to train and test. Also from the customer's perspective, a shipment is more appropriate than a sailing as the customer does not care about other containers being on the same container vessel. So

even though it does not completely reflect reality, we choose to use a shipment as analytical perspective.

4.4 VERIFYING QUALITY OF DATA

Now we systematically explored the data, we have a better overview of the remaining attributes and how they behave. We are now going to measure their quality on ambiguity and missing values respectively.

4.4.1 Ambiguity

Although the LSP claims that they only process shipments from two ports (Rotterdam and Antwerp), the data show that the corresponding attribute 'Port of Loading' has more than two values. We decided to bring all the ambiguity in port names back to either Rotterdam or Antwerp. Table 4-2 below shows the changes we have made in the different port names that we found in the data.

Table 4-2: Changes that are made in the PoL attribute due to ambiguity

Initial port name	*Changed to*
Antwerp	N.A.
Rotterdam	N.A.
Terminal Antwerp	Antwerp
Terminal Rotterdam	Rotterdam
Maasvlakte Rotterdam	Rotterdam
Port of Rotterdam	Rotterdam
Antwerpen	Antwerp
Lommel	Antwerp
Vlissingen	Antwerp

Note that we also changed orders from the ports 'Lommel' and 'Vlissingen' to 'Antwerp'. We can justify this by the fact that the ports represent almost the same area from a geographical perspective. Thereby, there are only a few orders (63 and 37 respectively) that are shipped from these two ports, and the sub group they form would be too small to make statements (let alone predictions) about.

We also found ambiguity in the different notations for the same carrier, e.g., difference in capitalization and punctuation. Using the FILTER function in Excel, the records with double notation are changed to one overarching name. Given the small number of records with ambiguity, we chose to manually adjust the names in the CSV file.

We also encountered that sometimes a ship that kind of belongs to a specific carrier (i.e., OOCL Korea is a ship from the carrier OOCL), is also found in shipments from other carriers. We do not know if these are errors, or that different carriers are cooperating in the sense that they rent space for containers on each other's ships. Due to the lack of additional information about ships belonging to carriers, we neither changed nor deleted the concerning shipments.

We lastly found some formatting ambiguity in date columns, with a number of rows following *dd-mm-yyyy HH:mm* and other rows following *yyyy-mm-dd HH:mm:ss.SSS*. Or worse, the column containing a date had a 'text' cell format. We manually adjusted all the date columns to one format.

4.4.2 Missing values

As stated before, a lot of missing values are present in the departure- and arrival times. Almost two third of the shipments do not have an STD and consequently have no STA, ETA nor ATA. We do not know if these values are just missing or that the corresponding shipment was never really executed. Because no single reference exists to the time when the shipment was executed (if it was actually executed at all), it is hard to estimate this period in time. Moreover, because this applies to more than two third of all shipments, it would not reflect reality anymore if more values are estimated than that were initially available and from which we are sure about the correctness. We therefore choose to exclude shipments with missing departure- and arrival times. Also, some missing values are found in the columns Port of Loading (26 rows); Port of Delivery (54 rows); and Ship (99 rows). From a data completeness perspective, we chose to only include shipments that have no missing values in any one of the aforementioned columns. Rows with a missing value in one of these columns, are removed. Even though it is possible to estimate missing values using several modelling techniques, we try to find correlations by only using real data first. Fortunately, the amount of data that remains after cleaning allows us to do so. The result after cleaning is a dataset consisting of 4781 shipments.

4.5 Conclusion

In this chapter, we collected the initial data to discover first insights to potentially formulate hypotheses from. We extensively described the data, including discussing the columns present in the CSV file with historical order data. After determining all relevant columns in the context of our research, we focused on the structure of the data where we distinguished between a shipment and a sailing, where a sailing is a dimension of a shipment which is a transit of a container ship between two ports. One sailing can contain multiple shipments. In order to systematically process the data, we used MySQL Server as database management system. In MySQL Workbench we designed the structure of the database using a star scheme. The tables in the star scheme are filled using an ETL flow. After systematically exploring the data, we verified their quality. We evaluated the quality based on the presence of ambiguity and missing values, respectively. Especially in Port names, we noticed a lot of ambiguity. Also at the Carriers, Ships and date columns ambiguity is discovered. We removed all ambiguity. Last, we evaluated missing values. A lot of missing values were present in the departure- and arrival times: almost two third of the shipments did not have an STD and consequently have no STA, ETA nor ATA. We do not know if these values are just missing or that the corresponding shipment was never really executed. Because this affected more than half of the shipments, we chose to exclude them. We are aware of the fact that this is a rough deletion, but we chose to do it anyway in favour of reliability.

5 DATA PREPARATION

Now we have gained more insight in the data that is available at the LSP and how it is structured, we are going to further prepare the data. We do this by consecutively deriving new attributes from existing ones and then trying to make hypotheses about which attributes are possibly good predictors for the deviation in arrival time. After we have constructed a dataset, we are going to apply feature selection in Section 5.4 to find the best subset of attributes for input of our prediction model. We answer the second part of research question II during this chapter: *Which attributes can we derive from the original dataset and which ones are included in the final prediction model?*

5.1 CHOICE OF ATTRIBUTES

Before we are going to prepare our data in order to create the most accurate prediction model, we follow the theory of Shmueli and Koppius (2011) who define two constraints in case of choosing variables as input of a prediction model: they should be 'available at time of prediction' and they should be of good quality. They also stated that the choice of potential predictors is often wider than in a resulting explanatory model. We come back to this statement in Section 5.4.

5.2 DERIVED ATTRIBUTES

In the process of deriving attributes, we use our knowledge about the domain to calculate (derive) new attributes from existing attributes. The following attributes are derived as we intuitively think these are potentially valuable attributes for predicting the deviation in arrival times: the Delta, on-time performance, period in time, transit time, region and the hurricane season.

5.2.1 The delta

The most important derived attribute is the Delta: the deviation in arrival time, which we denote by Δ_s in the mathematical notation. As we have three types of arrival times, we can calculate more Delta's from the available data: *ATA-ETA; ATA-STA; ETA-STA*. In our case, we are interested in the deviation of the actual versus scheduled arrival time *in advance of* the shipment *s,* measured in days. We chose to calculate this attribute on the level of days (instead of hours, minutes) because the arrival time information from the carrier in advance of and during shipment is also provided in days. We think this attribute is the most important one because it says something about the accuracy: the further away from zero, the more deviation and the less accurate the arrival time information was.

$$\Delta_s = ATA_s - STA_s \quad \forall\, s \in 1, \dots, S \tag{5.1}$$

5.2.2 On-time performance

With this newly derived attribute, we are also capable of calculating the on-time performance $OTP(\Delta_s)$, which is a binary attribute and can take the values 0 (not on time) and 1 (on time). We based the value on the threshold of seven days, as the LSP uses this

value in their analyses about their on-time performances: an order is defined to be 'on-time' when the delta deviates less than a week.

$$OTP(\Delta_s) = \begin{cases} 0, & -7 > \Delta_s > 7 \quad \forall s \in 1,..,S \\ 1, & -7 \leq \Delta_s \leq 7 \quad \forall s \in 1,..,S \end{cases} \tag{5.2}$$

As one can notice, the Δ_s also has a direction: a negative value means that the order arrives earlier– a positive value means that the order arrives later than estimated. We will further elaborate on the existence, meaning and consequences of this split in Chapter 7.

5.2.3 Period in time

We also want to give each order a value that represents a specific period in time, in such a way that we can divide the orders with a specific departure- and arrival time in categories. As categories, we chose day of the week, week of the year and month of the year and we calculated them for both the scheduled departures (STD) and the scheduled arrivals (STA), for each shipment $s \in 1,..,S$.

$$ArrDay_s = DAY(STA)_s \quad DepDay_s = DAY(STD)_s \tag{5.3}$$

$$ArrWeek_s = WEEK(STA)_s \quad DepWeek_s = WEEK(STD)_s \tag{5.4}$$

$$ArrMonth_s = MONTH(STA)_s \quad DepMonth_s = MONTH(STD)_s \tag{5.5}$$

5.2.4 Transit time

Intuitively, the transit time can be an important indicator of orders not arriving on time: the longer the transit time, the more 'space' for disruptions and the more deviation in arrival time is expected. The transit time can be determined in two ways. First, by simply calculating the difference between the actual arrival- and departure time (actual transit time). Second, by following the sailing schedules that are published by the executive carrier and are collected by the LSP in a separate CSV file (scheduled transit time). Each row in that file represents a unique combination of PoL-PoD and has a certain transit time. However it is more time-consuming to integrate this separate file into our historical order dataset than simply subtracting the arrival time from the departure time, we chose this as method to determine the transit time because it better fits the context of our research. We want to make a prediction *in advance of* the shipment, meaning that we can only base our prediction on attributes that are known at that moment in time. Since the actual arrival time of the order is not known yet at this moment of predicting (simply because this is what we aim to predict), we decided to take the scheduled transit time rather than the actual transit time as potential explanatory attribute.

5.2.5 Region

In the same separate CSV file as mentioned above, every record in the file also contains the PoDs region. The LSP works with six 'trade areas' as regions, where each trade area represents a continent. As it is possible that from a more aggregate level of detail other patterns in the data become visible, we choose to include the region of each PoD in our dataset. Using Visual Basic in Excel, we loop through all the records in the file searching for a specific PoD – region combination, and add the corresponding region in a new column.

$$Region_s \in \{Africa, Asia, Middle\ East, North\ America, South\ America, Oceania\}$$

5.2.6 Hurricane season

Although several authors who researched arrival time prediction models encountered that 11weather is not such a good predictor for arrival times as always expected, the management of the LSP has expressed their interest in studying the influence of hurricane seasons. We therefore included a binary variable that indicates a hurricane season (1) or not (0). Because hurricane seasons often affect bigger areas than just one port, we decided to determine the value (0/1) of this variable on the variable 'region'. For each region, we determined the months of the hurricane season (source: http://landen.net/weer-klimaat/orkaan.html). Thus, each record in the dataset either got a 1 or 0 depending on the combination of the region and whether the STA falls in a hurricane season (1) or not (0). Table 5-1 shows the periods of hurricane season determined per region.

Table 5-1: Determination of hurricane season per region

Region	Hurricane season	From	To
North-America	Atlantic Hurricane season	01/06	01/12
South-America	Atlantic Hurricane season	01/06	01/12
Africa	Atlantic Hurricane season	01/06	01/12
Asia	Great ocean hurricane season	01/05	01/12
Australia	Great ocean hurricane season	01/11	01/04

We are aware of the fact that we do not know whether there has actually been a hurricane in one of the areas at the time of a shipment was sent to that area, or that an order passed an area where actually has been a hurricane. Although the quality will not be excellent, we are nevertheless curious about the effect of the binary variable in our model.

5.3 HYPOTHESIS TESTING

In this section, we are going to elaborate to what extent we may assume that the variable is of influence on the deviation in arrival time. We do this for the following attributes: the carrier, the ship, the port of loading, the port of delivery, the period in time and the transit time. We now have the following initial attributes available, which are listed in Table 5-2.

Table 5-2: Summary of initial data

Attribute	*Description*	*Data type**
Carrier	The carrier with which the order is shipped	CHARACTER
Ship	Name of the ship with which the order is sent	CHARACTER
Port of Loading (PoL)	The port from where the order is loaded	CHARACTER
Port of Delivery (PoD)	The port where the order is delivered to	CHARACTER
STD	Scheduled Time of Departure	DATE
ETD	Estimated Time of Departure	DATE
ATD	Actual Time of Departure	DATE
STA	Scheduled Time of Arrival	DATE
ETA	Estimated Time of Arrival	DATE
ATA	Actual Time of Arrival	DATE

*Following R statistical software's data types.

Table 5-3 consists of the attributes that are derived from the initial dataset from Table 5-2. Now we have enriched our dataset by deriving attributes from existing ones, some initial attributes lose their quality. This counts for the six DATE variables of the different departure- and arrival times (STD, ETD, ATD, STA, ETA and ATA), which are now represented by the more aggregated 'period in time' attributes. We choose to exclude the initial departure- and arrival time attributes.

Table 5-3: Summary of derived data

Derived attributes	***Calculation***	***Data type***
Delta (Δ_s)	$\Delta_s = ATA_s - STA_s$	INTEGER
On-time performance	$OTP(\Delta_s) = \begin{cases} 0, & -7 > \Delta_s > 7 \\ 1, & -7 \leq \Delta_s \leq 7 \end{cases}$	LOGICAL
Departure Day	$\text{DepDay}_s = DAY(STD)_s$	CHARACTER
Departure Week	$\text{DepWeek}_s = WEEKNUMBER(STD)_s$	CHARACTER
Departure Month	$\text{DepMonth}_s = MONTH(STD)_s$	CHARACTER
Arrival Day	$\text{ArrDay}_s = DAY(STA)_s$	CHARACTER
Arrival Week	$\text{ArrWeek}_s = WEEKNUMBER(STA)_s$	CHARACTER
Arrival Month	$\text{ArrMonth}_s = MONTH(STA)_s$	CHARACTER
Transit	$T_s = STA_s - STD_s$	INTEGER
Region	$Region_{s=} \in \{Africa, Asia, Middle\ East, North\ America, South\ America$	CHARACTER
Hurricane	Yes / No	LOGICAL

In this section, we are going to explore to what extent the different variables support our hypothesis, that it, are of predictive value with regard to the order's arrival time. But for a proper prediction model, we first need a response (target) variable. We determine the earlier introduced Delta to be the response variable as we want to know to what extent we can predict the deviation in arrival time. This deviation is implied in the Delta: the more this value is away from zero, the more deviation between the scheduled and actual arrival time (also denoted by 'anomalous ETA').

In the following section, we will further elaborate on the behaviour of the Delta. Subsequently, we will go through the other attributes to hypothesize which ones can explain and possibly predict this anomalous ETA.

5.3.1 Target variable Delta

We first analysed the Delta by plotting all values in histogram, which can be found in Appendix F. When looking at the distribution of Delta using this histogram, almost all records are in the range of [-10, 10]. However, there are some records in the range [-30, -10]. This also counts for positive Delta's from [10, 51]. We also performed an outlier test to check whether these values (outside the range of [-10, 10]) are indeed outliers. The result, which also can be found in Appendix F, gives us a reason to assume that all values outside the aforementioned range are considered to be outliers.

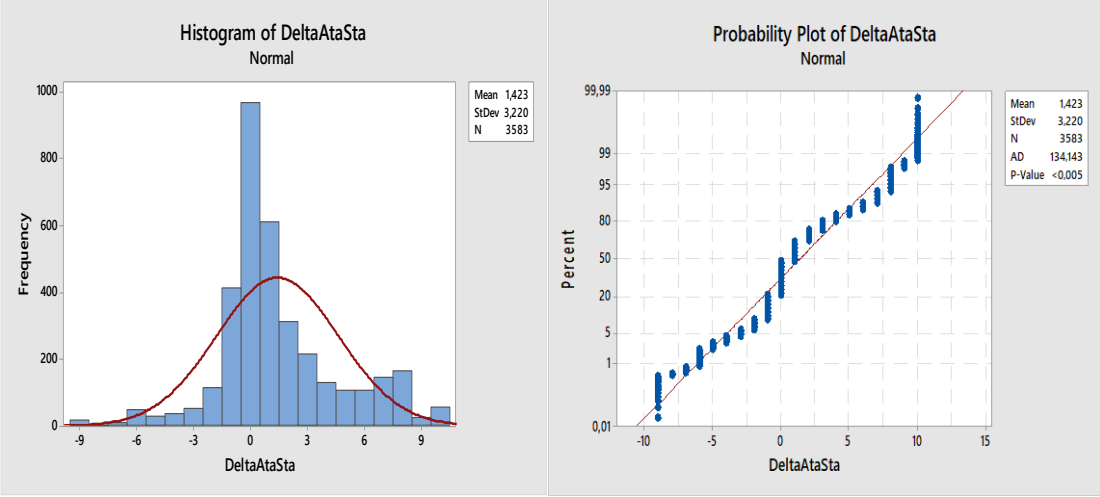


Figure 5-1: Histogram and probability plot of target variable Delta

However this seems a radical deletion, it is not. We can justify this by the fact that shipments that we removed because their delta falls beyond this range, only represents 0.7% of the dataset.

5.3.2 The carrier

When we plot the number of shipments each carrier has performed against the amount of delayed and on-time orders in figure 5-2, there are two remarkable observations. First, the distribution of trade volume between the carriers is skewed: there is a large difference between the amount of orders performed by each carrier. Second, the skewness also reveals the relatively large number of carriers that only performed a couple of orders (some carriers only performed one).

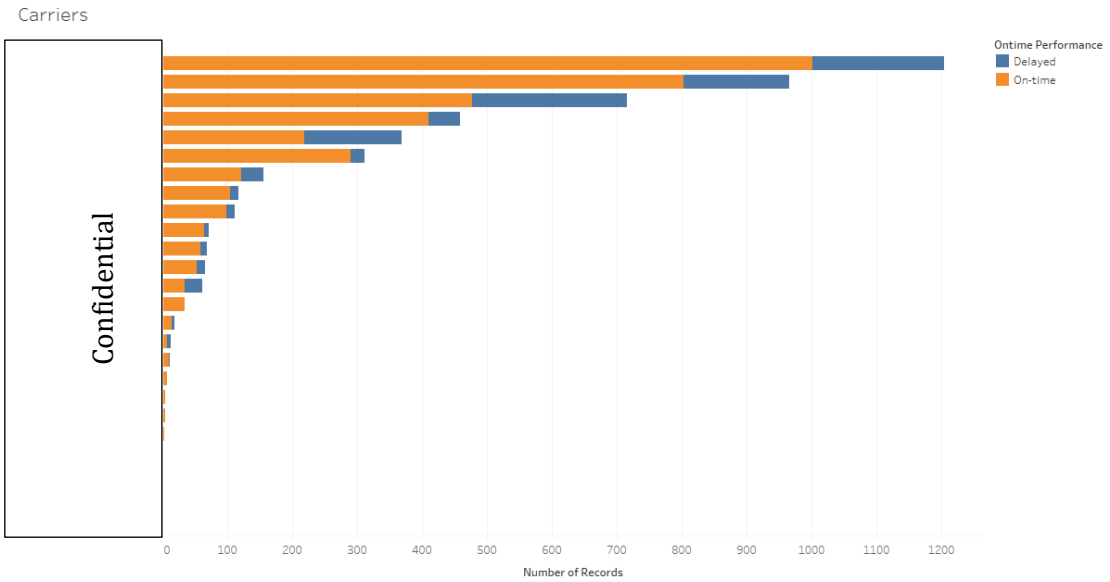


Figure 5-2: Carrier's trade volume, broken down to their on-time performance

The skewed distribution of trade volumes between the carriers, has as disadvantage that small classes are really hard to make predictions from. We therefore chose to exclude these 'rarely occurring' carriers from the dataset. The number of carriers included is reduced from 26 to 10: the carriers with the biggest trade volume remain. This maybe feels like a rough exclusion, but the 10 biggest carriers still represent 95% of total trade volume. It is for that reason that we chose this 95% as split point. Moreover, we justify this exclusion by the fact that if the carrier did not perform that much shipments historically, there is no reason to believe that it will perform much shipments in the future. And if so, the effect of not being able to predict the arrival time would not be that big, simply because there is good chance that it then only affects one order.

The graph also reveals that the amount of delayed orders seem to differ between the carriers. But no doubt that a carrier with much shipments overall also has more shipments not arriving on time. For this reason, we computed ratios of on-time performance to get a more absolute judgement. The ratio of each carrier *c* is calculated as follows:

$$Ratio\ (c) = \frac{\sum_{s=1}^{S} OTP(\Delta_{s,c})}{\sum_{s=1}^{S} S} \qquad \forall\, c \in 1, \dots, C \qquad (5.6)$$

It is the sum of the on time shipments performed by carrier c, divided by the total amount of shipments performed by carrier *c.* The higher the ratio, the better the carrier performs. The results are listed in Table 5-4.

Table 5-4: Carrier's ratio of on-time performance

Carrier	On-time	Total	Ratio	Diff	LB	UB	p-value
Confidential	506	716	0.706	0.10	0.03	0.17	0.005*
	99	112	0.884	0.07	-0.16	0.01	0.081
	412	458	0.900	0.09	-0.16	-0.02	0.008*
	1045	1204	0.868	0.06	-0.12	0.00	0.068
	69	72	0.958	0.15	-0.22	-0.07	0.000*
	126	156	0.808	NA	NA	NA	NA
	113	117	0.966	0.16	-0.23	-0.09	0.000*
	867	966	0.898	0.09	-0.15	-0.03	0.007*
	292	311	0.939	0.13	-0.20	-0.06	0.000*
	244	368	0.663	0.15	0.07	0.22	0.000*

[1] All carriers' proportions are compared to X.

But more important in this case, is the difference between the ratios: significant difference between the proportions means that it does matter which carrier executed the shipment and the carrier is thus a possibly explanatory variable. To investigate this, we included the 95% confidence interval of the difference between two proportions based on Altman, Machin, Bryant and Gardner (2003). In this table we chose X to be the reference group: we compared every carrier with X in the table. We chose this carrier, because the on-time performance of this carrier (0.808) most closely matches the total on-time performance of the LSP, thus reflects reality the most.

From the confidence intervals, we can conclude that 7 of the 9 carriers differ from the reference carrier X, as the zero does not lie within the lower- and upper bound of the confidence interval. For the other two, Y and Z, counts that the on-time performance compared to the reference carrier not differs. If the p-value is lower than or equal to our significance level of .005, we reject the null hypothesis of the two ratios being equal. Based on the p-values, we have proven that at 7 of the 9 carriers, the difference in ratios is significant at a 95% confidence level. That the on-time performance significantly differs that often, gives us reason to believe that the carrier is of influence for the deviation in arrival time.

5.3.3 Ship

According to the theory of Shmueli and Koppius (2011), who define two constraints in case of choosing variable as input of a prediction model, we cannot include the attribute ship in our prediction model, because this attribute is not available at time of prediction.

5.3.4 Port of Loading

As we have already established in Section 4.2, the LSP ships from only two ports: Antwerp and Rotterdam. When looking back at Figure 2-7 in the global export analysis, one can intuitively conclude that the loading port will not be a very good predictor because almost all orders are sent from the port of Rotterdam (and only a few from the port of Antwerp). We will use the variable as input for our feature selection model, but our hypothesis is that the variable will be of no influence of the deviation in arrival time.

5.3.5 Port of Delivery

When computing the same graph as we did for the carriers, but now for the delivery ports available in the historical order dataset, at total of 99 ports are counted. Since this graph

takes up a whole page, it can be found in Appendix H. The graph reveals the same pattern as the one found in the graph of carrier trade volumes: a large difference between the amount of orders shipped to each delivery port and a relatively large number of delivery ports where only a couple of orders are shipped to. It is not remarkable that the same pattern is found, since exclusively one carrier is contracted to a specific delivery port per quarter, meaning that a certain one to one relation exist between carrier and delivery port. In case of delivery ports, the skewed distribution has the same disadvantage that small classes are hard to make predictions from. We therefore reduce the number of delivery ports from 99 to 28. This again sounds like a rough exclusion, but the busiest 28 delivery ports are still responsible for 80% of total trade volume.

We could perform the same analysis as we did for the carriers, which is making an overview of the on-time performance per port and test if the difference between ports is significant. With this analysis we could make our hypothesis about ports potentially being good predictors of the arrival time of an order. However, as there is one carrier contracted on a specific port for a certain period, the analysis would not add much new insights. We therefore chose not to perform the analysis here.

5.3.6 The period in time

When focusing on the trade volumes over the year for several time dimensions, a clear difference becomes visible when splitting the period in weeks of the year. When focusing on the number of shipments executed at each week of the year, broken down to on-time and delayed orders, there is a difference visible in the number of shipments that arrive either on time or too late. There is a clear difference visible in trade volumes between the different weeks of the year. This argues for the hypothesis that the departure week is of influence on the anomalous ETA.

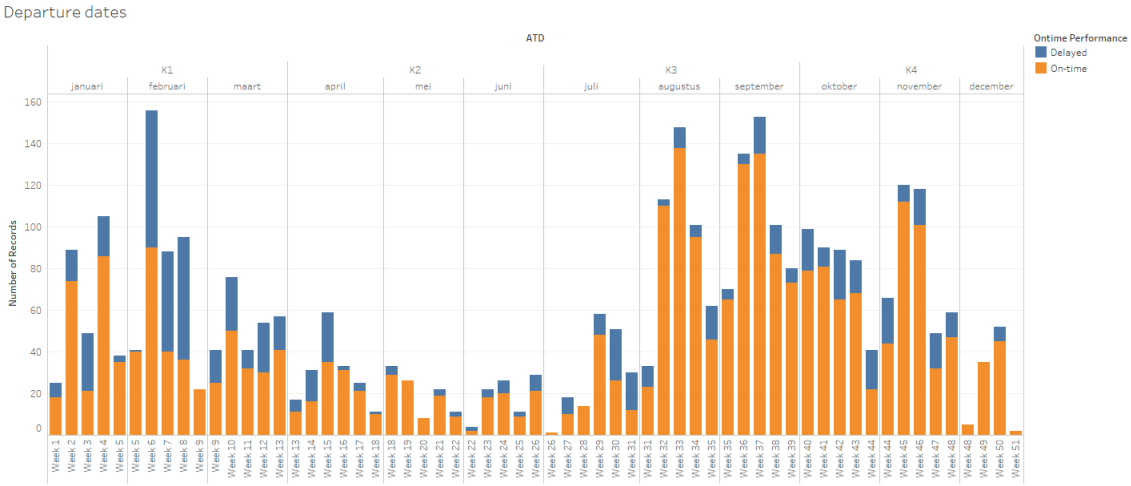


Figure 5-3: Departed orders per week of the year, broken down to their on-time performance

The same holds for the week of arrival: a pattern appears to be visible in which the arrival week is of influence on the anomalous ETA.

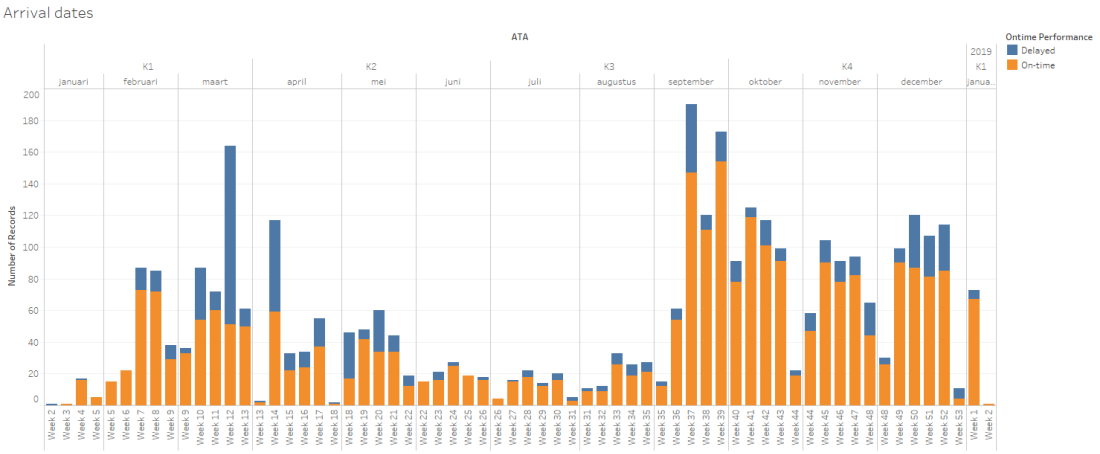


Figure 5-4: Arrived orders per week of the year, broken down to their on-time performance

This arouses our interest to further explore the effect of the departure- and arrival week on the deviation in estimated versus actual arrival time. To further explore this appearance, we calculated the average Delta of all shipments departing in week w ∈ 1,…,52. We did the same for all shipments arriving in week w ∈ 1,…,52 for the years y ∈ 2016,…,2019. Not for all weeks all the four years are included, as we do not have a full four years on historical order data. The table in Appendix J shows which months and weeks are included from each year, broken down to departure- and arrival week. In the figures below, the week factor's distribution is visualized. The blue shading around the orange dot represents the 95% confidence interval. The grey constant line (where Delta equals zero) indicates there is no difference in the scheduled and actual arrival time.

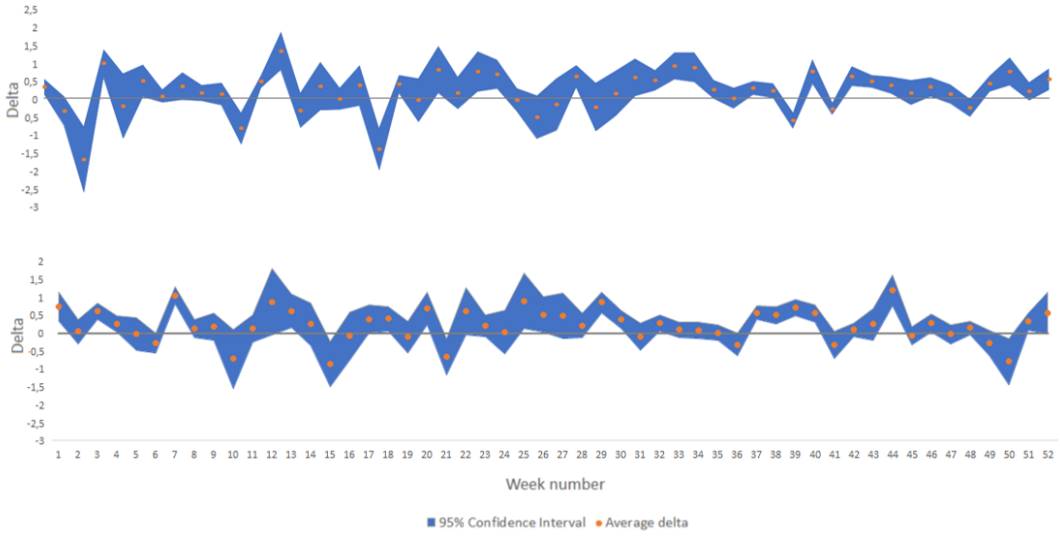


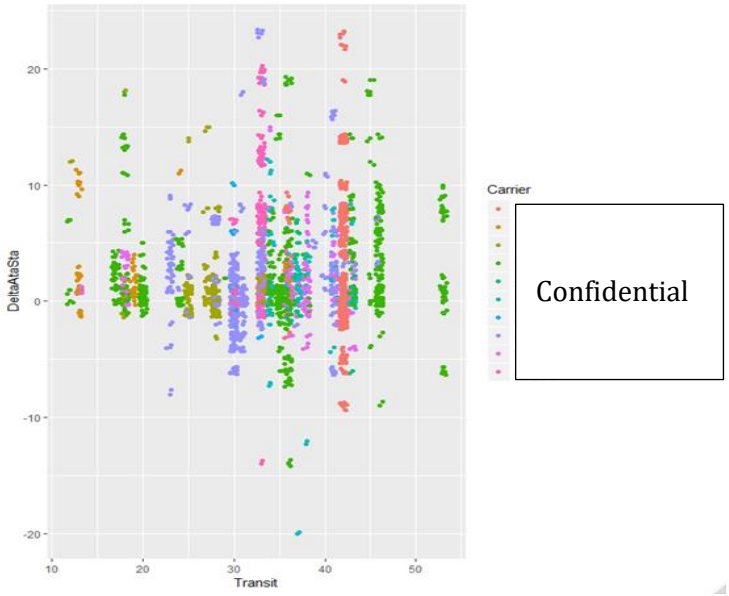
Figure 5-5: Average delta per arrival- (upper) and departure week

In the upper graph with average Delta's per *arriving* week and in the lower graph with average Delta's per *departure* week w ∈ 1,…,52, we see a similar pattern: in the lower graph for example, one can count that in 21 of the 52 weeks, the Delta is significantly above or below the zero point, meaning that there is evidence to believe that the departure week is of influence on the Delta. We did the same for the departure- and arrival day and the departure and- arrival month. These graphs can be found in Appendix I, but also gave evidence for the period in time being of influence on the deviation in arrival time.

5.3.7 The transit time

When looking at the scatter plot of the distribution of transit times against the average Delta, this variable seems to be a good predictor for the Delta. Supported intuitively but also reflected in the scatter plot, the average Delta becomes higher when the transit time is longer. We also added the carrier component by colouring the dots per carrier who executed the order. We clearly see a pattern in which the vertical lines have the same colour, implying that a specific carrier often ships to a limited set of (the same) delivery ports. This can be concluded, because a vertical line means that multiple orders are shipped to a port with the same transit time, thus assuming the ports are in the same region. If this vertical line is than of one uniform colour, only one carrier was responsible for the transport to this (set of) ports.

Figure 5-6: Scatter plot of transit time and delta, dots coloured per carrier



5.4 FEATURE SELECTION

In this section, we are going to apply the feature selection method as described in the literature review. This section answers the research question: *Which attributes are included in the final model?* We now have the following attributes available as input for our model: Carrier, PoL, PoD, ArrDay, ArrWeek, ArrMonth, DepDay, DepWeek, DepMonth, Transit, Region and Hurricane.

As we have already discussed in the literature review, more features (attributes) as input for your model increases dimensionality which makes training and testing methods more difficult. So, more attributes do not necessarily lead to a better prediction model: a reduced set of attributes can prevent the model from including noise, which positively affects the accuracy of the model. We have performed several feature selection methods with different algorithms, to check which subset of attributes leads to a model with the highest predictive power. Besides the random forest algorithm, we decided to add two other tree-based algorithms for completeness. Of each algorithm, we ran the three different search methods forward, backward and bi-directional. Table 5-5 shows the results of the experiments. In Appendix K, the selected attributes per type of algorithm and search method are mentioned.

Table 5-5: Results of feature selection experiments with different algorithms and search methods

Algorithm	*Search method*	*Evaluated subsets*	*# Selected attributes*	*Accuracy*
RandomForest	Forward	212	6	0.704
RandomForest	Backward	116	7	0.704
RandomForest	Bi-directional	267	**7**	**0.712**
RandomTree	Forward	292	11	0.698
RandomTree	Backward	235	9	0.700
RandomTree	Bi-directional	261	12	0.696
REPTree	Forward	208	8	0.709
REPTree	Backward	195	7	0.709
REPTree	Bi-directional	231	7	0.709

The one with the highest accuracy is the RandomForest algorithm with bi-directional search method. The model selected seven attributes as input and scored an accuracy 71.2%. However, one can notice that the accuracies of the models with different attribute subsets are all very close to each other. We are therefore going to perform an additional test on the selected seven attributes (Carrier, PoD, ArrDay, ArrWeek, DepDay, DepWeek and Transit)to prevent choosing a suboptimal attribute subset. We did this by fitting a basic linear regression .0.0.model and evaluated the model's analysis of variance in Figure 5-7.

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
3,61835	37,76%	35,13%	31,30%

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Regression	150	14279,4	95,196	14,30	0,000
Transit	1	8,1	8,131	1,22	**0,269**
PoD	26	1730,6	66,563	10,00	0,000
ArrDay	6	464,3	77,381	11,63	0,000
ArrWeek	51	2174,1	42,630	6,40	0,000
DepDay	6	846,0	141,004	21,18	0,000
DepWeek	51	2812,0	55,136	8,28	0,000
Carrier	9	269,6	29,960	4,50	0,000

Figure 5-7: Model summary and analysis of variance of linear regression

To determine whether the individual attributes explain the variation in the target variable, we check the p-values. Low p-values indicate significance (which is a good thing) and high p-values show that the variable is not a significant contribution to the prediction. When looking at the table, we see that the transit time does not seem to be a good predictor with an insignificant p-value of 0.269. We decide to exclude the attribute Transit from the model.

5.4.1 Correlation between model variables

Now we almost have a causal model, it is essential to address the extent to which explanatory variables correlate with each other. This occurrence is something to be referred to as multicollinearity (Walpole, 2016), and causes misinterpretations of the model. When two explanatory variables highly correlate with each other, it is possible that the variables are explaining each other rather than the target variable. If his happens, a high predictive performance can be achieved without really addressing the target variable. In Table 5-6, we listed the correlation coefficients by their Lambda. Lambda is a measure of association that reflects that proportional reduction in error when values of one variable are used to predict the other (James et al., 2013). Lambda is suitable when working with nominal variables, which is our case. The value ranges from 0 to 1 and gives in indication of the strength of the relationship between two variables: 0 indicates that there is nothing to be gained to include the variable to predict the other, 1 indicates that the variables perfectly predict each other.

Table 5-6: Lambda's measure of association between explanatory variables

Explanatory variables	*Lambda*	*Measure*[1]	*P-value*
Carrier * PoD	0.545	Moderate	0.000*
Carrier * ArrWeek	0.078	Little	0.000*
Carrier * DepWeek	0.072	Little	0.000*
Carrier * ArrDay	0.074	Little	0.000*
Carrier * DepDay	0.103	Little	0.000*
PoD * ArrWeek	0.075	Little	0.000*
PoD * DepWeek	0.070	Little	0.000*
PoD * ArrDay	0.029	Little	0.000*
PoD * DepDay	0.051	Little	0.000*
ArrWeek * DepWeek	0.462	Moderate	0.000*
ArrDay * DepDay	0.050	Little	0.000*
ArrWeek * ArrDay	0.025	Little	0.000*
DepWeek * DepDay	0.031	Little	0.000*
DepWeek * ArrDay	0.035	Little	0.000*
ArrWeek * DepDay	0.073	Little	0.000*

[1] The size of the Lambda is interpreted as follows: .00 to .19 "little to no relationship" .20 to .39 "weak relationship" .40 to .59 "moderate relationship" .60 to 1.00 "strong relationship" (Goodman & Kruskal, 1979).

As the table shows, the two sets of explanatory variables *Carrier * PoD* and *ArrWeek * DepWeek* moderately correlate with each other, which can be explained intuitively. It makes sense that the Carrier correlates with the PoD, as for each quarter just one carrier is contracted to ship to a specific destination port. Thus, the same combination of a carrier shipping to a specific delivery port, occurs often. The same counts for the ArrWeek that correlates with the DepWeek. Since each shipment is in transit for approximately 35 days, there is always roughly the same amount of days between the ArrWeek and the DepWeek. Now the question arises to what extent the two correlated explanatory variables can form a threat for the performance of our prediction model. Because if so, we may need to change the causal model by excluding correlated explanatory variables. However, the applied technique to predict the target variable is of great influence on the extent to which multicollinearity forms a threat for model performance. As Random forests can handle some multicollinearity in explanatory variables, we conclude that we do not have to change our causal model.

5.5 CONCLUSION

In this chapter, we prepared a dataset as input for predicting the Delta. We first elaborated potentially predictive variables that we derived from original variables, which are several periods in time, the transit time, the region and the hurricane season. Also our response variable, the Delta, is a derived attribute which we have discussed extensively. Subsequently, we discussed the attributes that were available from the original dataset. We manipulated some variables if we were convinced of the advantages of doing so, for example we determined all delta's smaller than -10 and bigger than 10 to be outliers and therefore removed them from the dataset. We did the same for carriers and ports with low trade volumes. By doing this, we aim to be better capable of making accurate predictions. We ended this chapter with the feature selection process in which we selected the optimal subset of features to be used as input of our prediction model. We therefore executed multiple experiments with varying algorithms and search methods. All tests were evaluated by their R^2, from which one best model is chosen with seven attributes. Because the results were very close to each other, we performed an extra test to check if these seven attributes were actually a good subset of predictors: by checking the p-values we decided to exclude one more attribute from the model.

The final model for predicting the response variable Delta consists of six predictor variables, namely the departure day (DepDay), the departure week (DepWeek), the arrival day (ArrDay), the arrival week (ArrWeek), the Carrier (Carrier) and the port of delivery (PoD). Together they have the goal of predicting the target variable Delta. Note that all predictor variables are categorical while the target variable is numerical.

6 MODELLING

Now we have a causal model, we are able to actually build our prediction algorithm. We load our dataset, that is prepared and manipulated using MySQL, into R statistical software from which we are going to build our prediction model. Section 6.1 discusses an extra test to prove the fitness of our choice for working with random forests. We follow some machine learning fundamentals and discuss our cross validated experimental design, parameter tuning and the performance of the final model. In this chapter, the following question is answered: *What are the characteristics of the prediction model?*

6.1 THE MODELLING TECHNIQUE

Because our research is not about comparing the performance of several machine learning techniques but about being able to predict the deviation in arrival time estimation as accurate as possible, we think it is more valuable to execute one algorithm in detail rather than deploying multiple ones superficially. From the literature review, we already concluded that Random Forest would be the most appropriate modelling technique. To be even more certain of applying this technique and thus excluding other possibly appropriate techniques, we performed an extra test. This test uses a tool (RapidMiner) that is able to automatically perform multiple algorithms on one dataset and compare them based on their performance and run time. After selecting the task (which is prediction for supervised learning, since all data has a known outcome), the platform automatically suggests relevant machine learning techniques based on characteristics of the input data and the task. Based on our input, RapidMiner suggested the following machine learning techniques: Generalized Linear Model, Deep Learning (based on multi-layer ANN), Decision Tree, Random Forest and Support Vector Machine. Note that these machine learning techniques highly correspond with the models that we discussed earlier in our literature study: Generalized Linear Model is an extension of the linear regression that we elaborated in the literature review in Chapter 3 and Deep Learning (based on a multi-layer ANN) is a specific type of Neural Networks that we discussed in the same chapter. Also the Random Forest and Support Vector Machine are suggested, which we elaborated earlier. Because during our literature review we already discussed the relevance of these machine learning algorithms, we chose to run the test on these. Table 6-1 shows the results of comparing these machine learning techniques.

Table 6-1: Results of comparing multiple machine learning techniques

ML technique	*RMSE*	*Sigma (σ)*	*Runtime*
Generalized Linear Model	3.562	0.128	5 sec
Deep Learning (ANN)	2.965	0.255	2 min 13 sec
Decision Tree	2.871	0.242	3 sec
Random Forest	2.420	0.222	2 min 40 sec
Support Vector Machine	3.377	0.155	5 min 38 sec

This experiment, that compares the performance of five machine learning techniques on their error and runtime, shows that no technique outperforms the others on both criteria (error and runtime). When taking both measures into account simultaneously, both Decision Tree and Random Forest are good candidates: the Decision Tree turns out to be a good technique since it has the lowest runtime and the second best error rate. The

Random Forest algorithm is the one with the lowest error, but has second highest runtime. Because we think that in this business case, the lowest possible prediction error is more important than a low runtime, we choose Random Forest to be the best modelling technique. This experiment supports our choice to apply Random Forest, which we suggested earlier as a result of extensive literature research. An additional benefit of this technique is that it is capable of dealing with correlated predictor variables.

6.2 GENERATION OF EXPERIMENTAL DESIGN

Although we have already claimed that random forests are robust to overfitting, it does not mean that random forest cannot overfit. Overfitting is a significant problem in machine learning as the machine learning model's ultimate goal – to be able to predict unseen data – fails due to fitting too well on the train data. The most frequently used technique to prevent a machine learning algorithm from overfitting, is k-fold cross validation. Using this technique, you split the data into k subsets (also referred to as folds). You train the model on all but one (k-1) of the subsets and then evaluate the model on the 'unseen' complementary (k) subset. In choosing the right value of k again requires a bias-variance trade-off: the conflict of simultaneously minimizing those two values – but this prevents the model from generalizing beyond the training set. Given these considerations, the most frequently used values are k = 5 or k = 10. This because these values have been shown empirically to yield test error rates that suffer neither from excessively high bias nor from very high variance (James et al., 2013). Because our dataset is large enough and we do not want the running time to be too long, we choose the value of k to be 5 and thus apply a 5-fold cross validation design. See Appendix L for a visual representation of how the dataset is split.

6.3 BUILD MODEL

For building the model, we use the aforementioned 5-fold cross validation test design. Figure 6-1 shows the pseudo code of the logic applied to generate the test design. We first shuffled our data to assure randomness and then divided our dataset in 5 equally sized slices (folds). This is done by assigning the first $1/5^{th}$ part of the data to 'fold 1', assigning the second $2/5^{th}$ part of the data to 'fold 2', etcetera. So each fold consists of 1/5 of the data, in which there is no overlap, meaning that each data point is used once and in only one of the 5 folds. Now, we loop through the folds and iteratively select one of the folds to be separated as test set. The remaining part of the data (the other 4 folds) is used for training the model. From this train set, we calculate the model performance. After 5 iterations, all the folds are separated for testing. In the end, we calculate an overall performance measure by summing all the individual results and dividing it by the number of folds. The pseudo code of this logic is found below; the R code belonging to this logic is put in Appendix M.

Pseudo code Cross Validation Algorithm

Read all input data $\boldsymbol{x_i}$ **(*i* = 1,2,….*n*)**
Shuffle input data
Initialize desired number of **folds (*k* = 1,2,.. *K*)**
Initialize size of each **subset** by: $\sum_{i=1}^{n} \boldsymbol{x_i}/\mathbf{K}$
Initialize **count** = 0
For j = 1:*K all k folds*
    Test data = (*subset*(j-1)+1) till (subset*j)* ← rows from shuffled input data
    Train data = *all data* ($\sum_{i=1}^{n} x_i$) − *test data*
        Fit model on train data
        Print model performance of j^{th} fold
        Count += model performance *j-1*th fold
End for
Calculate average model performance = *count / K*

Figure 6-1: Pseudo code of cross validation algorithm created in R statistical software (created by author)

6.3.1 Parameter tuning

The Random Forest algorithm is built on several parameters. Extensive discussion exists in literature about the influence of different parameters on the performance of a random forest algorithm, but the results are conflicting. We decided to follow the results of Strobl et al. (2008) as they worked with correlated predictor variables – which is also the case in our study – and found that in the case of correlated predictors, different values of mtry and ntree should be considered.

Mainly, the forest can be tuned on the number of trees the forest consists of (denoted by `ntree`) and the number of variables p to try as candidates for splitting in an individual tree (denoted by `mtry`). By default (following Breiman, 2001; Hastie et al., 2009; the R package `randomForest 4.6-14`), ntree set to 500 and mtry is set to $\lfloor p/3 \rfloor$, which in our case is $\lfloor 6/3 \rfloor = 2$. The node size is set to 5. However, the best combination of values in practice is highly case-dependent. The R package `randomForest` only has a built-in function to tune the mtry parameter; the ntree parameter cannot be tuned within this package. And if it was, it is not possible to tune different combinations of multiple parameters simultaneously using built-in packages. We therefore manually set up an

experiment to determine the optimal values of ntree and mtry. For ntree we tried 4 values: 250, 500, 750 and 1000. For mtry we tried all possible number of variables, from 1 (only including one random variable as candidate for splitting) to 6 (including all variables as candidate for splitting). We ran 4x6=24 experiments, each one with a different combination of parameter settings. We ran the experiments using a 10-fold cross validation. The R code for the experimental setup including detailed results are listed in Appendix N.

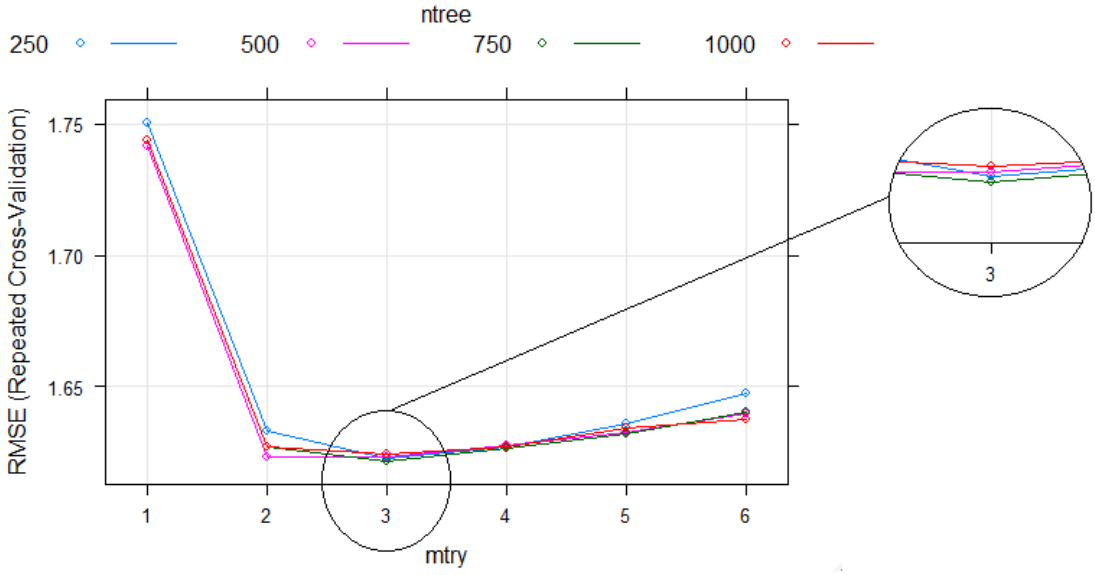


Figure 6-2: Results of parameter tuning with different values of mtry and ntree

As the graph in Figure 6-2 shows, the RMSE differs depending on the values of mtry and ntree. The RMSE is much lower when mtry is 2 and lowest for mtry is 3, and becomes high again from values bigger than 3. This can be explained intuitively: when there are more variables available to split on at each node, the model has more space to choose less important variables to split on which would increase the error. We choose `mtry=3` as first parameter setting as this value yields the lowest error. Furthermore, we see that the line from ntree = 750 is lowest at this value of mtry. In favour of the model's RMSE, which we prefer to be as low as possible, we choose `ntree=750` as second parameter setting.

6.3.2 Variable importance

The last step to get more insight in our model, is looking at the individual explanatory variables in terms of how well each variable contributes to the prediction of the Delta. We executed the so-called variable importance test, which is a built-in function in R from the `randomForest` package. It uses the percentage of increase in the model's Mean Squared Error (denoted as %IncMSE) as measure to scale the importance of the variables. The logic behind this measure is as follows: if an explanatory variable is important for the model in terms of predictive power, then randomly assigning other values than the real values for that variable will negatively influence model performance. You can then measure the MSE of the model in which that one variable is manipulated (and the other variables remain unchanged). Next to that, you take the MSE from the original dataset, where the original value is used for the given variable. Now the difference between the two models can be calculated based on their MSE. Intuitively, one expects the MSE from the original dataset to be smaller since feature selection already determined that this

subset of variables is most optimal in terms of predictive performance. The bigger the difference, the more important that explanatory variable is. Thus the higher the percentage of increase in MSE if the explanatory variables is *not* available in the model, the more important is that variable in terms of predictive performance. This logic is then applied for all the explanatory variables in the dataset. Figure 6-3 shows the variable importance graph of our dataset. The arrival week seems to be the most important variable with 188.44% increase in MSE if this variable was not included in the prediction model. Meaning that the MSE would increase with a factor of almost 2 when not including the arrival week as explanatory variable. The carrier seems to be the least important variable, but still has an increase in MSE of almost 54%. In other words, when a model has an MSE of 2 (see Table 6-2 for our model's MSE), this error can increase to 2.5 if the variable Carrier was *not* included in the model, which still is a significant difference.

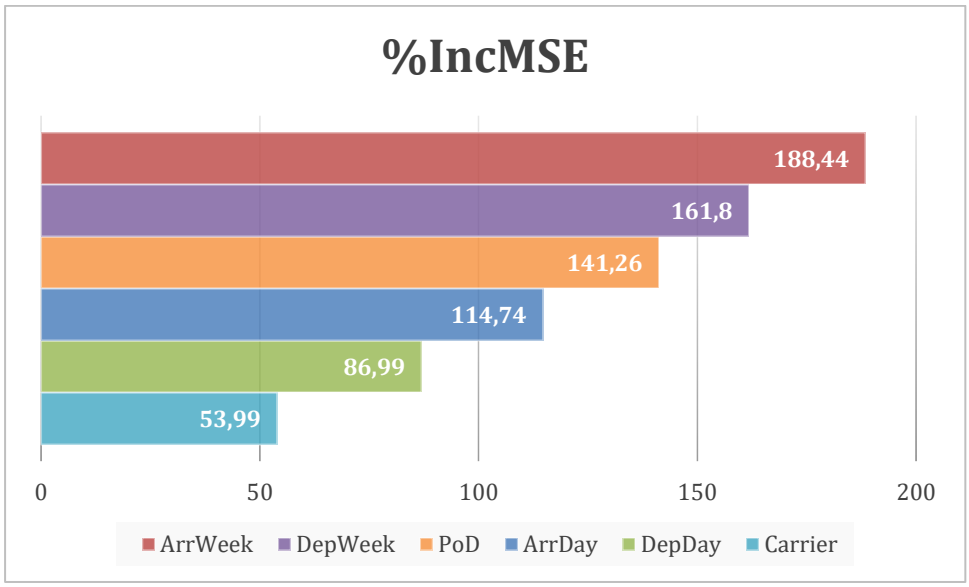


Figure 6-3: Variable importance measured by the percentual increase in MSE

6.4 ASSESS MODEL

In this section, we will fit the Random Forest algorithm on the five different training sets with the parameter settings as determined in the previous section. While training the machine learning model, one of the main goals is typically to prevent the model from overfitting. As discussed earlier, random forests are naturally robust to overfitting as they benefit from the bootstrapped aggregation. To be even more sure of this, we set up an experimental design using 5-fold cross validation. Two results are important to address now: the average performance of the model, which we evaluate based on the performance measures that we discussed in the literature study, and the extent to which the model tends to overfit. We first discuss the model performance measures that are listed in Table 6-2.

Table 6-2: Results per fold and on average of training the model

Fold	*MSE*	*RMSE*	*R*	*R^2*	*Adjusted R^2*
1	1.973	1.405	0.895	0.800	0.801
2	1.858	1.363	0.902	0.812	0.809
3	1.906	1.380	0.896	0.802	0.800
4	2.036	1.427	0.901	0.810	0.802
5	1.866	1.367	0.914	0.816	0.810
Average	**1.928**	**1.389**	**0.899**	**0.808**	**0.804**

The average RMSE of the model training is 1.4. We evaluate the model by discussing the RMSE here, as this measure is most interpretable because it is measured in the same units as our target variable Delta. The value shows that the algorithm is capable of predicting the target variable quite good, with an average error of only 1.4 from the Delta who can take values ranging from [-10,10].

A value of R^2 of 0.80 tells us that our explanatory variables can account for 80% of the variation in the Delta. In other words, if we are trying to explain why one shipment deviates more from the STA than others, we can look at the variation in the explanatory variables, which can together explain 80% of the variation. An adjusted R^2 gives us some idea of how well our model generalizes, and ideally we would like its value to be the same, or very close to the value of R^2. Taking a look at our results, the adjusted R^2 is very close to the R^2 (0.804 and 0.808 respectively) which is a sign of a good model that is able to generalize, and thus, does not tend to overfit.

As stated in Section 3.4, it is important to address the extent to which a model is overfitting (or even underfitting) as this is the main goal of the training phase. From the measures as discussed above one is quickly inclined to assume the model performs good, but if these results are not generalizable to unseen test datasets, the model is still worthless. As only comparing the R^2 with the adjusted R^2 is a little premature for evaluating a model on overfitting, it is necessary to take a look at a visual representation of the data. The behaviour of the actual against the predicted values in the train set can be derived from this. The results of one of the folds is shown in Figure 6-3. In Appendix O, one can find the other four plots. Since the arrival week turned out to be the most important variable, we added an extra dimension to the plots by colouring the dots based on the arrival week of the corresponding order. As the dots in the graph show, no patterns appear to be visible with regard to the error of the prediction and the arrival week.

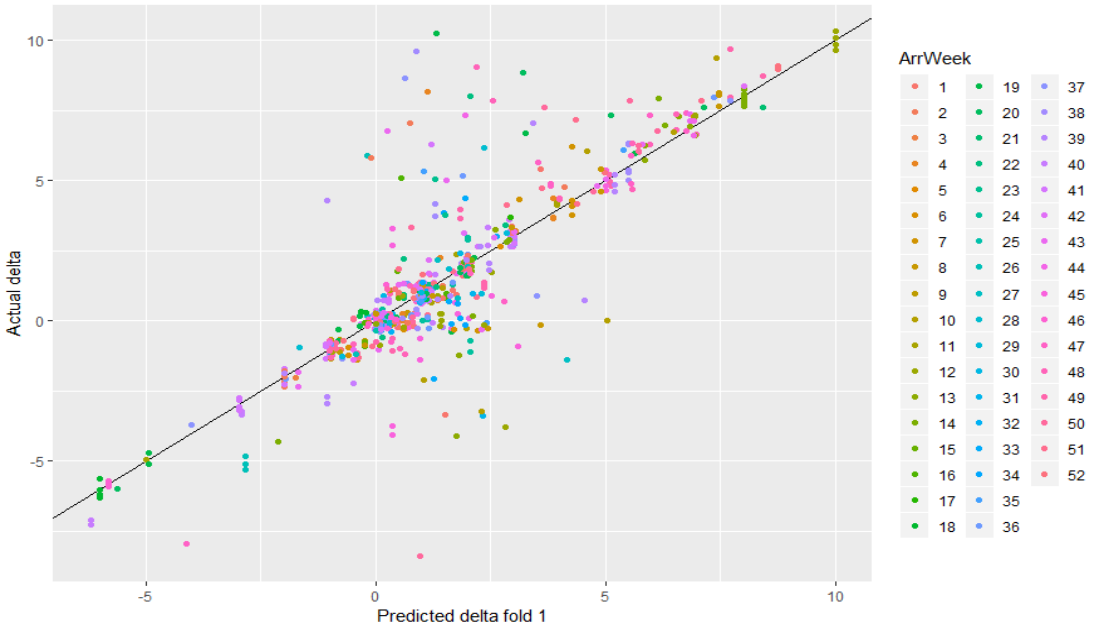


Figure 6-4: Scatter plot of predicted and actual Delta

Each dot represents a value, where the actual values are plotted on the y-axis and the corresponding predicted value is plotted on the x-axis. The line that is drawn through the dots, is the regression line and is determined by the lowest total sum of squared predictions: the closer the dots are to the line, the better the predictions. In an underfitted model, the individual dots would be far from the regression line; in an overfitted model, the dots would perfectly match the regression line. Both occurrences are not happening here. Also in the other four graphs, which are displayed in Appendix O these patterns are not present. From this we can conclude that our model is sufficient as it has good predictive power with an average RMSE of 1.389, and it does not tend to either overfit or underfit. We are now able to test our model on the test set.

6.5 VALIDATE MODEL ON TEST SET

We will now validate the performance of the model that we trained in the previous section. When we look at the quantitative performance measures in Table 6-3, the model seems to have good performance. The error is a bit higher than in the train phase (RMSE of 1.43 and 1.389, respectively) which can be expected because the model is now exposed to new data it has never seen before.

Table 6-3: Results per fold and on average of the test set

Fold	*MSE*	*RMSE*	*R*	*R^2*
1	2.145	1.465	0.877	0.769
2	2.389	1.546	0.899	0.808
3	2.084	1.444	0.894	0.800
4	1.877	1.370	0.896	0.802
5	1.764	1.328	0.904	0.817
Average	**2.052**	**1.430**	**0.894**	**0.800**

We visualize the model's predictive performance by plotting the actual and the predicted values in one graph. Figure 6-5 shows the first 142 orders of one fold of the test set. For visibility, we chose to only expose a part of the test dataset in a graph, but in Appendix Pone can find the remaining graphs of all test data.

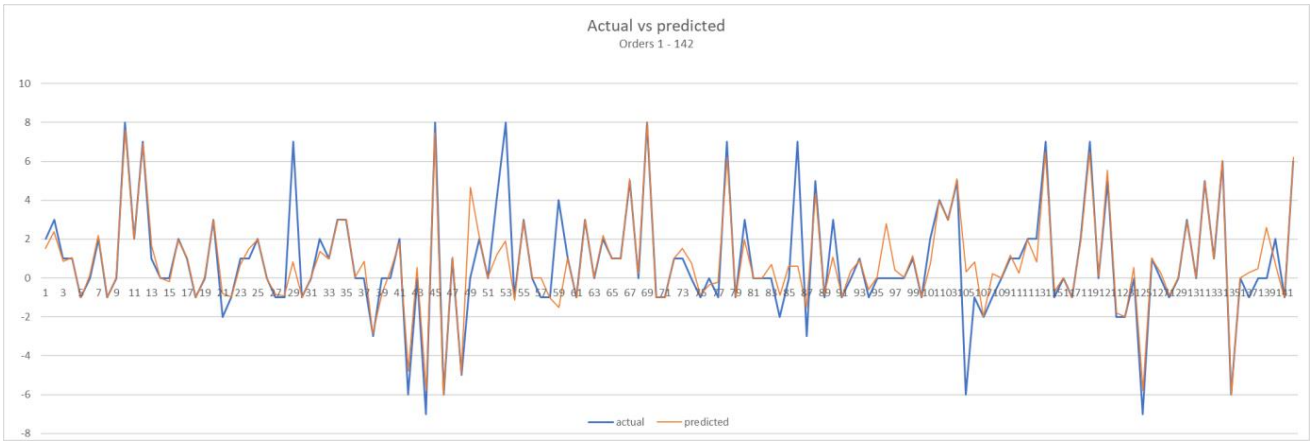


Figure 6-5: Actual and predicted values plotted for the first 142 data points

From the graph it becomes clear that the model is capable of making accurate predictions. For example, a relatively high deviation of 8 days (in the 10th record) is almost exactly predicted by the model. What becomes interesting, are the orders from which our model was *not* able to predict the deviation accurately. We therefore analysed the graphs and collected the orders whose predicted value visibly deviated from the actual value. We will call these orders 'outliers'. So from the part of the graph that is shown in Figure 6-5, the 29th, 53rd, 59th, 86th, 96th,105th, 106th and 139th order are labelled as outliers. We manually labelled all outliers from this fold of the test set and collected them in a separate dataset.

In total, 43 of the 717 shipments are classified as being an outlier. It is interesting to further explore the behaviour of this so-called 'outliers' dataset as all these orders appeared to be difficult to predict. In the next section, we will analyse this dataset (denoted by 'the outliers dataset') that exists of the shipments from which our model was not able of accurately predict the Delta.

6.5.1 The outliers dataset

The outlier dataset thus consists of 43 shipments. Ideally, one wants to find any pattern in this dataset of outliers to draw conclusions from. One obvious finding is that especially the higher Delta's are included in the outliers dataset, concluding that the model apparently finds it harder to predict a shipment with a bigger Delta. This can be visually represented by the histogram and boxplot in Figure 6-6. There are few shipments in the outliers dataset with a Delta around zero: more shipments have a Delta of 4 or more. Especially shipments with a Delta around 8 are popular in the outliers dataset. However, because the dataset consists of only 43 orders, we suspect that this occurrence is based on coincidence. As the boxplot shows, also the standard deviation of the dataset is high (4.7). This can be explained intuitively by the fact that shipments in these dataset are the outliers and thus do not behave in a perfect manner.

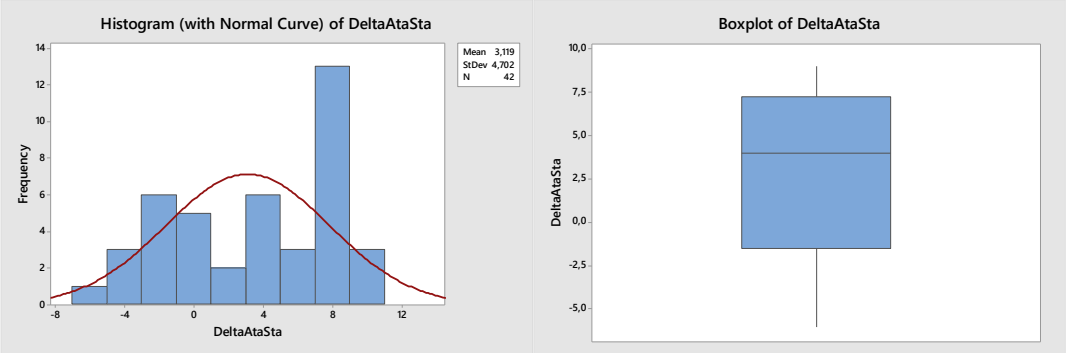


Figure 6-6: Histogram and boxplot of Delta from outliers set

Unfortunately, the data does not reveal a pattern in connection to the explanatory variables. The explanatory variables (weeks and days of departure and arrival, the carrier and the port of delivery) seem to be randomly distributed over the outlier dataset with no extreme outliers.

6.6 CONCLUSION

In this chapter, we answered the sub-question how the prediction model looks like. We started with the choice of the right machine learning technique, which became the random forest algorithm. Next, we treated the machine learning fundamentals concerning the bias-variance trade-off and the extent to which random forest is able to anticipate to that. We discussed the experimental test design, which consists of a 5-fold cross validation before we executed the actual random forest algorithm. During the parameter tuning phase, we discovered that our model generates best results with the number of features to possibly split on set to 3, and the number of trees to grow set to 750. During training the model, we avoided overfitting by showing an average R^2 of 0.808 and an average adjusted R^2 of 0.804. The fact that the values are close to each other is an indication for a good fit. We also visualized the model's goodness of the fit using a scatter plot. The model also has a good predictive performance with an average RMSE of 1.389. The validation of the model on the test set is a third indication for a good model fit, because the error is not extremely better or worse than in the train set. The model performs well, with an average RMSE of 1.430. The model is able of accurately predicting the deviation in arrival time. Only a few outliers were detected. We performed an extra analysis on these outliers, but unfortunately did not find any revealing patterns.

7 EVALUATION

This chapter is meant to quantify the effects of using the predicted deviation in the arrival time in communication to the customer. The following research question is answered in this chapter: *How can we translate a better prediction to improved business processes?* As stated in the problem description, the provision of an inaccurate ETA to the customer negatively influences business processes for both the LSP and the customer. Especially the customers are financially affected as they are responsible for all the operational activities from the moment that the order arrives at the destination port, meaning that they are also affected financially.

7.1 CONSEQUENCES OF A DEVIATION IN THE ETA

In the current situation, the LSP communicates the arrival time to the customer that is provided by the executive carrier in their sailing schedule in advance of actual shipment. This estimation of arrival time, however, seems to deviate significantly from the actual arrival time. When the arrival time appears to deviate, the delta gets a value: for example, the value is 8 when the order arrives 8 days later than the estimated arrival time using the STA, or the delta becomes negative when the order arrives earlier than estimated using the STA. As stated in the problem description, a deviated arrival time has consequences for both the customer and the LSP in terms of decreased efficiency and increased costs. Because costs are more easily quantified, we decided to construct a cost savings' model as tool to explore the effects of a better prediction on an improved business situation. We also decided to address the cost savings' model from the perspective of the customer as they are more financially affected by the consequences of a deviation in arrival time. We distinguish the following operational activities that are directly related to costs due to a deviated arrival time: demurrage fees, rescheduling costs and costs for the probability of running out of stock. In the next sections, we will individually discuss all three factors.

7.1.1 Demurrage for import

When referring to the incoterms described in Section 2.3.1, it becomes clear that the customer of the LSP (the consignee) is responsible for the operations and associated costs from the moment the goods arrive at the destination port. It is for that reason that we will only discuss the demurrage and detention fees for the import, thus from the perspective of the customer.

Demurrage fees are charged when import containers are still full and standing in the terminal after the 'free time for pick up' has expired; the container is not yet been picked up by the consignee then. The free time is usually 4 days, depending on the carrier, and starts when the container has been discharged from the vessel to the terminal. Demurrage is applied after this free time period has ended and the one who is charged depends on the incoterms (see Section 2.3.1).

Detention occurs when the consignee holds onto the carrier's container outside the port, terminal or depot beyond the free time, which starts after container pick up. When the container have been picked up but the container (either still full or empty) is still in the possession of the consignee and has not been returned to the carrier within the free time allotted to it, detention fees are charged.

Referring to these two charges, a deviation in arrival time does intuitively affect the chance on being charged for demurrage. When an order arrives earlier than estimated, there is a chance that the customer (consignee) is not able to pick up the goods earlier. When this time exceeds the free time for pick up, demurrage is charged. Since the incoterms indicate that the customer is responsible for all the operations and associated costs from the moment the goods arrive at the destination port, the customer is charged demurrage fees then. Demurrage fees are heavily depending on several factors, namely on the type of cargo, the size of the container, the carrier who executed the shipment and the port of import. Not only do different carriers charge different fees and handle different number of days of free time, also the country in which the port lies determines the demurrage fee, based on the currency in that country. The demurrage fees are, for instance, less in Thailand than in Japan.

7.1.2 Rescheduling costs

Rescheduling tasks, when an unexpected event occurs, affect the smooth operation of the customers' business if the initial schedule cannot be taken into account anymore. This also happens when an order arrives earlier or later than estimated by the carrier. The customer had accounted for a certain day that the order would arrive in the port, but if it turns out to be on another day, the schedule cannot be applied anymore. Extra costs are then incurred: rostered employees and to be used equipment that were planned must be cancelled. Also, new resources need to be planned for the new arrival day. This often needs to be done on an ad-hoc basis within a very short planning horizon, resulting in higher costs for employees and equipment. However, since we have no data available about costs for cancelling a pickup and rescheduling a new one on an ad-hoc basis, we will estimate them.

7.1.3 Running out of stock

If an order arrives later than planned, there is a chance that the customer runs out of stock. Of course, there are costs related to this occurrence. We refer here to the costs of missed sales: if a customer runs out of stock, it is not able to sell the fries anymore. The more delayed an order is, the more chance of running out of stock. We chose to increase this chance with a factor of 0.1 per day that the order is delayed (up to 1 after 10 days). We multiply this chance by the related costs of being out of stock. But what these cost then exactly are, is hard to determine. Questions like 'What is the daily trade volume of the customer?' and 'How many safety stock does the customer handles?' are relevant here. Because the costs are highly dependent on the customer who runs out of stock, we do not have exact values. We therefore have to estimate them.

7.2 ASSUMPTIONS

Because we have to scope the research due to limited time and because not all necessary information is available, we make some assumptions for constructing the cost savings' model. These are as follows:

- In case the free time exceeds, we assume that the customer pays demurrage costs or rescheduling costs. The customer either chooses to pick up the container on an ad-hoc basis and pays rescheduling costs, or the customer chooses to wait until the initial communicated date has arrived when he picks up the container then. In the latter, the customer is charged demurrage fees per day that the container is in the port after the free time has passed.
- Because the demurrage fees and days of free time are heavily depending on multiple factors and there is no centralized source where demurrage statistics are collected, we take an average amount on demurrage fees per day and the days of free time. Because the highest trade volume is reached around Singapore, we base our estimation on their current demurrage fees of an arbitrary chosen carrier. In Appendix Q, one can find the overview of the demurrage fees belonging to this example. We set the days of free time to 4 days.
- Rescheduling costs are the same for orders that arrive too early as that arrive too late. Besides, de costs do not depend on the number of deviated days: once the customer is forced to reschedule, the same costs are incurred.

7.3 COST SAVINGS' MODEL

Because all costs associated with deviations in estimated arrival times are unknown, we must estimate them. We will take a low and a high value of each cost parameter, and executed the calculation for each combination of parameter values. Following this logic, we create an experimental design with $2^3 = 8$ combinations of parameter settings. The values for the cost parameters are estimated per container and are as follows:

	Parameter	Low	~	High
•	Demurrage costs	75	~	150
•	Rescheduling costs	250	~	550
•	Out of stock costs	150	~	500

The test set over which we execute the calculations for our cost savings' model, consists of 717 orders. As in 2018 the yearly trade volume of the LSP was 3,422 orders, our test set represents roughly one fifth of a year. So the costs that we list here, are calculated over a time period of approximately 10 weeks.

	ArrDay	ArrWeek	DepDay	DepWeek	Carrier	PoD	DeltaAtaSta	predictrounded	residual
1574	3	15	3	10	HAPAG-LLOYD	Valparaiso	1	1	0
536	3	40	5	37	HAMBURG SUD	Rio de Janeiro	0	0	0
2706	6	42	7	37	APL	Lat Krabang	-1	-1	0
2124	3	37	1	33	MSC	Tanjung Priok	0	0	0
3111	1	48	4	44	OOCL	Shanghai	-1	0	-1
2805	2	44	2	39	OOCL	Shanghai	7	2	5

Figure 7-1: Preview of the test dataset on which the cost savings' model is based

We will explain the logic behind the model using a concrete example from our test set. We therefore take the last shipment of the preview of the dataset from Figure 7-1 (row number 2805). In our cost savings' model, we simulate two situations: the current situation and the new situation. The savings are then the difference in the costs between the current and the new situation.

In the **current** situation, the customer gets an ETA that is equal to the STA that the carrier published in their sailing schedule, thus

$$ETA = STA$$

The value in the column `DeltaAtaSta`[2], that has a value of 7, is the actual delta and indicates that the order arrived 7 days later than the STA initially indicated. The customer is then required to pay several costs over this 7 days of deviation. We thus base the costs, directly related to the deviation in the current situation, on the column `DeltaAtaSta`.

In the **new** situation, we assume that the predicted arrival time (ETA_{pred}) is communicated to the customer in advance of actual shipment (instead of the arrival time from the carrier), where

$$ETA_{pred} = STA + \varepsilon$$

The value in the column `predictrounded`, that has a value of 2, is the ε and is the output of our prediction model. However the actual delta was 7, our prediction model was able to predict a deviation of $\varepsilon = 2$. This means that in the new situation, the ETA_{pred} is 2 days closer to the actual delta, that was 7, but there are still 7 – 2 = 5 days of deviation left. This value of 5 can be found in the column `residual` and is the subtraction of actual

[2] Note that this variable is our target variable during our report.

delta minus predicted delta. The customer is then required to pay several costs over this 5 days of deviation in arrival time. We thus base the costs, directly related to the deviation in the new situation, on the column `residual`.

Our model first loops through the `DeltaAtaSta` column and through the `residual` column for calculating the costs for the current and new situation respectively. The pseudo-code of looping through these columns is as follows:

Pseudo code Costs savings' algorithm

Read all input data x_i ***(i = 1,2,….n)***
Add column with predicted values from testing
Initialize ***X*** ← *target column, either* ***DeltaAtaSta*** *or* ***Residual***
Initialize ***demurrage*** costs
Initialize ***reschedule*** costs
Initialize costs ***out of stock***
for *i* = 1:*nrow*
    if *X* <= -4 **then**
        j = X + 4
        costs = | *j* * *demurrage* |
    else if *X* > -4 & *X* <= -1 **then**
        costs = *reschedule*
    else if *X* = 0 **then**
        costs = 0
    else if X >= 1 **then**
        p = *X*/10
        costs = *reschedule* + (*p* * *out of stock*)
    end if
        Total costs += *costs* *i*-1th iteration
end for
        Savings = *Total costs* (current) – *Total costs* (new)

Figure 7-2: Pseudo code for looping through the data for determining cost savings

We executed the algorithm 8 times, one for each unique combination of cost parameter settings. Depending on the number of days of deviation, certain costs are charged. For example, if the number of days of deviation exceeds 4, demurrage fees are charged for each extra day that the arrival time deviates.

At the end of each run, we calculated the savings by extracting the costs from the new situation from the costs in the current situation. The costs related to each combination of parameter settings (low and high value for demurrage, reschedule, and out of stock costs) per situation (current and new) are listed in Table 7-1.

Table 7-1: Results of cost saving's model for different parameter settings and for the current and new situation

	Parameter settings								
Demurrage	Low	High	Low	Low	High	Low	High	High	
Reschedule	Low	Low	High	Low	High	High	Low	High	
Out of stock	Low	Low	Low	High	High	High	High	Low	
	Costs (in euros)								**Average**
Current situation	148,980	152,805	298,380	197,175	350,400	346,575	201,000	302,205	249,690
New situation	58,520	59,795	123,020	66,675	132,450	131,175	67,950	124,295	95,485
Savings	90,460	93,010	175,360	130,500	217,950	215,400	133,050	177,910	**154,205**

To be able to generate a realistic estimation of costs, we included a low and a high value for each parameter. The average savings over the 8 combinations of cost parameter settings can then be calculated and is €154,205. As discussed earlier in this chapter, these savings cover a period of approximately 10 weeks. When multiplying it by 5, an average amount of €771,025 can be saved on a yearly basis.

7.3.1 Translation to the LSPs perspective

Since we stated in the problem description that also the LSP is negatively affected by the deviation in arrival time, it is worthwhile to also evaluate the advantages of the LSP. When exploring the current situation, we concluded that the LSP is negatively affected in terms of efficiency and by the potential loss of customers rather than being financially affected. However less easy quantifiable, we will measure the effects of the predicted ETA by the times the LSP does not have customer contact. Customer contact happens when an order deviates more than 4 days in the form of an informative mail, and when the order deviates more than 6 days in the form of a telephone call. Both occurrences can be referred to as 'issue resolving', because customer contact is needed to explain and apologize for the deviation. We simply counted the times that the LSP is requested to have customer contact, which is, from a deviation of 4 days or more in the form of a mail and from a deviation of 6 days or more in the form of a phone call. When we then compare the current with the new situation, we see the following improvements, summarized in Table 7-2.

Table 7-2: Number of times customer contact is required in current and new situation, broken down to type of contact (email at 4 days and telephone at 6 days), including the percentual improvement

	Current situation	**New situation**	**% Improvement**
Deviation >= 4 days	180 (25%)	34 (4%)	+ 84%
Deviation >= 6 days	131 (18%)	16 (2%)	+ 89%
Of total orders	717	717	NA

In the current situation, the LSP was required to have customer contact in the form of a mail in 180 of the 717 orders. In the new situation, for only 34 of the 717 orders the LSP was requested to have customer contact. So the LSP goes from 25% of the times to 4% of the times being busy with issue resolving by mail, which is a percentual improvement of 84%. For a deviation of 6 days or more, when the LSP is required to contact the customer by a phone call, the percentual improvement is 89% when comparing the current and the new situation.

The decrease in issue resolving is not only preferable in terms of increased efficiency (less customer contact), it also implies the LSP having a more proactive attitude towards arrival times. The communicated arrival time is not solely based on the carrier's sailing schedule anymore, but also other factors are incorporated which makes the arrival time more robust to deviations. Then, being able to more proactively communicate an arrival time that captures multiple factors, results in less issue resolving, which eliminates the concern of potential loss of customers.

7.4 CONCLUSION

In this chapter, we constructed a cost savings model to quantify the effects of communicating the predicted ETA instead of the arrival time purely based on the schedule of the carrier. Because all costs directly resulting from a deviation in arrival time remained unknown while executing this research, we were forced to estimate them and we decided to include two extreme values for each cost parameter. We addressed the customers' perspective in the cost savings' model and found that the average yearly savings are around €771,025. We subsequently also translated this logic to the LSPs perspective and found that their efficiency can be increased by 84% less customer contact being busy with issue resolving. This also positively affects the LSPs reputation as the customer's need for more proactive and accurate arrival time provision is granted. With this, the LSPs concern of potential loss of customers is also eliminated.

8 DEPLOYMENT

In this chapter, we are going to develop a prototype that can be deployed for actual use in the Logistics Management System (LMS) that the LSP uses. This LMS is an application that is developed using Mendix, which is a low-code software platform that provides tools to build, test, deploy and iterate applications (Rao, 2009). As discussed earlier when exploring the current situation, the LMS is a platform that integrates, processes and delivers relevant data for planning, executing and optimizing transportation fleets. The global forwarding department of the LSP uses this platform for the daily planning and controlling of all shipments. We will first elaborate where we implemented our prediction model in the already existing LMS in Section 8.1, which is all about the prototype. In Section 8.2, we will delve deeper into the underlying data architecture which makes it possible to call a Random Forest algorithm from within a Mendix application.

8.1 THE PROTOTYPE

The output of our prediction algorithm is visible at two pages within the LMS. The first one is at the page where the transport planner sees an overview of available schedules for an order he/she wants to book at a specific carrier. This overview refers to the 'process of booking a sailing', described in Section 2.2.2. Figure 8-1 shows this environment in the LMS. Where currently only the STA from the carrier's sailing schedule was notated, the overview now has a field 'Predicted ETA' which displays the output (ETA_{pred}) of our prediction model.

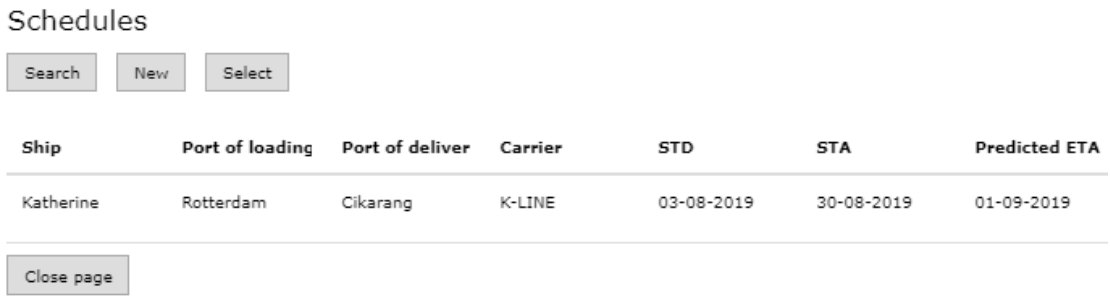


Figure 8-1: Overview of carrier's available sailing schedules

In the example we displayed here, our prediction model evidently predicted a delta of 2 for this order, as the predicted ETA is 2 days later than the STA. The predicted ETA is displayed here, because then the transport planner can include this knowledge in the decision-making process of choosing the most appropriate sailing.

The second page where our prediction is visible, is at the general order overview. This overview refers to the 'process of ETA communication', described in Section 2.2.4. From the moment the order is booked at and confirmed by the carrier, the order is visible in this overview. The order is visible in this overview until it gets an ATA (and is thus actually arrived at the destination port). In the overview in Figure 8-2, the additional field 'Predicted ETA' is visible, in which the output (ETA_{pred}) of our prediction model is displayed. For reference, in Figure 2-3 of this report, one can find a screenshot of how the

general order overview currently looks like, so without the additional 'Predicted ETA' field.

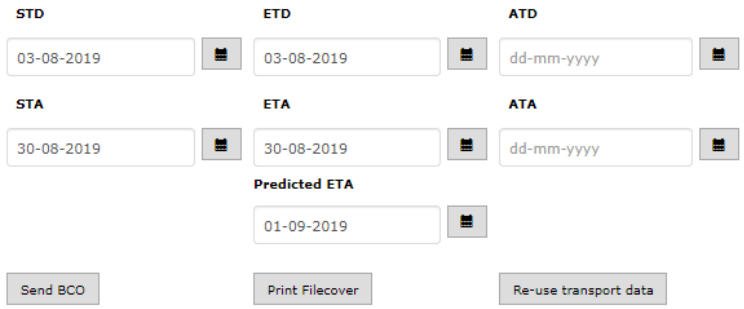


Figure 8-2: Departure- and arrival times in the order overview, including the added 'Predicted ETA' field

From the information available at this page, the customer is informed about the arrival time. Currently the ETA (where ETA = STA) is communicated to the customer, but with the additional field that displays a predicted ETA that is closer to the actual arrival time than the ETA, the LSP is now capable of communicating a more accurate arrival time to the customer.

8.2 DATA ARCHITECTURE

To make sure the Mendix application is able to 'communicate' with the R statistical software in which we programmed our Random Forest prediction model, we used an Amazon Web Service (AWS) to create and run a virtual machine in the cloud. Within this virtual machine in the cloud, which is also referred to as an instance, the R script can be executed and the result can exported back to the Mendix application. The logic that needs to be built in Mendix to make sure the application is able to export data to the instance and receive data from it, is built in a so-called 'microflow'. A microflow can perform actions like creating and updating objects, extracting data and making decisions. As stated earlier, Mendix is a low-code platform and a microflow is a visual way of expressing actions that traditionally end up in a textual algorithm. The microflow that we created for extracting the relevant order data receiving the result of the prediction model, can be found in Appendix R. The data architecture of the deployment of this prototype is visually represented in Figure 8-3.

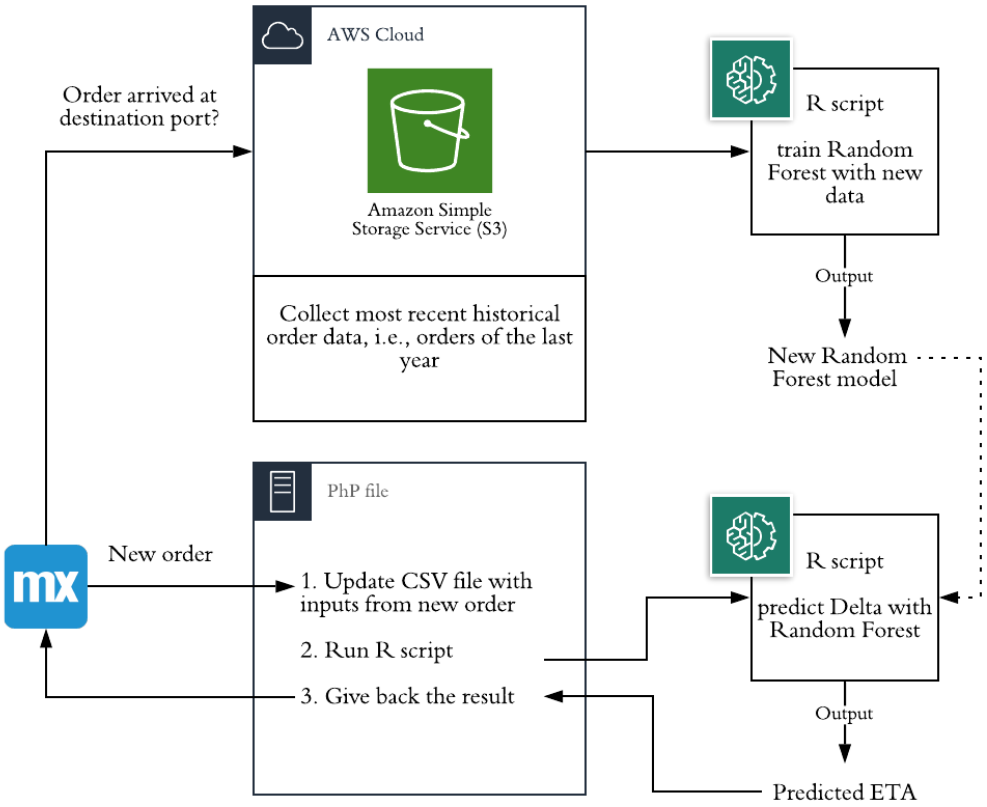


Figure 8-3: Data architecture for communicating between Mendix (working environment of the LMS) and R statistical software (prediction model's environment)

The logic behind this implementation consists of two parts: the prediction of a new, individual order and training the Random Forest algorithm on a historical order dataset once a while. We will discuss the training and the individual predictions in the following two sections respectively.

8.2.1 Training the algorithm

The iterative aspect of retraining a machine learning model is important because as it is exposed to new data, it becomes smarter as they learn from previous computations to produce more reliable outputs (James et al., 2013). For the prototype, we use the model that is fitted on all 2.5 years of historical order data. Continuously (re-)training the model is not necessary as the throughput time of an order is not that fast. However, when the prediction model is in production for a while, and new data has become historical, the model needs to be retrained. Considering the average transit time of an order, which is 33 days[3], we recommend retraining the model once a month. If you decide to train the model more often, it is expected that the train dataset will not differ enough from the previous one. Also, not all historical order data should be included in the train dataset. The bucket in which train data is collected in the cloud, should consist of only a selection of historical order data. Considering the average yearly trade volume of the LSP, which counted 3422 orders in 2018, we recommend only including orders in the train dataset that are at most one year old. Older orders can contain obsolete information, for example a port where the LSP does not ship to anymore. Besides, when training the model with a dataset of this length (of approximately 3422 orders), the train set is expected to be big enough to let the model learn new patterns. Thereby, because Random Forests are not known for their short running time, the train set is small enough to not let the running time explode.

8.2.2 Individual predictions

When a new order comes in, a prediction about its ETA must be made. We therefore created the microflow that first extracts all features from the order that our prediction model needs as input (ArrDay, ArrWeek, DepDay, DepWeek, Carrier, PoD). Then, the input is sent to the virtual machine that calls the Random Forest algorithm which produces the prediction. The prediction model uses the most recent version of the retrained Random Forest algorithm. The output (which is of type integer, as our prediction model predicts the Delta), is sent back to the microflow in the Mendix application. The last step is converting this integer value to a date so it becomes the predicted ETA (ETA_{pred}). This is done by adding the Delta (in number of days) to the STA. The microflow belonging to this logic can be found in Appendix R.

[3] Based on the 2.5 years on historical order data.

8.3 CONCLUSION

In this chapter, we deployed the Random Forest prediction model as a prototype in the already existing LMS. To let the application, that is developed in Mendix, communicate with our prediction model that is developed in R, we use an AWS instance that can execute the R script from the cloud, and exports the output back to the Mendix application. (Re-) training the algorithm can also be executed in the cloud on the AWS instance, from which we recommended to do once a month on a train dataset that contains one year of historical order data.

9 CONCLUSION, RECOMMENDATIONS AND LIMITATIONS

In this research, we have constructed – and implemented – a prediction model for a Dutch LSP that is responsible for the outbound global logistics of frozen potato products. All products are shipped by containerized freight transport from the ports of Rotterdam and Antwerp to ports all around the world. The LSP currently solely bases their arrival time information on the sailing schedule of the executive carrier. However, this source historically appears to be unreliable as 20% of all shipments did not arrive on time according to the published schedules by the executive carrier, based on a threshold of 6 days deviation before a shipment is classified as being 'not on time'. From a customer survey, the need became visible for a more proactive provision of more accurate arrival time information. The LSP collects order data since October 2016, and we used this historical data to make a prediction model that is able to predict the deviation in scheduled arrival time in advance of actual shipment.

9.1 CONCLUSION

In this section, the main findings and answers to the research (sub-)questions as introduced in Section 1.5 are listed. We will individually address the sub-questions to fully provide an answer to our research question:

To what extent can the operational efficiency at the global forwarding department of an LSP be increased by predicting the accuracy of the ETA provided by the carrier at the moment of booking?

The answer to this question is directly derived from answering the sub questions, which we briefly list in the following sections. After extensively answering the research questions, we address the recommendations, followed by the limitations in which we do suggestions for future research.

9.1.1 Current situation at the LSP

The first sub-question addressed the current situation at the LSP. From the operational processes that are extensively elaborated, we found the detailed processes of 1) booking an order from which the arrival time is taken from the sailing schedule of the executive carrier, 2) obtaining tracking information from an active order and 3) communicating the ETA information to the customer. The tracking updates during shipment are obtained in three ways: by the cloud scraper, via INTTRA and manually. If an order's ETA deviates more than 4 days from the initial arrival time from the sailing schedule, a mail is sent to the customer to inform the deviation in arrival time. If an order's ETA deviates 6 days or more, the customer is being called with the new arrival time information. Besides exploring their daily operational proceedings, we exposed a global export analysis in which we provided first insights into the data that is available from the LSP. From this we concluded that the LSP is experiencing a rapid growth of 222.92% increase of average monthly trade volume between 2017 and 2018. Where the LSP was initially a relatively small player, they are now an active player in the domain of small-medium logistic businesses with serious competitors right around the corner. The LSP works with two

types of incoterms, namely CIF and CFR. Both are very similar and differ only from the insurance agreements. From the incoterms it becomes clear that the LSP is only responsible for the operations associated with the shipment until unloading the container in the port of import. The incoterms are a good argument for the customer's concerns of inaccurate ETA information as they become fully responsible from this moment.

9.1.2 Historical order data

In this sub-question we focused on the historical order dataset and to what extent the available attributes in the dataset are usable in the context of predicting the deviation in arrival time in advance of actual shipment. First, we explored the dimensionality and quality of the data, from which we made a distinction between a sailing and a shipment: a shipment is one order by a customer; a sailing is a transit of a ship between two ports. One sailing can contain multiple shipments. Because the LSP uses a shipment as operational and analytical perspective, we chose to base our further analysis and prediction model on the shipment dimension. The quality of the data is addressed on the presence of ambiguity and missing values. All quality issues were resolved both manually and systematically.

The second part of this sub-question focused on finding the best attributes to make predictions from. To this end, we also derived new attributes from existing ones from which we intuitively think they would be of good predictive power. We analysed all (both initial and derived) attributes one by one and made hypotheses about which attributes are possibly good predicts for the deviation in arrival time. We also explored the behaviour of our target variable, the Delta. From some attributes (Carrier, PoD) and from the target variable Delta, we removed the outliers. In the context of constructing a prediction model that predicts *in advance of* actual shipment, we constrained the attributes to be 'must be available at time of prediction'. Following this logic, the attribute Ship must be excluded from the dataset. We end with a dataset of 12 attributes readily available to predict the target. After applying feature selection, we end with a set of 7 explanatory attributes: DepDay, DepWeek, ArrDay, ArrWeek, Carrier, PoD and Transit. After performing an additional experiment, Transit appears not to be of good predictive power and we choose to exclude this attribute.

9.1.3 The prediction model

In this sub-question, we designed the actual prediction model. The random forest machine learning algorithm came out to be the algorithm with the best predictive performance out of a small experiment. We also concluded this earlier from our extensive literature research. One of the main advantages of random forest is that it can handle correlated predictor variables. Although the random forest is robust to overfitting, it does not mean the model always avoids overfitting. We therefore set up a k-fold cross validation experiment to prevent the model from overfitting. The parameters mtry and ntree are tuned as we concluded from literature that these two parameters are worth tuning. The final model is constructed using values of ntree = 750 and mtry = 3. To provide the reader some insight into the model, we addressed the relative importance of the input variables. ArrWeek turned out to be the most important variable in terms of percentual increase in MSE when not including this variable. We trained our model using the aforementioned cross validated experimental design, an evaluated the model on predictive performance and the extent to which it tends to overfit. The results are satisfying: the average RMSE is 1.40 and when we look at the scatter plot, the model does not seem to overfit. This appears to be the case when we tested our trained model on an unseen dataset. The error is quite higher, but still very low, with an average RMSE of 1.43.

The graph of predicted against actual values clearly reflected which orders were more difficult to predict. We executed an extra analysis on the orders for which the model was not able to make a good prediction. Unfortunately, the data does not reveal a pattern in relation to the explanatory variables. The weeks and days of departure and arrival, the carrier and the port of delivery seem to be randomly distributed over the outlier dataset with no extreme outliers. What we did conclude was that from the orders that are harder to predict, the average Delta is also higher than the average Delta from the initial dataset (3.2 and 1.4, respectively).

9.1.4 Deployment of the prediction model in the LMS

In the LMS, which is a Mendix application, we deployed a prototype for actual use. The output of the prediction algorithm (thus, the predicted ETA) is visible at two pages within the LMS: at the page where the transport planner sees an overview of available schedules for an order he/she wants to book at a specific carrier, and at the general order overview after the order is booked at and confirmed by the carrier. These are the two phases within the broader process in which having the knowledge of a deviation in arrival time, is crucial. The first one when searching for the most appropriate sailing, and the second one at the overview page from which customer are informed with an arrival time. Currently the ETA (where ETA = STA) is communicated to the customer, but with the additional field that displays a predicted ETA that is closer to the actual arrival time than the ETA, the LSP is now capable of communicating a more accurate arrival time to the customer.

9.1.5 Translation to improved business processes

In this sub-question, we studied the effect of using the forecast, i.e., the effect of using the improved prediction of the arrival time on the business situations for both the customer and the LSP. We chose to address the cost savings' model from the perspective of the customer as they are financially more affected by the consequences of a deviation in arrival time. Because all costs directly resulting from a deviation in arrival time remained unknown when executing this research, we were forced to estimate them and we decided to include two extreme values for each cost parameter. Since we had 3 cost parameters with all 2 possible values, we calculated $2^3 = 8$ cost savings' models. On average, it is expected to save a total amount of €771,025 euros on a yearly basis. We also addressed the improved business processes from the LSPs perspective. It is expected their efficiency will be increased by 84% less customer contact being busy with issue resolving. This would also positively affect the LSPs reputation as the customer's need for more proactive and accurate arrival time provision is granted. The LSPs concern of potential loss of customers would also be eliminated with this.

9.2 RECOMMENDATIONS

During the execution of this research, but also after the research goal has been reached, a number of recommendations for the LSP were found, which we will discuss in this section.

9.2.1 Data usage

During the data understanding and data preparation part of the research, it became clear that the data that is stored, is of low quality. A lot of changes, deletions, merging operations and other manipulations had to be done, otherwise the data could not be used for analysis. Also ambiguity plays a fatal role to this end: multiple denotations for one (level of a) variable is misleading when analysing the data. There are, for example, more than 5 ways for denotating the port of loading: RTM, Port of Rotterdam, Rotterdam Port, RTDM; all referring to the same port of loading. A human would understand the similarity between these, but it would be harder for machines. We therefore recommend to be consistent in the way of notating information about an order. A suggestion for this, is to use drop down menus from which the transport planner is forced to choose one of the options and is not able to fill in a name by him/herself anymore.

9.2.2 Data collection

We also recommend to collect more data, preferably as much as possible. We are still at the very beginning of a digital age, in which decision-making is increasingly data-driven. In the context of this research, for example, the knowledge about an order requiring one or more transhipments would be valuable information for predicting the deviation in the arrival time. However, because this information was not collected and stored in the historical order dataset, we were not able to include this into our prediction model. This is unfortunate, because this data is available at the LSP; it is just not stored in the right place. We thus recommend the LSP to include data like this in the historical order dataset.

We further recommend to divide the dataset into shipments by road and by sea. Currently, data of all orders is collected in one CSV file. The fact that the European distribution (road) and the global forwarding (sea) are two completely separate departments within the LSP, is reason enough for separating the datasets. Moreover, it avoids the issue of having a lot of variables (columns) that only relate to one of the two types of shipment, which results in an unnecessarily big and complex dataset with more than 40 columns, in which more than half of the columns only belong to one of the two types of shipment.

9.2.3 Use predicted ETA rather than the carrier's sailing schedule

As the results of our prediction model indicated, it is good in predicting the deviation of an order whose arrival time is solely based on the carrier's sailing schedule. We proved that our model can predict the arrival time of an order more accurately, which leads to improved business situations in terms of saved costs and increased efficiency. We therefore strongly recommend to communicate the predicted ETA to the customer instead of the arrival time as stated in the carrier's sailing schedule.

9.2.3.1 Retrain the model

The iterative aspect of retraining a machine learning model is important because as it is exposed to new data, it becomes smarter because they learn from previous computations to produce more reliable outputs. To be able to keep the predictions based on historical order data accurate, the model needs to be retrained once in a while. Continuously (re-) training the model is not necessary as the throughput time of an order is not that fast. We therefore recommend to train the model once a month on the most recent historical order dataset that consists all orders of the twelve months. It is not preferable to add more

historical orders in the train dataset as this can add noise. Orders from more than one year ago can contain obsolete information. For example, they can contain a carrier that does not even have contracts with the LSP anymore. Besides, as random forests are not the fastest models to train, it would unnecessarily increase the running time. We therefore recommend to add an extra bucket[4] in which only the historical order data is stored that is available for training. This bucket is then dynamic in the sense that every day, an order becomes older than one year and is therefore removed from the dataset.

[4] The LSP works with Amazon Simple Storage Service (S3), which is an object storage service in which historical order data is stored in so-called 'buckets'.

9.3 LIMITATIONS

The research is subject to several limitations. As already discussed in the recommendations, some key data was not available for including in our prediction model. Intuitively, the knowledge about an order having a transhipment or being send directly would be of great value for predicting the deviation in the arrival time. Because a delay (and thus a deviation) is influenced by the need of a transhipment, this knowledge could lead to a much more accurate prediction model. As soon as the LSP has started to collect this order characteristic, future research can include this variable to check whether the inclusion leads to significant more accurate predictions.

9.3.1 Assumption of costs

A second limitation lies in the translation to improved business situations. Because all costs directly resulting from a deviation in arrival time remained unknown while executing this research, we were forced to estimate them. This makes the improved business processes difficult to interpret and to generalise. We suggest that future research focuses more on the translation to business processes and delves deeper into the cost components.

9.3.2 Prediction interval

A third limitation is about the type of prediction that is generated. We generated a point estimate rather than a prediction interval. A prediction interval is a type of a confidence interval, which provides a range of possible values. It says something about how sure the model is about the predicted value: if the prediction interval is wide, the model's prediction is not very accurate. This especially becomes interesting with respect to the use of the forecast: we might want to inform customers differently in case of forecasts with large prediction intervals. Ideally, orders with large prediction errors as we also encountered in our analysis, also have a high prediction interval. If this is the case, one can more easily decide to exclude this type of order from the prediction model in favour of the model's overall predictive performance. For those difficult orders, we might still resort to the schedules from the carriers.

9.3.3 Times series forecasting with machine learning

The last thing worth mentioning, is not necessarily a limitation but rather a recommendation for future research. We built a static prediction model that uses a static target variable, which is the Delta, based on a historical order dataset. This Delta is derived from the ETA, which equals the STA published in the carrier's sailing schedule. As we described in Section 2.2.3, the carrier publishes track and trace updates along the way. With the creation of our prediction model, we ignored this and assumed that the ETA is static and is determined once. However, this not completely reflects reality. We therefore recommend a future research where time series forecasting is applied instead of machine learning with a static target variable. A time series analysis has the time (t) as an independent variable and a target dependent variable (y_t). The output of the prediction model is the predicted value for y at time t ($\hat{y}_t$)(Brown, 2018). Now, assuming that the ETA is updated at regular intervals of time, a time series forecasting using machine learning can be performed (Brownlee, 2016). The time series adds an extra dimension to the problem because of its temporal component. The sliding window that time series data entails, can be phrased by restructuring the data as follows. First, the tracking updates are collected and are the output of the time dimension, see Table 9-1 for a concrete example.

Table 9-1: Example time series data

Time	Measure (ETA)
1	29-08-2019
2	29-08-2019
3	29-08-2019
4	31-08-2019
5	01-09-2019

Now, this data can be restructured to be used in a machine learning problem by using the value at the previous time step to predict the value at the next time step. Restructuring the data following this logic, the data will look as follows, see Table 9-2.

Table 9-2: Restructuring time series to machine learning

X	y
?	29-08-2019
29-08-2019	29-08-2019
29-08-2019	29-08-2019
29-08-2019	31-08-2019
31-08-2019	01-09-2019
01-09-2019	?

Notice that the previous time step has become input (**X**) and the next time step is the output (**y**) when transforming the time series to our machine learning problem. Also the order between the observations, that must be preserved in order to let it function as time series, is preserved. As in the first row does not have a previous value and the last row does not have a next value, these rows should be deleted when start modelling. Intuitively, when performing time series analysis and putting the results in a graph, a decreasing trend (with respect to the target variable Delta) should become visible. This because after several tracking updates are published, and the vessel is approaching its destination port (implying the further we are in the time dimension), the more the carrier is expected to accurately predict the arrival time. It is expected that a time series approach of this research reveals other interesting patterns in the data which we can make knowledge from, because the model is not solely based on historical data but also on online tracking updates. Moreover, if the prediction model is capable of adapting to new tracking updates in making predictions, way more accurate predictions can be made which results in even more saved costs and time.

BIBLIOGRAPHY

Altinkaya, M., & Zontul, M. (2013). Urban bus arrival time prediction: A review of computational models. *International Journal of Recent Technology and Engineering (IJRTE), 2*(4), 2277-3878.

Altman, D., Machin, D., Bryant, T., & Gardner, M. (1989). *Statistics with confidence* (Second edition ed.). Bristol: J W Arrowsmith Lt.

Bin, Y., Zhonzhen, Y., & Baozhen, Y. (2006). Bus Arrival Time Prediction Using Support Vector Machines. *Journal of Intelligent Transportation Systems, 10*(4), 151-158.

Blackhurst, J., Craighead, C. W., Elkins, D. & Handfield, R. B. (2005). An empirically derived agenda of critical research issues for managing supply-chain disruptions. *International Journal of Production Research*, *43*(19), 4067–4081.

Breiman, L. (2001). Ramdom Forest. *Machine Learning, 45*, 05-32.

Brown, M. (2018). *Time Series Forecasting with Machine Learning: An example from Kaggle.* Retrieved from RPubs: https://rpubs.com/mattBrown88/TimeSeriesMachineLearning

Brownlee, J. (2016). *Time Series Forecasting as Supervised Learning.* Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/time-series-forecasting-supervised-learning/

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinart, T., Shearer, C., & Wirth, R. (2000). CRISP–DM Step-by-Step Data Mining Guide, http://www.crisp-dm.org/

Chien, S.I.J., Ding, Y., Wei, & C. (2002). Dynamic Bus Arrival Time Prediction with Artificial Neural Networks, *Journal of Transportation Engineering, 128*(5), 429-438.

Chopra, S., & Sodhi, M. (2004). Managing Risk to Avoid Supply-Chain Breakdown. *MIT Sloan Management Review, 46*(1), 53-61.

Craighead, C. W., Blackhurst, J., Rungtusanaham, M. J., & Handfield, R. B. (2007). The severity of supply chain disruptions: Design characteristics and mitigation capabilities. *Decision Sciences, 38*(1), 131-156.

Dashko, L. (2017). Travel time prediction. Distributed System Seminar (pp. 1-6). Tartu: University of Tartu.

Diakopoulos, N., & Koliska, M. (2017, 8 9). Algorithmic Transparency in the News Media. *Digital Journalism, 5*(7), 809-828.

Dobrkovic, A., Iacob, M., Van Hillegersberg, J., Mes, M., & Glandrup, M. (2015). Towards an approach for long term AIS-based prediction of vessel arrival times. In A. Dobrkovic, M. Iacob, J. Van Hillegersberg, M. Mes, & M. Glandrup, *Logistics and Supply Chain Innovation: Bridging the Gap between Theory and Practice.*

Dobrkovic, A., Iacob, M.-E., & van Hillegersberg, J. (2015). Using machine learning for unsupervised maritime waypoint discovery from streaming AIS data. *Proceedings*

of the 15th International Conference on Knowledge Technologies and Data-driven Business.

Dobrkovic, A., Liu, L., Iacob, M., & Van Hillegersberg, J. (2016). Intelligence amplification framework for enhancing scheduling processes. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).*

Drewry Shipping Consultants. (2015). *Container Performance Insight.* Retrieved from: http://www.drewry.co.uk/

Dunham, M. *Data Mining - Introductory and Advanced Topics.* Prentice Hall.

Elkins, D., Handfield, R., Blackhurst, J., & Craighead, C. (2005). *18 Ways to Guard Against Disruption.*

Fancello, G., Pani, C., Pisano, M., Serra, P., Zuddas, P., & Fadda, P. (2011). Prediction of arrival times and human resources allocation for container terminal. *Maritime Economics and Logistics, 13*(2), 142-173.

Freedman, D. (2005). *Statistical models : theory and practice.* Cambridge University Press.

Freitas, A. (2002). *Data Mining and Knowledge Discovery with Evolutionary Algorithms.*

Golbraikh, A., & Tropsha, A. (2002). *Beware of q 2 !*

Goodman, L., & Kruskal, W. (1979). *Measures of Association for Cross Classifications.* New York: Springer-Verlag.

Gurning, R. (2011). *Maritime Disrutions in the Australian Indonesian Wheat Supply Chain; an Analysis of Risk Assessment and Mitigation Strategies.* Institute of University of Tasmania.

Guyon, I., & De, A. (2003). *An Introduction to Variable and Feature Selection André Elisseeff.*

Haykin, S., & Haykin, S. (2009). *Neural networks and learning machines.* Prentice Hall/Pearson.

Heerkens, H., & Winden, A. v. (2017). *Systematisch Management Problemen Oplossen.* Groningen/Houten: Noordhoff Uitgevers.

Hodge, V., & Austin, J. (2004). A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review, 22*(2), 85-126.

International Chamber of Shipping, 2018. *Shaping the future of shipping.* Retrieved from: http://www.ics-shipping.org/shipping-facts/shipping-and-world-trade

James, G., Witten, D., Trevor, H., & Robert, T. (2013). *An introduction to statistical learning: with applications in R.* New York: Springer.

Jantawan, B., & Tsai, C.-F. (2007). A Comparison of Filter and Wrapper Approaches with Data Mining Techniques for Categorical Variables Selection. *International Journal of Innovative Research in Computer and Communication Engineering (An ISO, 3297*.

Jeong, R. (2004). *The prediction of bus arrival time using automatic vehicle location systems data.* Texas A&M University, Texas.

Jin, Y., & Sendhoff, B. (2008, 5). Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 38(3)*, 397-415.

Johansson, U. (2007). *Obtaining Accurate and Comprehensible Data Mining Models-An Evolutionary Approach.* Linköping Studies in Science and Technology .

Kantardzic, M. (2011). *Data Mining: Concepts, Models, Methods and Algorithms* (2nd edition ed.). New Jersey: John Wiley & Sons, Inc.

Karabulut, E., Özel, S., & İbrikçi, T. (2012, 5 23). A comparative study on the effect of feature selection on classification accuracy. *Procedia Technology, 1*, 323-327.

Karegowda, A., Manjunath, A., & Jayaram, M. (2010, 6 15). Feature Subset Selection Problem using Wrapper Approach in Supervised Learning. *International Journal of Computer Applications, 1*(7), 13-17.

Kutner, M., Nachtsheim, C., Neter, J., & Li, W. (2004). *Applied Linear Statistical Models.* New York: McGraw Hill.

Larose, D. (2005). *Discovering Knowledge in Data: An Introduction to Data Mining.* Hoboken, New Jersey: John Wiley & Sons Inc.

Laxhammar, R. (2007). *Artificial Intelligence for Situation Assessment.* Royal Institute of Technology , Stockholm.

Markou, M., & Singh, S. (2003). Novelty detection: A review - Part 1: Statistical approaches. *Signal Processing*.

Marsland, S. (2015). *Machine Learning - An Algorithmic Perspective.* Boca Raton: CRC Press.

Martineau, E., Roy, J., & Valcartier, D. (2011). *Maritime Anomaly Detection: Domain Introduction and Review of Selected Literature.* Defence R&D Canada-Valcartier.

Mascaro, S., Nicholson, A., & Korb, K. (2014). Anomaly detection in vessel tracks using Bayesian networks. *International Journal of Approximate Reasoning, 55*, 84-98.

Pathak, A., & Vashistha, J. (2015). Classification Rule and Exception Mining Using Nature Inspired Algorithms. *International Journal of Computer Science and Information Technologies, 6*(3), 3023-3030.

Piatetsky, G. (2014). CRISP-DM, still the top methodology for analytics, data mining, or data science projects. Retrieved January 15, 2019, from KDnuggets: http://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-datascience-projects.html

Portnoy, L. (2001). *Intrusion detection with unlabeled data using clustering.* Data Mining Lab.

Rao, L. (2009). *Mendix Helps Enterprises Streamline Application Deployment.* Retrieved from Techcrunch: https://techcrunch.com/2009/04/08/mendix-helps-enterprises-streamline-application-deployment/

Rao, S., & Goldsby, T. J. (2009). Supply chain risks: A review and typology. *The International Journal of Logistics Management, 20*(1), 97-123. doi:10.1108/09574090910954864

Recknagel, F. (2001). Applications of Machine Learning to Ecological Modeling. *Ecological Modeling, 146*(1), 303-310.

Riveiro, M. (2011). *Visual Analytics for Maritime Anomaly Detection.* Orebro University.

Rogalewicz, M., & Sika, R. (2016). Methodologies of knowledge discovery from data and data mining methods in mechanical engineering. *Management and Production Engineering Review, 7*(4), 97-108.

Salleh, N., Riahi, R., Yang, Z., & Wang, J. (2017). Predicting a Containership's Arrival Punctuality in Liner Operations by Using a Fuzzy Rule-Based Bayesian Network (FRBBN). *Asian Journal of Shipping and Logistics*.

Shalaby, A., & Farhan, A. (2004). Bus Travel Time Prediction for Dynamic Operations Control and Passenger Information Systems. *82nd Annual Meeting of the Transportation Research Board.* Washington D.C: National Research Council.

Shavlik, J., Mooney, R., Towell, G., & Quinlan, J. (1991). *Symbolic and Neural Learning Algorithms: An Experimental Comparison.*

Sipser, M. (2006). *Introduction to the Theory of Computation* (Second Edition ed.). Thomson Course Technologies.

Smith, B., Demetsky, M., & Smith, B. (1995). Short-term Traffic Flow Prediction: Neural Network Approach. *Transportation Research Record, 1453*, 98-104.

Spekman, R. E., & Davis, E. W. (2004). Risky Business: Expanding the Discussion of Risk and the Extended Enterprise. *International Journal of Physical Distribution & Logistics Management, 34,* 414-433.

Tan, P.N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining.* Boston: Pearson Education, Inc.

Vespe, M., Visentini, I., Bryan, K., & Braca, P. (2012). Unsupervised learning of maritime traffic patterns for anomaly detection. *9th IET Data Fusion & Target Tracking Conference (DF&TT 2012): Algorithms & Applications.*

Vernimmen, B., Dullaert, W., & Engelen, S. (2007). Schedule unreliability in liner shipping: Origins and consequences for the hinterland supply chain. *Maritime Economics & Logistics, 9*(3), 193-213.

Walpole, R., Myers, R., Myers, S., & Ye, K. (2016). *Probability and Statistics for Engineers and Scientists* (Ninth edition ed.). Prentice Hall.

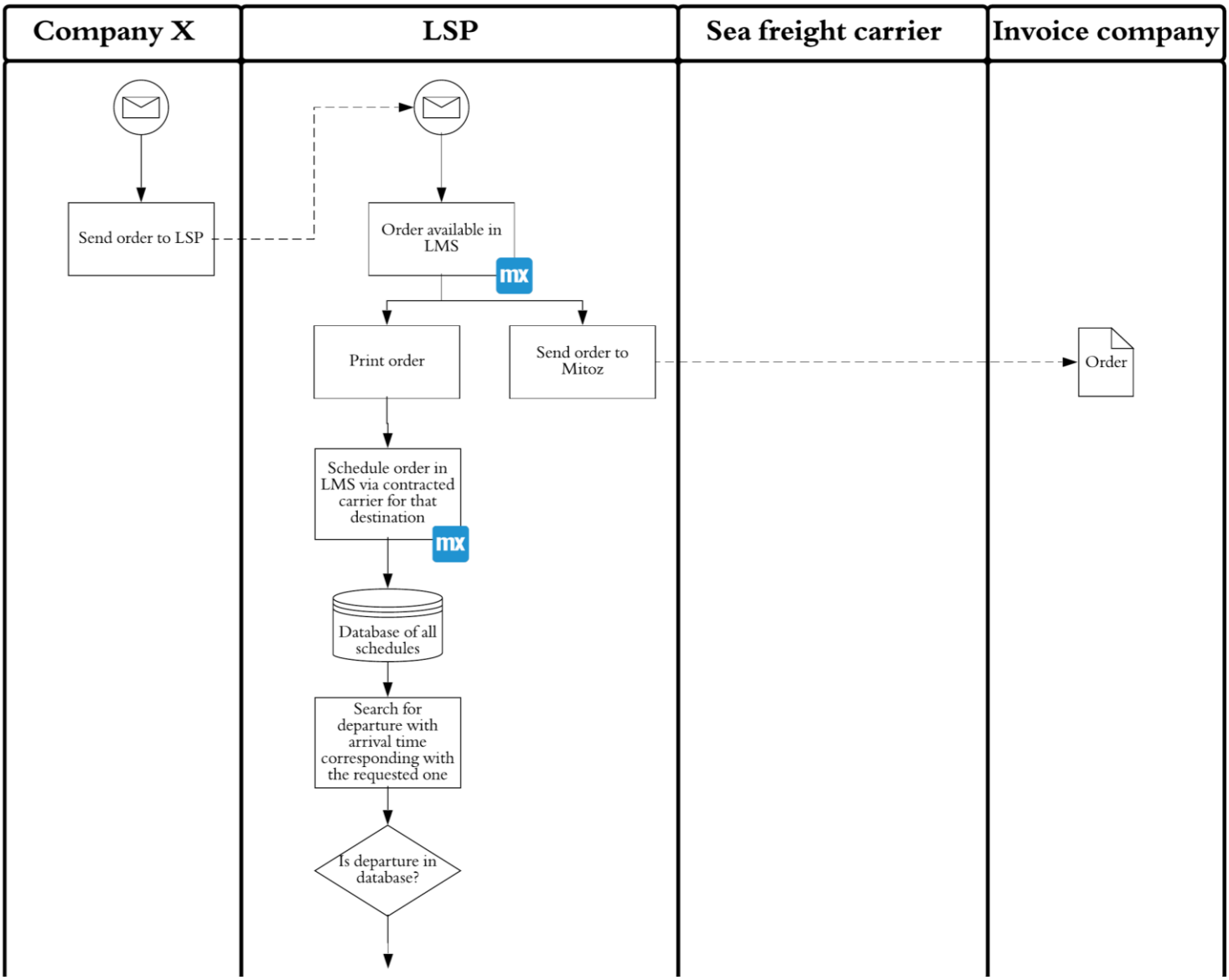
Wiering, M., Embrechts, M., Stollenga, M., Wiering, M., Van Der Ree, M., Embrechts, M., . . . Schomaker, L. (2013). *The Neural Support Vector Machine Image retrieval View project Continuous learning in robot navigation using virtual categorization and reinforcement learning, including robotic arm View project The Neural Support Vector Machine.*

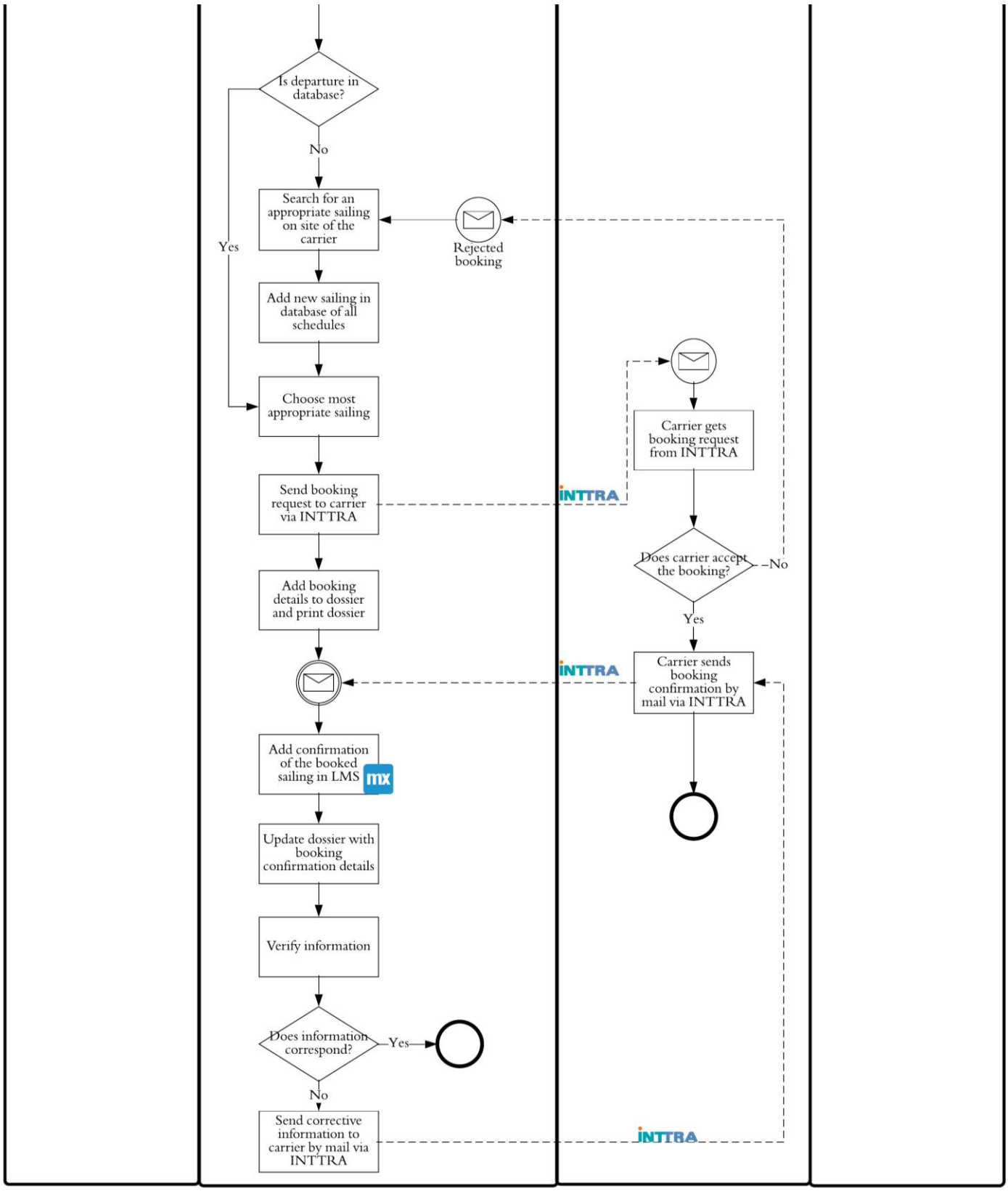
Williams, B., & Hoel, L. (2003, 11). Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results. *Journal of Transportation Engineering, 129*(6), 664-672.

Wu, C., Wei, C., Su, D., Chang, M., & Ho, J. (2003). Travel time prediction with support vector regression. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. 2*, pp. 1438-1442. Institute of Electrical and Electronics Engineers Inc.

APPENDIX A

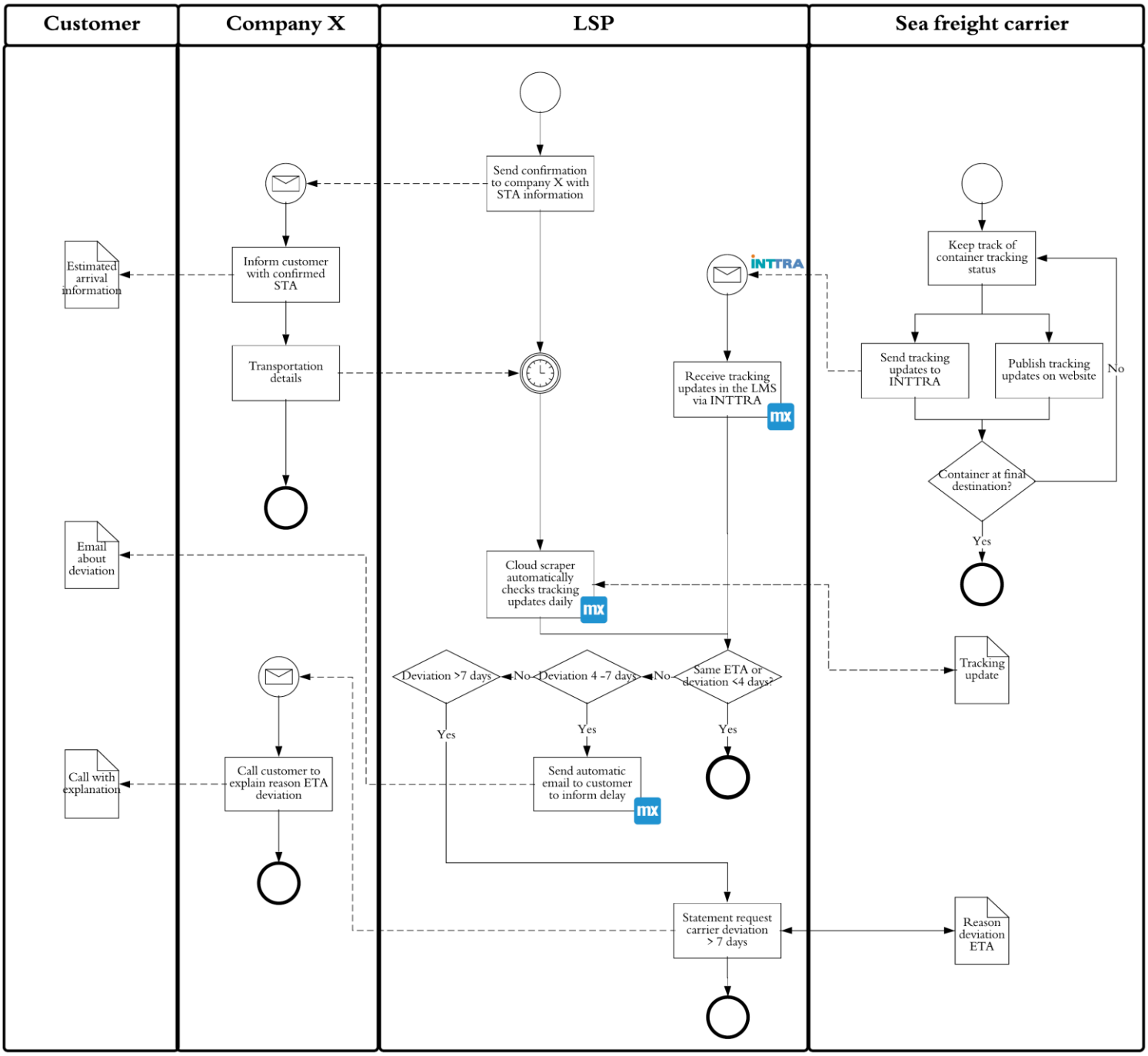
Carrier booking process




Is departure in database?
No
Yes
Search for an appropriate sailing on site of the carrier
Rejected booking
Add new sailing in database of all schedules
Choose most appropriate sailing
Send booking request to carrier via INTTRA
Add booking details to dossier and print dossier
Add confirmation of the booked sailing in LMS
Update dossier with booking confirmation details
Verify information
Does information correspond?
Yes
No
Send corrective information to carrier by mail via INTTRA
Carrier gets booking request from INTTRA
Does carrier accept the booking?
No
Yes
Carrier sends booking confirmation by mail via INTTRA
INTTRA
INTTRA
INTTRA

APPENDIX B

ETA communication process



APPENDIX C

Available order data

Name	Description	Data type
Shipment number	The number of the shipment	INTEGER
Consignee	Person/company to whom the order is delivered	CHARACTER
Freight Term	CFR or CIF?	CHARACTER
Cold Store	The cold store where freight is loaded	CHARACTER
Terminal Closing Date	Due date for container to arrive at terminal	DATE
VGM Due Date	Due date for delivering VGM information to carrier	DATE
Loading Date	The date on which container is loaded at the cold store	DATE
Carrier	The carrier with which the order is shipped	CHARACTER
Ship	Name of the ship with which the order is sent	CHARACTER
Port of Loading	The port from where the order is loaded	CHARACTER
STD	Scheduled Time of Departure	DATE
ETD	Estimated Time of Departure	DATE
ATD	Actual Time of Departure	DATE
Port of Delivery	The port where the order is delivered to	CHARACTER
STA	Scheduled Time of Arrival	DATE
ETA	Estimated Time of Arrival	DATE
ATA	Actual Time of Arrival	DATE
Shipping instructions	Date of sending shipping instructions	DATE
Final B/L	Date of sending Bill of Lading	DATE
Comments	Additional comments / complaints	N.A.

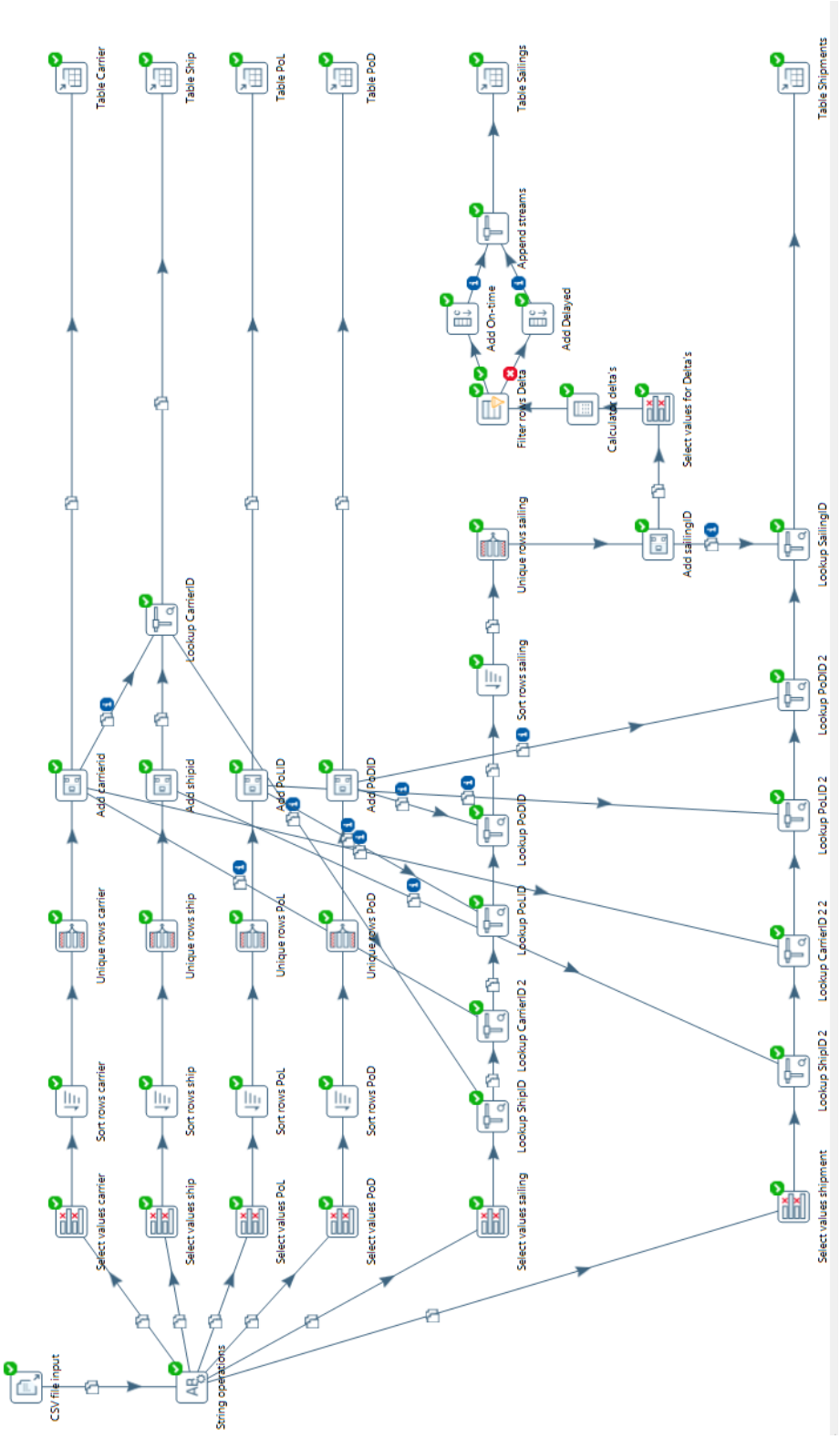
In- or exclusion of attributes based on the capability of possibly predicting the anomalous ETA

Attribute	*In-/excluded*	*Reason*
Shipment number	Excluded	Already using another unique identifier
Delivery address code	Excluded	Already implied in the PoD
Loading address code	Excluded	Already implied in the PoL
Product type	Excluded	Not applicable to sea shipments
Distance	Excluded	Not applicable to sea shipments
Total gross weight	Excluded	Formalities for booking container
Total net weight	Excluded	Formalities for booking container
Transport type	Excluded	Dataset already filtered on only sea shipments
Delivery date	Excluded	Already implied in the STA
Loading date	Excluded	Already implied in the STD
Customer code	Excluded	Already using another unique identifier
Port of Loading (PoL)	Included	Intuitively explanatory variable
Creation date	Excluded	Already implied in the STD
Cold store	Excluded	Not applicable to sea shipments
Port of Delivery (PoD)	Included	Intuitively explanatory variable
Container number	Excluded	Intuitively not explanatory
Seal number	Excluded	Intuitively not explanatory
Booking number	Included	To keep a unique identifier in the dataset
Document status	Excluded	Intuitively not explanatory
Carrier	Included	Intuitively explanatory variable
Ship	Included	Intuitively explanatory variable

Reference	Excluded	Already used another unique identifier
VGM	Excluded	Formalities for booking container
Batch number	Excluded	Already used another unique identifier
Freight term	Excluded	Intuitively not explanatory
Actual loading date	Excluded	Already implied in the ATD
Terminal closing date	Excluded	Intuitively not explanatory
VGM due date	Excluded	Formalities for booking container
BCO sent	Excluded	Formalities for booking container
Bills of Lading sent	Excluded	Formalities for booking container
Booking request sent	Excluded	Formalities for booking container
Booking confirmed	Excluded	Formalities for booking container
STD	Included	Intuitively explanatory variable
STA	Included	Intuitively explanatory variable
ETD	Included	Intuitively explanatory variable
ETA	Included	Intuitively explanatory variable
ATD	Included	Intuitively explanatory variable
ATA	Included	Intuitively explanatory variable
Full return depot	Excluded	Intuitively not explanatory
Empty pick-up depot	Excluded	Intuitively not explanatory

APPENDIX D

ETL flow of order data



APPENDIX E

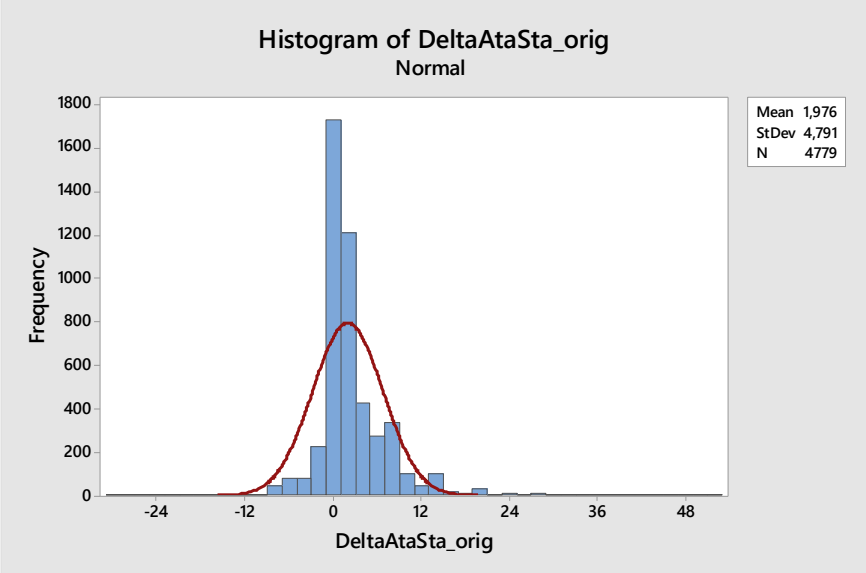
Ports per Region

Africa	Asia	Middle East/India	North America	South America	Oceania
Abidjan	Dalian	Ad Dammam	New York	Antofagasta	Auckland
Apapa	Kobe	Ashdod	Baltimore	Arica	Christchur
Casablanca	Nagoya, Aichi	Aqaba	Jacksonville	Malboa	Lautoka
Cape Town	Osaka	Bahrain	Manzanillo	Barranquilla	Lyttelton
Cotonou	Port Klang	Doha	Puerto Cortes	Buenos Aires	Napier
Dakar	Qingdao	Haifa	Antigua	Callao	Port More
Der es Salaam	Shanghai	Hama port	Barcadera	Cartagena	Suva
Durban	Singapore	Jebel Ali	Bridgetown	Corinto	Tuaranga
Djibouti	Tanjung Priok	Karachi	Castries	Iquique	Townsville
Freetown	Tokyo	Sharjah	Caucedo	Mariel	Wellingto
Tema	Xingang	Shuwaikh	Fort de France	Montevideo	Brisbane
Lekki Lagos	Yangon	Sohar	Freeport	Puerto Cabello	Fremantle
Matadi	Yantian Pt	Chennai	Guayaquil	Progreso	Melbourn
Port Louis	Yokohama	Colombo	Kingston	Puerto Progresso	Sydney
Luanda	Belawan Sumatra	Jeddah	La Guaira	Punta Arenas	Riverwoo
Alexandria	Bintulu Sarawak	King Abdullah port	Oranjestad	San Antonio	
Canical	Busan	Sudan	Paramaribo	San Vicente	
Las Palmas	Cat Lai	Umm Qasr	Philipsburg	Santa Marta	
Santa Cruz de T	Cebu	Kolkata	Pointe a Pitre	Valparaiso	
Marsaxlokk	Chittagong	Mersin	Port of Spain	Vera Cruz	
	Cikarang		Puerto Barrios	Zarate	
	Davao, Mindanao		Puerto Limon	Itajai	
	Guangzhou		Puerto Moin	Itapoa	
	Hakata/Fukuoka		Puerto Santo Tomas	Manaus	
	Hakata/Fukoka		Rio Haina	Navegantes	
	Hong Kong		Roseau	Parangua	
	Ilnchon		San Juan	Pecem	
	Kaohsiung		St George's	Rio de Janeiro	
	Keelung		St. John's	Rio Grande	
	Kobe		Willemstad	Salvador	
	Kota Kinabalu, Sabah		Balboa	Santos	
	Kuching Sarawak			Suape	
	Laem Chabang			Vitoria	
	Lat Krabang			Acajutla	
	Manila North Harbour			Guadalajara	
	Mawei			Georgetown	
	Miri			Paranagua	
	Pasir Gudang				
	Penang				
	Phnom Penh				
	Semarang				
	Sibu				
	Sihanoukville				
	Surabaya				
	Xiamen				
	Yantai				

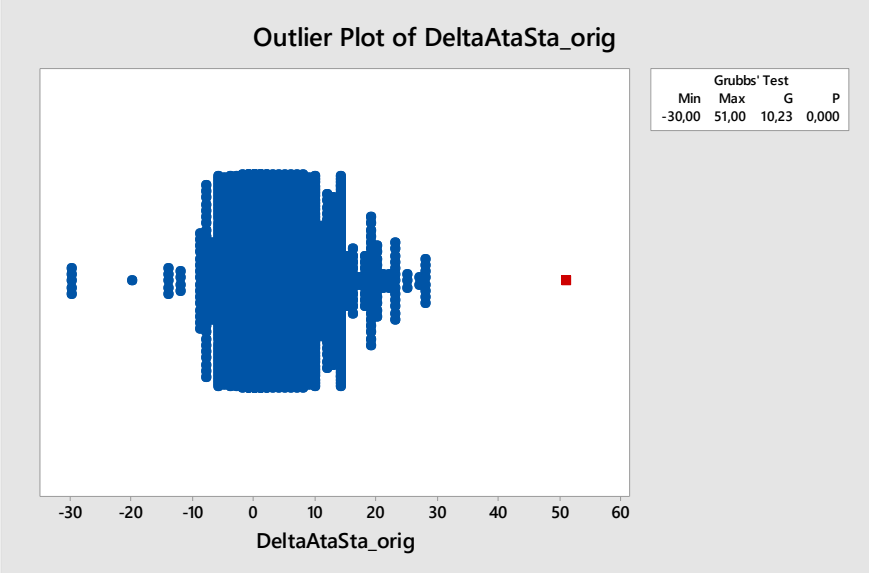
APPENDIX F

Analysis of target variable Delta

Histogram



Outlier test



APPENDIX G

Hypothesis test difference in carrier ratios

Difference between proportions of carriers' on time performance

The standard error of $p_1 - p_2$ is

$$\mathrm{SE}(D) = \sqrt{\frac{p_1(1-p_1)}{n_1} + \frac{p_2(1-p_2)}{n_2}}.$$

The confidence interval for the population difference in proportions is then given by

$$D - z_{1-\alpha/2} \times \mathrm{SE}(D) \quad \text{to} \quad D + z_{1-\alpha/2} \times \mathrm{SE}(D),$$

Hypothesis test: Difference between proportions

H_0: $P_1 = P_2$

H_1: $P_1 \neq P_2$

For all the carriers compared to X, except Y and Z, we can reject the null hypothesis and conclude that the on-time performance differ significantly at an significance level of .05.

X – Y

Descriptive Statistics

Sample	N	Event	Sample p
Sample 1	156	126	0,807692
Sample 2	716	506	0,706704

Estimation for Difference

Difference	95% CI for Difference
0,100988	(0,030725; 0,171252)

CI based on normal approximation

Test

Null hypothesis	H_0: $p_1 - p_2 = 0$
Alternative hypothesis	H_1: $p_1 - p_2 \neq 0$

Method	Z-Value	P-Value
Normal approximation	2,82	0,005
Fisher's exact		0,010

X – Y

Descriptive Statistics

Sample	N	Event	Sample p
Sample 1	156	126	0,807692
Sample 2	112	99	0,883929

Estimation for Difference

Difference	95% CI for Difference
-0,0762363	(-0,161933; 0,009460)

CI based on normal approximation

Test

Null hypothesis	H_0: $p_1 - p_2 = 0$
Alternative hypothesis	H_1: $p_1 - p_2 \neq 0$

Method	Z-Value	P-Value
Normal approximation	-1,74	0,081
Fisher's exact		0,128

X – Y

Descriptive Statistics

Sample	N	Event	Sample p
Sample 1	156	126	0,807692
Sample 2	458	412	0,899563

Estimation for Difference

Difference	95% CI for Difference
-0,0918710	(-0,159566; -0,024176)

CI based on normal approximation

Test

Null hypothesis	H_0: $p_1 - p_2 = 0$
Alternative hypothesis	H_1: $p_1 - p_2 \neq 0$

Method	Z-Value	P-Value
Normal approximation	-2,66	0,008
Fisher's exact		0,005

X – Y

Descriptive Statistics

Sample	N	Event	Sample p
Sample 1	156	126	0,807692
Sample 2	1204	1045	0,867940

Estimation for Difference

Difference	95% CI for Difference
-0,0602479	(-0,124982; 0,004487)

CI based on normal approximation

Test

Null hypothesis	H_0: $p_1 - p_2 = 0$
Alternative hypothesis	H_1: $p_1 - p_2 \neq 0$

Method	Z-Value	P-Value
Normal approximation	-1,82	0,068
Fisher's exact		0,049

X – Y

Descriptive Statistics

Sample	N	Event	Sample p
Sample 1	156	126	0,807692
Sample 2	72	69	0,958333

Estimation for Difference

Difference	95% CI for Difference
-0,150641	(-0,227812; -0,073471)

CI based on normal approximation

Test

Null hypothesis	H_0: $p_1 - p_2 = 0$
Alternative hypothesis	H_1: $p_1 - p_2 \neq 0$

Method	Z-Value	P-Value
Normal approximation	-3,83	0,000
Fisher's exact		0,002

The normal approximation may be inaccurate for small samples.

X – Y

Descriptive Statistics

Sample	N	Event	Sample p
Sample 1	156	126	0,807692
Sample 2	117	113	0,965812

Estimation for Difference

Difference	95% CI for Difference
-0,158120	(-0,228184; -0,088056)

CI based on normal approximation

Test

Null hypothesis	H_0: $p_1 - p_2 = 0$
Alternative hypothesis	H_1: $p_1 - p_2 \neq 0$

Method	Z-Value	P-Value
Normal approximation	-4,42	0,000
Fisher's exact		0,000

The normal approximation may be inaccurate for small samples.

X – Y

Descriptive Statistics

Sample	N	Event	Sample p
Sample 1	156	126	0,807692
Sample 2	966	867	0,897516

Estimation for Difference

Difference	95% CI for Difference
-0,0898232	(-0,154558; -0,025088)

CI based on normal approximation

Test

Null hypothesis	H_0: $p_1 - p_2 = 0$
Alternative hypothesis	H_1: $p_1 - p_2 \neq 0$

Method	Z-Value	P-Value
Normal approximation	-2,72	0,007
Fisher's exact		0,003

X – Y

Descriptive Statistics

Sample	N	Event	Sample p
Sample 1	156	126	0,807692
Sample 2	311	292	0,938907

Estimation for Difference

Difference	95% CI for Difference
-0,131214	(-0,198545; -0,063884)

CI based on normal approximation

Test

Null hypothesis	H_0: $p_1 - p_2 = 0$
Alternative hypothesis	H_1: $p_1 - p_2 \neq 0$

Method	Z-Value	P-Value
Normal approximation	-3,82	0,000
Fisher's exact		0,000

X – Y

Descriptive Statistics

Sample	N	Event	Sample p
Sample 1	156	126	0,807692
Sample 2	368	244	0,663043

Estimation for Difference

Difference	95% CI for Difference
0,144649	(0,066182; 0,223116)

CI based on normal approximation

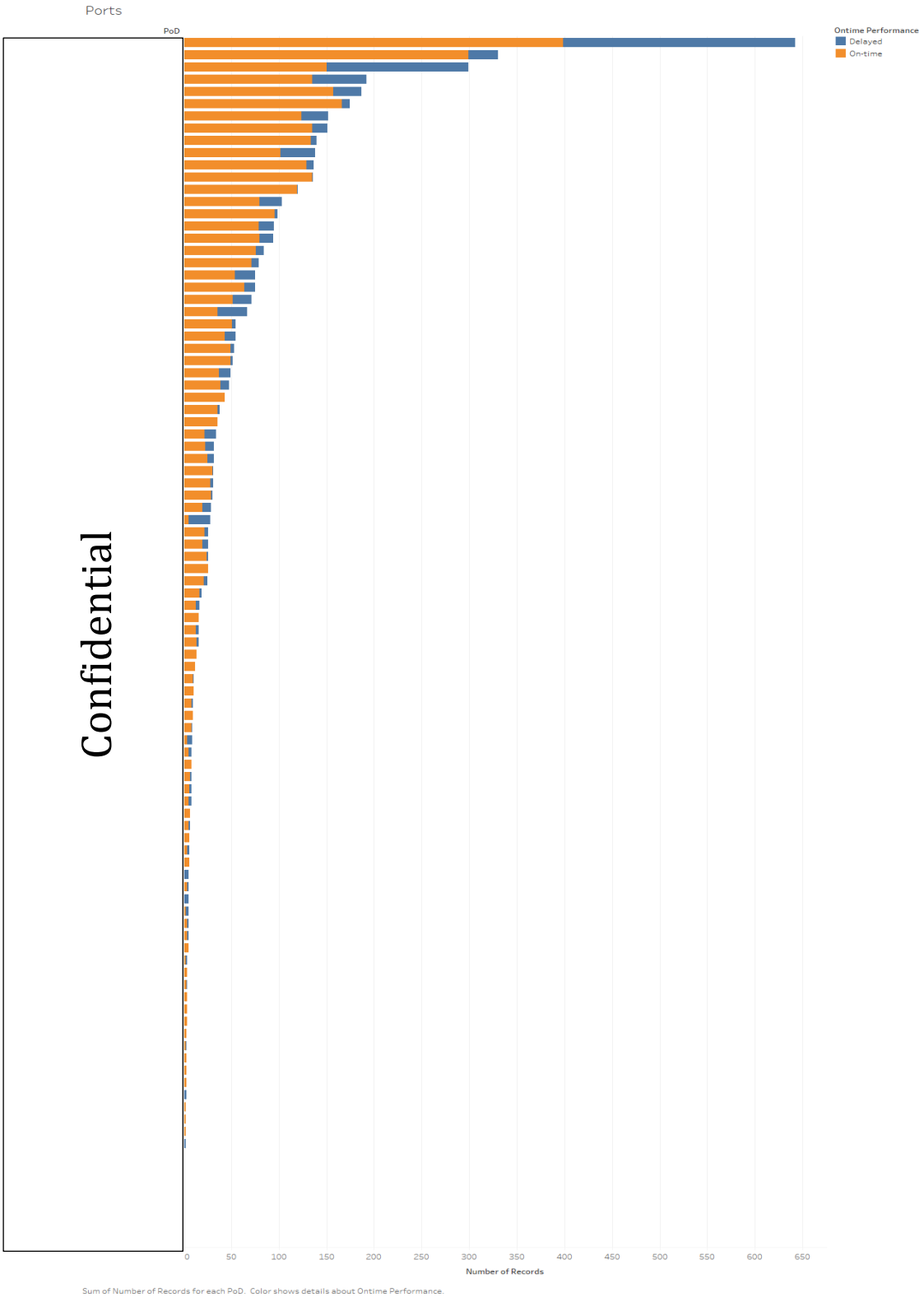
Test

Null hypothesis	H_0: $p_1 - p_2 = 0$
Alternative hypothesis	H_1: $p_1 - p_2 \neq 0$

Method	Z-Value	P-Value
Normal approximation	3,61	0,000
Fisher's exact		0,001

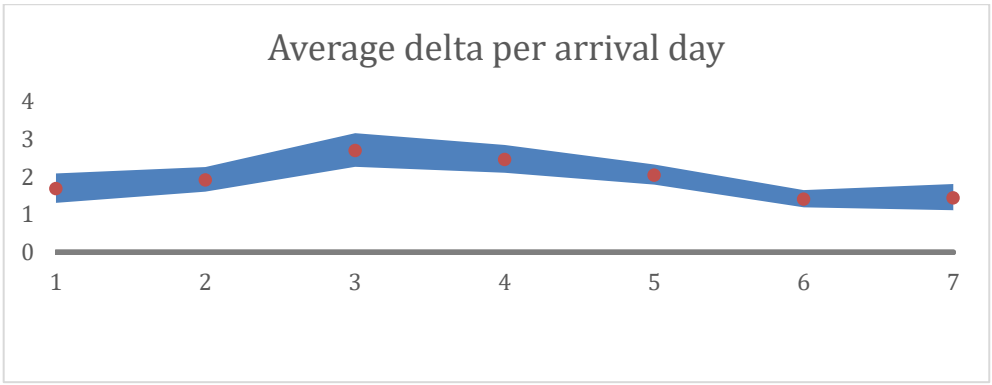
APPENDIX H

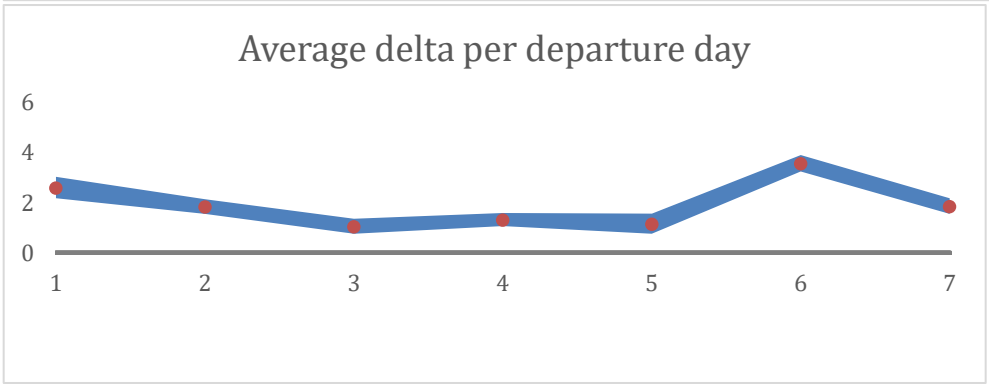
Ports of Delivery

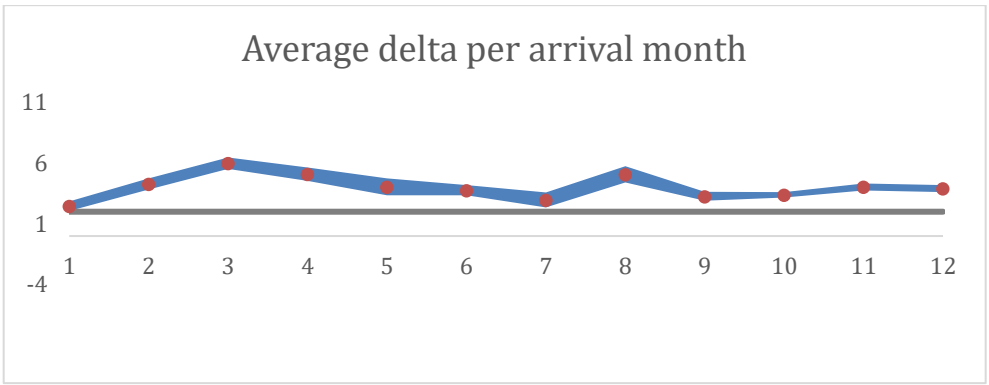


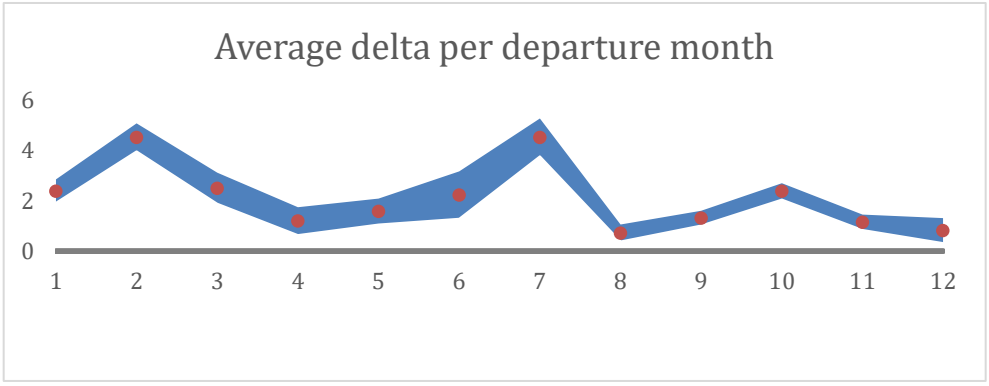
APPENDIX I

Average Delta per arrival- and departure day and month









APPENDIX J

Week numbers included in the dataset, broken down to week of arrival and week of departure

Departure week	*Months included*	*Weeks included*
2016	{October, November, December}	40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52
2017	{January, February, March, April, May, June, July, August, September, October, November, December}	1,…,52
2018	{January, February, March, April, May, June, July, August, September, October, November, December}	1,…,52
2019	N.A.	N.A.

Arrival week	*Months included*	*Weeks included*
2016	{November, December}	44, 45, 46, 47, 48, 49, 50, 51, 52
2017	{January, February, March, April, May, June, July, August, September, October, November, December}	1,…, 52
2018	{January, February, March, April, May, June, July, August, September, October, November, December}	1,…,52
2019	{January}	1, 2

APPENDIX K

Selected attributes from feature selection

RANDOM FOREST FORWARD

=== Run information ===

Scheme: weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Relation: WekaFeatureSelection-weka.filters.unsupervised.attribute.Remove-R3,6,9-11,13,15-29

Instances: 4781

Attributes: 6

PoD

ArrWeek

ArrMonth

DepWeek

Transit

Region

Test mode: 10-fold cross-validation

RANDOM FOREST BACKWARD

=== Run information ===

Scheme: weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Relation: WekaFeatureSelection-weka.filters.unsupervised.attribute.Remove-R3,6,9-11,13,15-29

Instances: 4781

Attributes: 7

PoL

PoD

ArrDay

ArrWeek

DepDay

DepWeek

Transit

Test mode: 10-fold cross-validation

RANDOM FOREST BI-DIRECTIONAL

=== Run information ===

Scheme: weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Relation: WekaFeatureSelection-weka.filters.unsupervised.attribute.Remove-R3,6,9-11,13,15-29

Instances: 4781

Attributes: 7

Carrier

PoD

ArrDay

ArrWeek

DepDay

DepWeek

Transit

Test mode: 10-fold cross-validation

RANDOM TREE FORWARD

=== Run information ===

Scheme: weka.classifiers.trees.RandomTree -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Relation: WekaFeatureSelection-weka.filters.unsupervised.attribute.Remove-R3,6,9-11,13,15-29

Instances: 4781

Attributes: 11

Carrier

PoL

PoD

ArrDay

ArrWeek

ArrMonth

DepDay

DepWeek

Transit

Region

Hurricane

Test mode: 10-fold cross-validation

RANDOM TREE BACKWARD

=== Run information ===

Scheme: weka.classifiers.trees.RandomTree -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Relation: WekaFeatureSelection-weka.filters.unsupervised.attribute.Remove-R3,6,9-11,13,15-29

Instances: 4781

Attributes: 9

Carrier

PoL

PoD

ArrDay

ArrWeek

DepDay

DepWeek

Transit

Region

Test mode: 10-fold cross-validation

RANDOM TREE BI-DIRECTIONAL

=== Run information ===

Scheme: weka.classifiers.trees.RandomTree -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Relation: WekaFeatureSelection-weka.filters.unsupervised.attribute.Remove-R3,6,9-11,13,15-29

Instances: 4781

Attributes: 12

Carrier

PoL

PoD

ArrDay

ArrWeek

ArrMonth

DepDay

DepWeek

DepMonth

Transit

Region

Hurricane

Test mode: 10-fold cross-validation

REP TREE FORWARD

=== Run information ===

Scheme: weka.classifiers.trees.REPTree -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Relation: WekaFeatureSelection-weka.filters.unsupervised.attribute.Remove-R3,6,9-11,13,15-29

Instances: 4781

Attributes: 8

Carrier

ArrDay

ArrWeek

ArrMonth

DepDay

DepWeek

DepMonth

Transit

Test mode: 10-fold cross-validation

REP TREE BACKWARD

=== Run information ===

Scheme: weka.classifiers.trees.REPTree -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Relation: WekaFeatureSelection-weka.filters.unsupervised.attribute.Remove-R3,6,9-11,13,15-29

Instances: 4781

Attributes: 7

Carrier

PoD

ArrDay

ArrWeek

DepDay

DepWeek

Transit

Test mode: 10-fold cross-validation

REP TREE BI-DIRECTIONAL

=== Run information ===

Scheme: weka.classifiers.trees.REPTree -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Relation: WekaFeatureSelection-weka.filters.unsupervised.attribute.Remove-R3,6,9-11,13,15-29

Instances: 4781

Attributes: 7

PoL

PoD

ArrDay

ArrMonth

DepWeek

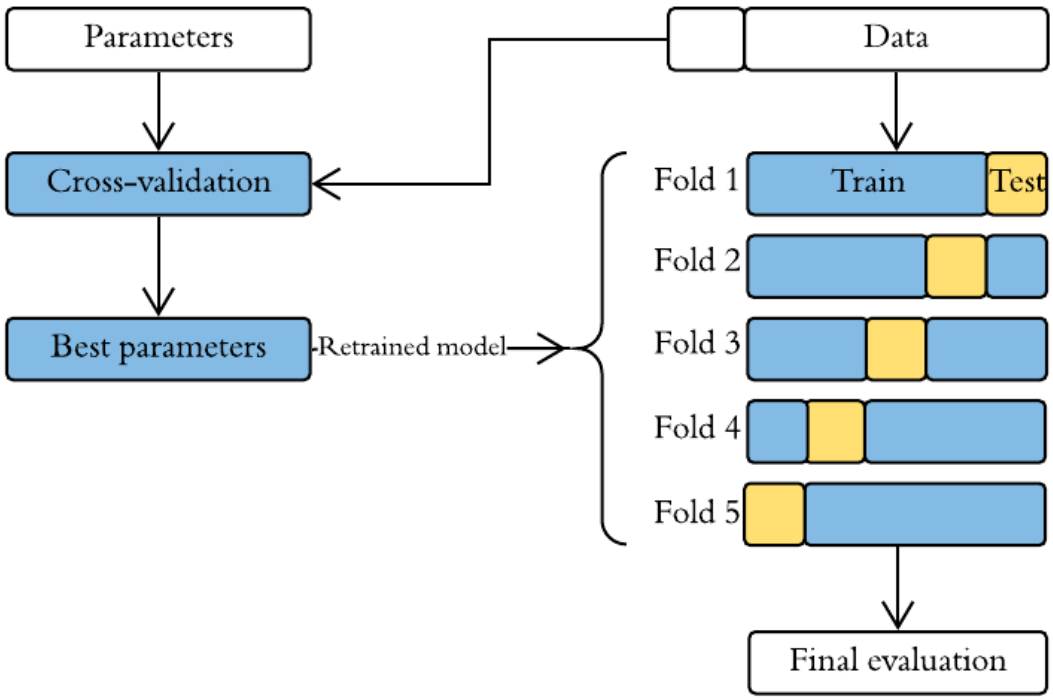
Transit

Region

Test mode: 10-fold cross-validation

APPENDIX L

The k-fold cross validation setup for parameter tuning, training and testing.



APPENDIX M

R code experimental setup

```r
#import data
alldata <- read.csv("C:/Users/CAPE/OneDrive/Master Thesis/Data/alldata10.csv", header = TRUE, sep = ";")
#read data
str(alldata)
head(alldata)

#convert datatype from NUM to NOM
alldata$ArrDay <- as.factor(alldata$ArrDay)
alldata$ArrWeek <- as.factor(alldata$ArrWeek)
alldata$DepDay <- as.factor(alldata$DepDay)
alldata$DepWeek <- as.factor(alldata$DepWeek)

#TRAINING THE ALGORITHM USING 5 FOLD CROSS VALIDATION
#shuffle data before splitting into folds
alldata2 <- alldata[sample(nrow(alldata)),]
#check if shulffled correctly
head(alldata)
head(alldata2)

#reset row index (ascending) #-> to check if CV split is made
rownames(alldata2) <- seq(length=nrow(alldata2))
head(alldata)
head(alldata2)

#initialize
library(randomForest)
library(ggplot2)
set.seed(seed)
library(Metrics)
folds <- 5
k <- c(1,2,3,4,5)
rftot <- 0
msetot <- 0
slice <- ceiling(nrow(alldata)/folds)
```

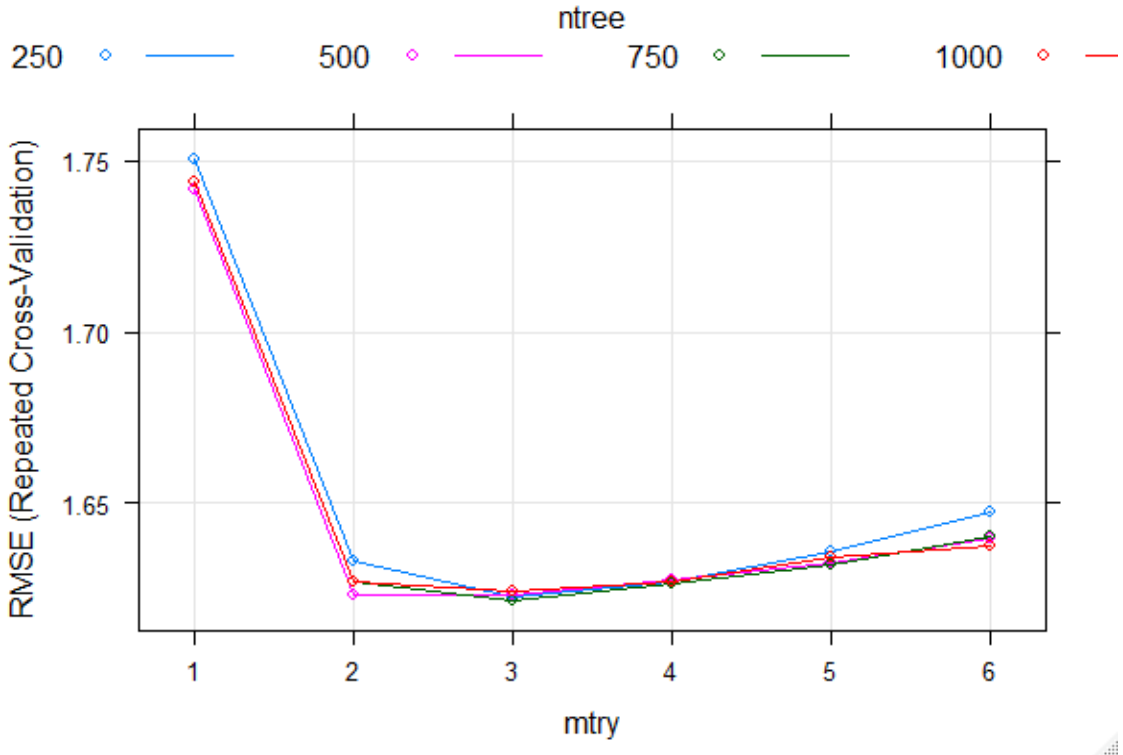
```r
for (i in k) {
  #divide shuffled dataset into 5-folds for cross validation (0.80 train - 0.20 test)
  test <- alldata2[c((slice*(i-1)+1):(slice*i)),c(1:8)]
  train <- alldata2[-c((slice*(i-1)+1):(slice*i)),c(1:8)]

  #TRAIN THE MODEL
  rf <- randomForest(DeltaAtaSta ~ ArrDay + ArrWeek + DepDay + DepWeek + Carrier + PoD, data = train, mtry = 3, ntree = 750, importance = TRUE)
  #displays the Rsquared of the ith fold
  r2 <- tail(rf[["rsq"]], n = 1)
  print(append("r2",r2))
  #displays the MSE
  mse <- tail(rf[["mse"]], n=1)
  print(append("mse",mse))
  #displays the correlation
  print(append("cor", cor(rf$predicted, rf$y)))
  #plot predicted vs actual values
  print(ggplot(train, aes(x =rf$predicted, y = rf$y, color = ArrWeek)) + geom_abline() + geom_jitter() + xlab(paste("Predicted delta fold",i)) + ylab("Actual delta"))
  #plot graph for residuals
  residual <- rf$y - rf$predicted
  print(ggplot(train, aes(x = rf$predicted, y = residual, color = ArrWeek)) + geom_abline(slope=0) + geom_jitter() + xlab(paste("Predicted delta fold",i)) + ylab("Actual delta"))
  #write results of ith fold to csv file
  write.table(rf$predicted, file = "trainRF5", append = TRUE, row.names = FALSE, col.names = FALSE)
  write.table(rf$y, file = "trainRF5", append = TRUE, row.names = FALSE, col.names = FALSE)
  #calculates average R-squared of 5 folds from training
  rftot = rftot + r2
  #TEST THE MODEL
  pred <- predict(rf, test, predict.all = TRUE)
  #write prediction results of new data from test set to file
  write.table(test, file = paste("actualvalues test",i), row.names = TRUE, col.names = TRUE)
  write.table(pred, file = paste("test",i), row.names = TRUE, col.names = TRUE)
  #plot predicted vs actual values
  print(ggplot(test, aes(x =pred$aggregate, y = test$DeltaAtaSta, color = ArrWeek)) + geom_abline() + geom_jitter() + xlab(paste("Predicted delta fold",i)) + ylab("Actual delta"))
  #plot graph for residuals
  residualtest <- test$DeltaAtaSta - pred$aggregate
  #plot(pred$aggregate, residualtest, xlab = "Predicted delta", ylab = "Residuals")
  #abline(h=0, col="blue")
  print(ggplot(test, aes(x = pred$aggregate, y = residualtest, color = ArrWeek)) + geom_abline(slope=0) + geom_jitter() + xlab(paste("Predicted delta fold",i)) + ylab("Actual delta"))
  #remove NAs if needed
  a <- na.omit(test$DeltaAtaSta)
  b <- na.omit(pred$aggregate)
  #print R and MSE from test set without NAs
  print(append("cor test", cor(a, b)))
  print(append("testmse", mse(a,b)))
}
print(rftot/folds)
```

APPENDIX N

R code parameter tuning

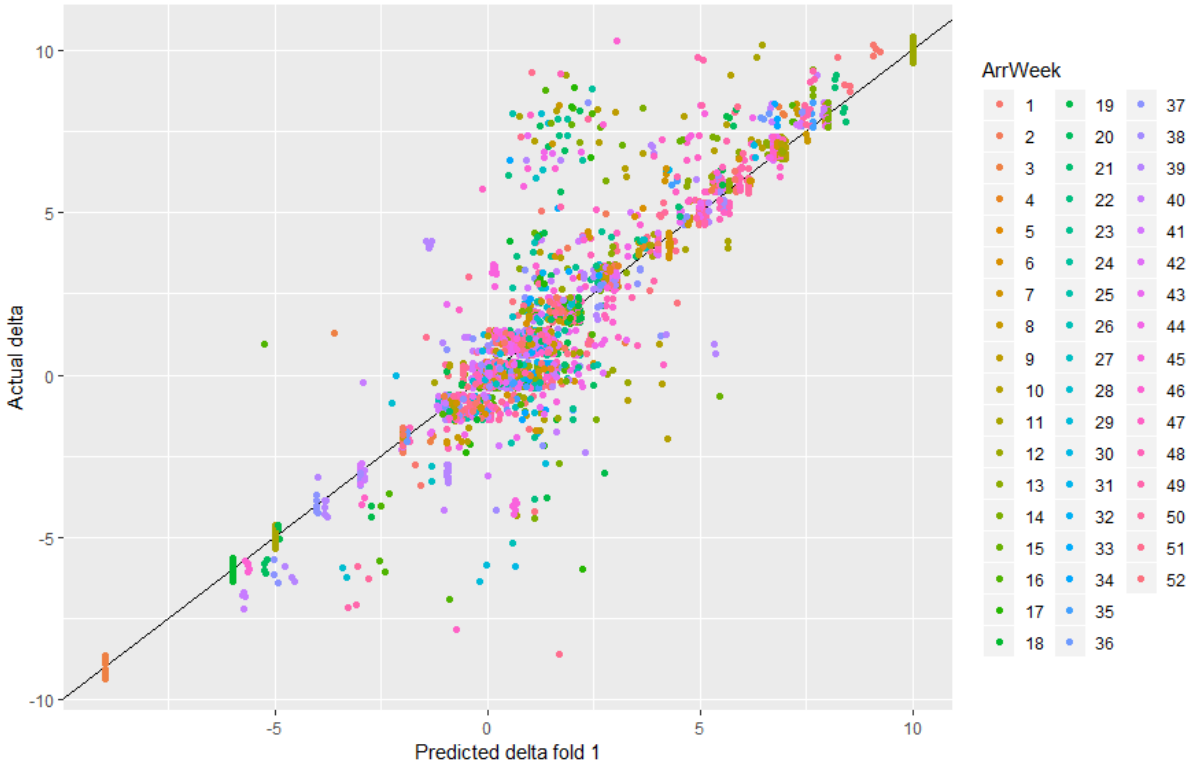
```r
#create a test for parameter tuning
library (randomForest)
library(mlbench)
library(caret)

#create model with default parameters
control <- trainControl(method ="repeatedcv", number = 10, repeats = 3)
seed <- 7
metric <- "rmse"
set.seed(seed)
mtry <- floor(sqrt(ncol(alldata)))
tunegrid <- expand.grid(.mtry=mtry)
rf_default <- train(DeltaAtaSta~., data= alldata, method = "rf", metric = metric, tuneGrid = tunegrid, trControl = control)
print(rf_default)

#create customized RandomForest for tune manually
customRF <- list(type = "Regression", library = "randomForest", loop = NULL)
customRF$parameters <- data.frame(parameter = c("mtry", "ntree"), class = rep("numeric", 2), label = c("mtry", "ntree"))
customRF$grid <- function(x, y, len = NULL, search = "grid") {}
customRF$fit <- function(x, y, wts, param, lev, last, weights, classProbs, ...){
  randomForest(x, y, mtry = param$mtry, ntree=param$ntree, ...)
}
customRF$predict <- function(modelFit, newdata, preProc = NULL, submodels = NULL)
  predict(modelFit, newdata)
customRF$prob <- function(modelFit, newdata, preProc = NULL, submodels = NULL)
  predict(modelFit, newdata, type = "prob")
customRF$sort <- function(x) x[order(x[,1]),]
customRF$levels <- function(x) x$classes

# use customized RF as method to train algorithm for finding optimal parameter settings
control <- trainControl(method="repeatedcv", number=10, repeats=3)
tunegrid <- expand.grid(.mtry=c(1:7), .ntree=c(250, 500, 750, 1000))
set.seed(seed)
custom <- train(DeltaAtaSta~., data= alldata, method=customRF, metric=metric, tuneGrid=tunegrid, trControl=control)
summary(custom)
plot(custom)
print(custom)

```

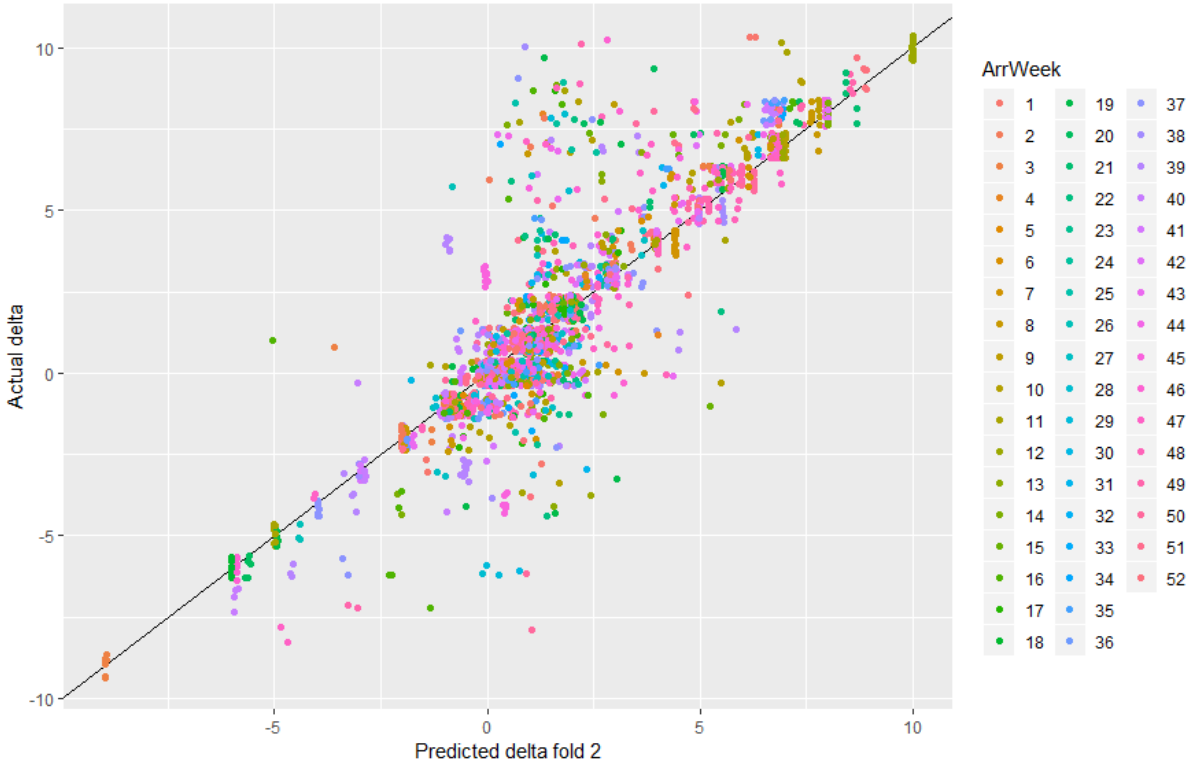
```
No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 1792, 1791, 1793, 1790, 1793, 1790, ...
Resampling results acrosvs tuning parameters:

  mtry  ntree  RMSE      Rsquared   MAE
  1      250   1.750364  0.7356944  1.0956155
  1      500   1.741503  0.7358945  1.0811818
  1      750   1.743872  0.7362318  1.0846178
  1     1000   1.743965  0.7363699  1.0863844
  2      250   1.632829  0.7477807  0.8807892
  2      500   1.623392  0.7511253  0.8758641
  2      750   1.626929  0.7498391  0.8780842
  2     1000   1.626971  0.7498715  0.8785970
  3      250   1.622664  0.7497092  0.8610310
  3      500   1.623391  0.7493993  0.8628481
  3      750   1.621741  0.7499266  0.8609223
  3     1000   1.624239  0.7491131  0.8624207
  4      250   1.626985  0.7476161  0.8590629
  4      500   1.627375  0.7474587  0.8594737
  4      750   1.626530  0.7477300  0.8587408
  4     1000   1.627062  0.7475520  0.8587074
  5      250   1.635691  0.7442464  0.8619679
  5      500   1.632428  0.7453541  0.8586271
  5      750   1.632026  0.7454871  0.8584897
  5     1000   1.634298  0.7447734  0.8607299
  6      250   1.647253  0.7401717  0.8646757
  6      500   1.639456  0.7427972  0.8620552
  6      750   1.640347  0.7425206  0.8623544
  6     1000   1.637653  0.7434316  0.8604939

RMSE was used to select the optimal model using the smallest value.
The final values used for the model were mtry = 3 and ntree = 750.
```

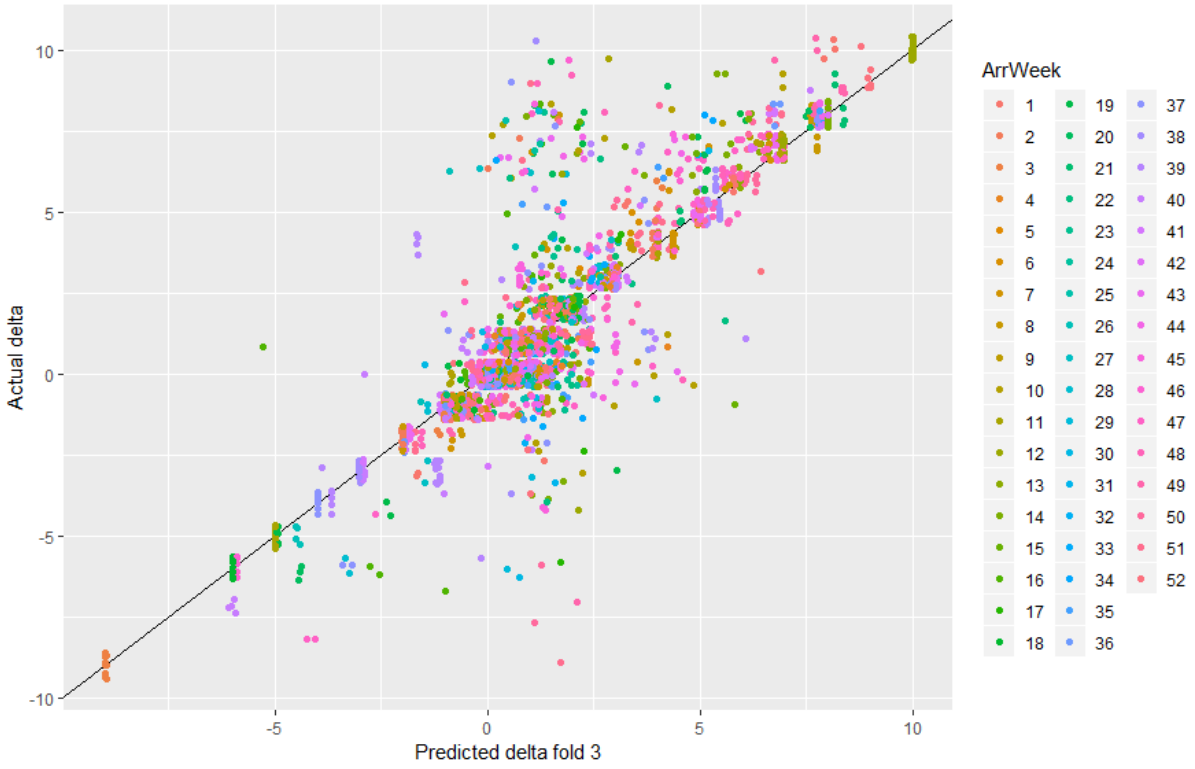


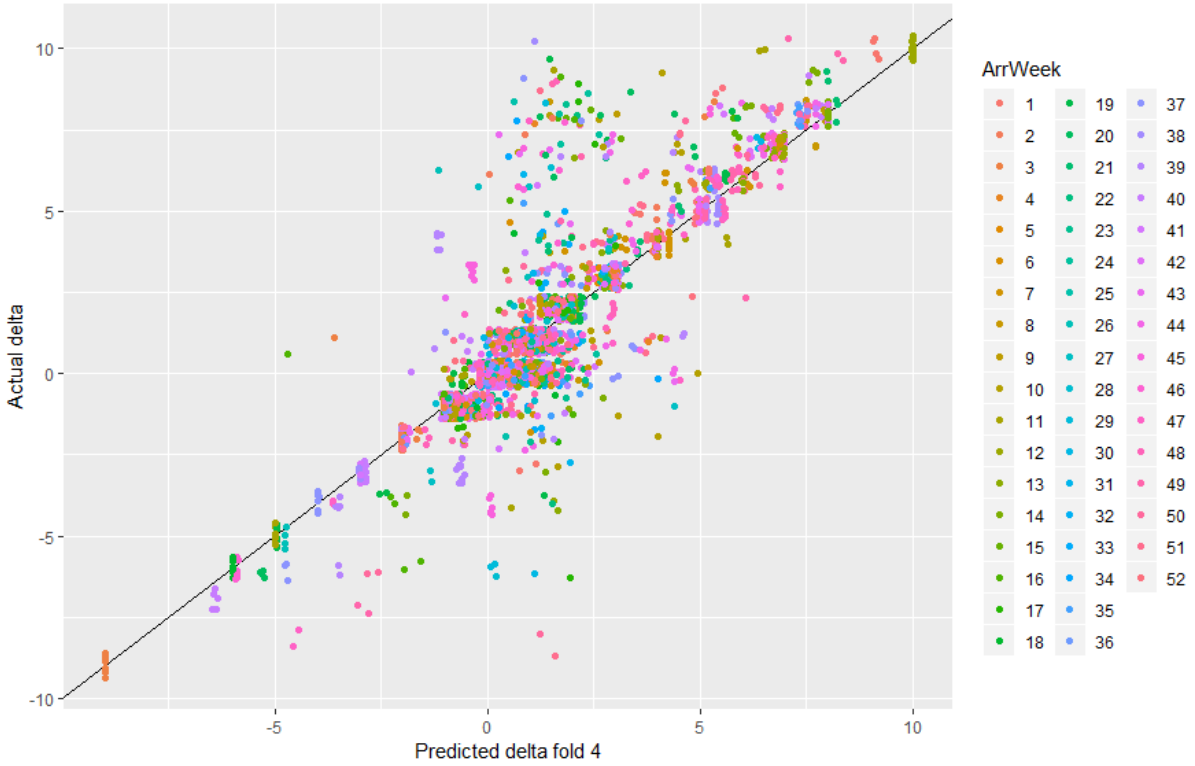
APPENDIX O

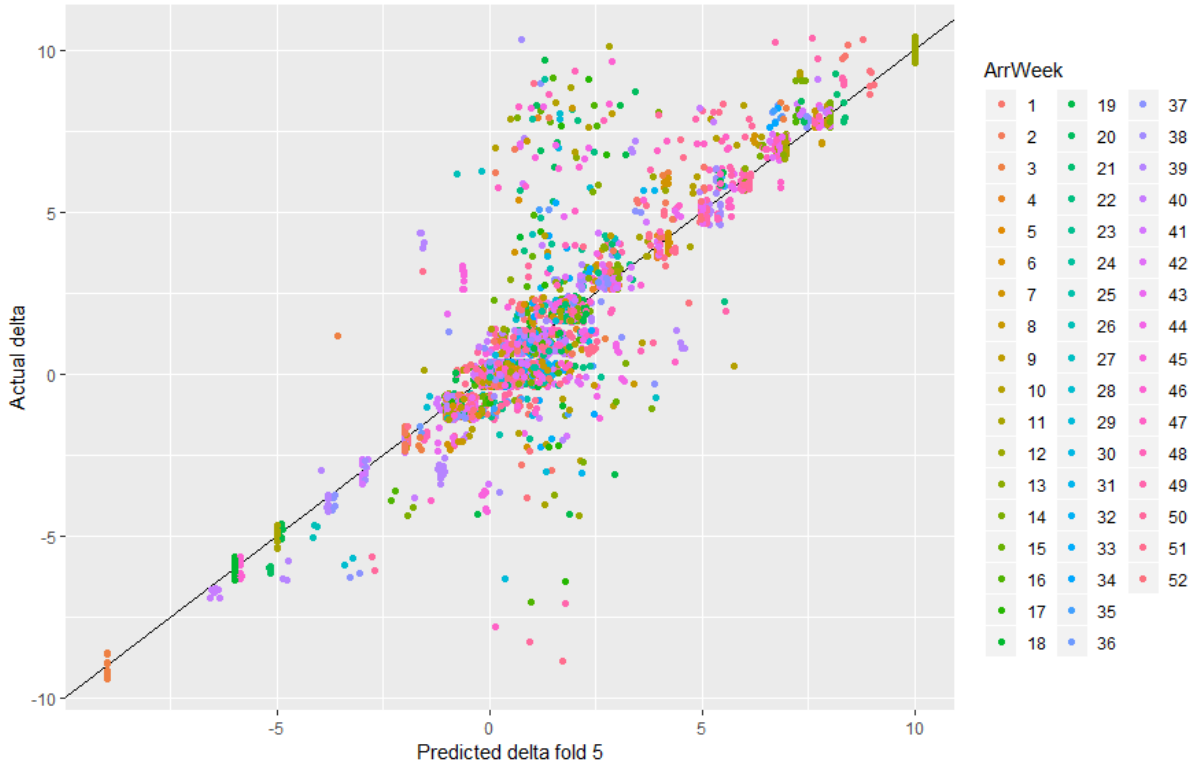
Scatter plots of all folds of train set





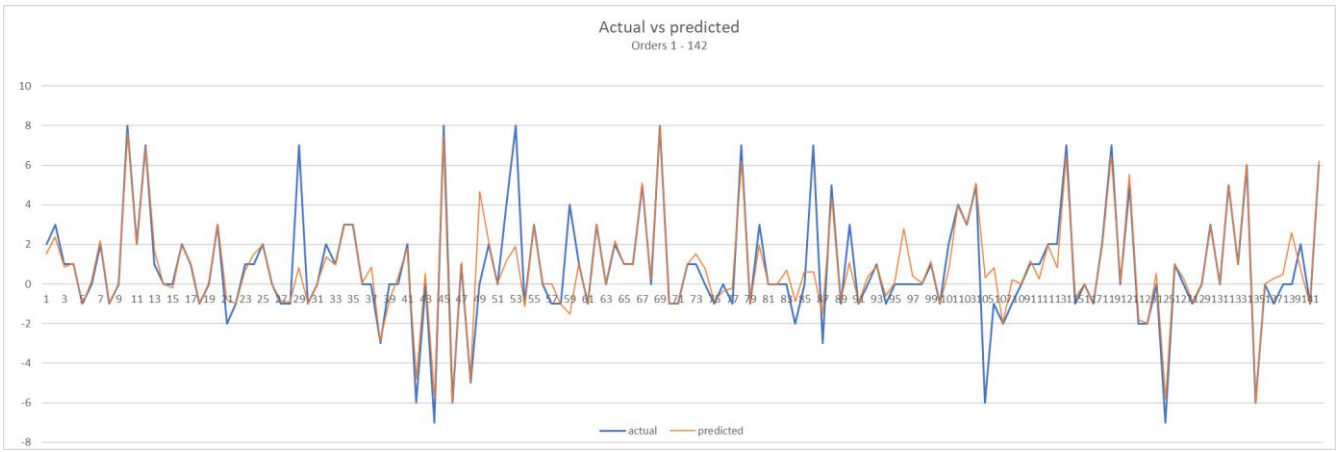

Actual delta
Predicted delta fold 3
ArrWeek
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52

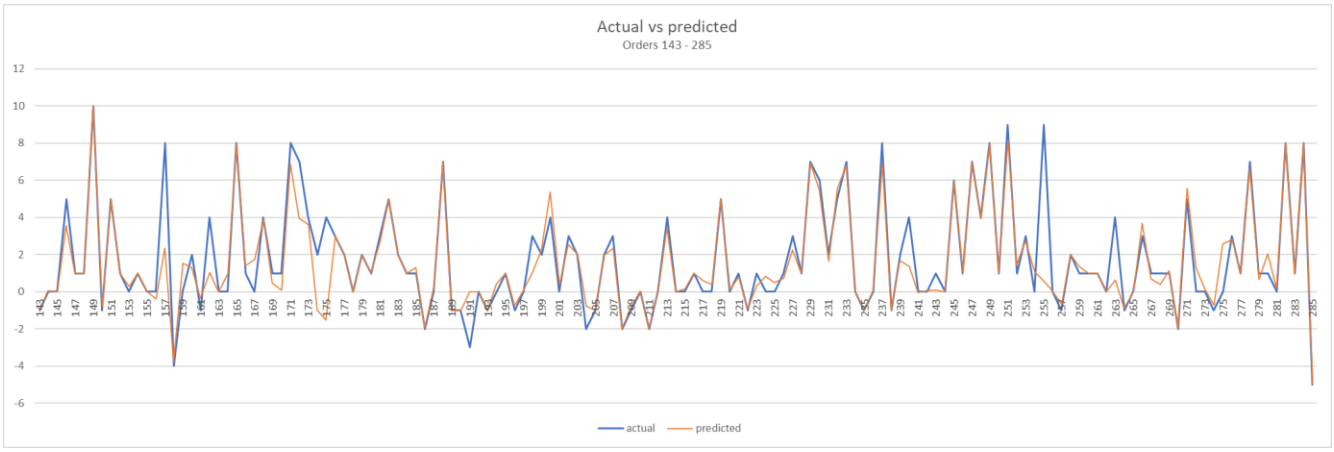

Actual delta
Predicted delta fold 4
ArrWeek
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52

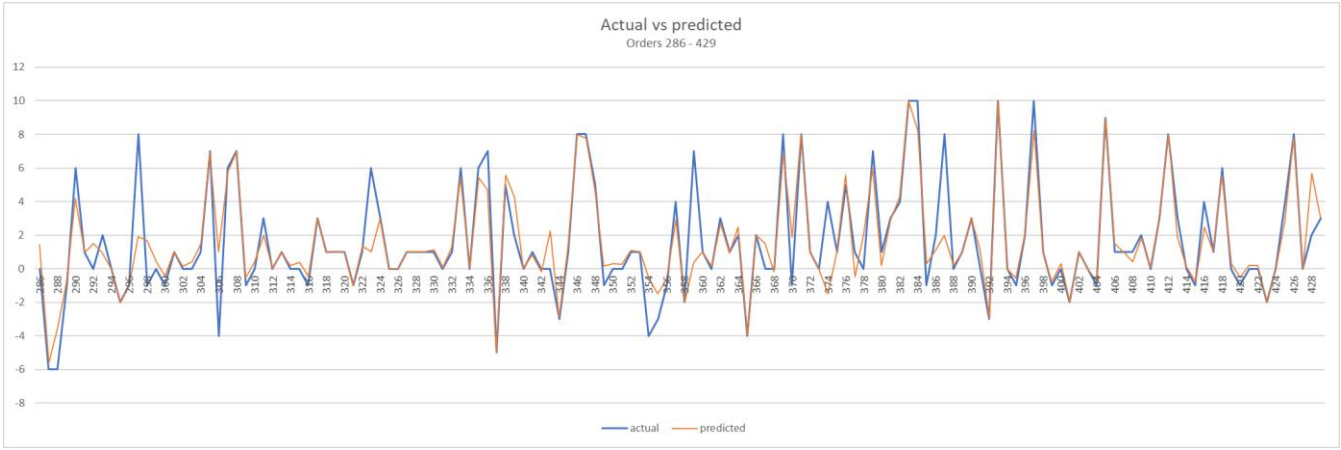

Actual delta
Predicted delta fold 5
10
5
0
-5
-10
-5
0
5
10
ArrWeek
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52

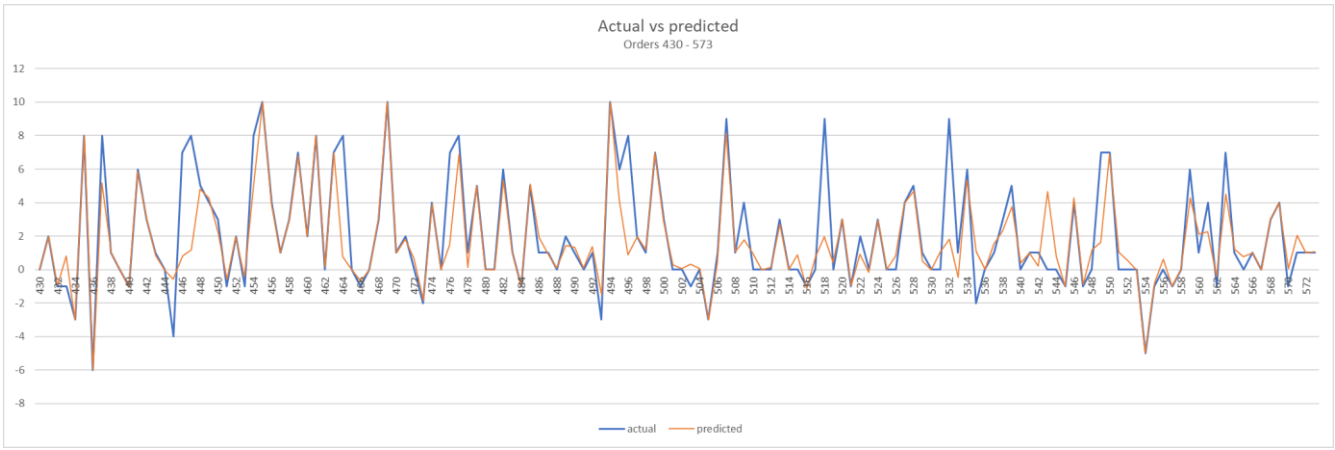
APPENDIX P

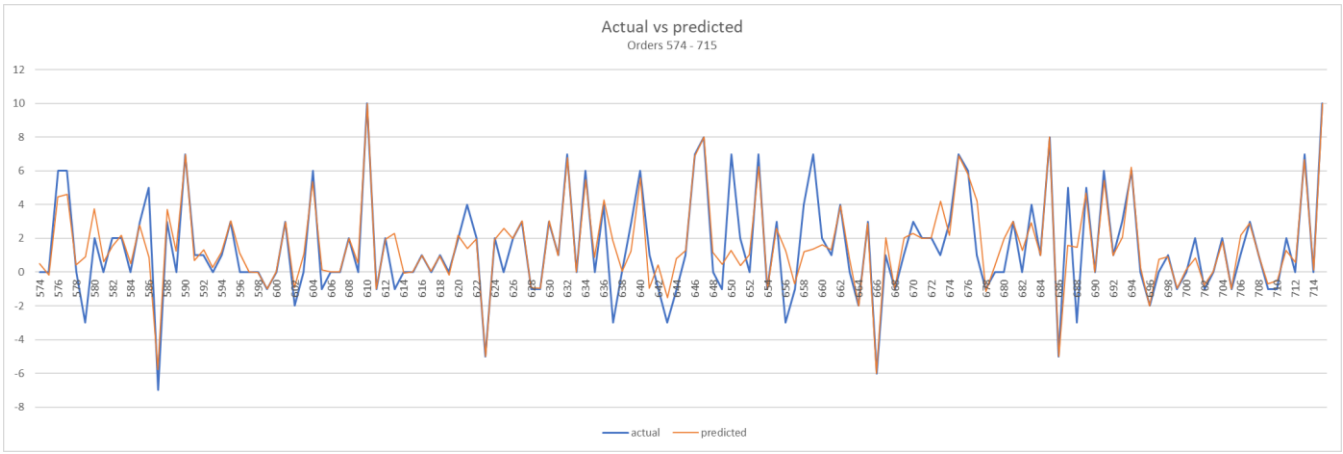
Actual versus predicted values test sets








Actual vs predicted
Orders 430 - 573
actual
predicted


Actual vs predicted
Orders 574 - 715
actual
predicted

APPENDIX Q

Example demurrage fees

APL

Demurrage and detention tariff
SINGAPORE
Import/Export : IMPORT

Currency : SGD
Freetime in : Calendar
Places : All
Excess time : Calendar

Type	Size	After Freetime Day	Freetime	Charge	Start Validity	Charge Type
GP	20	From 4th To 5th	3	30	02-JAN-18	Demurrage
GP	20	From 6th Onwards	3	50	02-JAN-18	Demurrage
GP	20	From 4th To 5th	3	30	02-JAN-18	Detention
GP	20	From 6th Onwards	3	50	02-JAN-18	Detention
GP	40	From 4th To 5th	3	60	02-JAN-18	Demurrage
GP	40	From 6th Onwards	3	100	02-JAN-18	Demurrage
GP	40	From 4th To 5th	3	60	02-JAN-18	Detention
GP	40	From 6th Onwards	3	100	02-JAN-18	Detention
RF	20	From 4th To 7th	3	90	02-JAN-18	Demurrage
RF	20	From 8th Onwards	3	180	02-JAN-18	Demurrage
RF	20	From 4th To 7th	3	90	11-DEC-16	Detention
RF	20	From 8th Onwards	3	180	11-DEC-16	Detention
RF	40	From 4th To 7th	3	110	02-JAN-18	Demurrage
RF	40	From 8th Onwards	3	220	02-JAN-18	Demurrage
RF	40	From 4th To 7th	3	110	11-DEC-16	Detention
RF	40	From 8th Onwards	3	220	11-DEC-16	Detention

The selected lines are applicable to our research as the container where the LSP ships the frozen potato products in, is a RF (cooling container) of 40 ft.

APPENDIX R

Microflow in Mendix for ETA prediction algorithm