November 23, 2022

Delft University of Technology

2022.TIL.8734

Mike van der Meer 4456750

Machine learning applications in supply chain management: A case study at MPO

Preface & acknowledgements

This report has been realized as a component of a masters thesis for the program Transport, Infrastructure and Logistics. Which is a masters program for the university of Delft and faculty of Civil Engineering and Geosciences.

I would like to thank the following supervisors from the TU Delft, Dr. J.M. Vleugel, Ir. M.B. Duinkerken and Prof.dr. R.R. Negenborn for their guidance during the project. In addition, I would also like thank my supervisors from MPO, Paul van Dongen and Martin Verwijmeren. Furthermore, I would like to thank Nick Roggeveen for his guidance, encouragement and kind words during our weekly meetings. Also, I would like to thank the interviewees who contributed with their knowledge and expertise, as well as everyone who made time available at MPO and outside of MPO for my many questions. Finally, I would like to thank my girlfriend who supported me a lot over the last months during the work on my thesis.

Mike van der Meer

November 23, 2022

Executive summary

The costs for logistics has been increasing over the last few years, especially since the pandemic. This is felt in many supply chains and therefore it is important to explore avenues to reduce these costs. In this research a case study will be performed at MPO. MPO is a supply chain management company that would like to improve their services through the use of machine learning. Accordingly, in this research the main research question that will be answered is: "How can supply chain management be improved through the use of machine learning?". This study is split into five sub-questions where the first two sub-questions are aimed at the literature of supply chain management and machine learning respectively. The third sub-question dives into the case of MPO and selects an element, which will be looked at further in the research. To answer sub-question four, a conceptual model for the chosen element is made. The fifth and final sub-question measures the impact of conceptual model variants on the results of experiments performed on the selected element.

In order to discuss what supply chain management is, it is first important to establish what a supply chain is. A supply chain is a set of three or more entities directly involved in the up- and downstream flows of products, services, finances and/or information from a source to a customer. To manage such a supply chain eight different key businesses processes are defined, which are:

- Customer relationship management
- Customer service management
- Demand management
- Order fulfillment
- Manufacturing flow management
- Procurement / supplier relationship management
- Product development and commercialization
- Return logistics

Of these eight key business processes MPO is mostly involved with order fulfillment. This process is defined as beginning with receiving a customer order and finishing with the final items being delivered.

Machine learning on the other hand is a computing process that utilizes input data to accomplish a goal without explicitly being written to do so. These algorithms adjust or adapt their architecture as a result of repetition to get better and better at executing their specific task. This process of adaptation is known as training, and it involves providing samples of input data along with desired outputs. The algorithm then optimizes its configuration such that it can not only provide the intended result when given the training inputs, but also generalize to achieve the desired result when given new, previously unseen data. There are three categories of machine learning and which is best used depends on the training data. These categories are supervised learning, where the data is labelled, semi-supervised learning, where the output is partly labelled, and unsupervised learning where unlabelled data is used.

As mentioned earlier, MPO performs the process of order fulfillment. MPO does this process for their clients through the use of their platform. These clients, which are themselves companies, have customers who place

orders for products. The customer orders that are placed are relayed to the MPO platform. Through a set of steps, outlined in chapter 4, a path is determined, a carrier is selected and contacted and an estimated arrival time is provided. MPO's clients have visibility over the whole process from the arrival of a customer order to the confirmation by the carrier that the delivery has been completed. Within the scope of MPO five possible elements were encountered, where possible improvements could be made through the use of machine learning. These elements were found through a set of interviews (see appendix B) and analysis of the platform. These topics were:

- On-time estimation
- Order cost estimation
- Carrier performance
- Return of product
- Demand forecast

The data for order cost estimation and product returns was unsatisfactory. For carrier performance it was unclear why machine learning should be used, alternative methods would most likely achieve similar or better results. Demand forecast and on-time estimation were both strong candidates, however, the wider applicability of demand forecast is questionable. The market a client would operate in would have a very large effect on the importance of some variables. Important variables for the demand forecast in one market might be useless in another. Therefore on-time estimation was selected to be the topic for further research in this study. Previous work on on-time estimation used almost exclusively regression and classification as methods to solve this problem, which both fall under the machine learning category of supervised learning. Moreover, every paper exclusively focused on a single mode for the research. Furthermore, often the weather and traffic data was used and a single route and origin was assumed. In this research the mode will not be assumed and neither will weather and traffic be considered, since that data is not available.

The basis for the conceptual model was found in the previous work for on-time estimation. The conceptual model can be viewed in figure 1. There were many variables that could be significant when trying to predict whether a delivery or transportation will be on time or not. Along with the previous work and interviews with the experts, a list of relevant variables was made. Since this was an extensive list it would have to be filtered down in order to narrow the amount of variables down to a reasonable amount. Since most of the variables were categorical this meant that the filtering methods had to be able to accommodate that. For that purpose the chi-squared test and mutual information were chosen. Afterwards a wrapper will be applied that uses recursive feature elimination. Similarly to the previous work, classification and regression will be used as methods to solve on-time estimation. Classification is a method that outputs a class, which in this case will be either on-time or not on-time. Alternatively, regression is a method that outputs a number. The algorithms used to perform these methods of machine learning are random forest and neural network. These algorithms were selected based on the rankings provided by Kotsiantis, Zaharakis, Pintelas, et al. (2007) and three criteria which were:

- Accuracy
- Speed of classification
- Explainability and transparency

Since accuracy and explainability were (almost) mutually exclusive one algorithm was chosen for both sides of the spectrum. For explainability random forest was chosen and for accuracy neural network.

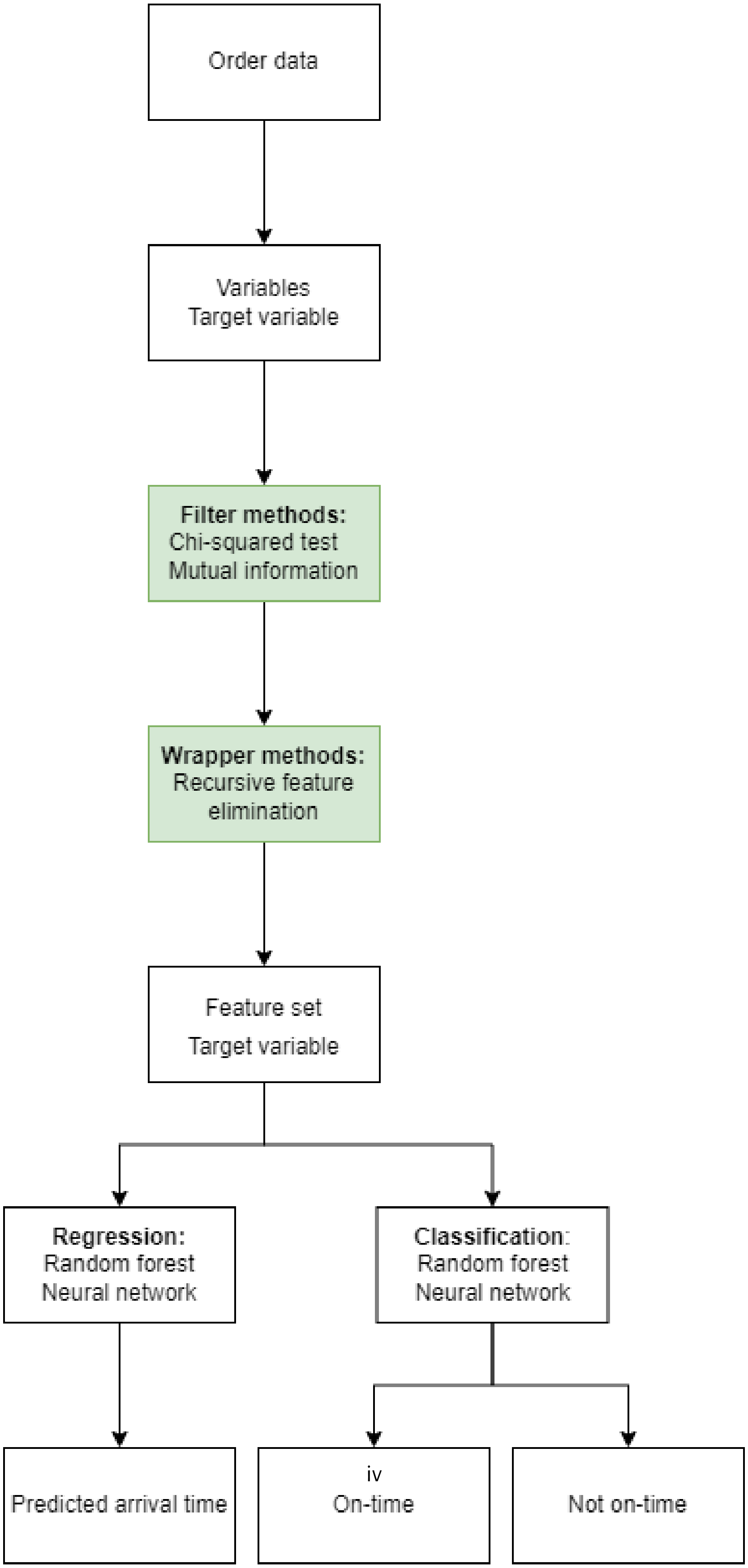


Figure 1: The filled in conceptual model.

A set of variables was derived from the order data based on the conceptual model discussed previously. On these variables the process of feature selection was performed. Both of the aforementioned algorithms have a set of parameters that can be chosen which alter the performance and thereby the results of the algorithm. For the experiment plan the one factor at a time approach (OFAT) was used due to constraints in computation power and time. For this approach one parameter is altered whilst the rest are kept constant in order to measure the impact of a parameter on the result. Therefore, a baseline of parameters had to be established. The experiments are used to determine improved values for the parameters relative to the baseline. For neither classification nor regression the baselines of the algorithms were significantly improved by the adjusted variants. The first reason for this is that the experiment approach, OFAT, does not measure the relation between the parameters, thus adjusting multiple parameters at the same time for the adjusted variant did not yield significantly better results. Secondly, in the individual experiments the baseline values for the parameters already performed rather well, being very close to the best values for the parameters. Therefore, it can be concluded that the baseline chosen was relatively good compared to the other tested values for the parameters. For classification the results were around 72-73% accuracy, which might still need some work for broad use in practice. Supply chains with expensive shipments or where it is very important that deliveries are made on-time still might make use of it. However, clearly a relation was found between the features and the target variable, otherwise results would have been a lot closer to an accuracy of 50%, which would be random. For regression on the other hand, the MAE, expressed in hours, was between four and five. In more explainable terms, a time-frame could be created of six hours around the prediction which would contain 70% of the predictions. This time-frame could be useful to communicate to customers to give a better indication when the delivery can be expected.

The main research question of this thesis is: "How can supply chain management be improved through the use of machine learning?". Since it is impossible to solve all the problems of the field of supply chain management in the course of a thesis, a number of elements were discussed and one of these was selected. A model was proposed with the aim of improving this element and said model was executed. The results as discussed before were two-fold in the form of classification and regression. Although classification and regression stem from the same idea of on-time estimation their applications are different. Namely, the output of the classification model (binary) is too limited in information to be something that could be communicated to a client. It could, however, be very useful as an internal feedback loop of the planning system to correct it or signal that someone needs to look at a certain planning. The regression model, on the other hand, could be used to replace the projected arrival date or to replace it with a time-window instead with a certain confidence value. Moreover, there are angles that can be explored to further improve the results of these models. One avenue for further research is to perform experiments combining different sets of parameters, which especially for the neural network might affect results significantly. Furthermore, using larger sample sizes that used in this research might also affect the results. More data could allow the algorithm with more cases to learn, but could also result in overfitting.

All in all, a model was proposed which resulted in results for on-time estimation. This model could be applicable to other data sets within MPO but also to other logistics companies who have the data available outlined in the conceptual model. Moreover, this model and the approach taken to determine this model could be applied to different elements within supply chain management to improve other elements.

Contents

Chapter 1

Introduction

1.1 Background

Supply chains have been around for thousands of years. The first well-documented global supply chain is over two thousand years old and it was called the silk road (History, 2021). The supply chain received its apt name from the silk trade, where the silk originated from China and was brought to markets as far as the Roman Empire through the ports on the eastern Mediterranean sea. This lasted for about 1500 years until the fall of Constantinople in 1453 to the Ottomans who then boycotted trade with China (History, 2021). This boycott is one of the reasons for seafaring European nations such as Portugal to take to the seas in the pursuit of trade. In 1488 Bartolomeus Diaz rounded the now called Cape of Good Hope on the southern tip of Africa for the first time and by doing so found a new way to Asia (Randles, 1988). This route increased the efficiency of the supply chain for silk, spices and other goods by allowing to carry it in larger bulks and in shorter trips. This was further decreased by opening a new shorter passage, the Suez Canal, between the Indian Ocean and the Mediterranean in 1869 (Bogaars, 1955). There were also technological advancements over time which allowed for the trade in spices and silk to pick up. The boats used for the transportation of goods slowly changed from wind-powered to steam-powered and finally to the engine-powered ships that are used today. Supply chains throughout history have come a long way. From crossing the entirety of Asia with a caravan of camels to sailing through the Suez canal with a container ship with thousands of containers. With these technological advancements the costs to bring goods from one place to another has ever been decreasing over time.

The world is becoming increasingly connected by the year, and it is shown for example through the amount of packages that are ordered and delivered. In the year 2021 for example, PostNL delivered a record number of 384 million packages in a country (Netherlands) with a population of only 17 million people. The amount of deliveries increased by 14% compared to 2020 (NOS, 2021). The COVID pandemic, however, heavily impacted supply chains across the world, due to the disruptions in trade that were caused. Many countries enforced stricter customs or shut their borders entirely. As a result in late 2021 77% of the worlds largest ports still faced backlogs (Blake, 2021). In the United States there is a shortage of 80.000 truckers which could more than double by 2030 Duffy (2021). This is felt by retailers, since they have to pay 30% more to move goods by truck in 2020 relative to 2019 Premack (2020). The cost of shipping a 40-foot container from Shanghai to Los Angeles, for example, has increased by 75% in a year and is expected to continue rising in 2022 (Smith, Berger, & O'Neal, 2021). These are merely a few examples for the increases in logistics cost. It is therefore time for supply chains to use the newest technologies available to suppress the rise in logistics cost, in order to stay ahead of or in line with the competition.

1.2 Problem statement

Multi-Party Orchestration (MPO) is a company that aids in this process by providing a supply chain cloud platform for its customers that is used for supply chain management (SCM). Supply chain management is the integration of key business processes across the supply chain, more on this definition in chapter 2. MPO facilitates this process by organizing incoming, outbound and reverse flows across multiple parties in dynamic supply chain business networks. Through real-time analytics, planning, execution and cost control businesses can boost both customer experience and operational excellence through the usage of the platform. At the heart of MPO is the customer chain control. This system creates a micro supply chain for every individual customer order to provide continuous, real-time control over service levels, inventory, transportation and costs. How MPO works and what it does is further elaborated upon in section 4.1. Now, MPO would like to further improve their services by integrating machine learning techniques into their platform to better serve their customers.

1.3 Scope

This research is conducted at MPO which is a supply chain management company, as previously described, and is the studied system of this research. As per their request, the premise of this research is that the method used to improve an element of the MPO system is machine learning. The element that is to be improved is found through the use of interviews and analysis of the existing system. These elements will then be evaluated based on whether machine learning could facilitate improvement. One element will then be selected for further study in this research. The data in order to perform this machine learning will be provided by one of MPO's largest clients. This data will be thoroughly examined and finally used to train a machine learning model, in order to attempt to improve the performance of the chosen element.

1.4 Relevance of study

Although there are papers being published about machine learning with respect to supply chain management, the possibilities are not being fully exploited and the attitude for new research is too conservative (Ni, Xiao, & Lim, 2020). Moreover, Bertolini, Mezzogori, Neroni, and Zammori (2021) discuss that in terms of ML applications, supply chain management is not a domain that has been much explored. Furthermore, according to Wenzel, Smit, and Sardesai (2019) further research could be done to examine the relevant literature to determine which machine learning algorithms are most appropriate for certain SCM tasks. The gap addressed by this research will therefore be to gain insights in how machine learning can be used to improve supply chain management.

1.5 Research objective

By combining the problem statement as provided by MPO with the research gap encountered in the literature, the following research question can be formulated:

How can supply chain management be improved through the use of machine learning?

The main research question is split up in a few sub-questions in order to incrementally define the progress of the project. These sub-questions are defined as follows:

1. What is supply chain management?
2. What is machine learning?

3. What are the characteristics of the studied system and what are elements of the system where improvements could be made?
4. What would a model look like that would solve the problem of the chosen element?
5. What is the impact of these model alternatives on the performance of the chosen element?

The research starts with a very wide angle from the entirety of supply chain management which answers the first sub-question. Afterwards, machine learning is taken a closer look at as well as its relation to supply chain management, which answers the second sub-question. In the following chapter, the studied system is discussed. In this chapter MPO is placed within the scope of supply chain management. Furthermore, elements where improvements could be made within MPO are discussed. These elements are found through interviews with experts within MPO as well as through analysis of the studied system. In addition, one of these elements is selected. To answer sub-question four different machine learning methods will be discussed that can improve the performance of the chosen element. Finally sub-question five is answered by employing different machine learning methods and comparing the results.

1.6 Research approach and document outline

First, it is necessary to establish the context for the research, which is supply chain management. This occurs in chapter 2. Following that, a brief overview of machine learning will be given, in chapter 3. In the next chapter, which is chapter 4, the studied system will be discussed. In addition, various elements for improving the studied system will be examined in this chapter. These elements were brought up during interviews with experts. Furthermore, an element will be chosen and the relevant literature for that element will also be discussed. Different methods for improving the performance of the chosen element are proposed and selected in chapter 5. Afterwards, in chapter 6 the results of the selected methods are described and compared. Finally, in chapter7 there is a conclusion that answers the main research question. Furthermore, the chapter includes a discussion and recommendations for future research.

Chapter 2

Literature study: Supply chain management

In this chapter sub-question one will be answered which is: "What is supply chain management?". In order to answer this question it is first important to establish what a supply chain is. This will be done in section 2.1, afterwards will be discussed what supply chain management is in section 2.2. The following section, section 2.3 goes into the relations between the different processes that SCM consists of. Finally in section 2.4.

2.1 Supply chain

Supply chains is a broad field, with that come many different definitions of what it is. The paper by Mentzer et al. (2001) made an effort to come to a unified supply chain definition based on previously existing definitions. For this paper the definition provided by Mentzer et al. (2001) will be used, which is:

A supply chain is defined as a set of three or more entities (organizations or individuals) directly involved in the upstream and downstream flows of products, services, finances and/or information from a source to a customer.

Within this concept, there are three levels of supply chain complexity: direct supply chain, extended supply chain and ultimate supply chain. These three types are also illustrated in figure 2.1. A direct supply chain is made up of a company, a supplier, and a customer who are involved in the upstream and/or downstream movement of goods, services, funds and/or information. An extended supply chain comprises suppliers to immediate supplier as well as customers to the immediate customer, all of whom are involved in the upstream and/or downstream flows of products, services, funds, and/or information. An ultimate supply chain consists of all companies involved in the upstream and downstream flows of products, services, funds, and information from the ultimate supplier to the ultimate client. Mentzer et al. (2001)

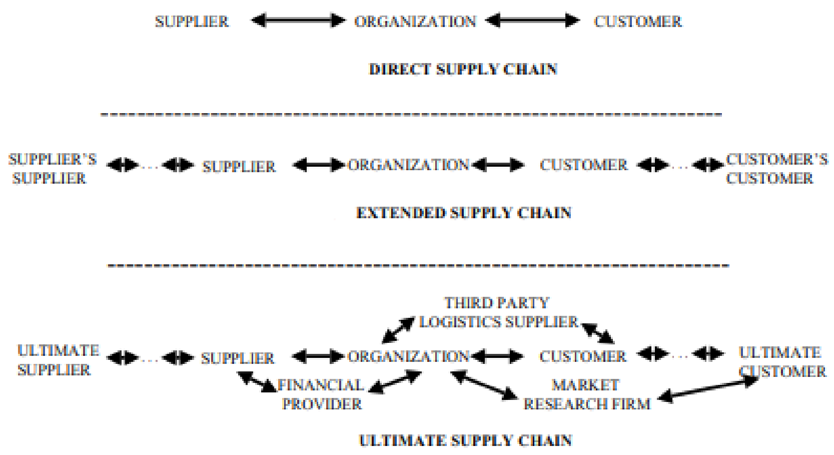


Figure 2.1: Types of supply chain complexity (Mentzer et al., 2001)

2.2 Supply chain management

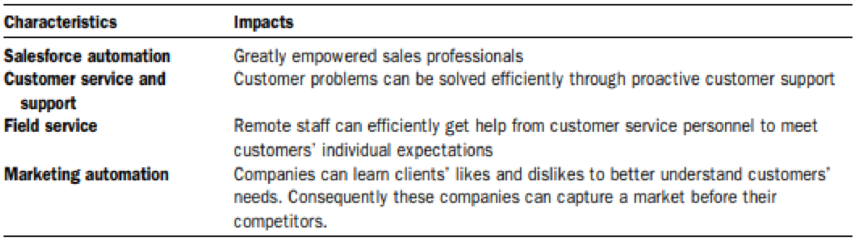
There are many ways to discuss the field of supply chain management. The paper by Mentzer et al. (2001) sets the philosophies next to each other and shows, in their words, a representative sample of the definitions. In this paper the method by Croxton, García-Dastugue, Lambert, and Rogers (2001) will be used. In that paper, supply chain management is described as the integration of key business processes across the supply chain. It furthermore defines eight key processes that make up the core of supply chain management, which are:

- Customer relationship management
- Customer service management
- Demand management
- Order fulfillment
- Manufacturing flow management
- Procurement / supplier relationship management
- Product development and commercialization
- Return logistics

All of these topics will be further elaborated upon in their respective subsection. The following section goes into the relations between these key processes.

2.2.1 Customer relationship management

Customer relationship management (CRM) stands for the methodologies, software, and usually internet capabilities that help a company manage customer relationships in an organized way (Xu, Yen, Lin, & Chou, 2002). The main goal of CRM is how a company can maintain its most profitable customers while also lowering expenses and increase the value of interaction to maximize profits. In order to do so it is important for a company to determine what the important drivers are for current and future customers. Customers can be grouped together based things like age or gender but can also be either a company or individual dependent on the classification method. In order to maintain a good relationship with the customer in order to retain them as customers resources have to allocated accordingly with the wishes and wants of current and future customers. The areas of investment can differ based on the product and or customer group such as customer services, pricing strategies, product development and others. In figure 2.2 the areas in which CRM has its main contributions is shown. By empowering the Salesforce and interfacing well with the customer sales could positively benefit. As a result, this could impact the demand for products.



Characteristics	**Impacts**
Salesforce automation	Greatly empowered sales professionals
Customer service and support	Customer problems can be solved efficiently through proactive customer support
Field service	Remote staff can efficiently get help from customer service personnel to meet customers' individual expectations
Marketing automation	Companies can learn clients' likes and dislikes to better understand customers' needs. Consequently these companies can capture a market before their competitors.

Figure 2.2: The main four characteristics of CRM (Xu et al., 2002).

2.2.2 Customer service management

Customer service management (CSM) is the company's face towards the customer. It serves as a single source of information for customers, such as product availability, shipment dates, and order status. The consumer receives real-time information through interfaces with the firm's company's operations such as manufacturing and logistics (Croxton et al., 2001). An example of this is that webshops like Amazon and Bol send an email once the an order has been received. On top of that, once the package has been sent a track and trace code is usually attached. Through this interface customers can see when their package will arrive as well as if there are delays of any kind.

Customer service is an essential aspect of a business since retention of customers highly depends on it. Namely 32% of customers stop interacting with a brand they love after a single bad experience (Puthiyamadam & Reyes, 2018). On the other hand good customer service can boost sales by 20% or more (Nolinske, 2022). The important aspects with customer service to 'get right', according to a survey by Puthiyamadam and Reyes (2018), are speed, convenience, easy payment, knowledgeable help and friendly service. In this survey 80% of the American respondents pointed to these aspects as being the most important for a positive customer experience. In figure 2.3 the aspects mentioned by respondents of the study can be viewed. On the x-axis level of importance for the customer experience is plotted, whereas on the y-axis which percentage of the respondents thought it was worth paying for that aspect.

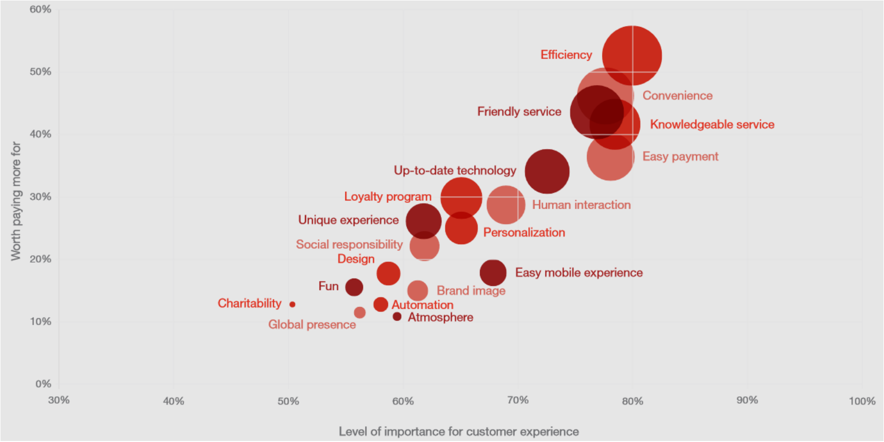


Figure 2.3: The important aspects related to customer service according to a survey by Puthiyamadam and Reyes (2018).

2.2.3 Demand management

Croxton et al. (2001) describes demand management as the method that must balance the wants of the consumers with the firm's supply capabilities. Forecasting demand and synchronizing it with production, procurement and distribution are all a part of this.

Forecasting demand is about predictions of future demand. This is done through the use of models, which require data in order to base those predictions on. The paper by Suganthi and Samuel (2012) describes eleven different models that can be used to forecast energy demand, amongst which regression, econometric models, artificial systems and more. The data for these demand forecast models can be anything relevant to the demand. Usually the data used depends on the chosen model to do the forecasting and the type of product that the forecast is for. Examples of information that can be used to perform such forecasts are historical data, market research, sales projections, promotion plans, market share data, and more Helms, Ettkin, and Chapman (2000). Another important factor to consider when forecasting demand is the time frame for the forecasts. These should be aligned with the wants and needs of the company. A demand forecast for a manufacturer of winter clothing for example can vary wildly from season to season.

Once a forecast is made some form of synchronization has to follow in order to match the demand forecast to the company's sourcing, production capabilities and distribution. Due to the uncertainty associated with forecasts in general it is also important to develop contingency plans in case of events that disrupt the balance of supply and demand either internally or externally. The paper by Towill and McCullen (1999) describes that the supply chain that accomplishes the most in reducing uncertainty and variability is likely to succeed the most in enhancing its competitive position.

2.2.4 Order fulfillment

The order fulfillment process begins with receiving a customers orders and finishes with the final items being delivered. The order fulfillment cycle time is therefore defined as the time it takes from the receipt of an order to the delivery of a product. The order fulfillment process is a complex task, because it is consists of multiple activities that are carried out by different functional entities, and are highly interconnected among the tasks, resources, and agents involved (Davenport, 1993). It has been estimated that up to 80% of the total cost of the final product is determined in the design of the network Harrison (2001). Therefore evaluating the logistics network of order fulfillment has a significant influence on the cost and performance of a system.

The paper by F.-R. Lin and Shaw (1998) outlines the primary activities of the order fulfillment process as follows:

- Order management is the process of receiving orders from customers and committing order requests.
- Manufacturing involves production planning, material planning, capacity planning, and shop floor control.
- Distribution takes into account issues such as inventory and transportation.

The distribution and order management aspect of order fulfillment is the bread and butter of what MPO does and will therefore be more thoroughly discussed in that respective chapter, which is chapter 4.

2.2.5 Manufacturing flow management

Manufacturing flow management is a process that focuses on converting raw materials and components into marketable finished items. The manufacturing flow management process differs from company to company and depends on the choices that the company makes. One pivotal choice that needs to be made is the manufacturing strategy that is to be employed. (Goldsby & García-Dastugue, 2003)

The paper by Goldsby and García-Dastugue (2003) delineates five generic types of manufacturing strategies that can be employed:

- Ship to Stock (STS): Products are standardized and pre-positioned in the market; consumers expect instant availability, which supports the maintenance of speculative safety stock at all stages of distribution.
- Make to Stock (MTS): Products are standardized but not necessarily assigned to specific locations; demand is expected to be stable or readily forecastable at an aggregate level.
- Assemble to Order (ATO): Products can be modified in a variety of ways, generally based upon a basic platform; final form of the products is postponed until demand is ascertained.
- Make to Order (MTO): Raw materials and components are ubiquitous, yet they may be combined to create a vast range of products.
- Buy to Order (BTO): Products desired by customers can be unique right down to the raw material level; product diversity is almost endless, albeit lead time is lengthy while resources are manufactured, processed into finished goods, and delivered.

One can see from these different strategies that the further down one goes through this list the more the manufacturing flexibility increases. Figure 2.4 illustrates this concept.

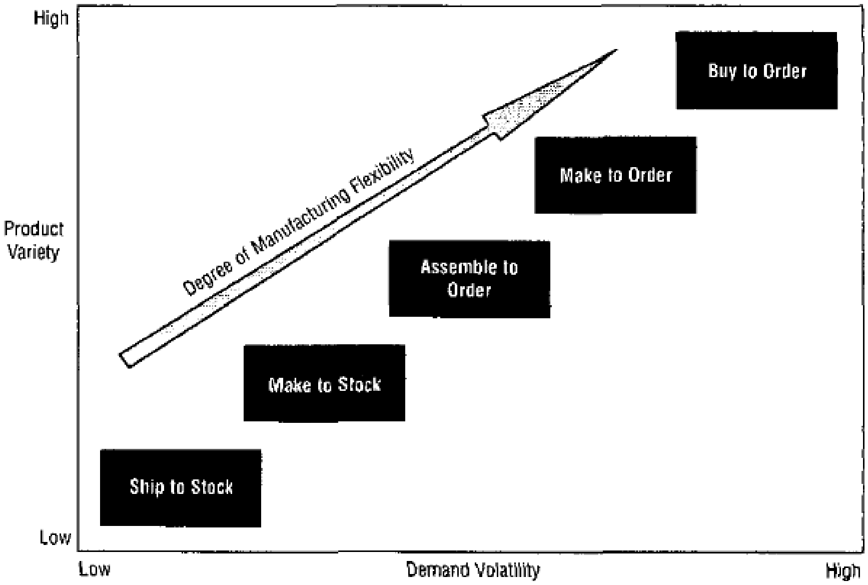


Figure 2.4: The five generic manufacturing strategies and their flexibility (Goldsby & García-Dastugue, 2003)

.

These manufacturing strategies also determine where the decoupling point is for a certain organization in their respective supply chain. The decoupling point is separates the part of the supply chain/organization oriented towards customer orders from the part of the supply chain/organisation based on planning (Hoekstra, Romme, & Argelo, 1992). The decoupling point is also where strategic stock is frequently kept as a buffer between variable client requests and/or product variety to smooth manufacturing output (Ben Naylor, Naim, & Berry, 1999). Figure 2.5 illustrates where the different decoupling points are often located for the generic types of manufacturing strategies.

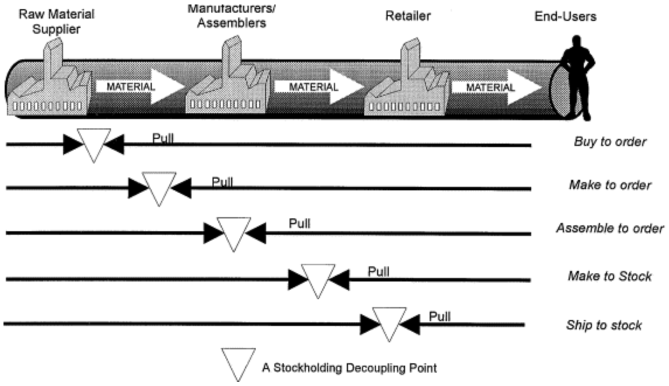


Figure 2.5: Illustration of the decoupling points for the five generic manufacturing strategies (Ben Naylor et al., 1999)

.

2.2.6 Supplier relationship management

Supplier relationship management, also often referred to as procurement, is similar to the customers relationship management. However, in this case it is mirrored in that the suppliers come before instead of after the company in the supply chain. These relationships ensure that the materials that are required for production are available (Croxton et al., 2001). When it comes to procurement different strategies can be employed dependent on the companies wishes and or possibilities.

One such strategy is multiple sourcing, which means that the materials come from multiple sources (suppliers) who are often pitted against each other to achieve the lowest prices (Zeng, 2000). A drawback of this strategy is that there is usually a large base of suppliers and the contracts are often short-term. Having new and/or many suppliers may cause for delays and result in disturbed production schedules. Besides multiple sourcing there is also single sourcing which uses a single supplier (for a specific good). An advantage of single sourcing is that there is improved communication and that prices can be suppressed from reduction in costs from the improved stability. One big disadvantage is that the company is very dependent on the supplier when choosing to go with a single one (Zeng, 2000). It is therefore critical that supplier partnerships be mutually beneficial especially in the case of single sourcing. If one of the parties does not benefit from the connection, the reason to stay in it diminishes and the partnership will likely end (Croxton et al., 2001).

2.2.7 Product development and commercialization

Product development is crucial to the companies long-term success. Rapidly developing new products bringing them to the market in an efficient manner is a critical component of company success Croxton et al. (2001). Because it is so vital for success the time to market has become a critical objective of this process. In turn, the development of innovative products must be accelerated and is already accelerating whilst on the other hand the product life cycle is decreasing Bullinger, Warschat, and Fischer (2000). In figure 2.6 the decrease in product life cycles and product development time is shown.

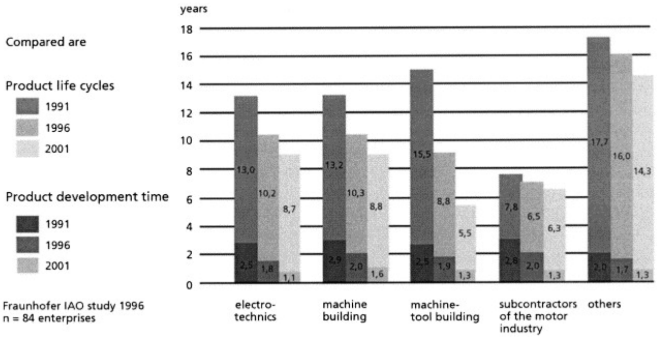


Figure 2.6: Product life cycle and development time shown for a few different branches. (Bullinger et al., 2000)

.

In order to minimize time to market, supply chain management should integrate customers and suppliers into the the product development process. To remain competitive as product life cycles shrink, the correct products must be designed and effectively launched in ever-shorter time constraints Croxton et al. (2001). Whereas before marketing and commercialization strategies had years to prepare and entice the customer before the release or during the life cycle of a product, the intervals for this also have become increasingly shorter.

2.2.8 Returns logistics

Returns logistics, also known as reverse logistics, is concerned with the management of the recovery of products once they are no longer desired (end-of-use) or can no longer be utilised (end-of-life) by the consumer (Rubio & Jiménez-Parra, 2014). This process as further described by (Rubio & Jiménez-Parra, 2014) consists of three main activities.

The first activity concerns the collection of products which can be viewed as the starting point of the system. There are different approaches to the collection of products as it can be done by the manufacturer itself, through retailers and distributors or through third-party logistics providers. The second activity concerns itself with the classification of products. This step is required due to the uncertainty associated with the state of the recovered products. Whilst some products could be reused with minor repairs others are completely unusable and would have to be broken down to raw materials in order to serve a purpose. This boils down to the third and final activity, which concerns itself with the recovery process of the products. The second and third activity are illustrated in figure 2.7 which delineates the different states of products and how they are handled.

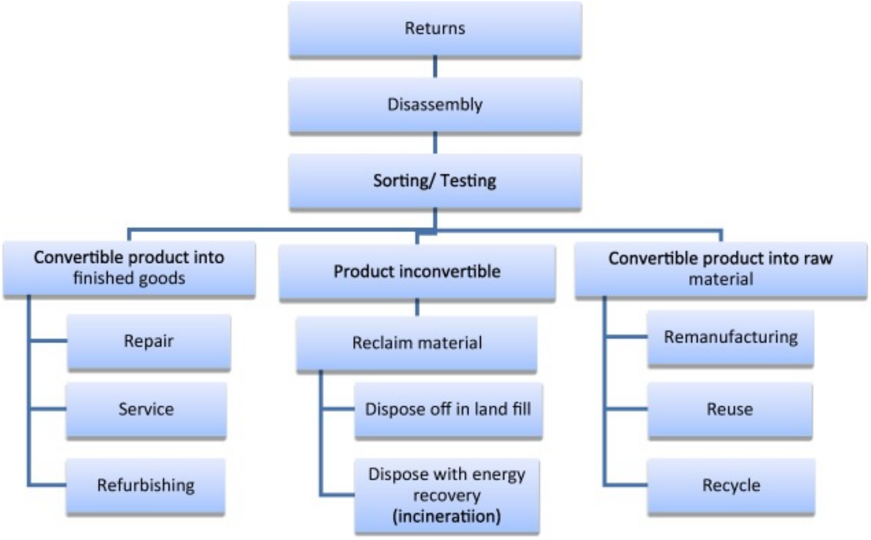


Figure 2.7: Generalized reverse logistics structure. (Prajapati et al., 2019)

There are also clear advantages associated with implementing reverse logistics systems. The economic advantage is that the returned product can still have some value that can be reused (de Brito & Dekker, 2003). There is also an environmental advantage in that it is good for the environment. The added benefit is that showing environmentally responsible behaviour could lead to improved relations with customers, resulting in more economic advantages (Thierry, Salomon, Van Nunen, & Van Wassenhove, 1995).

Besides the advantages there are also some difficulties associated with establishing a reverse logistics system. The infrastructure required to perform the reverse logistics can be costly in order to function correctly, these costs might not outweigh the monetary gains. Moreover, some companies will have to redesign their product in order to make more (or any) use out of recovered products (Thierry et al., 1995).

2.3 Interactions of the key processes

In figure 2.8 and **??** examples are given of interactions between the key processes. These examples are chosen since they are tangible and easy to grasp. There are definitely far more interactions. It would, however, go too deeply into the fundamentals of supply chain management for this study. For a more detailed overview one can look at the paper by Croxton et al. (2001). In this paper every key process is split up in sub-processes which connect to the other key processes.

Figure 2.8 illustrates the connections that are made when a firm wants to make a forecast. Because demand management must balance customers requirements with the firms supply capabilities. Order fulfillment and customer service management supply the information needed to construct the forecast since they are closest to the client and hence have the critical information. These forecasts are then conveyed to the various process teams affected by them, such as customer service management, order fulfillment, manufacturing flow, and product development and commercialization. Usually, supplier relationship management is also brought into the fold since fluctuations in demand is also important knowledge for this process team Croxton et al. (2001).

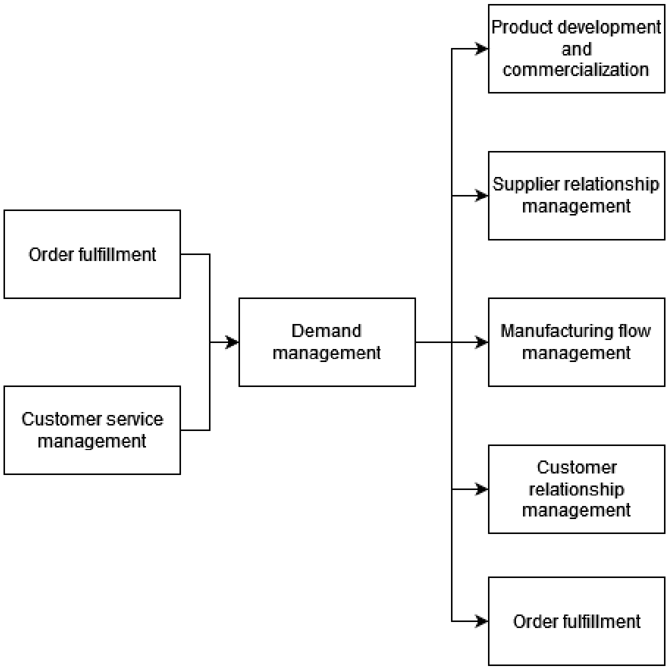

Product development and commercialization
Supplier relationship management
Order fulfillment
Demand management
Manufacturing flow management
Customer service management
Customer relationship management
Order fulfillment

Figure 2.8: Connections between key processes when it comes to a demand forecast. Inspired by Croxton et al. (2001)

.

Figure 2.9 illustrates the connections that are made when a firm wants to evaluate their logistics network. When evaluating a logistics network it is necessary to look into: which plants generate which goods, where warehouses, plants and suppliers are situated, and which means of transportation should be employed. Important information in this case comes from the demand management and returns key processes. The resultant network is supplied to the manufacturing flow process Croxton et al. (2001).

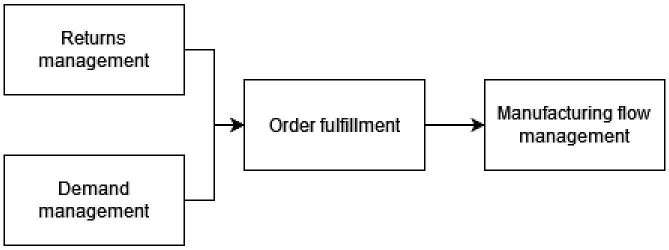


Figure 2.9: Connections between key processes when it comes to evaluating the logistics network. Inspired by Croxton et al. (2001)

.

2.4 Conclusion

This chapter provides a brief overview of supply chain management. The answer to sub-question one is the definition provided by (Croxton et al., 2001) as: "The integration of key business processes across the supply chain". These key processes of supply chain management were all discussed. Furthermore, a few examples were discussed of how they are connected. In the next chapter a look will be taken at machine learning and how this goes together with supply chain management.

Chapter 3

Literature study: Machine learning

This chapter will begin with an introduction to machine learning in section 3.1. For more in-depth information regarding the algorithms used in this research refer to chapter 5. After the introduction to machine learning the chapter continues with section 3.2, where the combination of the fields ML and SCM is looked at. The chapter concludes with section 3.3 where the research question: "What is machine learning?" is answered.

3.1 Machine Learning

A machine learning algorithm is a computing process that utilizes input data to accomplish a goal without being explicitly written(i.e. "hard coded") to do so. These algorithms are "soft coded" in the sense that they automatically adjust or adapt their architecture as a result of repetition to get better and better at executing the specific task. The process of adaptation is known as training, and it involves providing samples of input data along with desired outputs. The algorithm then optimizes its configuration such that it can not only provide the intended result when given the training inputs, but also generalize to achieve the desired result when given new, previously unseen data. The "learning" aspect of machine learning is this training. (El Naqa & Murphy, 2015)

Figure 3.1 shows four different classifications of machine learning. Supervised, unsupervised, semi-supervised and reinforcement learning. These different classifications depend on the data, as can be seen in figure 3.2. The more human interaction there is with the data, in the form of labels, the more supervised the learning becomes.

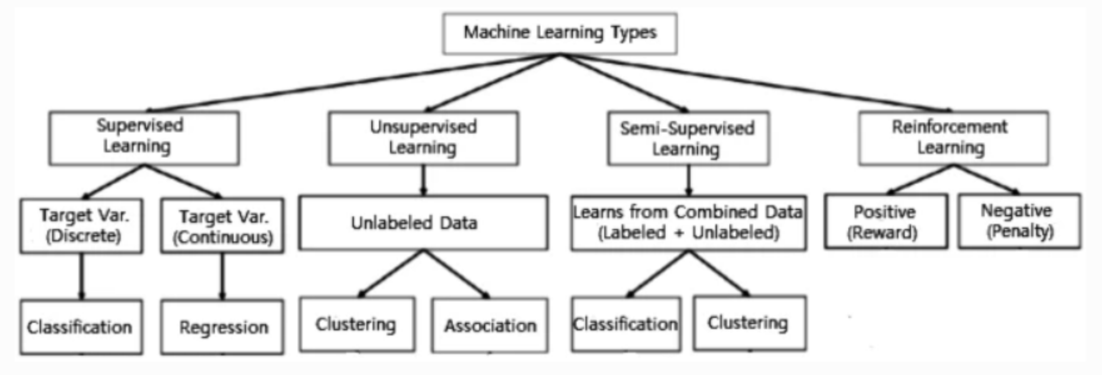


Figure 3.1: Different machine learning methods and the learning categories they belong to by I. Sarker (2021).

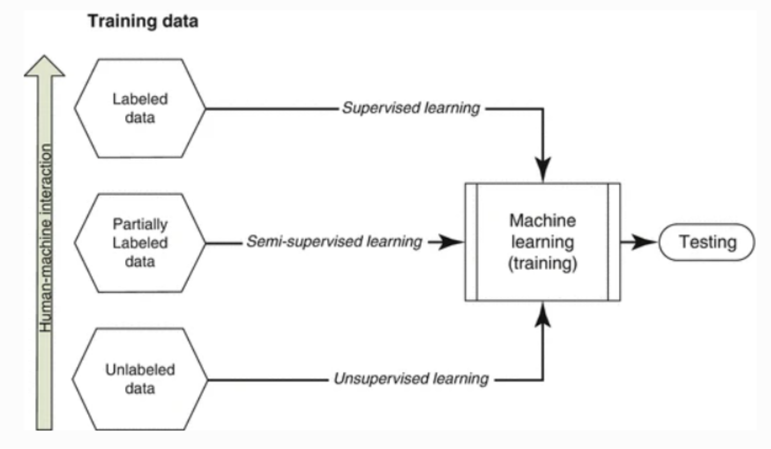
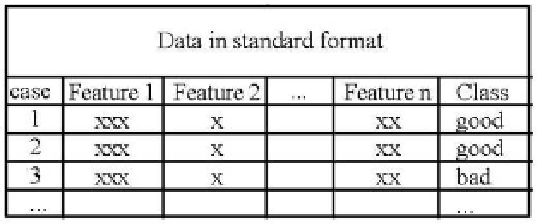


Figure 3.2: Classification of learning techniques El Naqa and Murphy (2015).

Supervised learning is a machine learning task that generally involves learning a function that translates an input to an output based on sample input-output pairs (Han, Pei, & Kamber, 2011). To infer a function, it employs labeled training data and a set of training examples. Supervised learning is carried out when certain goals are identified to be achieved from a specific set of inputs (I. H. Sarker et al., 2020). More simply put, the goal of the machine is to produce the desired output based on a given new input. The most typical supervised tasks are classification, which divides the data, and regression, which fits the data. For example, predicting the class label or sentiment of a piece of text, such as an movie review, is an example of this. Figure 3.3 illustrates what the data would look like in such a case, whilst figure 3.4 shows how the process of learning works in that case.



Data in standard format

case	Feature 1	Feature 2	...	Feature n	Class
1	xxx	x		xx	good
2	xxx	x		xx	good
3	xxx	x		xx	bad
...					...

Figure 3.3: Illustration of the data form in supervised learning (Kotsiantis et al., 2007).

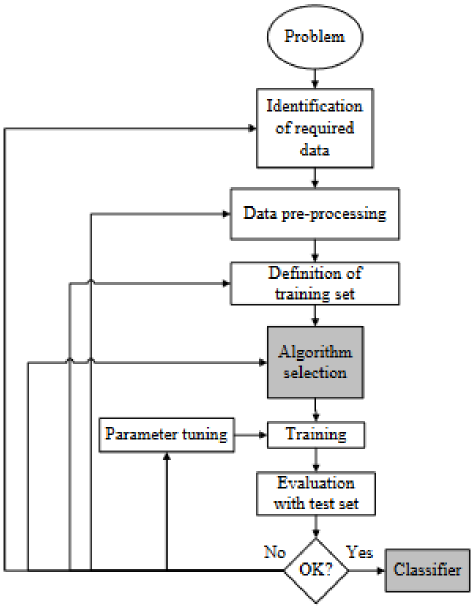


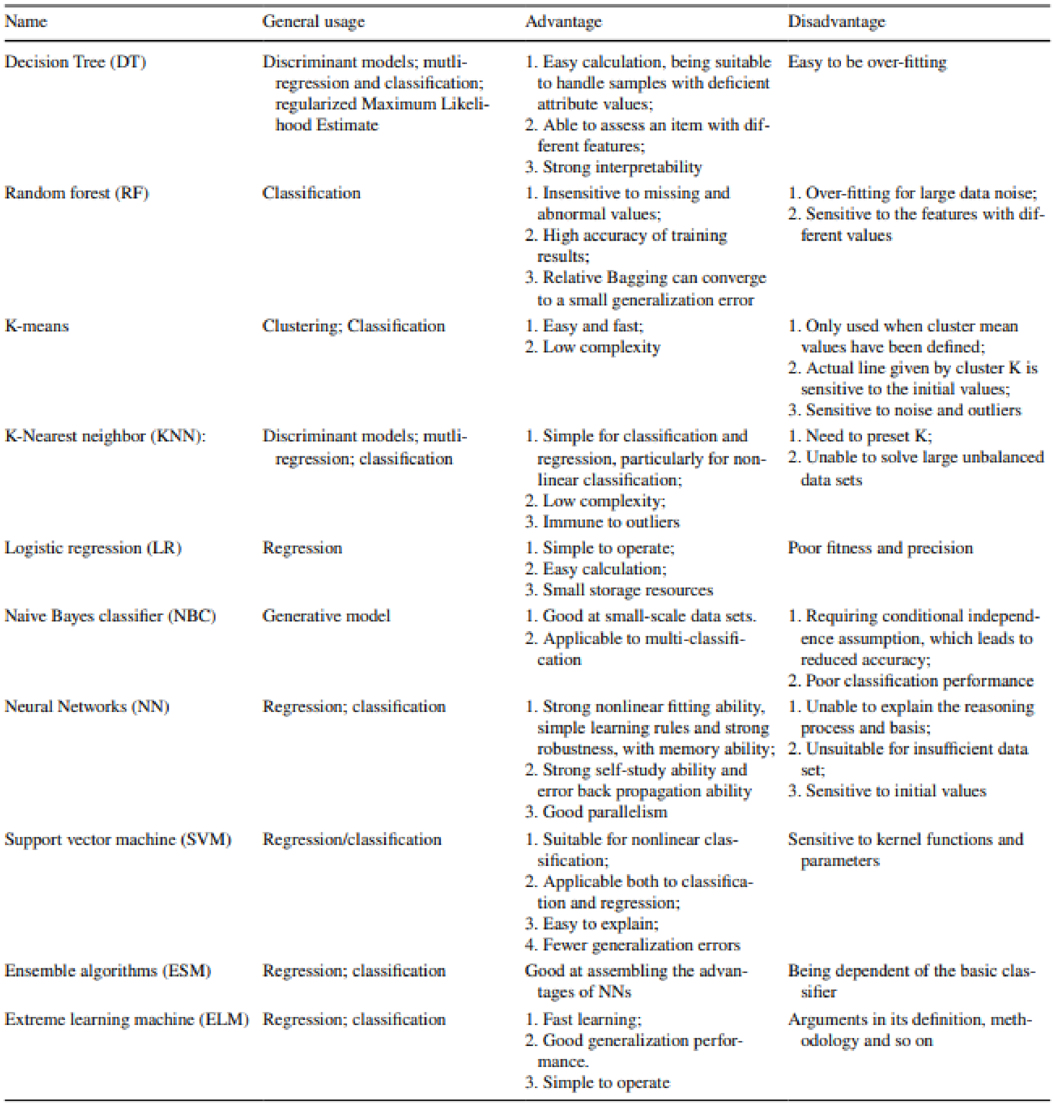
Figure 3.4: The process of supervised machine learning (Kotsiantis et al., 2007).

Unsupervised learning is a data-driven technique that examines unlabeled datasets without the requirement for human intervention (Han et al., 2011). This is commonly used for extracting generative features, discovering relevant patterns and structures, groups in results, and exploratory purposes. The most typical unsupervised learning tasks include clustering, feature learning, dimensionality reduction, density estimation and anomaly detection I. Sarker (2021).

Semi-supervised learning is a hybridization of the supervised and unsupervised approaches discussed above, as it acts on both labeled and unlabeled data (Han et al., 2011) (I. H. Sarker et al., 2020). As a result, it lies between learning without supervision and learning with supervision. In the real world, labeled data may be scarce in some circumstances, but unlabeled data is abundant, making semi-supervised learning helpful (Mohammed, Khan, & Bashier, 2016). The ultimate aim of a semi-supervised learning model is to offer a better prediction output than that obtained by the model utilizing only labeled input. Semi-supervised learning is utilized in a variety of applications, including machine translation, text classification, data labelling and fraud detection.

Reinforcement learning is a form of machine learning algorithm that allows software agents and machines to automatically evaluate the optimal behaviour in a certain context or environment in order to enhance efficiency (I. Sarker, 2021). This sort of learning is focused on reward or punishment, and its ultimate purpose is to use insights from an interaction with the environment in order to take action to maximize the reward or minimize the risk Mohammed et al. (2016).

In figure 3.5 a table can be found that gives a brief overview of ten frequently used machine learning algorithms in supply chain management (Ni et al., 2020). This table describes the advantages and disadvantages of every algorithm along with how it is generally used. The terms in the general use column mostly correspond to figure 3.2 that was presented earlier in this section. From there the type of learning can be linked to the corresponding algorithm.



Name	General usage	Advantage	Disadvantage
Decision Tree (DT)	Discriminant models; mutli-regression and classification; regularized Maximum Likelihood Estimate	1. Easy calculation, being suitable to handle samples with deficient attribute values; 2. Able to assess an item with different features; 3. Strong interpretability	Easy to be over-fitting
Random forest (RF)	Classification	1. Insensitive to missing and abnormal values; 2. High accuracy of training results; 3. Relative Bagging can converge to a small generalization error	1. Over-fitting for large data noise; 2. Sensitive to the features with different values
K-means	Clustering; Classification	1. Easy and fast; 2. Low complexity	1. Only used when cluster mean values have been defined; 2. Actual line given by cluster K is sensitive to the initial values; 3. Sensitive to noise and outliers
K-Nearest neighbor (KNN):	Discriminant models; mutli-regression; classification	1. Simple for classification and regression, particularly for nonlinear classification; 2. Low complexity; 3. Immune to outliers	1. Need to preset K; 2. Unable to solve large unbalanced data sets
Logistic regression (LR)	Regression	1. Simple to operate; 2. Easy calculation; 3. Small storage resources	Poor fitness and precision
Naive Bayes classifier (NBC)	Generative model	1. Good at small-scale data sets. 2. Applicable to multi-classification	1. Requiring conditional independence assumption, which leads to reduced accuracy; 2. Poor classification performance
Neural Networks (NN)	Regression; classification	1. Strong nonlinear fitting ability, simple learning rules and strong robustness, with memory ability; 2. Strong self-study ability and error back propagation ability 3. Good parallelism	1. Unable to explain the reasoning process and basis; 2. Unsuitable for insufficient data set; 3. Sensitive to initial values
Support vector machine (SVM)	Regression/classification	1. Suitable for nonlinear classification; 2. Applicable both to classification and regression; 3. Easy to explain; 4. Fewer generalization errors	Sensitive to kernel functions and parameters
Ensemble algorithms (ESM)	Regression; classification	Good at assembling the advantages of NNs	Being dependent of the basic classifier
Extreme learning machine (ELM)	Regression; classification	1. Fast learning; 2. Good generalization performance. 3. Simple to operate	Arguments in its definition, methodology and so on

Figure 3.5: Frequently used machine learning algorithms in supply chain management (Ni et al., 2020).

3.2 Machine learning in the field of supply chain management

There are many possible applications of machine learning in supply chain management. These applications range from supplier selection to inventory management and from risk management to circular economy and production (Babaee Tirkolaee, Sadeghi Darvazeh, Mooseloo, Rezaei Vandchali, & Aeini, 2021). The same example as in 2 will be utilized to demonstrate the usage of machine learning in supply chain management. This was demand forecasting, and this example is used since the required knowledge for explaining the problem was already established. Aamer, Yani, and Priyatna (2021) provides an overview of machine learning methods for demand forecasting. In this paper it is discussed that machine learning techniques can provide better accuracy and less computational cost compared to traditional models. The most prevalent ML technique to perform demand forecast was neural networks. Figure 3.6 shows that the number of papers published on demand forecasting by machine learning has steadily increased.

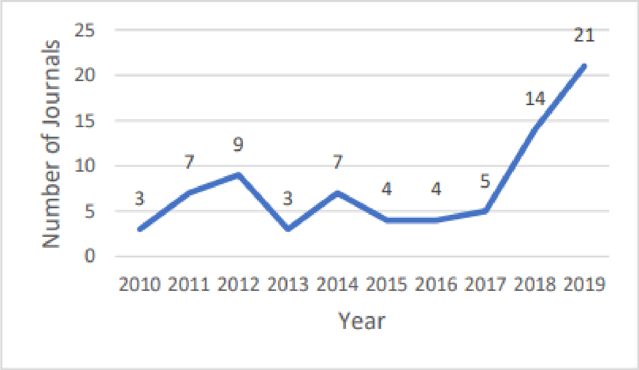


Figure 3.6: The increase in papers published on demand forecasting by machine learning Aamer et al. (2020).

Since 2015, the number of papers on supply chain management and machine learning has steadily increased (with minor exceptions). This is illustrated in figure 3.7. According to (Schroeder & Lodemann, 2021) the reason there appears to be a lower value for 2020 is because the literature search by this paper was conducted on the first of January 2021. As a result, the paper argues that not all papers published and submitted in 2020 had been entered into databases. (Kersten, See, Lodemann, & Grotemeier, 2020) argues that the combination of SCM and ML has grown in importance, which can be attributed to the increased importance of data in research and practise. (Mahraz, Benabbou, & Berrado, 2022) discusses that a study estimated that by 2023 half of global supply chain activities will be using AI and machine learning technology. Companies such as MPO must partake in order to stay relevant in this evolving field.

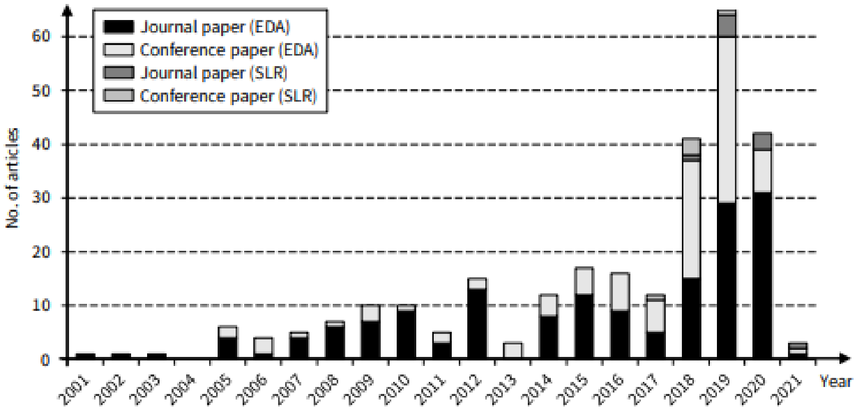


Figure 3.7: Papers published by type of publication in the field of supply chain management that involve machine learning according to Schroeder and Lodemann (2021).

3.3 Conclusion

This chapter provides a brief overview of machine learning, which answered sub-question two: "What is machine learning?". Furthermore, a look was taken into how machine learning is applied within supply chain management through the use of an example. Moreover, the general interest in terms of research of the combined fields of machine learning and supply chain management was described. In the next chapter the studied system of this research will be examined.

Chapter 4

Case study: MPO

Now that it is clear what supply chain management entails, it is possible to examine the studied system. In this chapter sub-question three will be addressed, which is: "What are the characteristics of the studied system and what are elements of the system where improvements could be made?". To answer this question, the begins starts by outlining what the MPO platform accomplishes and how it works in a general in section 4.1. This is done to provide a context for the elements of the system that will be explored in the following section, which is section 4.2. In this section one of these elements will be selected for further study in this research. The information described in these sections was obtained via internal documentation, the company's learning environment and interviews with experts. A list of interviewed experts can be found in appendix B. In section 4.4 the previous work with regards to the selected element is delved into. Finally, the chapter ends with a conclusion in section 4.5.

4.1 The MPO platform

Supply chain management is a very broad field that interacts with every aspect of the supply chain, as was described in chapter 2. When trying to fit MPO in one of the key processes as discussed in chapter 2 it becomes difficult, because the platform performs a multitude of different tasks. The process that MPO concerns itself most with is, however, order fulfillment and especially the distribution part of order fulfillment. In the platform many different management systems are combined in order to perform this distribution for their clients. How MPO operates is not generic for supply chain management companies. SCM ompanies operate with varying intelligence and scope levels dependent on the tasks they aim to accomplish. Therefore, MPO is only comparable to companies that operate with similar levels of intelligence and scope. Figure D.1 in appendix D depicts the space that MPO operates in.

Order fulfillment, as was stated in chapter 2, is defined as a process that begins with receiving customer orders and finished with the final items being delivered. Figure 4.1 depicts this process with a black box between the start and end of order fulfillment. MPO fits within order fulfillment in that sense that it receives an order from one of their clients and then arranges that said order is delivered. This is all done through the use of the MPO platform.

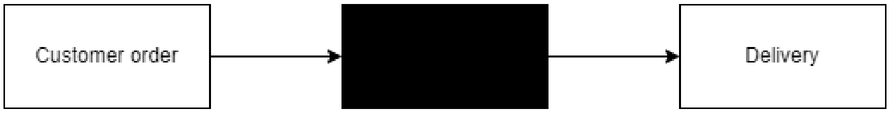


Figure 4.1: Illustration of the order fulfillment process.

Clients engage with the platform by placing a customer order. These customer orders might be in the form of a sale a buy or a return. It generally involves information on the origin and destination of the commodity, as well as any preferences, such as a deadline for the product's delivery or a preferred mode of shipping. The customer order is subsequently divided into one or more shipment orders based on different factors such as the availability of inventory or the required intermediate transportation steps. For example when goods from mainland China have to travel to for example Eindhoven there are multiple routes that can be taken. It can be flown in to Schiphol airport and then brought to Eindhoven by truck. It might alternatively be brought to one of the Chinese ports and then shipped to the port of Rotterdam and then taken by rail to Eindhoven. These are only a few examples of combinations; many more can be made. These shipment orders are divided into one or more service orders. These service orders are one of three things, a transportation, a booking or documentation. The first is transportation, which refers to the transport of an item from its origin to its destination. Second there is booking, which is a request for a carrier to perform a certain transportation. The third and last category is documentation, which refers to paperwork necessary, for example, to clear through customs or enter a warehouse. Figure 4.2 shows the different order tiers and how this process works in practice. These steps are logged on the platform and may be accessed by the client via each of the order levels their respective track and trace interface.

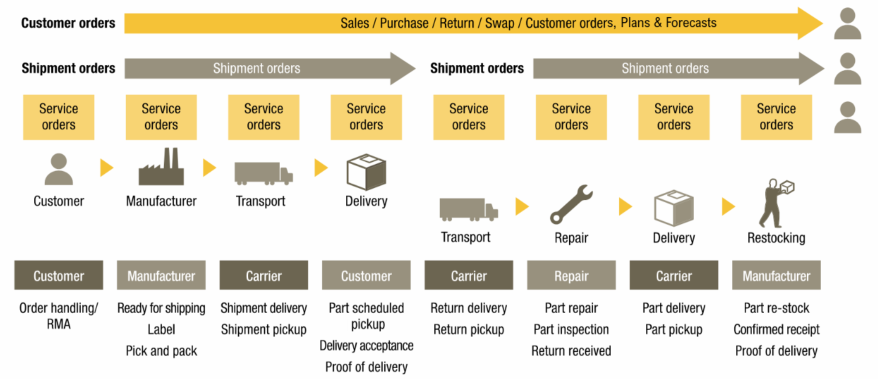


Figure 4.2: Example of a multi-leg transportation.

The path with the lowest 'cost' is chosen, where 'cost' is a heuristic that is based on a variety of different costs. These include monetary costs, environmental costs and time costs, among others. A path for a product from origin to a destination can also consist of more than one trip with either different or the same modes. The costs of these trips or of a whole path is also determined by the carrier that is utilized. There might be carriers that conduct the same route in less time for a higher cost and these are trade-offs that are considered according on the customers preferences. MPO does not negotiate service time and cost agreements with carriers. These are managed by the clients themselves, and the platform is aware of the prices that are utilized. As a result, the least expensive method is chosen for a client depending on the service time and cost agreed upon with their carriers.

Another important functionality that the platform offers is the inventory management. The platform displays which products are stored where and how many there are. Furthermore, the platform informs the client on how much more room there is in the warehouse for a certain product based on short, medium and long term usage. Furthermore, the platform notifies the client how much additional warehouse space is avaialable for a specific product depending on short, medium and long term consumption. Forecasts on stock levels contained within a given warehouse are also supplied. This is also accompanied with a map displaying the facilities of that client. When a client observes that stock levels should be replenished for a certain facility a customer order may be quickly produced, and the order management process is initiated.

Along with the aforementioned features, the platform provides an analytics suite. Because of the data provided by customers through usage of the platform, the suite is equipped with up-to-date information and may thus deliver intriguing insights. For example, the shipment order view in the analytics suite provides information on overall revenues and costs for a certain consumer, the number of orders of orders per country or client, as well as transportation performance as a percentage of on-time in full deliveries (OTIF). OTIF indicates that the delivery occurred before the specified deadline and that all items were present and undamaged (Waters, 2021). This might be useful information for a client if, for example, one or more of their present carrier(s) are not performing adequately. Overall, the analytics suite may provide a wealth of current information about trends as well as potential risks to customers, making it play a significant role in the MPO platform.

The unique value of the MPO platform lies that it combines a control tower system (CTS), supply chain visibility (SCV), order management system (OMS), return management system (RMS) and a transport management system (TMS) all in one place. It does so by providing the previously described multi-level order management, multi-tier inventory management and multi-leg transport management. As a result MPO incorporates variety of expertises as seen in appendix D, which includes figure D.1. This figure depicts the position of firms within supply chain management and the tasks they execute based on the amount of intelligence and scope with which they work.

4.2 Possible applications

In this section five elements of the platform are discussed where machine learning could make improvements. Each of these elements was found through analysis of the system and discussed with one or more of the experts during the interviews. A list of the interviewees can be found in appendix B. These elements will be placed under on of the pillars of supply chain management, which was discussed in chapter 2. The first element is on-time estimation, which is discussed in detail in subsection 4.2.1. The second element is cost estimation, which is covered subsection 4.2.2. In the subsection 4.2.3, the third element carrier preference is discussed. The fourth element is returns management, which is covered in subsection 4.2.4. The fifth and final element is demand forecasting, which can be found in subsection 4.2.5.

4.2.1 On-time estimation

On-time estimation can be placed under the pillar of order fulfillment, which was discussed in subsection 2.2.4. Knowledge of whether an order will be on time or not is of the utmost importance in supply chains. Since a delay may not only result in delay for the receiving party but may result in delays for all downstream parties in the supply chain. The most important data required for estimating whether a delivery is on-time are: the promised time that the delivery was supposed to happen and the actual delivery time. Other indicators that are useful are for example the path taken, the carrier and/or mode and more (Mahajan, Kiwelekar, Netak, & Ghodake, 2021) (Hildebrandt & Ulmer, 2022). As previously stated, MPO determines the order's path and informs the client of the expected delivery date. It is also known who will carry out the delivery. The carrier provides a proof of delivery that includes the date of the actual delivery. The most important data to make on-time estimations is therefore available, making this element an excellent candidate for further research.

4.2.2 Order cost estimation

Order cost estimation also falls under the pillar of order fulfillment, which was discussed in subsection 2.2.4. An order has a base cost that is agreed upon between the client and a carrier. This is called a rate service agreement. This is usually between zones or within a specific zone for different size or weight classes for a delivery. Alongside this base cost there are surcharges which are in most cases unknown to the client until the delivery is made. Examples of such surcharges are risk, hazard, remote area, congestion, waiting (during loading/unloading) and more. MPO already has a machine learning prediction algorithm for the remote area surcharge. This surcharge deals with deliveries to locations that fall within a certain zone but are difficult to reach or far from normal routes. Making predictions about other individual surcharges or about the entire cost of an order can be very useful. However, data with regards to these individual surcharges can sometimes be a bit incomplete, which would make predictions difficult to make.

4.2.3 Carrier performance

Carrier performance would fall under the pillar of order fulfillment and especially the distribution aspect of it, which was described in subsection 2.2.4. When a customer order is created the customer provides an origin/destination (also known as lane), arrival time and in some cases also preferred mode. The MPO platform then estimates which carrier can best fulfill this service order, using dynamic carrier selection, according to a set of soft and hard constraints. The hard constraint is that the carrier must comply with estimated delivery time (ETD). The soft constraints are monetary cost, environmental cost and service time. It does not, however, take the performance of a carrier into account.

A bad experience from a customer with a carrier can be characterised in many different ways, six of these ’negative’ performance parameters can be seen in figure 4.3 in yellow along with a proposed scoring system in the middle. There are, however, other algorithms than machine learning that can attach some form of ’risk’ value to using a carrier based on these proposed points. For example the paper by Y.-K. Lin and Yeh (2010) proposes a stochastic logistics network that defines a probability that freight is successfully transmitted to a customer for a certain carrier.

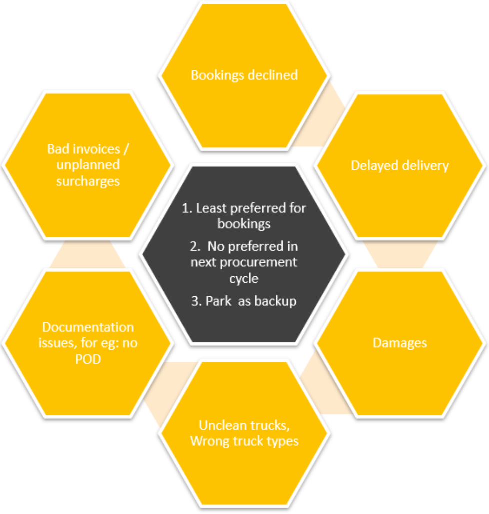

Bookings declined
Bad invoices / unplanned surcharges
Delayed delivery
1. Least preferred for bookings
2. No preferred in next procurement cycle
3. Park as backup
Documentation issues, for eg: no POD
Damages
Unclean trucks, Wrong truck types

Figure 4.3: Performance parameters of carriers and the (negative) ratings that can be given to a carrier for a specific lane and mode.

4.2.4 Return of product

Product return can easily be placed under the pillar of return logistics, which can be found in subsection 2.2.8. On a customer order, a client can specify whether it is a purchase, sale, or return. As a result, returns in the system are easily distinguished. However, (Lickert, Wewer, Dittmann, Bilge, & Dietrich, 2021) identifies critical data that is required in order to make accurate predictions, and some of this data is not available for MPO. These are: The product and/or batch number, condition of the product, delivery number, delivery date, customer information and supplier information. Because some of this data is unavailable, developing a well-performing prediction algorithm would be difficult.

4.2.5 Demand forecast

Within SCM, demand forecasting has its own pillar, which was described in subsection 2.2.3. The rudimentary data required to make a prediction about the estimated demand is available, since orders go through MPO. The necessary data, namely, is historic demand. Other features are highly dependent on the operating market. For example for a restaurant the weather, events (like holidays) and the location are important variables Tanizaki, Hoshino, Shimmura, and Takenaka (2019). Whereas for a retail company product categories, promotional information and special features of special days are important variables (Huber & Stuckenschmidt, 2020). Whilst in the case of water demand for a farm important variables are daily rainfall, average temperature, crop type, area size and many more. This phenomenon raises the question of the wider applicability of a demand forecasting algorithm for a company such as MPO. Because the demand forecast is highly dependent on the market in which a company operates. Since the clients of MPO that use the platform operate in very different markets. It might be the case that for each client a different demand forecast would have to be made, and in that case each demand forecast would have slightly different variables that are important. Still, the rudimentary data is available and therefore predictions could be made.

4.3 Element selection

In the previous section five elements were discussed that through the use of machine learning could improve the performance of an area of the MPO platform. The data for order cost estimation and product returns is unsatisfactory among these elements. For carrier performance it remains unclear why machine learning should be used and is therefore not a suitable candidate to be considered. Demand forecast and on-time estimation are both strong candidates, however, as was previously discussed the wider applicability of demand forecast is worrisome. Therefore, on-time estimation will be the topic for further research in this study.

4.4 Previous work

Only on-time/arrival time predictions will be considered for previous work. This is because, in the previous chapter, which is chapter 4, this element was selected as the focus of this research. Estimating arrival time or whether a transportation will be on time is a research area that has existed for some time. The literature will be reviewed in a chronological order with an emphasis on more recent literature. At the end of the section an overview, in table 4.1, an overview of the research area with respect machine learning applications can be found.

The reviewed literature is presented in chronological order, beginning in the relatively distant past. Reinhoudt and Velastin (1997) offers a method for estimating bus arrival time. The data used was given by countdown, a new technology that was being test in North London at the time. The arrival times of the preceding three buses on that route were utilized as data from this system. The error distribution of the static averaging approach was improved by up to 7% by using a Kalman filter.

Moving ahead in time, Y. Lin, Yang, Zou, and Jia (2013) conducts a case study on real-time bus arrival times in Jinan, China. To make these predictions, an artificial neural network (ANN) is trained using GPS and automated fare collecting data. This data provides variables such as the arrival and depart time at a certain stop as well as historical data of busses on the same stop. In order to balance prediction acuracy with computation efficiency only the data of the previous three busses was looked at. The results reveal that the ANN model outperforms the already existing Kalman filter. Under recurrent traffic situations, the relative prediction error within a 10 minute prediction time frame is less than 20% with a reliability probability of more than 85%, and the likelihood of having more than 40% relative prediction errors is no more than 7%. Furthermore, because of the low computation times the application may be used in real-time.

In 2016 this research area started getting more traction. In this year Yang, Chen, Wang, Yan, and Zhou (2016) presents a support vector machine with genetic algorithm (GA-SVM) to predict bus arrival times. The genetic algorithm is used to find the best parameters for the input vectors that were used in this study. These input vectors were: The length of the road, the bus speed, the rate of road usage, the weather and finally the character of the time period. The data from a single bus in the Shenyang region was used in order to validate the algorithm. It was then compared to a traditional support vector machine and an artificial neural network. It outperformed both of these aforementioned algorithms in accuracy.

In the same year Lee, Malik, and Jung (2016) compared a multitude of different machine learning algorithms on the taxi time of aircraft. The use case was Charlotte airport in North Carolina. The variables used for the predictions were: Terminal concourse, runway, spot, departure fix and weight class as well as weather conditions and traffic flow. The algorithms in this study were linear regression (LR), support vector machines (SVM), k-nearest neighbours (kNN), random forest (RF) and neural networks (NN). Of these linear regression and random forest had the lowest root mean square error. Furthermore, the paper discusses that uncertainty and operational complexity complicated the accuracy of predictions.

In the following year van der Spoel, Amrit, and van Hillegersberg (2017) presented a model to predict the arrival time of trucks originating from a single warehouse. The predictions were done by doing a classification of the expected arrival time in hours. Variables that were used for predictions were weather, accidents, congestion and time of day. For these predictions random forest, RPart, SVM, ADaboost and kNN were used of which ADaboost performed the best in terms of accuracy. However the authors argue that the predictive power of the model is limited. Furthermore, they argue that it is due to a gap in the literature where arrival time has barely been researched, whilst travel time has. With arrival time other factors, besides those important in predicting travel time, are important such as human factors like the intended arrival and departure time.

Pang et al. (2018) like Lee et al. (2016) presented an algorithm to predict the arrival times of busses. Historical trajectory data as well as statistics with regards to the infrastructure is used to train a recurrent neural network (RNN). On top of that, the arrival time of the bus at a stop is adjusted based on the arrival times of the bus at previous stops. The RNN was compared to state-of-the-arts methods, which were: kNN, SVM, LR, Additive mixed model (AMM), MLP and kalman filter. The recurrent neural network performs at least 10% better in terms of RMSE compared to the state-of-the-art techniques previously mentioned.

In the same year Barbour, Samal, Kuppa, Dubey, and Work (2018) compares different machine learning algorithms to estimate arrival times for freight trains in Tennessee in the US. In this paper support vector regression (SVR), random forest and deep neural nets are compared along with a statistical model on historical data. Variables used are train length, priority, weight and horsepower, remaining crew time, location, traffic and many others. The random forest managed to outperform all the other algorithms and models.

Moving forward in time, Wesely, Churchill, Slough, and Coupe (2021) developed a novel approach to predict aircraft arrival times. The first step predicts a landing time by selecting from among physics-based predictions using mediation rules. This data is available in the Federal Aviation Administration System Wide Information Management system. The second step employs a machine learning model, called XGBoost, that is based on the mediated predictions. Using features describing the current state of an airborne flight, the model is trained to predict the error in the mediated prediction. The XGBoost leads to substantial improvements in prediction quality compared to the mediated physics-based predictions (MPP) used in the first step.

Another paper in the same year by Basturk and Cetek (2021) proposes to predict estimated arrival times of aircraft. Random forest and deep neural networks (DNN) were trained in order to perform this task. Many different variables were used, for example: Scheduled time for departure and arrival, distance, aircraft type, airline, wind speed and visibility around airport and many more. Immediately after departure, both the DNN and RF algorithms predicted a delay with a mean absolute error (MAE) of less than 6 minutes.

The work by Park, Sim, and Bae (2021) provides a data driven methodology for estimating seagoing vessel arrival times for ports. In the first stage, an reinforcement learning (RL)L framework is proposed for path-finding of each vessel based on AIS data. This AIS data contains ship information such as the ships position with associated timestamp, the type of ship and the rate of turn, amongst other information. This data is employed in a reinforcement learning framework to find a Q-learning based path. This is paired with the ship's speed over ground (SOG) to calculate a estimated arrival time.

Finally, the study by Hildebrandt and Ulmer (2022) elaborates on a model for doing arrival time predictions for restaurant meal delivery. The study discusses two methods, both of which are accomplished through supervised learning. To produce predictions the offline technique employs gradient boosted decision trees (GBDT) on temporal, spatial, route and process data. Whilst the online-offline method uses an offline approximation of the runtime-expensive routing policy that is learned and supervised by a deep neural network to conduct simulations of all processes, including future demand and routing decisions, in real time.

Reference	Method	ML algorithms	Topic	Single mode	Single route/origin	Weather/traffic
\cite{reinhoudt1997dynamic}	Regression	Kalman filter	Bus arrival time	✓	✓	
\cite{lin2013real}	Regression	ANN	Bus arrival time	✓	✓	
\cite{yang2016bus}	Regression	GA-SVM	Bus arrival time	✓	✓	✓
\cite{lee2016taxi}	Regression	LR & RF	Taxi time of aircraft	✓	✓	✓
\cite{van2017predictive}	Classification	DT, KNN, SVM, ensemble classifiers & RF	Truck arrival time	✓	✓	✓
\cite{pang2018learning}	Regression	RNN	Bus arrival time	✓	✓	✓
\cite{barbour2018data}	Regression	SVR, RFR & DNN	Freight train arrival time	✓		✓
\cite{wesely2021machine}	Regression	XGBoost & MPP	Aircraft landing time	✓		✓
\cite{basturk2021prediction}	Regression	RF & DNN	Aircraft ETA	✓		✓
\cite{park2021vessel}	Reinforcement learning	Q-learning	Vessel ETA	✓		✓
\cite{hildebrandt2022supervised}	Regression/classification	GBDT & DNN	Arrival time of restaurant meal delivery	✓	✓	✓

Table 4.1: Previous work with regards to estimating time of arrival

Table 4.1 provides an overview of all the discussed papers with respect to on-time estimation. As can be seen in the table, there are numerous machine learning algorithms in use, but most of them have one thing in common: they perform classification, regression, or both and are therefore supervised learning methods. Furthermore, these classification and regression therefore either give an indication of whether a transportation is on-time or not in the case of classification whereas with regression an arrival time is estimated. Besides that, a large part of the literature with regards to on-time or arrival time estimation focuses on aircraft and busses. Another thing these papers have in common is that they all focus on a single mode. This study adds value because it combines several things that none of the previous studies have done.This relates to a difference in scope, when elements like mode and the precise route are ambiguous. In contrast, all of these publications are written from the perspective of a single mode, as can be seen in table 4.1. Additionally, practically all of them know the exact route in order use either weather or traffic data. Furthermore, a single route or origin is frequently used as the viewpoint in these articles. Finally, in none of the papers are the variables actually tested for relevance. It might be possible to achieve similar results with less variables, which is very relevant for this research since the model should be applicable to multiple sectors of industry. Namely, not every company has the same data available, therefore it is important to find out which variables are important and which are not important to establish when on-time estimation can actually be used.

4.5 Conclusion

In this chapter the studied system was discussed as well as the company that provides the data that will be used. Furthermore, five elements were discussed that were raised by experts where possible improvements could be made. These elements were discussed as well as limitations that might hamper the success. On-time estimation was deemed the most promising of the elements and will therefore be the focus of this research. In the following section a review of earlier work was performed and the added value of this research was discussed. The earlier work mainly used regression and classification as methods to solve the problem of on-time estimation. These methods both fall under the machine learning category of supervised learning, as discussed in chapter 3. Therefore, in the next chapter two models will be discussed that use supervised learning to solve the problem of on-time estimation. Moreover, this model that is inspired by the previous work will also add a way in which the relevance of features will be determined.

Chapter 5

The model

In the previous chapter the element of on-time estimation was reviewed, as well as its previous work. From this work it was concluded that the supervised learning methods classification and regression will be used to solve the problem of on-time estimation. In this chapter, a models will be discussed inspired by the previous work. Therefore, this chapter begins with section 5.1 where a conceptual model is proposed to answer the research question for this chapter. Which is: "What would a model look like that would solve the problem of the chosen element? ". Afterwards, section 5.2 describes how the features for the model could be obtained. Following that, section 5.3 discusses which algorithms will be used to perform the predictions. Subsequently, the metrics are discussed that will be used for measuring the performance of the models in section 5.4. The chapter concludes with section 5.5, where the answer to the previously stated research question will be given.

5.1 Conceptual model

The input and output of the model are shown in figure 5.1. The essence of the model is to convert raw order data into predictions. The input, are variables that should contain information relevant to the outcome. Predictions are the output of the model and convey whether a transportation will be on time or not.

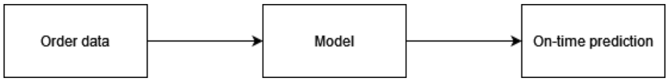


Figure 5.1: Schematic overview of the input and output of the model.

The conceptual model itself can be viewed in figure 5.2. It starts at the top with the order data from which variables are extracted. These are variables that have something to do with the transport of a good. From these variables a target variable is chosen. The target variable is the variable that predictions will aim to imitate. The other variables are going through a process of feature selection to determine which have predictive value with respect to the target variable and which do not. This process is highlighted in green since it is something that in previous work is found to be lacking and is therefore the novelty in this research, as discussed in the previous chapter. This selection process leads to a set of features that will be used to make the predictions. Two types of predictions can be made within the machine learning category of supervised learning. These are classification and regression, which were discussed in chapter 3. The output of classification is a class, which in this research will be either on-time or not on-time. Since there are only two classes it is a binary classification. For regression the output will be a number, which can be expressed as the predicted arrival time.

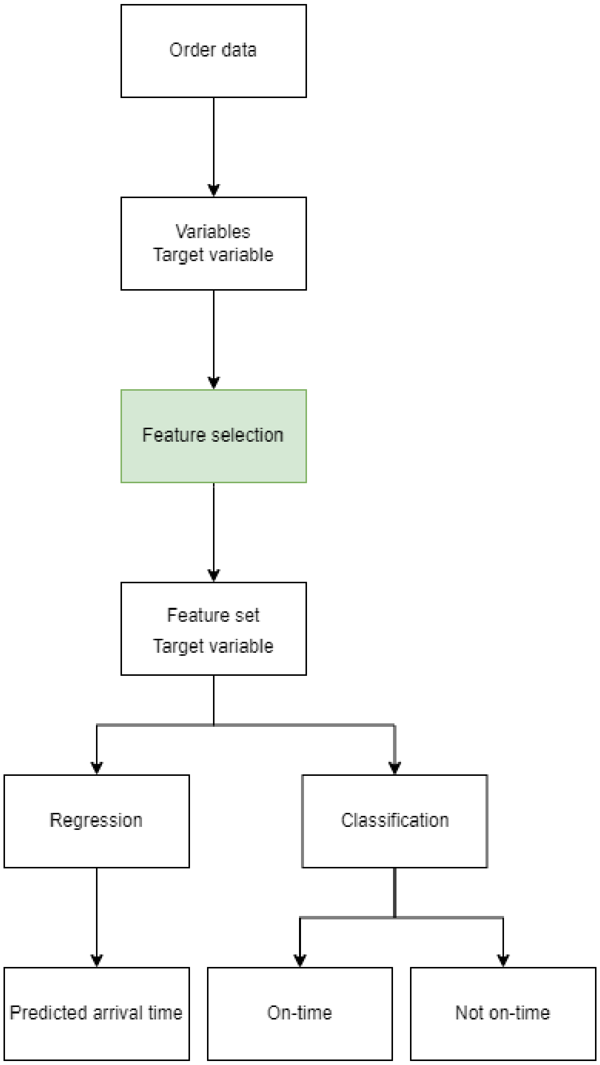


Figure 5.2: Schematic overview of the input and output of the model.

5.2 Feature selection

Before selecting a feature it is important to determine what exactly a feature is. In machine learning a feature is an individual measurable attribute or characteristic of a phenomenon. For example a model has to be created to determine whether something is a strip or a book based on some available data. This data can for example be the amount of words, sentences, pages or images contained in either the book or strip. In most cases books have more pages or less images than a strip. The combination of these two values could perhaps in most cases determine whether something is a strip or a book. These two determining values of the algorithm are then called features.

The amount of data in contemporary datasets in general requires clever algorithms and analytical thinking to determine which information is critical for solving the problem and which is not. Because, in order to find the best feature subset for a given aim, a feature selection technique must examine $2^n - 1$ subsets, where n is the total number of features in the dataset. Which is computationally infeasible even for a moderately large m (Jović, Brkić, & Bogunović, 2015). Furthermore, feature selection can also increase the accuracy of the performance (Fernandez Garcia, Iribarne, Corral, Criado, & Wang, 2018). It is therefore very important to select the right approach in determining a good subset of features.

Figure 5.3 shows an illustration of possible variables that could play a role in whether a shipment is on-time or not. From the origin this is information pertaining to which warehouse the shipment is from, the stock levels of the warehouse and location information such as country, city, address and postal code. Furthermore, information regarding the planned departure and actual departure. For the shipment this is information about the products to be carried, such as volume, weight, the number of products to be transported, and other product-related information. For transportation execution, the carrier, route, and distance to be traveled could be useful information. Similar location data as with the origin is available for the destination. Moreover, the planned and actual arrival times could be useful information. These variables were derived from discussions with the experts that were interviewed and the papers discussed in section 4.4, which is the section containing the previous work with regards to on-time estimation. Interesting to note is that most of this information is categorical instead of numerical. Categorical information is data that is divided into groups such as postal codes, carrier names or binary (yes or no) variables. The figure already contains 16 possible features, whilst it is a summarized version of the data. For example location information could be country, city, postal code or address. Which again indicates the importance of performing feature selection.

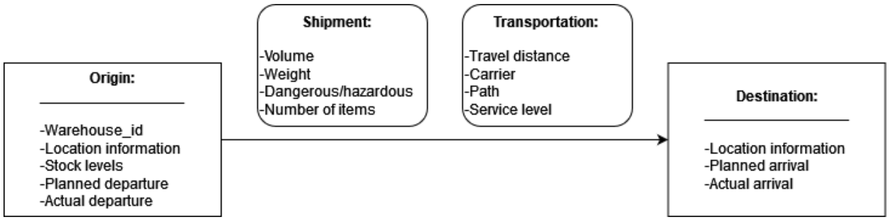


Figure 5.3: Illustration of the available data with regards to a shipment order.

There are three main types of feature selection which are filter, wrapper and embedded methods. Filter methods use variable ranking techniques as the main criteria for variable selection by ordering. The variables are scored using an appropriate ranking criterion, and variables below the threshold are eliminated (Chandrashekar & Sahin, 2014). For instance, variability may be one of these ranking criteria. The variable can be thought of as quasi-constant, for instance, when the variability is extremely low or even 0%. If that is the case it will probably not be helpful when making predictions, but will increase computation times. Therefore, these variables ought to be eliminated. Wrapper methods on the other hand evaluate feature subsets based on their performance using a modelling algorithm, which is used as a black box evaluator. Thus, a wrapper will assess subsets for classification tasks based on the performance of the classifier (Jović et al., 2015). Finally, embedded methods perform the feature selection during the models algorithm execution. Thus, these techniques are embedded as part of the algorithm's standard functionality or its expanded capabilities (Jović et al., 2015).

Wrappers perform more slowly than filters, yet it has been empirically proven that the subset they produce outperforms that of a filter. Due to the subsets being evaluated using a real modelling algorithm (Jović et al., 2015). Therefore, wrappers will be employed in this research because of their superior performance; nevertheless, to speed up computing, filtering techniques will also be used. The filter and wrapping methods used will be described in their respective subsection below.

5.2.1 Filter methods

As was previously discussed, the possible features are mostly categorical variables. The filtering method select will therefore have to be able to deal with that. Two commonly used filtering techniques that work well with categorical data are the chi-squared test and mutual information (Jantawan & Tsai, 2014) (Brownlee, 2019).

Chi-squared

Chi-square (X2) test is a nonparametric statistical test to examine if the two or more classifications of the samples are independent of one another. It is crucial to emphasize that the discovery of a statistical association using the chi-square method does not necessarily imply a causal connection between the variables being compared, but it does suggest that the association deserves further investigation (Zibran, 2007). The formula for the Chi-square test can be found in equation 5.1, Where O(i,j) is the observed value whilst E(i,j) is the expected value. Formula 5.2 can be used to compute the expected value, where $\sum_{i=1}^{c}$ is the sum of column i and $\sum_{k=1}^{c}$ is the sum of column k (Fernandez Garcia et al., 2018). The more dependent the variables are, the higher the X^2 value; conversely, the more independent the variables, the lower the value (Zibran, 2007).

$$X^2 = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{(O_{(i,j)} - E_{(i,j)})^2}{E_{(i,j)}} \tag{5.1}$$

$$E_{(i,j)} = \frac{\sum_{i=1}^{c} O_{(i,j)} \sum_{k=1}^{c} O_{(k,j)}}{N} \tag{5.2}$$

Mutual information

Mutual information is a non-negative measure of dependency between two variables (Chandrashekar & Sahin, 2014). To describe MI we must start with Shannons definition for entropy given by:

$$H(Y) = -\sum_{y} p(y)log(p(y)) \tag{5.3}$$

Equation 5.3 represents the uncertainty in output Y. Suppose we observe variable X then the conditional entropy is given by:

$$H(Y|X) = -\sum_{y\in Y}\sum_{x\in X} p(x,y)log(\frac{p(x,y)}{p(x)p(y)} \tag{5.4}$$

Equation 5.4 implies that by observing a variable X, the uncertainty in the output Y is reduced. The decrease in uncertainty is given as:

$$I(Y,X) = H(Y) - H(Y|X) \tag{5.5}$$

This gives the MI between Y and X meaning that if X and Y are independent then MI will be zero and greater than zero if they are dependent. This implies that one variable can provide information about the other thus proving dependency. From the above equations p(x,y) is the join probability function of X and Y, and p(x) and p(y) are the marginal probability distribution functions of X and Y respectively (Chandrashekar & Sahin, 2014) (Fernandez Garcia et al., 2018). The difference between the chi-square test looks at the number of occurrences, whilst mutual information looks at marginal and joint probability density functions (Richter, Knichel, & Moradi, 2019).

5.2.2 Wrapper methods

As was discussed previously wrapper methods use an algorithm to evaluate the performance of a subset of features. The papers by Wah, Ibrahim, Hamid, Abdul-Rahman, and Fong (2018) and (Poona & Ismail, 2013) both provide a comparison of different wrapper methods. In these comparisons recursive feature elimination either has the best performance or second to best. On top of that, recursive feature elimination is an algorithm that is widely available in different libraries. Because of these two reasons this is the wrapper method that will be used in this research. Important to note is that because a wrapper uses an algorithm to test the subset of features against, it is biased against that algorithm. A different algorithm should be used for validation of the derived subset of features (Jović et al., 2015).

Recursive feature elimination

Recursive feature elimination (RFE), as the name implies, is a recursive procedure that ranks features based on some metric of relevance. At each iteration, the relevance of each feature is assessed, and the least important one is discarded. The recursion is required because of the relative value of each feature might change significantly when examined across a different selection of features during the stepwise elimination procedure for specific metrics (especially for highly correlated features). A final ranking is constructed using the the (inverse) order in which features are removed (Granitto, Furlanello, Biasioli, & Gasperi, 2006). The ranking criterion could for example simply be the accuracy of the algorithm or some other metric, more on this in section 5.4.

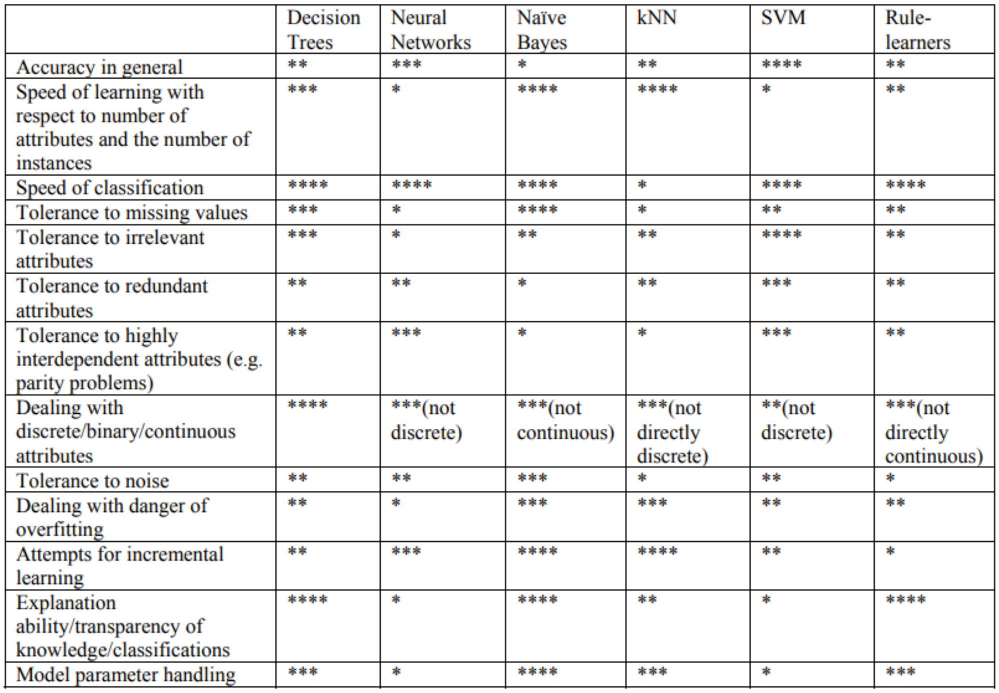
5.3 Algorithm selection

In this section the algorithms will be chosen that perform the task of classification and regression. The paper by Kotsiantis et al. (2007) provides an overview of learning techniques and performs a comparison based on a set of categories. This paper will be used as the basis to select which families of algorithms will be used. The comparison between the families of algorithms is shown in figure 5.4. As defined by MPO there are a number of selection criteria important, which are listed below:

- Highest accuracy should be the aim but not at the cost of all else.
- Speed of classification is important since these classifications are expected to be performed in real-time.
- Explainability/transparency is not as important as the aforementioned but would be nice to have. Being able to explain why certain predictions are made could solve underlying problems when for example one warehouse has more late deliveries than others according to the predictions.

Naïve Bayes has the lowest accuracy and since this is one of the criteria this group of algorithms falls off. KNN falls off due to the speed of classification, which, as previously described, is important to be performed in real-time. The other algorithm families score the highest score in that area. In the table there appears to be a trade-off between accuracy and transparency. The paper however describes that when it comes to categorical data logic-based algorithms perform the best (Kotsiantis et al., 2007). Since much of the data is categorical the features of this dataset will also most likely be mostly categorical. Therefore, either decision trees or rule learners would be a good choice. They perform similarly in terms of accuracy classification speed and transparency according to the table. However, when it comes to tolerance to irrelevant features and dealing with discrete/binary/continuous features decision trees perform better. Decision trees will therefore be selected for further examination in this research. Accuracy is still of the utmost importance in this research, and decision trees performs on par or worse than the remaining families. Therefore in comparison it would be interesting to select one of the best performing algorithm in terms of accuracy for comparison, which is neural networks.

It would also be interesting to further dive into the statement by Kotsiantis et al. (2007) that logic-based algorithms perform better with categorical data than other families of algorithms.



	Decision Trees	Neural Networks	Naïve Bayes	kNN	SVM	Rule-learners
Accuracy in general	**	***	*	**	****	**
Speed of learning with respect to number of attributes and the number of instances	***	*	****	****	*	**
Speed of classification	****	****	****	*	****	****
Tolerance to missing values	***	*	****	*	**	**
Tolerance to irrelevant attributes	***	*	**	**	****	**
Tolerance to redundant attributes	**	**	*	**	***	**
Tolerance to highly interdependent attributes (e.g. parity problems)	**	***	*	*	***	**
Dealing with discrete/binary/continuous attributes	****	***(not discrete)	***(not continuous)	***(not directly discrete)	**(not discrete)	***(not directly continuous)
Tolerance to noise	**	**	***	*	**	*
Dealing with danger of overfitting	**	*	***	***	**	**
Attempts for incremental learning	**	***	****	****	**	*
Explanation ability/transparency of knowledge/classifications	****	*	****	**	*	****
Model parameter handling	***	*	****	***	*	***

Figure 5.4: Comparison of learning techniques, where four stars represents the best and one star represents the worst performance (Kotsiantis et al., 2007).

5.3.1 Decision trees

Figure 5.5 depicts a simple decision tree model with a single binary target variable Y (0 or 1) and two continuous variables, X1 and X2, ranging from 0 to 1. A decision tree model's major main components are nodes and branches. During the development of these nodes and branches the most significant phases the model goes through are splitting, stopping and pruning.

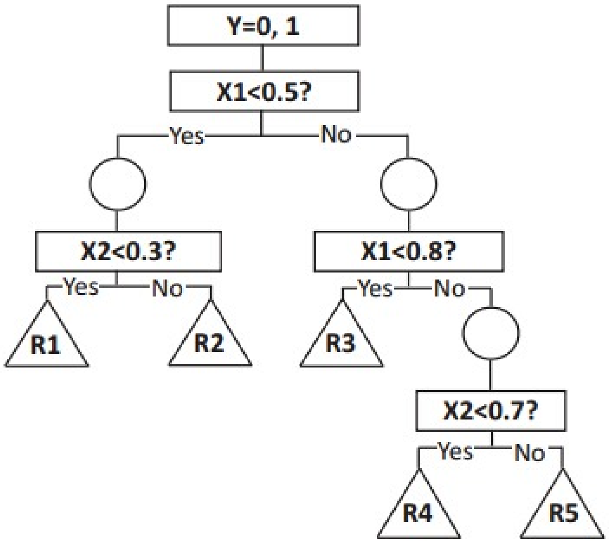


Figure 5.5: Example of a decision tree based on variables X1 and X2 and target variable Y (Song & Ying, 2015).

Nodes are classified into three categories. To begin, the root node, also known as a decision node, denotes a decision that will result in the partition of all records into two or more mutually exclusive subsets. Second, internal nodes, also known as chance nodes, indicate one of the options accessible at that time in the tree structure; the node's top edge is connected to its parent node, and the node's bottom edge is connected to its child nodes or leaf nodes. Finally, leaf nodes, also known as end nodes, describe the outcome of a series of decisions or occurrences.

Branches are events or occurrences that arise from root nodes and internal nodes. A decision tree is built using a branch hierarchy. Each path from the root node to the leaf node represents a categorization determination rule. These decision tree paths are often known as 'if-then' rules. What that means in accordance with the example in figure 5.5. If X1 = 0.3 and X2 = 0.4 then outcome R2 occurs.

The phase called splitting refers to the separation of parent nodes into purer child nodes of the target variable. In this only input variables relevant to the target variable are employed. It is possible to employ both discrete and continuous input variables. When developing the model, the most relevant input variables must first be identified, and then records must be separated into two or more categories or 'bins' based on the state of these variables at the root node and subsequent internal nodes. A set of characteristics related to the degree of 'purity' of the resultant child nodes are used such as entropy, classification error and Gini index among others. This splitting method is repeated until the homogeneity or stopping criteria are fulfilled. Most of the time, not all potential variables are used to form the decision tree model, and in some circumstances, a given input variable is used many times at different levels of the decision tree (Patel & Upadhyay, 2012).

The phase called stopping refers to the stopping of expanding the tree through splitting. Complexity and robustness are conflicting model qualities that must be evaluated concurrently while developing a statistical model. When used to anticipate future records, the more complicated a model is, the less reliable it is (Song & Ying, 2015). To ensure the balance between complexity and robustness stopping rules must be applied when building a decision tree. Stopping rules commonly employ the following paramaters: Minimum number of records in a leaf; the minimum number of records in a node prior to splitting; and the depth of any leaf from the root node. These stopping parameters must be chosen based on the analysis's purpose and the features of the dataset. Berry and Linoff (2000) advocate avoiding overfitting and underfitting by setting the goal fraction of records in a leaf node to be between 0.25 and 1% of the total training data as a rule of thumb.

The final phase is called pruning and is used in the cases that stopping rules do not operate effectively enough. Pruning is classifed into two types: pre-pruning (forward pruning) and post pruning (backward pruning). To avoid the formation of non-significant branches, pre-pruning use Chi-square tests or multiple-comparison adjustment techniques. After creating a full decision tree, post-pruning is used to remove branches in a way that enhances the overall classification Song and Ying (2015).

Random forest

A random forest is a group of decision trees that use a majority vote to determine the class that is used or in the case of regression an average. This process is illustrated in figure 5.6. Random forests have increased accuracy compared to decision tree whilst retaining (some of) the exaplainability (Qi, 2012). The accuracy is improved because a large number of relatively uncorrelated trees working together as a committee will outperform any of the individual constituent models (Yiu, 2019). To ensure that the trees are somewhat uncorrelated a process called bagging is used. Because decision trees are very sensitive to the data they are trained on, small adjustments to the training set can result in very different tree structures. This principle used in the process called bagging. By enabling each individual tree to randomly sample from the dataset with replacement and produce various trees as a consequence, random forest takes advantage of this Yiu (2019) Qi (2012). Due to the increase in accuracy whilst maintaining (some of the) explainability this algorithm will be used instead of a decision tree. In the next chapter, which is chapter 6, a number of parameters that can alter the performance of a random forest will be experimented with.

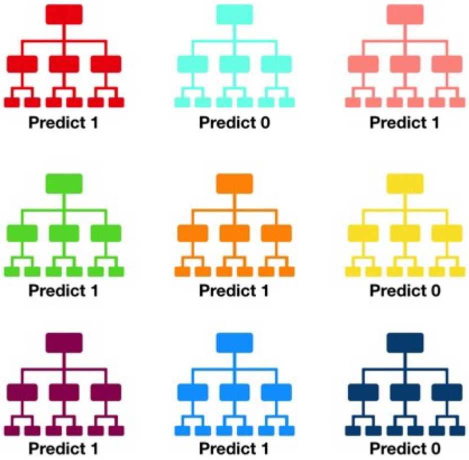


Figure 5.6: Illustration of a random forest which predicts either 0 or 1 (Yiu, 2019).

5.3.2 Neural network

Neural network is a machine learning method that mimics how the brain works and how it performs intelligent reasoning functions Mahesh (2020). Figure 5.7 provides an illustration of what this looks like in terms of machine learning. This figure shows an input layer at the bottom containing three inputs. This layer is connected to a hidden layer of 4 units. Hidden layers are the layers between the input and output, when there is more than one hidden layer the network is referred to as a deep neural network. The hidden layer finally is connected to an output layer of 5 units.

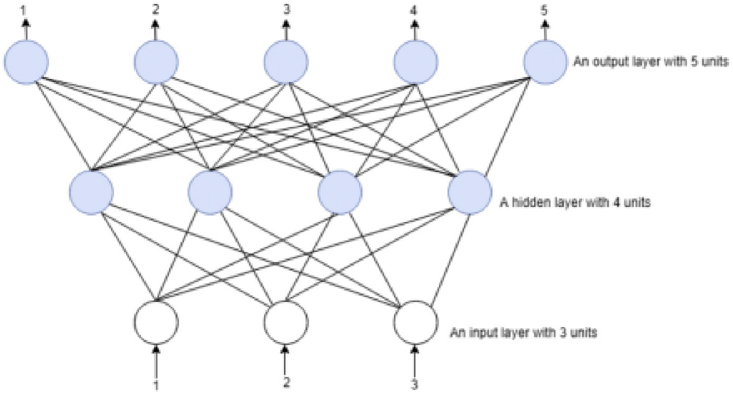

1
2
3
4
5
An output layer with 5 units
A hidden layer with 4 units
An input layer with 3 units
1
2
3

Figure 5.7: Illustration of a two-layered feedforward neural network (Abiodun et al., 2018).

The implementation that is used is a feed-forward neural network, just like figure 5.7. The exact library used for this implementation is discussed in the next chapter in section 6.2. A Feed-forward neural network means that there are only connections forward to the next layer and not back. Therefore, there is no backward flow, which is why it is called feed-forward. The opposite, called a recurrent neural network, also has a connection between the output layer and the input layer.

How these inputs are converted into outputs for a classification problem such as the one tackled in this research is through a set of calculated weights. Every connection between layers has an attached weight which is trained (Abiodun et al., 2018). The formula used to calculate the output for a single node is as follows:

$$\sum_{i=1}^{n} w_i X_i + bias \tag{5.6}$$

Where w is the weight of the connection and X are the outputs of the previous nodes. There are 1 to n connections between this node and its predecessors in the network. This output is then fed to the next layer in the network. The output of a layer has what is called an activation function which is used to convert this information to a classification (Cloud Education, 2019). The activation function:

$$output = f(x) = \begin{cases} 1 & \text{if} \sum w_i X_i + b \geq 0 \\ 0 & \text{if} \sum w_i X_i + b < 0 \end{cases} \tag{5.7}$$

The training of these weights in the H2O library is stochastic gradient descent using back-propagation. Back-propagation is the process of adjusting the weights based on the error rate, as is described by Naumov (2017). The error rate is calculated as the difference between the actual and predicted output, as follows:

$$\varepsilon = \frac{1}{2}(y - y^*)^2 \tag{5.8}$$

Where y is the actual and y* the predicted output. This error rate is used along with the learning rate and gradient descent to update the weights. This is shown in the formula below:

$$W^{i+1} = W^i - \alpha \Delta W^i b^{i+1} = b^i - \alpha \Delta b^i \tag{5.9}$$

The gradient descent refers to and ΔW^k and Δb^k of the previous formula. These are calculated as follows:

$$\frac{\vartheta \varepsilon}{\vartheta w_{ij}^{i}} and \frac{\vartheta \varepsilon}{\vartheta b_{ij}^{i}} \tag{5.10}$$

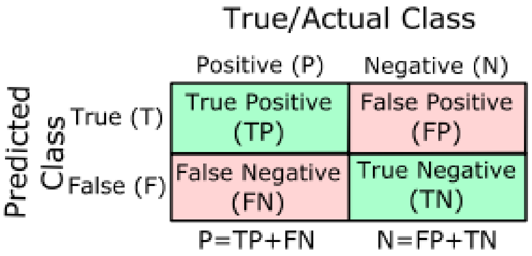
Therefore, during every iteration the a feed-forward network is used to calculate the outputs. Afterwards, back-propagation is used to calculate the error and adjust the weights using gradient descent. A number of parameters can be changed which impact the results of a neural network. Which parameters will be tested will be discussed in section 6.3 in chapter 6.

5.4 Performance metrics

The Oxford dictionary defines a metric as: "A set of numbers or statistics used for measuring something, especially results that show how well a business, school, computer program, etc. is doing." (Oxford, 2022). In this section we will be discussing performance metrics, which are used in order to validate the trained model (Tharwat, 2020). Because classification and regression have different types of results, a category and a number respectively, different ways of quantifying the results are necessary. Therefore, the performance metrics for classification and regression will be discussed separately.

5.4.1 Classification metrics

In this study the classification performed will have a binary result. In figure 5.8 the output for a binary classification problem is shown in what is called a confusion matrix. The green cells represent correct predictions, whereas the pink cells represent incorrect predictions. When a sample is positive and is accurately labeled as such by the model it is called a true positive (TP), whereas false negatives are samples that are positive but are incorrectly classified as negative, and thus a false negative (FN). On the other hand, when a sample is negative and is classified as such it is called a true negative (TN), whilst when it is labeled wrongfully it is called false positive (FP). Numerous common classification metrics are calculated using the confusion matrix and will be discussed below (Tharwat, 2020).



True/Actual Class

Predicted Class		Positive (P)	Negative (N)
	True (T)	True Positive (TP)	False Positive (FP)
	False (F)	False Negative (FN)	True Negative (TN)
		P=TP+FN	N=FP+TN

Figure 5.8: Illustrative example of a binary confusion matrix (Tharwat, 2020).

One of the most commonly used metrics is accuracy for classification performance. This performance metric is calculated through the following formula:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{5.11}$$

The accuracy metric looks at the overall picture of the predictions whilst there are also metrics that look at groups of predictions. One such metric is precision which looks at the ratio of correctly predicted positive samples compared to all actual positive samples. Precision is calculated through the following formula:

$$Precision = \frac{TP}{TP + FP} \tag{5.12}$$

Another such metric is called recall, which like precision looks at the amount of correctly predicted positive samples but then compares to the total amount of predicted samples that are positive. The formula for recall is:

$$Recall = \frac{TP}{TP + FN} \tag{5.13}$$

Another commonly used performance metric is F1 score, which combines the aforementioned precision and recall and produces a harmonized mean of the two. The formula for F1 score is described as:

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{5.14}$$

Except for accuracy these formulas only look at the positive samples of the predictions. Another metric that looks at the broader whole is called Matthews Correlation Coefficient (MCC) (Matthews, 1975). This metric represents the correlation between the observed and predicted classifications. The result is a real number between -1 and 1 where -1 means that the model predicts no better than random whilst 1 means that the model makes perfect predictions. MCC is calculated through the formula below:

$$MCC = \frac{TP * TN - FP * FN}{(TP + FP)(TP + FN)(TN + FP)(TN + FN)} \tag{5.15}$$

Since it is important to accurately judge the positive as well as negative cases the metrics that do not perform this task will not be taken into account. Furthermore, in order to accurately estimate the usability of classification it is necessary to establish how often the predictions are correct or in other words accurate. Accuracy is the discussed metric that performs these aforementioned tasks. MCC for example only gives a ratio between -1 and 1 and therefore does not give an indication of the usability. Therefore, accuracy will be used as metric for classification.

5.4.2 Regression metrics

The paper by Botchkarev (2018) provides an overview of performance metrics for regression. The most commonly used metrics according to this paper are MSE, RMSE, MAE and MAPE. These will be discussed to determine which one is most suited to this research. Mean square error (MSE) is a formula which measures the average squared difference between the predicted value and the actual value. It does so through the formula below:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - Y_i')^2 \tag{5.16}$$

By squaring the difference between the predicted and actual value, also known as the error, the result is always positive. In this formula the n stands for the total sample size and is used to get the average of all the errors, which is why this formula is called the mean square error. Root mean square error (RMSE) builds forth on this formula with the simple addition of a root. The formula for is shown below:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - Y_i')^2} \tag{5.17}$$

Whilst with MSE the error is squared, RMSE adds a root to the equation. Because of this added root it shows how much a prediction on average differs from the actual value, making it easier to understand. However, because the square is applied before the calculation and the root towards the end the RMSE formula gives a high weight to large errors. Mean absolute error (MAE) is a formula that does not use squares or roots but instead uses absolute values. As can be seen in the formula below:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - Y_i'| \tag{5.18}$$

Using this formula every error has the same weight. Which can be advantageous or disadvantageous dependent on whether it is important to minimize the outliers (higher errors) or the average. The final metric that to be discussed is the mean absolute percentage error (MAPE), as shown below:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\frac{Y_i - Y_i'}{Y_i}| \tag{5.19}$$

The disadvantage of MAPE is that it is unreliable when actual values are close to zero and does not work at all when values are close to zero. Because the actual values are in the denominator of the fraction. Furthermore, MAPE only makes sense when dealing with exclusively positive or negative data. Since in this research the predicted value can be either negative(before the estimated arrival time) or positive (after the estimated arrival tie) values MAPE would simply not work. Moreover, due to outliers having a large impact on the solution with RMSE and MSE these will also not be considered as metrics. With on-time estimation a delivery being a day, or in other words 24 hours, would have a far higher error than for example a delivery being 2 hours late. This is not a sought after effect, because that means that a single delivery being a day late weighs heavier than multiple packages being a couple of hours late.

5.5 Conclusion

In this chapter a conceptual model was discussed and in the following sections that conceptual model was was filled in, as can be seen in figure 5.9. First feature selection was discussed and two filter methods and one wrapper method were chosen to perform this task. Afterwards, the machine learning algorithms to perform the task of regression and classification were determined and discussed. Subsequently, the metrics for performance measurement were elaborated upon and for both classification and regression a metric was chosen to determine the success of the model. The resulting figure in 5.9 answers the sub-question: "What would a model look like that would solve the problem of the chosen element? ". Choices were made with regards to how the features would be determined and which algorithms would be used to make predictions for either classification and regression. The blocks highlighted in green are those that are a novelty in comparison to the reviewed previous work. These blocks correspond to the feature selection process where variables are tested for their relevance with respect to the target variable. In the next chapter experiments will be performed based on the discussed models.

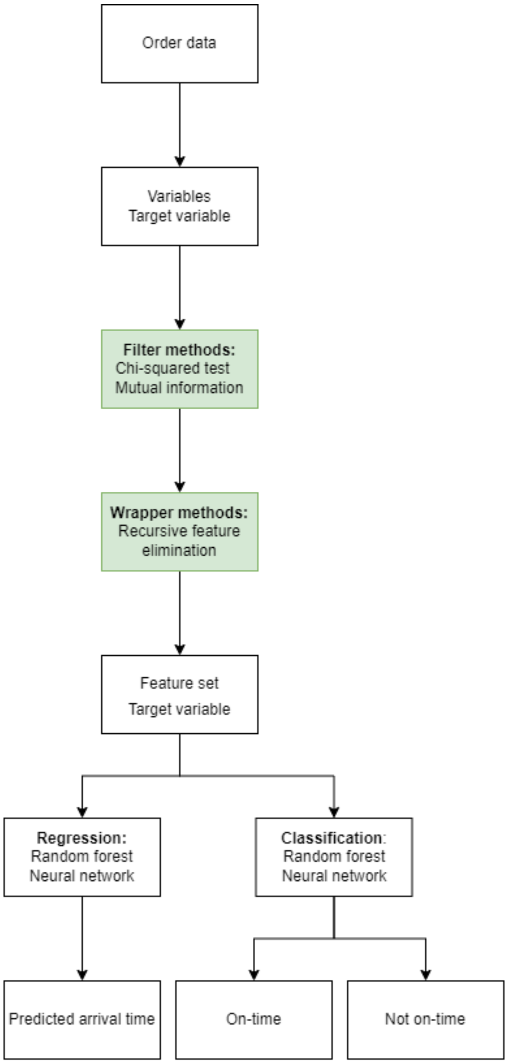


Figure 5.9: The filled in conceptual model.

Chapter 6

Experiments & results

In the previous chapter two models were proposed that can perform on-time estimation. In this chapter these two models will be experimented with. Furthermore, the answer will be given to the sub-question five for this chapter, which is: "What is the impact of these model alternatives on the performance of the chosen element?". The chapter begins with discussing the data that will be used for the models in section 6.1. Afterwards, the implementations for the algorithms will be discussed in section 6.2. Subsequently, how the experiment is performed discussed in section 6.3. Following that, the experiments and results for classification are discussed in section 6.4. For regression the experiments and results are discussed in section 6.5. Afterwards, for both regression and classification an experiment with sample sizes and an experiment with another dataset are discussed in section 6.6. Finally, the chapter ends with a conclusion in section 6.7, where also the answer to sub-question five will be given.

6.1 Order data

The data used in this research is provided by one of MPO's clients. From this dataset the data relevant to an order was filtered based in cooperation with the experts that were interviewed. This resulted in a list of variables, which can be found in table 6.1. This list contains in total 28 variables of which 4 are empty and two are nearly empty. Which is mentioned in the right-most column containing notes relevant to that specific variable. Of the 22 that are left there are twenty categorical features and two numerical features, which was expected as discussed in section C.2. The complete dataset was quite extensive and therefore filters were applied to remove incorrect data, to prioritize more recent data and to make a more manageable set of data. How this was done and what the data looks like is discussed in appendix C. This appendix also contains the performed feature selection which will be used in the coming experiments.

Name	Type	Notes
departure_month	**Categorical**	
arrival_month	**Categorical**	
departure_day	**Categorical**	
arrival_day	**Categorical**	
carrier_name	**Categorical**	
dangerous_goods_yn	**Categorical**	Boolean
from_address	**Categorical**	
from_city	**Categorical**	
from_location_id	**Categorical**	
from_postal_code	**Categorical**	
from_country_systemid	**Categorical**	
path_systemid	**Categorical**	
to_address	**Categorical**	Large number of unique entries
to_city	**Categorical**	Large number of unique entries
to_location_id	**Categorical**	Large number of unique entries
to_postal_code	**Categorical**	Large number of unique entries
to_country_systemid	**Categorical**	
same_country	**Categorical**	Boolean
country_combination	**Categorical**	
total_volume_m3	**Numerical**	
total_weight_kg	**Numerical**	
departure_difference_hours	**Numerical**	Mostly incomplete data.
departure_on_time_boolean	**Categorical**	Mostly incomplete data. Boolean.
actual_distance	-	Empty
to_region_state	-	Empty
nr_of_items	-	Empty
service_level	-	Empty

Table 6.1: The variables that are considered for features.

6.2 Implementation

Two different libraries will be used during this research to implement the discussed algorithms. These are H2O and Scikit-learn. H2O is chosen for three reasons. First, because it offers a wide selection of different machine learning algorithms, including those selected. Secondly, because it has employs many useful tools that will aid with the experiments. To name a few examples: Encoding of features, train and test set splits, multi-core processing and much more (H2O, 2022). Thirdly, because it is able to integrate with already existing MPO systems, because it runs on Java.

Scikit-learn on the other hand is also used because H2O does not have any feature selection methods. Scikit-learn is a widely used that offers many feature selection methods along with all the aforementioned that were discussed in section 5.2. The feature selection is discussed in appendix C.

6.3 Experiment setup

6.3.1 Parameters

A straightforward approach, although unfeasible for this research, would be to test every combination of parameters for a set of values. This is called a full factorial design. However, even for a small amount of values per parameter this would amount to a very large number of experiments. Since for example if every permutation for a neural network would be considered with only three different values per parameter this would equal to $3^5 = 243$ experiments. In order to keep the number of experiments feasible a different approach will be used. Which is called one factor at a time (OFAT) (Frey, Engelhardt, & Greitzer, 2003). In this approach first a baseline is created with a set of default values for the parameters. Afterwards, every parameter is tested for different values whilst all other parameters are kept constant at a default value that was chosen for the baseline. Finally best value for each parameter is chosen for an adjusted baseline. Using the OFAT approach the number of experiments does not increase exponentially, whilst every individual parameter is still tested. A drawback of this type of experiment is that the effects between parameters is not measured.

The default parameters used in this research are the recommended values by the H2O framework. These values are chosen since some of these parameters strongly corresponded to what is recommended in the machine learning literature (Bengio, 2012) (Oshiro, Perez, & Baranauskas, 2012). During the execution of the experiment plan these defaults will be compared to other values. For random trees the parameters and their selected default values can be found in table 6.2. For the first parameter, which are the number of trees, a default value of 50 was chosen. The research by Oshiro et al. (2012) compares the number of trees of a random forest across many different datasets. In this research the results only improve very slightly beyond either 32 or 64 trees. For the maximum depth of a tree a default value was 20 will be used. The third parameter is the minimum number of observations required to be at a leaf node, for which the default value will be one. For the final parameter the wrapper in the previous section determined that seven features was the best number of features to cease the removal of features, when wanting to maintain a high accuracy.

	Default value
Number of trees	50
Maximum depth of tree	20
Minimum observations for leaf	1
Number of parameters	7

Table 6.2: Default parameters for the random forest algorithm for the experiments.

As with random forest the default values are those suggested by the H2O framework. These can be found in table 6.3. These parameters were chosen since these can be altered in the H2O framework and since these are recommended to change since they (amongst others) can affect performance (Bengio, 2012). The first parameter is epochs, for which 10 was recommended, which are the number of passes over the training data. The following two parameters concern the layout of the network. Two layers (excluding input and output layer) with a 100 nodes of each were the recommended values. The next two parameters have to do with the learning scheme used by H2O which is Adadelta. This learning scheme is used ease the modelling experience. Namely, only two parameters instead of in some cases more than 7 can be altered, whilst maintaining similar computation times (Zeiler, 2012). The learning decay rate, is how much of the already learned information decays after every epoch, thus increasing the importance of more recent samples. Finally, the final parameter like with random forest was determined by the wrapper.

	Default value
Epochs	10
Neurons per layer	100
Amount of layers	2
Learning decay rate	0,99
Number of parameters	7

Table 6.3: Default parameters for the neural network algorithm for the experiments.

6.3.2 Sample size

For both of the to be tested algorithms there are many parameters that can be tuned experimented with. In order to keep computation times feasible when performing experiments with different parameters. A few experiments for run time were performed on both neural networks and random forests in a classification setting. This will also be in line with how long it would take for a regression experiment when using the same parameters. The results for the computation times can be seen in table 6.4. The numbers on the columns refer to the sample size of both on-time and not on-time sets. Therefore the total dataset is double the size that is reported on the column axis. The timestamps are in hours, minutes and seconds respectively.

	10k	**20k**	**50k**	**100k**
Random Forest	00:00:34	00:00:41	00:00:52	00:01:05
Neural network	00:11:05	00:25:05	01:17:49	03:09:21

Table 6.4: Computation times for random forest and neural networks when using different sample sizes for both on-time and not on-time. Therefore, the total size of the sample is doubled.

The conclusion that can be drawn from table 6.4 is that there is a stark contrast between the computation time between random forest and neural networks. In this timestamp the parameters, training and testing the model are all included. This difference between random forest and neural networks is most likely in large parts the training time. As was also indicated in figure 5.4, which was used to select the algorithms. To ensure the usability of a comparison between the two algorithms a similar sized data set should be used. Whilst taking in mind the limited resources and time available and the amount of parameters to be tested, an individual experiment should not exceed thirty minutes. By once again looking at table 6.4 a sample size of 20k allows for the largest set of training data whilst keep every experiment under half an hour.

6.3.3 Experiment plan

The number of samples as discussed in the previous subsection will be 20k of both classes for classification. For regression the same amount of samples will be used, which is 40k. These datasets will be the same across all experiments. Furthermore, the train test split will be partitioned in 80% and 20% respectively. Encoding of the categorical features is done through the default setting for H2O, which is enum encoding. This means that strings are mapped to integers, which means that every string will become a large integer, whilst the same strings will become the same integers. For retrieving and splitting the data the same seed will be used across all experiments for reproducibility. In table 6.5 and 6.6 the experiments for random forest and neural network respectively are shown. Also, which parameter will be tested and what values each parameter will take for each experiment.

,

Experiment	Number of trees	Maximum depth of tree	Minimum observations per leaf	Number of features
1	1, 2, 4, 8, 16, 32, 64, 128, 256	20	1	7
2	50	1, 2, 4, 8, 16, 32, 64, 128, 256	1	7
3	50	20	1, 2, 4, 8, 16, 32, 64, 128, 256	7
4	50	20	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10 , 11

Table 6.5: Experimental plan for the random forest. It shows all the parameters to be tested and which values every parameter will take for every experiment.

,

Experiment	Epochs	Neurons per layer	Amount of layers	Learning decay rate	Number of features
1	1, 2, 4, 8, 16, 32, 64, 128, 256	20	1	0.99	7
2	50	1, 2, 4, 8, 16, 32, 64, 128, 256	1	0.99	7
3	50	20	1, 2, 4, 8, 16, 32, 64, 128, 256	0.99	7
4	50	20	1	0.1, 0.3, 0.5, 0.8, 0.9, 0.95, 0.99, 0.999	7
5	50	20	1	0.99	7
6	50	20	1	0.99	1, 2, 3, 4, 5, 6, 7, 8, 9, 10 , 11

Table 6.6: Experimental plan for the neural network. It shows all the parameters to be tested and which values every parameter will take for every experiment.

The range of parameters was broadly selected in order to give an indication of the behaviour of a parameter relative to the performance metric. More than behaviour in broad terms, however, can not be concluded. For that more values for the parameters would have to be tested as well as experiments testing the relation between the variables. The performance metrics used are, as discussed in the previous chapter in section 5.4, accuracy for classification and MAE for regression. The binary incremental approach was used in the paper by Oshiro et al. (2012) for random forests and was useful in showing the behaviour of a parameter, which is why it is also used in this research. For amount of neurons per layer and the number of layers the same amount of neurons will be used across all layers. Because using the same amount of neurons for each hidden layer generally works better or equally as good as other network topologies (Bengio, 2012). The number of features is also a parameter to be tested, where the same order of removal is used as was suggested by the wrapper. That experiment can be used to validation the performance of the wrapper and to determine whether for a neural network and a random forest a similar pattern can be observed. All the experiments will be performed for both.

To ensure that these parameters are not only optimized for this specific set of the data or data split, a sensitivity experiment will be performed at the end on a combination of the newly found parameters as well as the earlier discussed baseline. This will be used as to validate the experiment plan and to compare it to the baseline evaluation.

6.4 Classification results

In this section the classification aspect of the model is handled. The baseline evaluation is done with the default values for the parameters. These can be found in table 6.2 for random forest and table 6.3 for the neural network. Table 6.7 contains the average accuracy based on ten experiments along with the minimum, maximum and standard deviation. In order to determine statistical significance a confidence interval of 95% will be used which can be calculated based on the standard deviation and number of experiments using a t-test statistic. In the table the upper and lower bound of this 95% confidence interval are also shown. From this confidence interval it can be concluded that the average result of the random forest result is not significantly different from that of the neural network. Furthermore, the results of the experiments with parameters would only be significant if they fall outside of the confidence interval bounds delineated in this table for their respective algorithm. There is no current existing system within the platform that performs this task, the orders are assumed to always arrive on-time. Since it is a binary classification problem the current system therefore has an accuracy of 50%. Both algorithms, therefore, perform significantly better. In the next subsection the results of the experiments outlined in the previous section are discussed. Afterwards, a sensitivity analysis adjusted parameters is compared to the baseline. More information of these results can also be inferred from the matrices that are shown in table D.1 and D.2 in appendix D.

	Average accuracy	**Minimum**	**Maximum**	**Standard deviation**	**Lower bound CI**	**Upper bound CI**
Random forest - baseline	72,1%	71,2%	73,1%	0,5	71,7%	72,5%
Neural network - baseline	72,2%	71,2%	73,1%	0,7	71,7%	72,6%

Table 6.7: Accuracy values for ten experiments of the baseline evaluation for the random forest and neural network algorithms.

6.4.1 Parameter experiments

In this subsection the results for the experiments as outlined by the experiment setup in section 6.3 are shown. For each experiment a graph is shown which contains three lines, the accuracy, the upper and the lower bound of the confidence interval of the default case for the respective algorithm.

Random forest

Figure 6.1 shows the accuracy relative to the number of trees. For the lower amount of trees a steep increase can be seen which reaches a plateau at around 16 trees from where there is not much movement. Afterwards, there is a very slight increase with the optimal results being achieved for 256 trees. This could be explained by the fact that more trees can capture more details, but that with 16 trees most correlations between the features and the target variable are already captured.

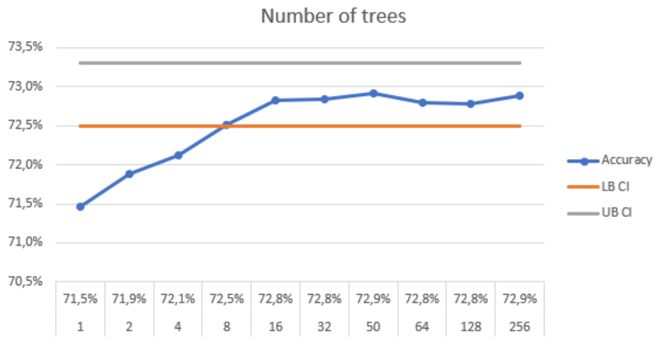


Figure 6.1: Accuracy relative to the number of trees.

Figure 6.2 depicts the accuracy as a function of the maximum tree depth. As with the number of trees there is a steep increase in accuracy that plateaus around a depth of 16. Afterwards from 20 onwards the accuracy slightly decreases. The maximum tree depth indicates the maximum number of decision nodes that can be encountered before a prediction is made. Having too few of those decision nodes means that the prediction is based off of too little information. However, trying to capture too much detail with too many decision nodes decreases accuracy, although ever so slightly. The best depth, therefore, is 16 for this experiment.

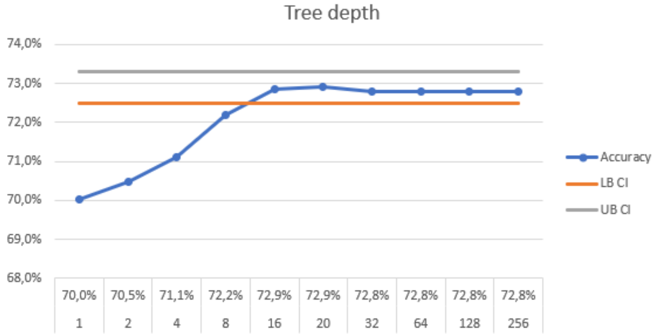


Figure 6.2: Accuracy relative to the maximum tree depth.

Figure 6.3 shows the accuracy relative to the minimum number of observations required for the algorithm to split a leaf node into a decision node. For example if this parameter is set to 10 and there are 10 true and 9 false samples, the algorithm can not split the leaf into a decision node because there have not been enough false samples encountered. The figure shows that results are either stagnant or decreasing when the minimum number of observations per leaf increases. This could be explained in that it limits the algorithm in using its own heuristics to determine whether something should be a leaf or not.

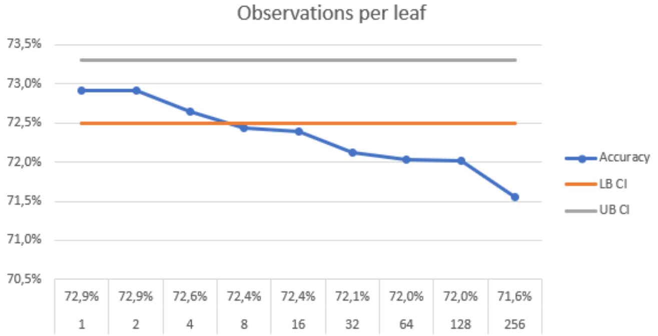


Figure 6.3: Accuracy relative to the minimum number of observations for a leaf to split.

In figure 6.4 the accuracy relative to the number of features is plotted. The more features there are the higher the accuracy becomes with the exception of the final (eleventh) parameter. Strangely enough the accuracy increases by removing this feature. This is a trend very similar to that encountered by the wrapper, which was discussed in appendix C.2. The similarity is not unsurprising since a random forest is made out of a number of decision trees. Similar to the wrapper, two features are good enough for the bulk of the accuracy. Until seven features the accuracy still increases slightly by a few percent. Between seven and ten features the increases in accuracy become even smaller. The highest accuracy is therefore found with ten features but compared to seven it is only an increase of 0.1%.

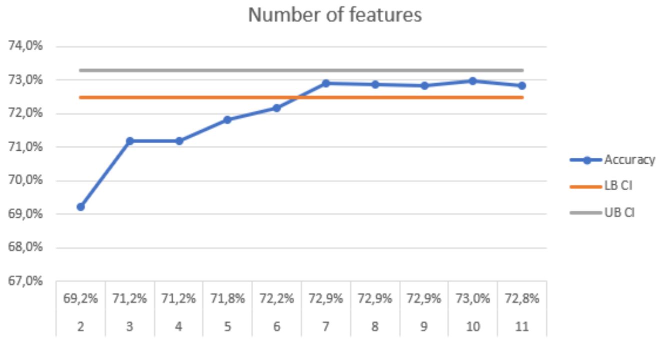


Figure 6.4: Accuracy relative to the number of features.

Neural network

Figure 6.5 shows the accuracy relative to the number of epochs. From two epochs until sixteen there is an increase in accuracy which peaks there. Afterwards, the accuracy dwindles by 0.4% for 32 epochs. The accuracy is stable for more epochs. The increase in accuracy can simply be explained in that the neural network is (correctly) learning until 16 epochs. Afterwards, the decrease in accuracy could be because of overfitting. The highest accuracy was, therefore, reached for 16 epochs.

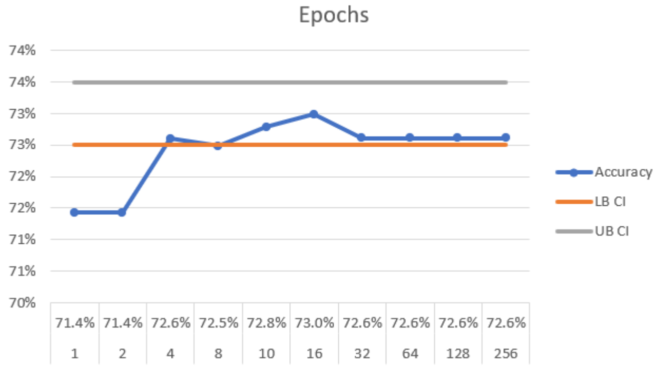


Figure 6.5: Accuracy relative to the number of epochs.

Figure 6.5 depicts the accuracy as a function of the number of neurons per layer. The accuracy trend in this figure is not very consistent with many steps up and down. The general trend, however, appears to be upward. The highest accuracy is reached for the default value of a 100 neurons per layer. The more neurons per layer the more complex relations can be made, which would allow for the model to capture more detail. However, with a 100 neurons most of this detail appears to be captured.

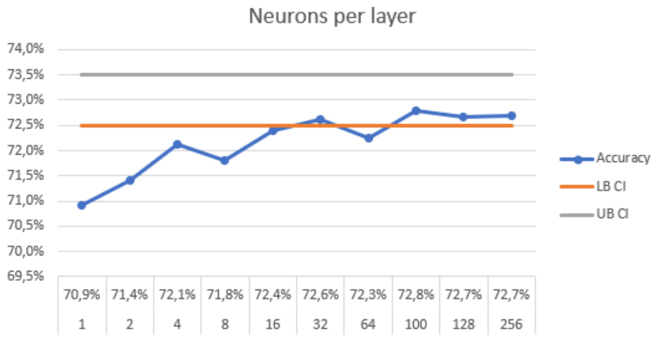


Figure 6.6: Accuracy relative to the number of neurons per layer.

In figure 6.7 the accuracy is plotted for the number of layers. The highest accuracy is reached with two layers, which is the default value. The impact of the number of layers does not appear to be as significant as for example epochs or the number of neurons per layer. The difference between the worst and best performing value is only 0.8%. After two layers, adding another layer only results in minor decreases or increases in accuracy between 0.1% and 0.2% until adding layer eight with a decrease of 0.3% in accuracy relative to layer seven. Therefore, adding more complexity to the model does not result in improved performance beyond two layers.

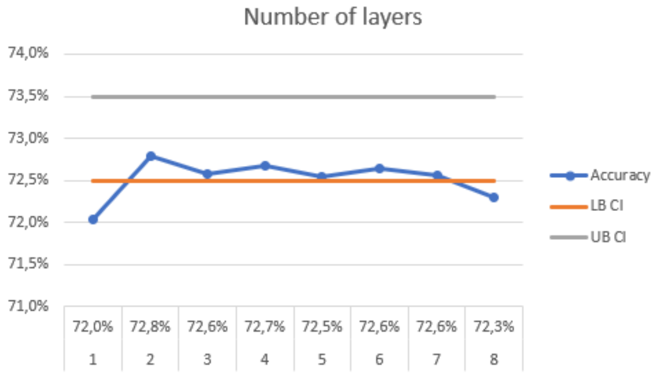


Figure 6.7: Accuracy relative to the number of layers.

The accuracy is shown against the learning decay rate in figure 6.8. The number 0.99, which is the default value for this parameter, indicates that the learning rate decreases by 1% after every epoch or in other words 99% of the learning rate is kept for the next iteration. The poorest scoring value here is a learning decay rate of 0.3 and 0.5, which indicates that decimating the learning rate after every epoch by half or more does not provide good results relative to the other decay rates. The highest accuracy is achieved for 0.99. Therefore, the slower the decay the better results. This might also have something to do with that according to the figure about the number of epochs the algorithm is not done learning until sixteen epochs. These parameters, in that sense, influence each other.

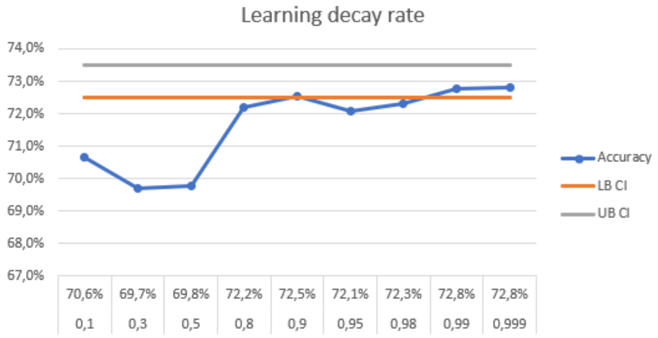


Figure 6.8: Accuracy relative to the learning decay rate.

Figure 6.9 depicts the accuracy in relation to the number of features. Similar to the wrapper and random forest with two features most of the accuracy can be reached. The highest accuracy is achieved with seven features, which is the default, with nine features being a close second. In that sense the line drawn with the help of the wrapper with seven features is validated for a neural network. However, in contrast with random forest the eleventh feature has a positive effect on accuracy, since removing it reduces accuracy by 0.5%.

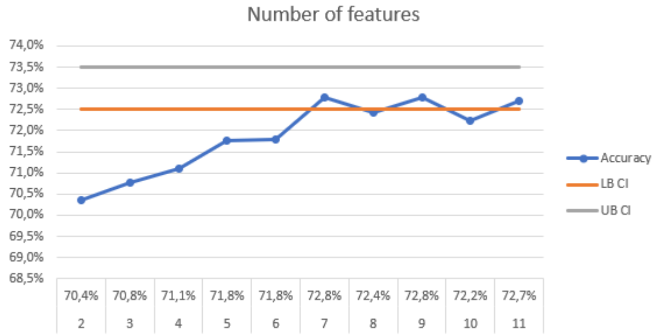


Figure 6.9: Accuracy relative to the number of features.

6.4.2 Sensitivity analysis

The sensitivity analysis was performed on the same sample size as the experiments. Namely, 20k samples for both on-time and not on-time. The test was performed ten different times. The samples were randomly selected samples every time. Afterwards these samples were randomly split into a train and test set. The results can be found in table 6.8. Interesting to note is that the average accuracy for the adjusted parameters for random forest is higher than the initial baseline, although not significantly so. Whilst for the neural network the adjusted parameters produce lower accuracy results on average and also not significantly so. A reason for this can be that even though for the adjust parameters the best parameter was individually chosen, that does not mean that the combination of these parameters improves results. Since the experimentation approach chosen in this research does not test the relations of the parameters relative to the accuracy, because only one parameter is adjusted at a time. When looking at the standard deviation it can clearly be seen that the adjusted parameters have a lower standard deviation than the initial baselines. Which indicates that for these ten samples the adjusted baselines are more robust to random samples of data than the initial baselines. All in all, with a confidence interval of 95% the results between the baseline and adjusted parameter variant are not significant. Neither is random forest significantly different from the neural network in performance. However, whilst taking run-time into account random forest performs much better than neural networks as it is a lot faster. The previously discussed table 6.4 shows that for this sample size random forest is more than 20 times faster than neural networks.

	Average accuracy	Minimum	Maximum	Standard deviation
Random forest - baseline	72,1%	71,2%	73,1%	0.5%
Random forest - adjusted	72,5%	71,9%	73,2%	0.4%
Neural network - baseline	72,2%	71,2%	73,1%	0.7%
Neural network - adjusted	71,8%	70,7%	72,8%	0.6%

Table 6.8: Sensitivity analysis on the baseline and adjusted baselines for random forest and neural networks. Shown is the average, minimum and maximum value that were encountered.

6.5 Regression results

In this section the regression aspect of the model is handled. The baseline evaluation is done with the default values for the parameters. These can be found in table 6.2 for random forest and table 6.3 for the neural network. In table 6.9 the results of this baseline evaluation are shown. As discussed in section 5.4, MAE will be the metric used to measure the performance of the results. Table 6.9 contains results of a sensitivity analysis of ten tests on different groups of samples of size 40k. In the table can be seen that the average error of a sample is around the four to five hour mark according to MAE. The table also contains, as with classification, an upper and lower bound of a 95% confidence interval, which was calculated with a t-test. From this it can quickly be seen that the average MAE of the random forest falls within the confidence interval of the neural network, but, interestingly enough, not the other way around. The standard deviation for random forest, however, is about half as much as neural network. Indicating that it is slightly more robust in its results although they are slightly worse, but not significantly so. The MAE of the current planning system for these samples is 2,49. Both algorithms, therefore, do not improve the results of the current system. In appendix D table D.3 can be found, which shows which time-frame would contain a certain percentile of the predictions. This table was created in order to give an indication for what these predictions could be used and how useful it is in the current state. In the next subsection the results of the experiments outlined in section 6.3 are discussed. Afterwards, a sensitivity analysis is performed on the adjusted parameter variants, which are compared to the baseline evaluations.

	Average MAE	Minimum	Maximum	Standard deviation	Lower bound CI	Upper bound CI
Random forest - baseline	4,83	4,66	4,99	0,1	4,76	4,90
Neural network - baseline	4,69	4,32	5,02	0,22	4,53	4,84

Table 6.9: Mean average error (MAE) in hours for ten experiments of the baseline evaluation for the random forest and neural network algorithms.

6.5.1 Parameter experiments

In this subsection the results for the experiments as outlined by the experiment setup in section 6.3 are shown. For each experiment a graph is shown which contains three lines, the MAE, the upper and the lower bound of the confidence interval of the default case for the respective algorithm.

Random forest

Figure 6.10 shows the mean average error relative to the number of trees. The figure starts with a steady decline by adding more trees. Which can be explained by more trees being able to capture more detail. Do take into account that there is an exponential increase in the number of trees on the x-axis so whilst it might look like an almost linear decline this is not the case. Beyond 32 trees the line starts to become more stagnant but the minimum MAE is achieved with 256 trees. This is, however, not a significant improvement from the base evaluation since it falls within the confidence interval.

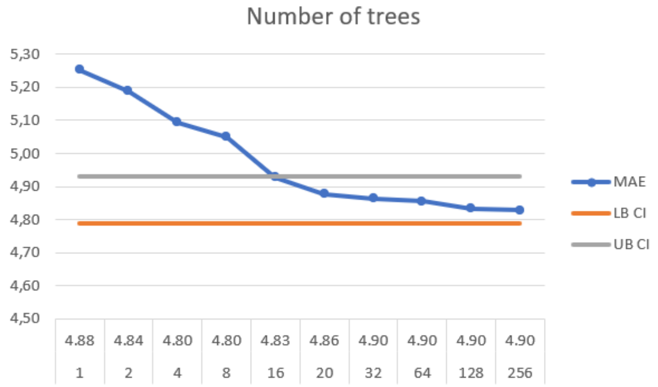


Figure 6.10: MAE in hours relative to the number of trees.

Figure 6.11 depicts the MAE as a function of the maximum tree depth. Whilst the figure starts with a very slight decrease (see y-axis). It increases beyond a depth of 8, although also very slightly. All the parameter values fall within the confidence interval and therefore this parameter does not have a significant impact on the result. The increase in error can be explained by that the trees create too many decision nodes of which the decisions do not achieve the right conclusions. More decision nodes do not guarantee a better result as is clearly shown here. The lowest MAE for these parameter values can be found for a maximum tree depth of 4.

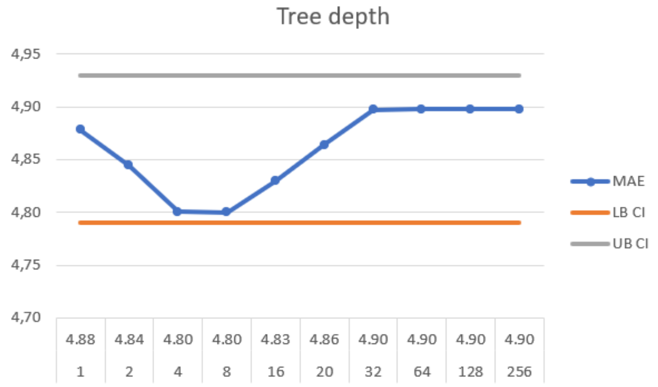


Figure 6.11: MAE in hours relative to the maximum tree depth.

Figure 6.12 shows the MAE relative to the minimum number of observations required for the algorithm to split a leaf node into a decision node. Similarly to the tree depth previously discussed this parameter for the most part does not have a significant impact. The significant impact that it does have have is only negatively since it increases the error instead of decreasing it. The MAE is relatively low in the figure until eight observations per leaf after which there is an steady increase. The lowest MAE can be found for two observations per leaf.

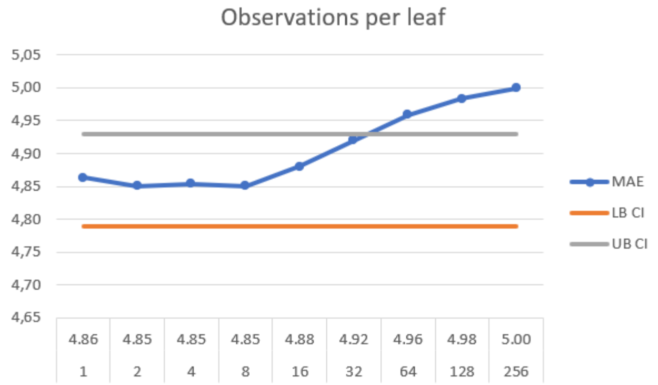


Figure 6.12: MAE in hours relative to the minimum number of observations required for a leaf to split.

In figure 6.13 the mean average error relative to the number of features is plotted. Whilst the wrapper was used on classification using the RF algorithm it performs similarly for regression, as can be seen in appendix C.2. In broad terms, the more features that are used the better the performance with not much difference in performance between seven and eleven features. The value of seven features ,which was selected from the wrapper process, is once again achieves good results and only being very slightly outperformed by ten features. Ten features therefore achieves the lowest MAE, although once again not significantly so.

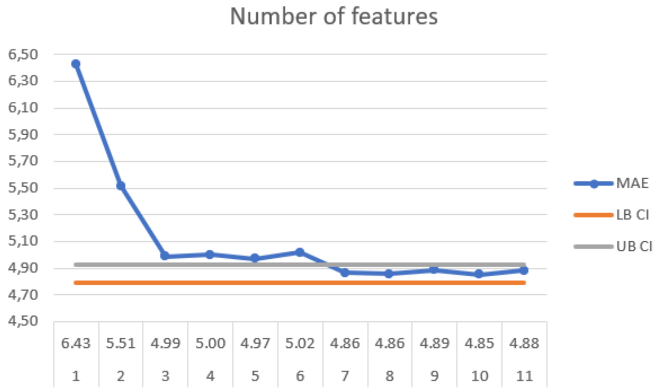


Figure 6.13: MAE in hours relative to the number of features.

Neural network

Figure 6.14 shows the MAE relative to the number of epochs. An increase in the number of epochs in general, with the exception of 32 epochs, reduces the error. The error stops decreasing at 64 epochs after which no decrease is measured. Which is not surprising, since it allows more samples for the weights to learn. The risk with iterating multiple times over the same data is overfitting. Which means that the weights are trained too much on the training set which results in a poor performance on a test set. This however did not materialize. The number of epochs with the minimum error therefore is 64 which is a significant improvement.

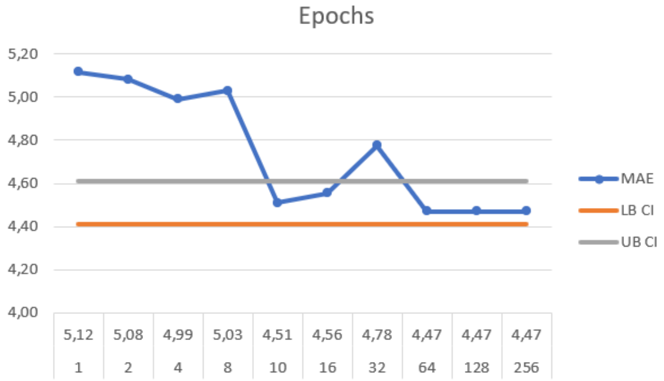


Figure 6.14: MAE in hours relative to the number of epochs.

Figure 6.15 depicts the mean average error as a function of the number of neurons per layer. The number of neurons has a very large effect on the error as can be seen in the figure. At the start the error is above 5 whilst towards the end it is around 4.3. Furthermore, the line does not seem to flatten, instead there appears to be a consistent decrease. Important to mention, however, is that the x-axis increases exponentially the decrease in error is therefore not linear. This figure still indicates that the error could decrease for a larger number of neurons per layer. The lowest error is achieved for 256 neurons, which is a significant improvement.

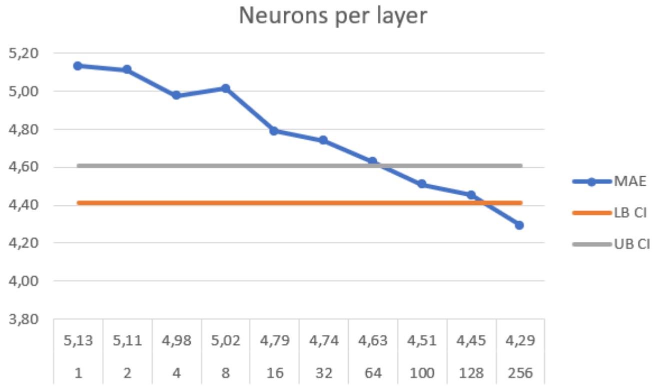


Figure 6.15: MAE in hours relative to the number of neurons per layer.

In figure 6.16 the MAE is plotted for the number of layers. The number of layers, contrary to epochs and neurons, do not seem to have too much of an impact on the results. Aside from using a single layer not giving relatively good results the other layers all perform in a similar fashion. The lowest error is achieved with seven layers.

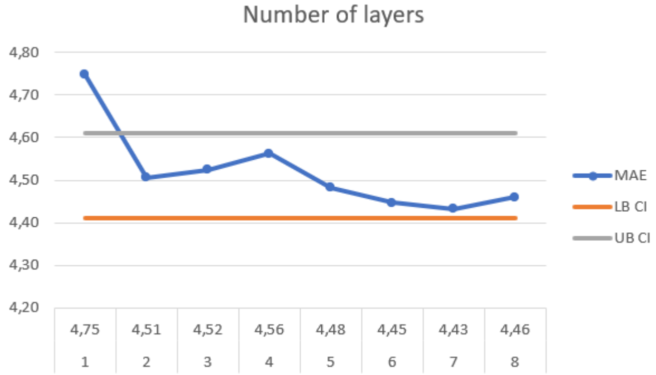


Figure 6.16: MAE in hours relative to the number of layers.

The MAE is shown against the learning decay rate in figure 6.17. The error is lowest the closer to 1 the learning decay rate is. Which essentially means that the learning from previous epochs is remembered (almost) entirely. The jump between error between 0.99 and 0.999 is relatively large to the others and might indicate that for an even higher learning decay rate a lower error might be achieved. The best value is found for 0.999 learning rate decay which is a significant improvement relative to the baseline.

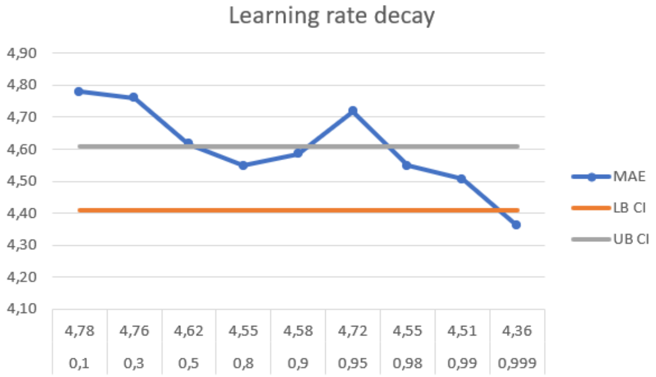


Figure 6.17: MAE in hours relative to the learning rate decay.

In figure 6.18 the MAE is plotted for the number of features. The number of features has slightly more of an impact on the error than with random forest. The parameter, however, once again performs very similarly compared to its classification counterpart and the wrapper. The lowest error is achieved with seven features which was the base value chosen based on the wrapper.

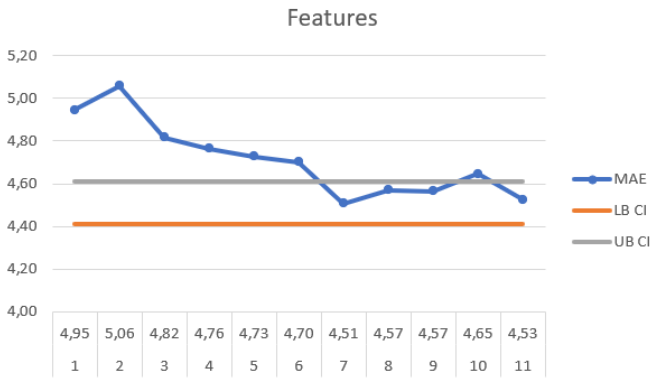


Figure 6.18: MAE in hours relative to the number of features.

6.5.2 Sensitivity analysis

In table 6.10 the results of the sensitivity analysis for the adjusted parameter variants can be found along with those of the baseline that were previously discussed. Once again a confidence interval of 95% will be used to compare, which is calculated by adding or subtracting twice the standard deviation. For both the adjusted variants the average error decreased. Whilst for random forest only very slightly and not significantly so. For the neural network quite a bit more and significantly so since the lower bound was 4,53 and the average MAE for the adjusted variant is 4,40. Interestingly enough the standard deviation for neural network increased by quite a bit, thus reducing robustness, whilst for random forest it remained the same.

	Average MAE	Minimum	Maximum	Standard deviation
Random forest - baseline	4,83	4,66	4,99	0,1
Random forest - adjusted	4,79	4,56	4,95	0,1
Neural network - baseline	4,69	4,32	5,02	0,22
Neural network - adjusted	4,40	4,03	4,97	0,29

Table 6.10: Mean average error (MAE) in hours for the baseline evaluation along with the adjusted variant for random forest and neural network.

6.6

6.6.1 Scale experiments

In table 6.11 the accuracy is shown for the different variants and scale sizes of the experiments. Using the confidence interval determined earlier, there are no significant differences. The variants all behave similarly and will therefore be discussed as such. An increase is seen between the 40k and 100k sample sizes across all the variants and from there a slight decrease for the 200k and 500k sample sizes. The highest accuracy is achieved by all variants on 100k. This indicates that the correlation between the features and the target variable is the strongest for this sample size.

	40k	**100k**	**200k**	**500k**
Random forest - default	72,9%	73,4%	72,9%	72,9%
Random forest - adjusted	73,0%	73,3%	72,9%	72,8%
Neural network - default	72,8%	73,1%	72,6%	72,7%
Neural network - adjusted	72,6%	73,2%	72,8%	72,8%

Table 6.11: Showing the accuracy for the scale experiment. The number on the columns show the size of the train/test set which was split 80%/20% respectively.

In table 6.12 the MAE is shown for the different variants and scale sizes of the experiments. Using the confidence interval determined earlier, significant differences can be seen in the table. For the default random forest there is a significant improvement between 40k and 500k sample sizes, but a steady decrease in the error can be seen for the increases in sample size. The adjusted random forest on the other hand is barely affected by changes in sample size and no significant change is measurable. This could be due to the parameters being selected for the 40k sample size and relatively underperforming on the other sample size. The default neural network is does not appear to be as robust or consistent as the random forest, as indicated previously as well. The adjusted variant of the neural network performs very strangely with an error of almost 14 for the 100k sample size. The error slowly seems to stabilize for the 200k and 500k error. Most likely the parameters chosen perform well on the 40k sample size but are very bad for the other sample sizes. For example the learning rate for the neural network is increased for the adjusted variant which might work well on a small sample size, but as the sample size increases a high learning rate might not be good.

	40k	**100k**	**200k**	**500k**
Random forest - default	4,86	4,73	4,67	4,66
Random forest - adjusted	4,78	4,73	4,72	4,74
Neural network - default	4,51	4,93	4,53	5,06
Neural network - adjusted	4,92	13,93	6,18	5,28

Table 6.12: Showing the MAE in hours for the scale experiment. The number on the columns show the size of the train/test set which was split 80%/20% respectively.

6.6.2 Dataset comparison

Table 6.13 shows the accuracy of the classification model of the original dataset that was used for feature selection and creating the variants. Also shown is the dataset that is used for comparison and validation, which is named alternative dataset. There is a slight increase in accuracy measurable of about 2-3% across all variants. For the existing system 50% accuracy is chosen, because there is no existing system and is therefore always assumed to be on-time. In a binary classification problem with a balanced dataset, which is used for these experiments, it would be correct 50% of the time. Whilst the original dataset already improved upon the existing system, the alternative dataset performs significantly better across all variants.

	Existing system	**RF - default**	**RF - adjusted**	**NN - default**	**NN - adjusted**
Original dataset	50%	72,9%	72,8%	73,0%	72,6%
Alternative dataset	50%	75,9%	75,6%	74,8%	75,2%

Table 6.13: Showing the accuracy for the original and other dataset.

Table 6.9 shows the MAE of the regression model of the original dataset that was used for feature selection and creating the variants. Also shown is the dataset that is used for comparison and validation, which is named alternative dataset. Unlike classification there was an existing system that performed a task similar to the regression. The error of the original dataset is about halve that of the variants. Although the error for the variants is a lot higher for the alternative, so is the existing system error. The error ratio between existing system and variants almost stays the same at around twice as much between both datasets. Although, the existing system is not improved upon the results are consistent across different datasets, which at least validates the model in that sense.

	Existing system	**RF - default**	**RF - adjusted**	**NN - default**	**NN - adjusted**
Original dataset	2,57	4,86	4,78	4,92	4,51
Alternative dataset	10,69	20,78	21,48	20,58	19,78

Table 6.14: Showing the MAE in hours for the original and other dataset.

6.7 Conclusion

The chapter starts with discussing the data that will be used in order for the machine learning algorithms to make the predictions. Afterwards, experiment plan was discussed on how the experiments will be performed. A baseline set of parameters was determined for both algorithms and the experiments are used to find out if other values for parameters could improve the results. Subsequently, the experiments were conducted in the following two sections for classification and regression respectively. These results will now be used to answer the final sub-question of this research, which is: "What is the impact of these model alternatives on the performance of the chosen element?".

First the classification results were discussed with an average accuracy for both algorithms in the 72-73% range. The usage of these algorithms would be much hampered by this accuracy. The around 30% wrong predictions can be split into two categories. Either false positives, which mean that the delivery is predicted to be on time while it will be late, or false negatives, which mean that the delivery is predicted to be late while it will be on time. A false positive is what the system currently in place does, since there is no feedback loop concerning projected arrival time. This would therefore not be problematic. However, a false negative would indicate a problem that does not exist. This would require the attention of an employee to figure out whether action is actually needed, since an employee would not be able to tell a true negative from a false negative without looking into the problem. From the confusion matrices in table D.1 and D.2 in appendix D it can be inferred that false negatives do occur regularly. For random forest this occurs 13% of all predictions and for neural network 10%. If in 10% of the deliveries an employee is looking into a delivery that is not a problem it is a large amount of wasted resources. This alone means that it is unlikely to see broad use in practice in its current form. Supply chains with expensive shipments or where it is very important that deliveries are made on-time still might make use of it. The (lack of) accuracy could be explained by the fact that the target variable had to be created with a number of assumptions, outlined in appendix 3.3, in order to correspond to the selected element's purpose, which was on-time estimation. In doing so, the relation between the features and the target variable might have been lost or reduced, if it was ever there. However, an accuracy of a binary classification problem that achieves 72-73% accuracy on tens of thousands of samples clearly indicates that it is not random since that accuracy would most likely be closer to 50%. So it can be concluded that a correlation was found by both machine learning algorithms. The scale experiment showed that there is barely any impact in increasing the sample size. On the other hand, for the alternative dataset that was used to validate the results there was a significant increase in results in about 2-3%. Clearly, the correlation between the variables and target variable used is stronger for that dataset than in the original one. Furthermore, since feature selection was not performed on the alternative dataset there might still be room for even more improvement.

For regression the MAE can be interpreted as hours since that is how the target variable was expressed as.

Therefore the results can be interpreted as being on average wrong by about four to five hours. One way in which this could be useful is for example when outliers are (accurately) predicted to see why this is the case by interpreting the random forest. Interpretation of a neural network is very difficult or near possible, depending on the structure of the network, as is discussed in 5.3. Determining the causes of outliers could be useful in solving underlying logistical problems. Another possible use is to the prediction and to establish, based on a certain percentile for example, a time-frame within which the delivery will take place. Table D.3 in appendix D shows a table for the baseline that contains this information. For example a time-frame of 6 hours would cover 70% of the samples. All in all, classification will most likely not be very useful in practice due to the limitations in accuracy, but mainly the amount of false negatives. On the other hand, although for regression the adjusted parameters were not significant, the results were still adequate enough to be of use in practice. For example in the form of a time-frame. The scale experiment, only baseline evaluation for random forest showed significant improvement with more data. The others were either consistent or performed slightly worse. For the neural network variants it seemed like the algorithm had not converged to a solution yet. The dataset comparison, on the other hand, showed a very consistent error between the original and the alternative dataset relative to teh existing system error. Namely, the error of the variants relative to existing system error was about 1.8-1.9 for both dataset. This validates that the model works similarly on another dataset and presumably will also do so on others.

Chapter 7

Conclusion, discussion & recommendations

7.1 Conclusion

In this thesis the main research question to be answered is: "How can supply chain management be improved through the use of machine learning?". To answer this question the research began with a literature study of supply chain management as well as machine learning to gather insights of what these fields entail and how they might interact. Following this literature study a deeper look was taken at what MPO does and where machine learning may be used to improve elements of the present system. A series of interviews, as well as analysis of the platform, were undertaken, yielding five elements where machine learning might be applied to to facilitate improvements. This answers the prerequisite of the main research question, namely that there are actually topics where improvements may be made within MPO. However, it does not address how these improvements can be made. Since it would not be possible in the span of a thesis to address all these problems, one had to be selected. Therefore, one topic out of these five topics was selected based on a set of criteria, which can be found in section 4.2. The topic selected was on-time estimation. Previous work on the topic of on-time estimation was presented, and two machine learning models were used to solve this problem. These are classification and regression. These two models were incorporated into a conceptual model.

Through the use of more interviews, analysis and the previous work; a set of variables was deemed relevant to consider for feature selection. These variables were chosen as features based on how well they performed in two filter methods (chi-square and mutual information) and a wrapper method (recursive feature elimination). Classification and regression were tested using two algorithms: Random forest and neural network. These algorithms were chosen using a set of criteria, detailed in section 5.3. The data, which will be used for the algorithms, provided by MPO's client is substantial. Therefore, the number of features had to be reduced in order to decrease the complexity and computation times. Both algorithms have a set of parameters that influence their results. Since it was not possible to evaluate most or all combinations of parameters, a different approach was employed. In this approach one parameter is altered while the others are held constant to examine how the changing parameter affects the result. After performing this process for every parameter a new set of parameters could be chosen based on their individual optimal performance.

For classification the results between the adjusted parameters and the baseline parameters was not significant. This is most likely because the experimentation approach used ignores the relationships between parameters. Moreover, for the individual experiments the baseline values for the parameters already performed rather well, either being the best tested value for a parameter or very close. On the other hand, there were multiple values for every parameter that performed significantly worse than the baseline evaluation. Therefore, it can be concluded that the baseline parameter values chosen were relatively good compared to the other tested values for the parameters. The results of classification accuracy was around 72-73%. The existing system that performs the task similar to classification achieves a 50% accuracy. Therefore the machine learning model has a large and significant increase in accuracy. However, for broad usage in practice the results might have to be improved, more on how this could be done in the discussion. Supply chains with expensive shipments or where it is very important that deliveries are made on-time still might make use of it. The (lack of) accuracy could be explained by the fact that the target variable had to be created with a number of assumptions in order to correspond to the selected element's purpose, which was on-time estimation. Therefore, there may have been a loss of a clear relationship between the variables and the target variable. In contrast, the target variable for regression is much more straightforward. As it is is simply the difference between the actual arrival time and planned arrival time. For regression the baseline parameters also performed rather well, once again most of the values for the parameters had only a negatively significant impact on the results. The adjusted variant for random forest performed similarly to the baseline evaluation whilst for neural network the adjusted variant significantly improved results compared to the baseline neural network evaluation. The average mean error (MAE) for both algorithms was between four and five hours. A different representation of the results, however, could be by giving a time-frame within which a certain percentage of the predictions lie. For both algorithms 70% of the predictions lie within a six hour time-frame (three hours before and three hours after the prediction). This could be used in order to further clarify to the customer when to expect a package. It would also be more informative than the currently existing system which just provides a specific hour which is closer to a rough estimate of when to expect a delivery. The existing system for had an MAE of 2.57 and the variants had an error that was about 1.8 or 1.9 times larger. The model was also tested on another dataset where the ratio of error between existing system and variants was similar, thus validating at least the consistency of results across multiple datasets. Due to the MAE for all variants being higher than that of the existing system the machine learning model for regression could not replace the existing system, but could exist alongside it to evaluate the achievability plannings. However, for companies that do not have a planning program as extensive as MPO, this model might provide more accurate arrival times.

The main research question of this thesis is: "How can supply chain management be improved through the use of machine learning?". A model was proposed that performs on-time estimation. In addition, the research gap was addressed by doing so without data concerning mode, route, traffic or weather. However, this data was not used due to unavailability, since it might very well improve the results of the model if that data is included. Furthermore, this model could be applied to other data sets within MPO, as was experimented with, and achieve similar results. This could also be done by other companies that have similar data available, as discussed in this report. Moreover, the proposed model and the approach taken to determine this model could be used as a basis to be applied to different elements within supply chain management, but for this purpose the content of parts of the model might have to be altered depending on the problem.

7.2 Discussion & Recommendations

The abundance of data available would in most cases be considered a positive thing. However, due to the limits in time and computation power only a small subset of the entire dataset could be considered at once. Therefore, some variables that would have been interesting to consider might have performed worse during example feature selection, which lead to them being not selected. For example the variable to_city had a large number of unique cities in the dataset who had only a single or a couple of samples. If the entire dataset had been used this might not have been a problem and that variable might have produced significant predictive power. Furthermore, one of the filters applied in order to reduce the dataset to a more manageable size was was the date the first of January 2021. Variables that for example looked at months or years would most likely not have a lot of predictive power, since only 1 one and a half year was contained in the dataset (out of 6 years). A solution to this problem would have been to randomly select samples that could have been taken from every month/year combination relative to the total population of the dataset. This would be an interesting angle to continue this research to see whether these variables have predictive power.

The wrapper method used for feature selection was a very computationally intensive method, which is why it was only performed once. Ideally, it would have been performed for both classification and regression as well as for both neural network and random forest. Due to it being so computationally and time intensive a more basic algorithm was selected to perform this task. A decision tree was used since it is what a random forest is made up of and was therefore expected to achieve similar results. When looking at the parameter experiment for random forest the results were similar to that of the wrapper performed by the decision tree. However, for neural network these results were slightly different. Therefore, the order of removal for the variables might have been different. For further research it could be interesting to see if there would be a difference between the wrapper performance between the two algorithms but also between the two models. If so, whether this would affect results at all.

The experiment approach chosen was one factor at a time (OFAT), had clear limitations. This approach was chosen to limit the number of possible experiments, the disadvantage is that the interaction between the parameters for the experiments is not taken into account. Therefore, the results for the adjusted parameters for neural network and random forest were, presumably, not significantly different from the baseline evaluation. It would be interesting with more computation power to test combinations of parameters or achieve optimal parameters through the use of, for example, a grid search. Furthermore, to test whether these new parameters then significantly improve results or that using this approach better results can be achieved.

The features found in this research were encountered for one of MPO's clients. This does not mean that these features have the same predictive power for another dataset. This is already somewhat shown in the experiment with the alternative dataset where the accuracy is about 2-3% higher. However, with other parameters results for this dataset might achieve even a far higher result. Therefore, the process of feature selection would have to be executed for a different dataset, to determine whether similar features are deemed important across different datasets or whether the derived feature set is more unique.

The trained random forest for both classification and regression could be used in an alternative way than experimented with. Since every tree of the random forest makes its own prediction it could be said that if a large number of the trees, for classification for example, predict the same thing (i.e. more than 90%) it might indicate a certain confidence in that decision. It would be interesting to see if there is a correlation between a high confidence by a large of amount of trees relative to correct predictions. In other words, to only show predictions with a certain amount of confidence by the random forest. Furthermore, these trees could be used to extrapolate data about poorly performing deliveries and what the reasons for that are. These trees could be extrapolated for interesting connections that have been learned that lead to predicting late deliveries. For example, the tree might make connections such as: If origin is warehouse x , carrier is y and country is z then the delivery will be late. This could be used to solve logistics problems that might not be easy to recognize or to find poor performing aspects.

For regression the target variable had outliers that were very far from the average, for example 100 hours instead of 5. These might have had an effect on the entirety of the learning process, yielding worse results. It would be interesting to see if, for the training set, these outliers were not used whether the average MAE for the testing set would drop significantly.

Finally, the main research question was answered within the scope of MPO. There are many more possible areas within supply chain management where improvements can be made through machine learning. This research, therefore, only answers this question for one company on the very large spectrum of supply chain management. It can serve as an example of the possibilities that companies or researchers can explore.

Appendix A

Scientific paper

Machine learning applications in supply chain management: A case study at MPO

Mike van der Meer

Committee:
Ir. M.B. Duinkerken TU Delft, supervisor
Dr. J.M. Vleugel, TU Delft, supervisor
Prof.dr. R.R. Negenborn, TU Delft, chairman
Dr M. Verwijmeren, MPO, supervisor
Ir. P. van Dongen, MPO, supervisor

Abstract— **In the world of supply chains the costs of logistics and demand for logistics have been rising over the last few years and are expected to continue to rise. It is therefore important for supply chain companies to use the newest technologies available in order to reduce these costs and meet the ever increasing demand for logistic solutions. In this paper machine learning will be used to approach aspects of this problem. This research was performed at a company called MPO. Which is a supply chain management company that maintains a platform that mainly concerns itself with order fulfillment for their clients. In this paper five elements were discussed within the field of supply chain management and scope of MPO that could be improved with machine learning, of these on-time estimation was chosen for further research. On-time estimation can reduce costs in the sense that before the execution phase of a delivery, orders that will not arrive on time can be flagged and further looked at. For this element a model was proposed and executed with two different outputs, namely, classification and regression. These tasks are tested by a neural network and a random forest. The dataset used for this research was provided by MPO and is the dataset of one of their largest clients. The platform had no system in place that performed a similar task to classification. Whereas for regression, the existing system performed better than the machine learning solution. Whilst the neural network performed significantly better than the random forest it was not as robust.**

I. Introduction

The world is becoming increasingly connected by the year, and it is shown for example through the amount of packages that are ordered and delivered. In the year 2021 for example, PostNL delivered a record number of 384 million packages in a country (Netherlands) with a population of only 17 million people. The amount of deliveries increased by 14% compared to 2020 [NOS, 2021]. The COVID pandemic, however, heavily impacted supply chains across the world, due to the disruptions in trade that were caused. Many countries enforced stricter customs or shut their borders entirely. As a result in late 2021 77% of the worlds largest ports still faced backlogs [Blake, 2021]. In the United States there is a shortage of 80.000 truckers which could more than double by 2030 [Duffy, 2021]. This is felt by retailers, since they have to pay 30% more to move goods by truck in 2020 relative to 2019 [Premack, 2020]. The cost of shipping a 40-foot container from Shanghai to Los Angeles, for example, has increased by 75% in a year and is expected to continue rising in 2022 [Smith et al., 2021]. These are merely a few examples for the increases in logistics cost. It is therefore time for supply chains to use the newest technologies available to suppress the rise in logistics cost. MPO in this story is a supply chain management company that would like to improve their services through the use of machine learning. In this paper elements within the scope of said company and supply chain management are found where machine learning might provide improvements. From these elements one is selected and a conceptual model is proposed and executed. The dataset for the machine learning algorithm was provided by MPO and contains data of one of their largest clients.

The structure of the paper is as follows: In section II the problem statement is discussed, in section III the related work of said problem is described, section IV describes the proposed model, in section VI the results of the model are discussed and finally section VIII outlines the conclusion and discussion.

II. Problem definition

Within the field of supply chain management, MPO concerns itself mostly with the process of order fulfillment. This process is defined as beginning with receiving a customer order and finishing with the final items being delivered. MPO performs this for their clients through the use of their platform. These clients, which are themselves companies, have customers who place orders for products. The customer orders that are placed are relayed to the platform. Through a set of steps, a path is determined, a carrier is selected and contacted and an estimated arrival time is provided. The clients have visibility over the whole process from the arrival of a customer order to the confirmation by the carrier that the delivery has been completed. Within the scope of this platform five possible elements were encountered, where possible improvements could be made through the use of machine learning. These elements were found through a set of interviews and analysis of the platform. These topics are:

- On-time estimation
- Order cost estimation
- Carrier performance
- Return of product
- Demand forecast

The data for order cost estimation and product returns was unsatisfactory. For carrier performance it was unclear why machine learning should be used, alternative methods would most likely achieve similar or better results. Demand forecast and on-time estimation were both strong candidates, however, the wider applicability of demand forecast is questionable. The market a client would operate in would have a very large effect on the importance of some variables. In other words, important variables for the demand forecast in one market might be useless in another. On-time estimation had none of these shortcomings and was thus selected to be the topic for further research in this study.

III. Related work

Table I contains the reviewed papers for on-time estimation. For example, [Yang et al., 2016] presents a support vector machine with genetic algorithm (GA-SVM) to predict bus arrival times. The genetic algorithm is used to find the best parameters for the input vectors that were used in this study. These input vectors were: The length of the road, the bus speed, the rate of road usage, the weather and finally the character of the time period. The data from a single bus in the Shenyang region was used in order to validate the algorithm. It was then compared to a traditional support vector machine and an artificial neural network. It outperformed both of these aforementioned algorithms in accuracy. Another model was presented by [van der Spoel et al., 2017], which predicts the arrival time of trucks originating from a single warehouse. The predictions were done by doing a classification of the expected arrival time in hours. Variables that were used for predictions were weather, accidents, congestion and time of day. For these predictions random forest, RPart, SVM, ADaboost and kNN were used of which ADaboost performed the best in terms of accuracy. However the authors argue that the predictive power of the model is limited. Furthermore, they argue that it is due to a gap in the literature where arrival time has barely been researched, whilst travel time has. With arrival time other factors, besides those important in predicting travel time, are important such as human factors like the intended arrival and departure time. Another interesting model is proposed by [Basturk and Cetek, 2021]. In this paper the arrival times of aircraft are estimated. Random forest and deep neural networks (DNN) were trained in order to perform this task. Many different variables were used, for example: Scheduled time for departure and arrival, distance, aircraft type, airline, wind speed and visibility around airport and many more. Immediately after departure, both the DNN and RF algorithms predicted a delay with a mean absolute error (MAE) of less than 6 minutes. In this field, as shown in the discussed examples, almost exclusively regression and classification are used as methods to solve this problem. Moreover, every paper exclusively focused on a single mode for the research. Furthermore, often the weather and traffic data was used and a single route and origin was assumed. In this research the mode will not be assumed and neither will weather and traffic be considered, since MPO operates at a higher scope level this information is not available. Additionally, in none of the papers are the variables actually tested for relevance. It might be possible to achieve similar results with less variables, which is very relevant for this research since the model should be applicable to multiple sectors of industry. Namely, not every company has the same data available. Therefore, it is important to find out which variables are important and which are not important to establish when on-time estimation can actually be used.

IV. Conceptual model

The conceptual model itself can be viewed in figure 1. By following the individual steps this model can be used to perform on-time estimation. An important part of this model is the feature selection which is highlighted in green, since it is something that in previous work is found to be lacking and is therefore the novelty in this research. The model starts with the order data at the top from which variables are extracted. These are variables that have something to do with the transport of a good. From these variables a target variable is chosen, which is the variable that predictions will aim to imitate. The other variables are going through a process of feature selection to determine which have predictive value with respect to the target variable and which do not. This selection process leads to a set of features that will be used to make the predictions. Chi-squared test and mutual information were chosen as the filter methods, because most of the variables are categorical. For the wrapper recursive feature elimination was chosen, because it performed well in reviewed studies that compare wrapper techniques [Wah et al., 2018] [Poona and Ismail, 2013]. Two types of predictions can be made within the machine learning category of supervised learning. Similarly, to the previous work classification and regression will be used for this model. For both these models two algorithms will be used and the same algorithms will be used in order to make relevant comparisons. These algorithms were selected on the following three criteria:

- Highest possible accuracy.
- Speed of classification is important since these classifications are expected to be performed in real-time.
- Explainability/transparency is not as important as the aforementioned but would be nice to have.

Based on these criteria and the study by [Kotsiantis et al., 2007], which ranks families of machine learning algorithms on different categories. Since explainability and accuracy in that study were (nearly) mutually exclusive an algorithm on both sides of the spectrum was chosen. Neural network was chosen for accuracy and random forest was chosen for explainability. This would also be interesting to verify for this problem and dataset.

Reference	Method	ML algorithms	Topic	Single mode	Single route/origin	Weather/traffic
[Reinhoudt and Velastin, 1997]	Regression	Kalman filter	Bus arrival time	✓	✓	
[Lin et al., 2013]	Regression	ANN	Bus arrival time	✓	✓	
[Yang et al., 2016]	Regression	GA-SVM	Bus arrival time	✓	✓	✓
[Lee et al., 2016]	Regression	LR & RF	Taxi time of aircraft	✓	✓	✓
[van der Spoel et al., 2017]	Classification	DT, KNN, SVM, ensemble classifiers & RF	Truck arrival time	✓	✓	✓
[Pang et al., 2018]	Regression	RNN	Bus arrival time	✓	✓	✓
[Barbour et al., 2018]	Regression	SVR, RFR & DNN	Freight train arrival time	✓		✓
[Wesely et al., 2021]	Regression	XGBoost & MPP	Aircraft landing time	✓		✓
[Basturk and Cetek, 2021]	Regression	RF & DNN	Aircraft ETA	✓		✓
[Park et al., 2021]	Reinforcement learning	Q-learning	Vessel ETA	✓		✓
[Hildebrandt and Ulmer, 2022]	Regression/classification	GBDT & DNN	Arrival time of restaurant meal delivery	✓	✓	✓

TABLE I

REVIEWED WORK WITH REGARDS TO ESTIMATING TIME OF ARRIVAL AND THE RESEARCH GAP.

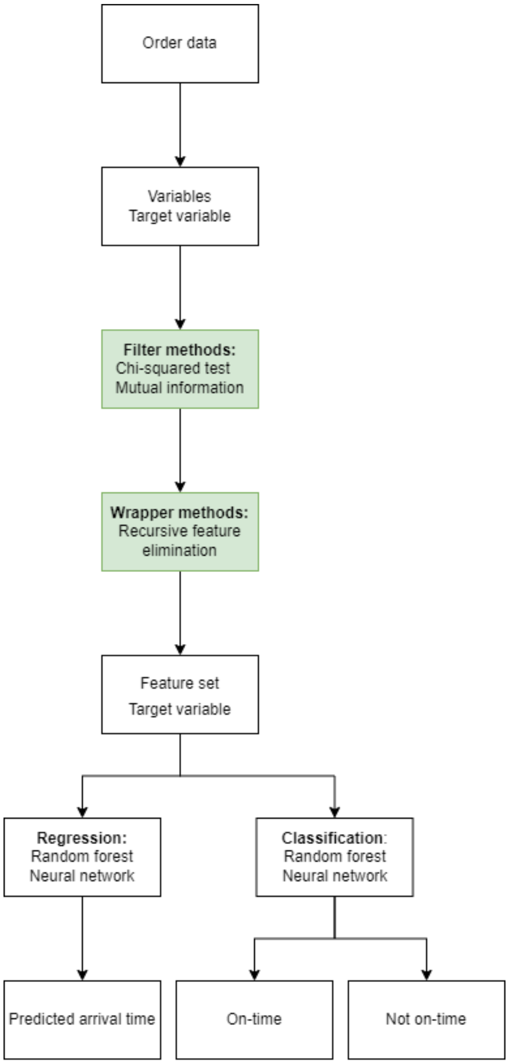


Fig. 1. The proposed model.

V. USE CASE

The dataset used in this research is by one of MPO's biggest clients, as previously mentioned. The first entries are from June 2016 and extend until July 2022, when this copy of the database was made. The dataset contains over 60 million entries. In order for the dataset to become tangible and manageable a time frame had to be selected, since the sheer size of the entire dataset is too much for this study. The start data of the selected time frame will be the first of January 2021 and the end of the time frame is until the end of the dataset. This time frame encompasses more than a year and a half of data. Noise in a dataset can be detrimental to any kind of data analysis [Xiong et al., 2006]. Therefore aside from a time frame, more filters must be applied to remove noise before the data can be properly analyzed. To perform this removal of noise the following filters were applied:

1) Entries more than two weeks late or early.
2) Entries that have no planned arrival or actual arrive time.
3) Internal deliveries.

The first filter was applied due to anomalies in the data. There were for example data entries that had a planned arrival of 2050 but an actual arrival of 2021 or 2022. A cut-off point had to be selected and after examination of the data and discussion with experts from the conducted interviews, two weeks was selected. Since if a delivery is more than two weeks early or late, for the given dataset, this is most likely an error of the system. The second filter concerns with entries that have no planned arrival or actual arrival time. Quite simply put, it is not possible to determine whether these entries were on-time or not and therefore are not applicable. These entries can for example be orders that have been cancelled or changed. The third and final filter corresponds to internal deliveries. These are deliveries from warehouses within the same warehouse or other warehouses. These deliveries are in a lot of cases incorrectly recorded, in contrast to outgoing orders, and are therefore also filtered. Even after applying all the filters the result is still a healthy data set of more than 15 million entries, as can be viewed in table II. The table further shows how many shipment orders arrive on time and how many do not. The percentages plainly reveal that the dataset is biased, with the vast majority being on time.

	Number of entries	Percentage
Not on-time	825.386	5,3%
On-time	14.777.731	94,7%
Total	15.603.117	100%

TABLE II

THE NUMBER OF SHIPPING ORDERS SEPARATED INTO THOSE THAT ARRIVE ON TIME AND THOSE THAT DO NOT.

VI. Results

A. Experiment setup

For the neural network there are five and for the random forest there are four parameters that were tested. This would be impossible using a factorial design since even for three values per parameter this would result in $3^5 = 243$ possible combinations. Therefore, the experiments will be conducted on the dataset using the one factor at a time (OFAT) approach, where one parameter is a altered whilst the others are kept constant. In this way the relation between the parameter and result can be captured, whilst limiting the number of experiments. Important to note is that with OFAT the relation between parameters is not captured [Frey et al., 2003]. For the experiments sample sizes of 40k total samples were used. For classification since the output is a binary problem and the dataset is skewed towards on time ($\geq$95%) a balanced dataset was chosen and therefore two sets of 20k samples were used. The train and test split used was 80% and 20% respectively. For the implementations the H2O framework was used. In table III and IV the default values can be seen for random forest and neural network respectively. Most of these were suggested by machine learning literature such as [Bengio, 2012] and [Oshiro et al., 2012]. The gaps were filled in by defaults provided by the H2O framework, since these corresponded closely to the suggested variables by the literature.

	Default value
Number of trees	50
Maximum depth of tree	20
Minimum observations for leaf	1
Number of parameters	7

TABLE III

Default parameters for the random forest algorithm for the experiments.

	Default value
Epochs	10
Neurons per layer	100
Amount of layers	2
Learning decay rate	0,99
Number of parameters	7

TABLE IV

Default parameters for the neural network algorithm for the experiments.

For the parameters number of trees, maximum tree depth, minimum number of observations per leaf, epochs and neurons per layer an exponential increase of 2^n was chosen, where n takes the values 1 to 8. Therefore, these parameters take the values 1, 2, 4, 8, 16, 32, 64, 128 and 256 along with their default value. For the number of parameters the same logic was used as with the wrapper in the previous section. Thereby, this parameter takes a value of 1 through to 11. Finally, the amount of layers will take a value from 1 through 10 and the learning decay rate will take a set of values between 0 and 1.

B. Classification

Table V contains the average accuracy based on ten experiments along with the minimum, maximum and standard deviation. For each of these experiments 20k positive and negative samples were randomly selected. In order to determine statistical significance a confidence interval of 95% is used, which was calculated based upon the standard deviation and number of experiments using a t-test statistic. In the table the upper and lower bound of this 95% confidence interval are shown. From this confidence interval it can be concluded that the average result of the random forest result is not significantly different from that of the neural network. Furthermore, the results of the experiments with parameters would only be significant if they fall outside of the confidence interval bounds. There is no current existing system within the platform that performs this task, the orders are assumed to always arrive on-time. Since it is a binary classification problem the current system therefore has an accuracy of 50%. Both algorithms, therefore, perform significantly better.

	Average accuracy	**Minimum**	**Maximum**	**Standard dev.**	**Lower CI**	**Upper CI**
Random forest - baseline	72,1%	71,2%	73,1%	0,5	71,7%	72,5%
Neural network - baseline	72,2%	71,2%	73,1%	0,7	71,7%	72,6%

TABLE V

Accuracy values for ten experiments of the baseline evaluation for the random forest and neural network algorithms.

For random forest only one parameter improved the results and that was the number of features which was slightly better with ten features, however not significantly so. For the neural network also only one feature improved the results and once again not significantly so. This parameter was the number of epochs and a better accuracy was achieved for 16 epochs. This indicates that the baseline values chosen already had relatively good performance. Especially when considering that a large number of the parameter values had a significant negative relation with the result. In other words, they decreased the accuracy. Table VI combines the previous sensitivity analysis with the adjusted parameter variants. Using the previously discussed confidence interval the average accuracy for the adjusted variant of random forest is on the upper bound. Whilst, the adjusted variant for neural network is not significant and even decreased the average accuracy. Therefore, it can be concluded that, whilst taking the limitations of the OFAT approach into account, the baseline results did not significantly improve as a result of the experiments.

	Average accuracy	Minimum	Maximum	Standard dev.
Random forest - baseline	72,1%	71,2%	73,1%	0.5%
Random forest - adjusted	72,5%	71,9%	73,2%	0.4%
Neural network - baseline	72,2%	71,2%	73,1%	0.7%
Neural network - adjusted	71,8%	70,7%	72,8%	0.6%

TABLE VI

SENSITIVITY ANALYSIS FOR THE BASELINES AND ADJUSTED BASELINES FOR RANDOM FOREST AND NEURAL NETWORKS.

C. Regression

Table VII contains the average mean average error (MAE) in hours based on ten experiments along with the minimum, maximum and standard deviation. For each of these experiments 40k samples were randomly selected. In order to determine statistical significance a confidence interval of 95% is used, which was calculated based upon the standard deviation and number of experiments using a t-test statistic. In the table the upper and lower bound of this 95% confidence interval are shown. From this confidence interval can be seen that the average MAE of the random forest falls within the confidence interval of the neural network, but, interestingly enough, not the other way around. The standard deviation for random forest, however, is about half as much as neural network. Indicating that it is slightly more robust in its results although they are slightly worse, but not significantly so. The MAE of the current planning system for these samples is 2,49. Both algorithms, therefore, do not improve the results of the current system.

	Average MAE	Minimum	Maximum	Standard dev.	Lower CI	Upper CI
Random forest - baseline	4,83	4,66	4,99	0,1	4,76	4,90
Neural network - baseline	4,69	4,32	5,02	0,22	4,53	4,84

TABLE VII

MEAN AVERAGE ERROR (MAE) IN HOURS FOR TEN EXPERIMENTS OF THE BASELINE EVALUATION FOR THE RANDOM FOREST AND NEURAL NETWORK ALGORITHMS.

For random forest changing all parameters except the number of features improved the results, but none of them significantly so. The number of trees was set at 256, the depth at 16 and the observed minimum number of nodes to 2. Similarly, for the neural network changing all parameters except the number of features improves results. Although only the number of neurons per layer provided significant improvements. The number of epochs was changed to 64, the nodes per layer to 256, the number of layers to 7 and the learning decay rate to 0,999. Even though almost all parameters were changed only one of these had a significant improvement, once again indicating that the baseline values chosen already had relatively good performance. Table VIII combines the previous sensitivity analysis with the adjusted parameter variants. For both the adjusted variants the MAE decreased. Whilst for random forest only very slightly and not significantly so. For the neural network quite a bit more and significantly so since the lower bound was 4,53 and the average MAE for the adjusted variant is 4,40. Interestingly enough the standard deviation for neural network increased by quite a bit, thus reducing robustness, whilst for random forest it remained the same.

	Average MAE	Minimum	Maximum	Standard dev.
Random forest - baseline	4,83	4,66	4,99	0,1
Random forest - adjusted	4,79	4,56	4,95	0,1
Neural network - baseline	4,69	4,32	5,02	0,22
Neural network - adjusted	4,40	4,03	4,97	0,29

TABLE VIII

MEAN AVERAGE ERROR (MAE) IN HOURS FOR THE BASELINE EVALUATION AND THE ADJUSTED VARIANT FOR RANDOM FOREST AND NEURAL NETWORK.

D. Scale experiments

In table IX the accuracy is shown for the different variants and scale sizes of the experiments. Using the confidence interval determined earlier, there are no significant differences. The variants all behave similarly and will therefore be discussed as such. An increase is seen between the 40k and 100k sample sizes across all the variants and from there a slight decrease for the 200k and 500k sample sizes. The highest accuracy is achieved by all variants on 100k. This indicates that the correlation between the features and the target variable is the strongest for this sample size.

	40k	100k	200k	500k
Random forest - default	72,9%	73,4%	72,9%	72,9%
Random forest - adjusted	73,0%	73,3%	72,9%	72,8%
Neural network - default	72,8%	73,1%	72,6%	72,7%
Neural network - adjusted	72,6%	73,2%	72,8%	72,8%

TABLE IX

SHOWING THE ACCURACY FOR THE SCALE EXPERIMENT. THE NUMBER ON THE COLUMNS SHOW THE SIZE OF THE TRAIN/TEST SET WHICH WAS SPLIT 80%/20% RESPECTIVELY.

In table X the MAE is shown for the different variants and scale sizes of the experiments. Using the confidence interval determined earlier, significant differences can be seen in the table. For the default random forest there is a significant improvement between 40k and 500k sample sizes, but a steady decrease in the error can be seen for the increases in sample size. The adjusted random forest on the other hand is barely affected by changes in sample size and no significant change is measurable. This could be due to the parameters being selected for the 40k sample size and relatively underperforming on the other sample size. The default neural network is does not appear to be as robust or consistent as the random forest, as indicated previously as well. The adjusted variant of the neural network performs very strangely with an error of almost 14 for the 100k sample size. The error slowly seems to stabilize for the 200k and 500k error. Most likely the parameters chosen perform well on the 40k sample size but are very bad for the other sample sizes. For example the learning rate for the neural network is increased for the adjusted variant which might work well on a small sample size, but as the sample size increases a high learning rate might not be good.

	40k	**100k**	**200k**	**500k**
Random forest - default	4,86	4,73	4,67	4,66
Random forest - adjusted	4,78	4,73	4,72	4,74
Neural network - default	4,51	4,93	4,53	5,06
Neural network - adjusted	4,92	13,93	6,18	5,28

TABLE X

Showing the MAE in hours for the scale experiment. The number on the columns show the size of the train/test set which was split 80%/20% respectively.

E. Dataset comparison

Table XI shows the accuracy of the classification model of the original dataset that was used for feature selection and creating the variants. Also shown is the dataset that is used for comparison and validation, which is named alternative dataset. There is a slight increase in accuracy measurable of about 2-3% across all variants. For the existing system 50% accuracy is chosen, because there is no existing system and is therefore always assumed to be on-time. In a binary classification problem with a balanced dataset, which is used for these experiments, it would be correct 50% of the time. Whilst the original dataset already improved upon the existing system, the alternative dataset performs significantly better across all variants.

	Existing system	**RF - default**	**RF - adjusted**	**NN - default**	**NN - adjusted**
Original dataset	50%	72,9%	72,8%	73,0%	72,6%
Alternative dataset	50%	75,9%	75,6%	74,8%	75,2%

TABLE XI

Showing the accuracy for the original and other dataset.

Table VII shows the MAE of the regression model of the original dataset that was used for feature selection and creating the variants. Also shown is the dataset that is used for comparison and validation, which is named alternative dataset. Unlike classification there was an existing system that performed a task similar to the regression. The error of the original dataset is about halve that of the variants. Although the error for the variants is a lot higher for the alternative, so is the existing system error. The error ratio between existing system and variants almost stays the same at around twice as much between both datasets. Although, the existing system is not improved upon the results are consistent across different datasets, which at least validates the model in that sense.

	Existing system	**RF - default**	**RF - adjusted**	**NN - default**	**NN - adjusted**
Original dataset	2,57	4,86	4,78	4,92	4,51
Alternative dataset	10,69	20,78	21,48	20,58	19,78

TABLE XII

Showing the MAE in hours for the original and other dataset.

VII. Results interpretation

The classification results for both algorithms was in the 72-73% range. The usability of these algorithms would be slightly hampered by this accuracy. The around 30% wrong predictions can be split into two categories. Either false positives or false negatives. A false positive is what the system currently in place does, since there is no feedback loop concerning the projected arrival time. This would therefore not be problematic. However, a false negative would indicate a problem that does not exist. This would require the attention of an employee to figure out whether action is actually needed, since an employee would not be able to tell a true negative from a false negative without looking into the problem. Time spent of employees obviously would inquire costs which these models are supposed to reduce. The trade-off would therefore need to be determined whether a 72-73% accuracy would actually save costs in delayed orders relative to time spent by employees checking the output of the algorithms. The confusion matrix for the baseline evaluation indicated that about 10% of all predictions are false positives. This would require one in ten orders to be looked at, because the prediction made is wrong. Dependent on the costs associated with a delayed shipment or the importance of a shipment arriving on time this still might be a good trade-off, but that would be market or company specific. This would most likely not be a good trade-off in every market or for every company. However, an accuracy of a binary classification problem that achieves 72-73% accuracy on tens of thousands of samples clearly means that some correlation has been found, since the performance of random predictions would most likely be closer to 50%. So it can be concluded that a correlation was found by both machine learning algorithms. The results were also validated on another dataset and performed significantly better, indicating that that dataset might be a better fit. However, both results already improved the accuracy of the existing system. Changes in sample size were also experimented with, these however had no significant impact on the results.

For regression the results were expressed in MAE in hours. Therefore, the results can be interpreted as being on average wrong by about four and a half to five hours. Whilst this does not speak to the imagination much it can also be expressed differently through time-frames. Where 50% of the predictions can fall within a 3.5 hour time-frame and 70% of predictions fall within a 6 hour time-frame. This could be very useful in practice when for example a time-frame is used and the determined deadline for an order falls within this time-frame. Moreover, the results were validated on another dataset and performed similarly relative to the error of the existing system. Namely, the error of the predictions was about twice as high as that of the existing system.

VIII. Conclusion & discussion

In this paper a machine learning model for on-time estimation of deliveries was proposed and executed. The output for this model was in the form of a binary classification that was on-time or not on-time and regression which was measured in mean average error (MAE). Neural network and random forest were the machine learning algorithms used to perform this task. The contribution of this research is that it can be concluded that it is possible to perform on-time estimation with no data about the mode used, the traffic, the route or weather. However, this data might have be helpful in improving results, but since this data was not available for this research this could not be tested. Moreover, the created model can be used as a basis to perform on-time estimation by others. In addition, the generic model can also be used as a basis to perform other machine learning problems or the discussed elements in logistics. However, for that the individual blocks of the model will most likely have to be adjusted to accommodate the different problem. For example, classification and/or regression might not be the best way to solve that problem.

The experiments performed in this research were on small samples relative to the size of the entire dataset due to time and resource constraints. Even in the largest sample size used, which was 500k, only a few percent of the entire available data was used. It would be interesting to see if results stay the same (which the scale experiment indicates that it will do) or if the robustness changes with far larger samples. A sensitivity experiment on a large portion or the entire dataset would be interesting. Furthermore, an experiment could be done to incorporate data such as route, mode, traffic or weather and to determine what the impact would be on the results.

References

[Barbour et al., 2018] Barbour, W., Samal, C., Kuppa, S., Dubey, A., and Work, D. B. (2018). On the data-driven prediction of arrival times for freight trains on us railroads. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2289–2296. IEEE.

[Basturk and Cetek, 2021] Basturk, O. and Cetek, C. (2021). Prediction of aircraft estimated time of arrival using machine learning methods. *The Aeronautical Journal*, 125(1289):1245–1259.

[Bengio, 2012] Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer.

[Blake, 2021] Blake, M. (2021). Moody's warns of 'dark clouds ahead' for the global supply chain as 77% of the world's largest ports face backlogs. Business Insider.

[Duffy, 2021] Duffy, K. (2021). Pay for truckers is soaring — one said his salary shot up to $70,000 from $40,000. but it's not enough to fill thousands of driver vacancies. Business Insider.

[Frey et al., 2003] Frey, D. D., Engelhardt, F., and Greitzer, E. M. (2003). A role for" one-factor-at-a-time" experimentation in parameter design. *Research in Engineering design*, 14(2):65–74.

[Hildebrandt and Ulmer, 2022] Hildebrandt, F. D. and Ulmer, M. W. (2022). Supervised learning for arrival time estimations in restaurant meal delivery. *Transportation Science*, 56(4):1058–1084.

[Kotsiantis et al., 2007] Kotsiantis, S. B., Zaharakis, I., Pintelas, P., et al. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24.

[Lee et al., 2016] Lee, H., Malik, W., and Jung, Y. C. (2016). Taxi-out time prediction for departures at charlotte airport using machine learning techniques. In *16th AIAA Aviation Technology, Integration, and Operations Conference*, page 3910.

[Lin et al., 2013] Lin, Y., Yang, X., Zou, N., and Jia, L. (2013). Real-time bus arrival time prediction: case study for jinan, china. *Journal of Transportation Engineering*, 139(11):1133–1140.

[NOS, 2021] NOS (2021). Postnl bezorgde recordaantal pakketten in 2021.

[Oshiro et al., 2012] Oshiro, T. M., Perez, P. S., and Baranauskas, J. A. (2012). How many trees in a random forest? In *International workshop on machine learning and data mining in pattern recognition*, pages 154–168. Springer.

[Pang et al., 2018] Pang, J., Huang, J., Du, Y., Yu, H., Huang, Q., and Yin, B. (2018). Learning to predict bus arrival time from heterogeneous measurements via recurrent neural network. *IEEE Transactions on Intelligent Transportation Systems*, 20(9):3283–3293.

[Park et al., 2021] Park, K., Sim, S., and Bae, H. (2021). Vessel estimated time of arrival prediction system based on a path-finding algorithm. *Maritime Transport Research*, 2:100012.

[Poona and Ismail, 2013] Poona, N. K. and Ismail, R. (2013). Reducing hyperspectral data dimensionality using random forest based wrappers. In *2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS*, pages 1470–1473. IEEE.

[Premack, 2020] Premack, R. (2020). Retailers are scrambling to find trucks amid the pandemic, generating record-smashing pay for drivers. Business Insider.

[Reinhoudt and Velastin, 1997] Reinhoudt, E. M. and Velastin, S. (1997). A dynamic predicting algorithm for estimating bus arrival time. *IFAC Proceedings Volumes*, 30(8):1225–1228.

[Smith et al., 2021] Smith, J., Berger, P., and O'Neal, L. (2021). Shipping and logistics costs are expected to keep rising in 2022. The Wall Street Journal.

[van der Spoel et al., 2017] van der Spoel, S., Amrit, C., and van Hillegersberg, J. (2017). Predictive analytics for truck arrival time estimation: a field study at a european distribution centre. *International journal of production research*, 55(17):5062–5078.

[Wah et al., 2018] Wah, Y. B., Ibrahim, N., Hamid, H. A., Abdul-Rahman, S., and Fong, S. (2018). Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy. *Pertanika Journal of Science & Technology*, 26(1).

[Wesely et al., 2021] Wesely, D., Churchill, A., Slough, J., and Coupe, W. J. (2021). A machine learning approach to predict aircraft landing times using mediated predictions from existing systems. In *AIAA AVIATION 2021 FORUM*, page 2402.

[Xiong et al., 2006] Xiong, H., Pandey, G., Steinbach, M., and Kumar, V. (2006). Enhancing data analysis with noise removal. *IEEE Transactions on Knowledge and Data Engineering*, 18(3):304–319.

[Yang et al., 2016] Yang, M., Chen, C., Wang, L., Yan, X., and Zhou, L. (2016). Bus arrival time prediction using support vector machine with genetic algorithm. *Neural Network World*, 26(3):205.

Appendix B

Expert interviews

There are six experts which were contacted for this research. In chapter 4, the findings are discussed. In table B.1 the list of the contacted experts can be found.

Expert 1	Tantely Raminososa	Lead business analyst
Expert 2	Paul van Dongen	Chief technology officer
Expert 3	Lekshmy Shaji	Director solutions
Expert 4	Joost Fieggen	VP center of excellence
Expert 5	Martin Verwijmeren	Chief executive officer
Expert 6	Pascal Dürr	VP product design
Expert 7	Fred Kaan	Solutions consultant
Expert 8	Sander Vervoordeldonk	Solutions architect

Table B.1: List of the interviewed experts

Appendix C

Data

C.1 Data

The dataset, which begins in 13th of June 2016 and extends until 21st of July 2022, is quite extensive. It contains over 60 million shipment orders, as shown in table C.1. The number of service orders is over 150 million. As was explained in chapter 4, there can be multiple service orders for a single shipment order since they can also be other things than a delivery. Which explains why there are two and half times more service orders than shipment orders. The final entry in the table shows the amount of service orders that correspond to delivery. The number of shipment orders and service orders that correspond to deliveries clearly differs. Shipment orders can also be consolidated into a single delivery, which would explain this disparity. This occurs when shipment orders are delivered to the same or nearby locations.

	Number of entries
Shipment orders	64.270.115
Service orders	153.684.263
Service orders - deliveries	63.097.446

Table C.1: Number of entries for different order categories.

In order for the dataset to become tangible and manageable a time frame had to be selected, since the sheer size of the entire dataset is too much for this study. The start data of the selected time frame will be the first of January 2021 and the end of the time frame is until the end of the dataset which, as previously mentioned, is late July in 2022. This time frame encompasses more than a year and a half of data. Noise in a dataset can be detrimental to any kind of data analysis (Xiong, Pandey, Steinbach, & Kumar, 2006). Therefore aside from a time frame, more filters must be applied to remove noise before the data can be properly analyzed. To perform this removal of noise the following filters were applied:

1. Entries more than two weeks late or early.
2. Entries that have no planned arrival or actual arrive time.
3. Internal deliveries.

The first filter was applied due to anomalies in the data. There were for example data entries that had a planned arrival of 2050 but an actual arrival of 2021 or 2022. A cut-off point had to be selected and after examination of the data and discussion with expert 4 and 8, two weeks was selected. If a delivery is more than two weeks early or late it is most likely an error of the system. The second filter concerns with entries that have no planned arrival or actual arrival time. Quite simply put, it is not possible to determine whether these entries were on-time or not and therefore are not applicable. These entries can for example orders that have been cancelled or changed. The third and final filter corresponds to internal deliveries. These are deliveries from warehouses within the same warehouse or other warehouses. These deliveries have an actual arrival date that corresponds to the system entry date, which is when the order was filed. The planned arrival date is usually a day after the system entry date and therefore these orders are always recorded as being on-time, due to being incorrectly recorded. Which is why these entries are also filtered. From now on, these filters will be applied to every component in this research. Even after applying all the filters the result is still a healthy data set of more than 15 million entries, as can be viewed in table C.2. The table further shows how many shipment orders arrive on time and how many do not. The percentages plainly reveal that the dataset is biased, with the vast majority being on time.

	Number of entries
Shipment orders	15.603.117
Service orders	64.641.083
Service orders - deliveries	15.603.117

Table C.2: Number of entries for different order categories after filters.

Figure C.1 shows the amount of orders per country. Deliveries are made to countries all over the world, for readability this chart only contains the countries that order the most. The Netherlands, Germany and France have the most amount of orders with a share of 27.7%, 24.7% and 18.6% respectively. Towards the end of the figure others can be seen which consists of all the countries not mentioned in the table with a share of 4.1%. Interesting to note is that all countries on this list, except the United Kingdom, are in the European Union. Deliveries are also made outside of Geographic Europe. Namely, overseas regions of France such as Guadaloupe and Martinique have 755 and 475 orders. There are also orders countries outside the European Union and geographic Europe such as New Zealand and China. Although the orders for those countries are on an entirely different scale as the ones depicted in the figure with 158 and 24 orders respectively.

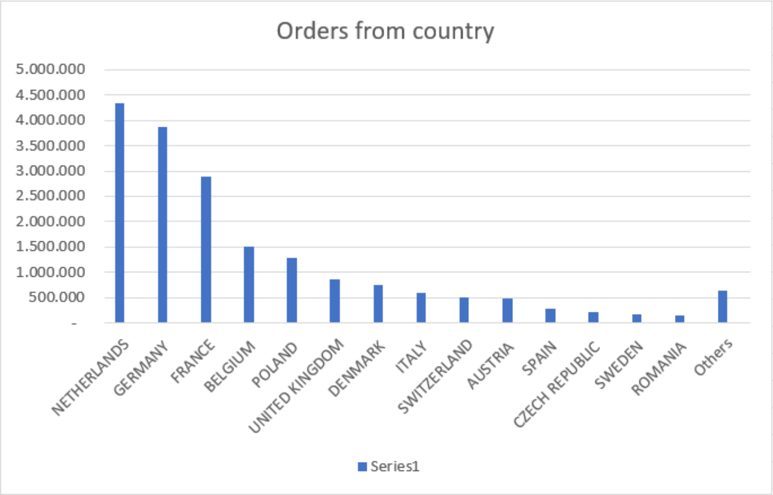


Figure C.1: Shipment orders per country.

Figure C.2 shows the amount of shipment orders from each location. These are ordered based on the amount of shipment orders with a Dutch, German and French location shipping the most orders, as can be seen by their country codes.

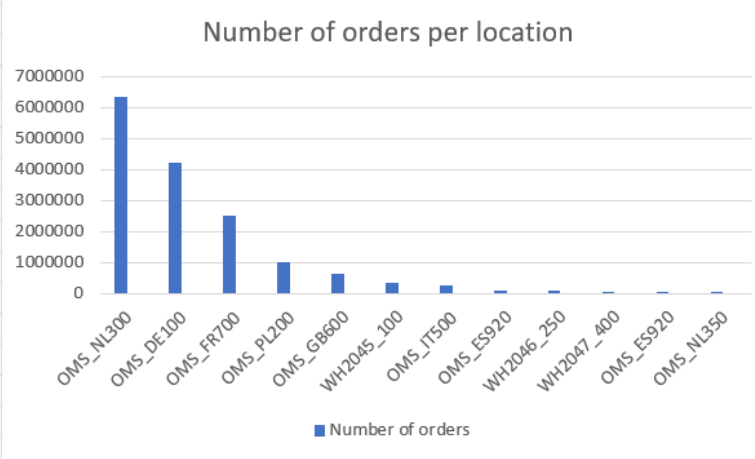


Figure C.2: Shipment orders per location.

Figure C.3 shows the amount of shipment orders per hour of the day. As can be seen in the figure the peak amount of deliveries occurs between 5 and 7 in the morning. For these early deliveries specialized night carriers are used that perform deliveries made before 6 or 8 in the morning.

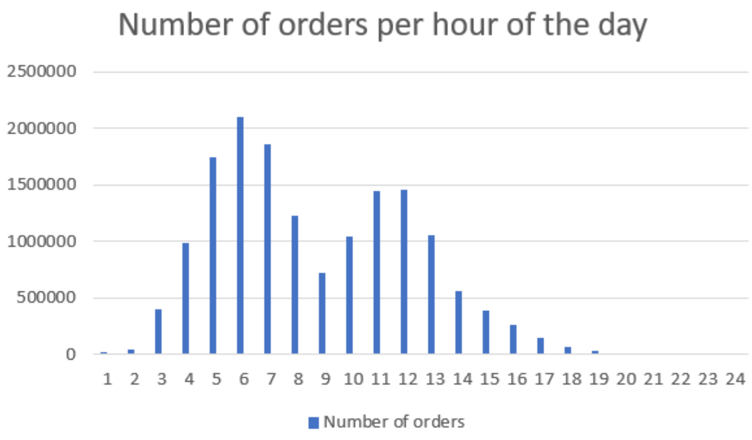


Figure C.3: Shipment order arrival time per hour.

C.1.1 Target variable

The target variable is the variable that the machine learning algorithm will aim to predict. In the case of classification it is whether an order will be on time or not. This variable is expressed in days, since same day delivery is viewed as on-time. It is calculated by deducting the arrival time from the planned arrival time. For example, an order is planned to arrive on the fifth day of the year and it arrives on the sixth day of the year it means that $6-5=1$ it is late by one day. Conversely, if the actual arrive would be on the fourth day of the year it would $4-5=-1$, which is negative. The difference in days for the shipment are shown in table C.3. This dataset contains a minimum of -14 and a maximum of 14, as per filter one. The total amount of shipment orders that are either on time or not is shown in table C.4. Here it can clearly be seen that the dataset is very imbalanced with only 5% being not on-time. The paper by Wei and Dunbrack Jr (2013) states that the best results for multiple different performance metrics (which are discussed in the next section, section 5.4) are achieved when the dataset is balanced. How a balanced dataset will be achieved is discussed in the next chapter, which is chapter6.

Difference in days	-5	-4	-3	-2	-1	0	1	2	3	4	5
Number of entries	416	7.108	75.895	124.051	249.933	14.319.246	576.673	65.648	106.318	41.068	15.449
Percentage	0,00%	0,05%	0,49%	0,80%	1,60%	91,77%	3,70%	0,42%	0,68%	0,26%	0,10%

Table C.3: Shipping orders separated into the day they arrive at relative to the planned arrival time.

	Number of entries	**Percentage**
Not on-time	825.386	5,3%
On-time	14.777.731	94,7%
Total shipment orders	15.603.117	100%

Table C.4: The number of shipping orders separated into those that arrive on time and those that do not.

For regression a the target variable will be calculated in the same manner but it will be expressed in hours instead of days. Moreover, it will not be split up in two sets of on-time or not on-time. Therefore, just like with classification the difference between the planned and actual arrival will be taken. However, this time it will be expressed in hours. Figure C.4 shows the distribution of this variable. From this figure can be seen that most of the deliveries are before the planned arrival time. Moreover, they occur in about a 12 hour window of the planned arrival time. This variable thus behaves very similarly to the previously encountered variable. The average is -2,57 with a standard deviation of 14,67.

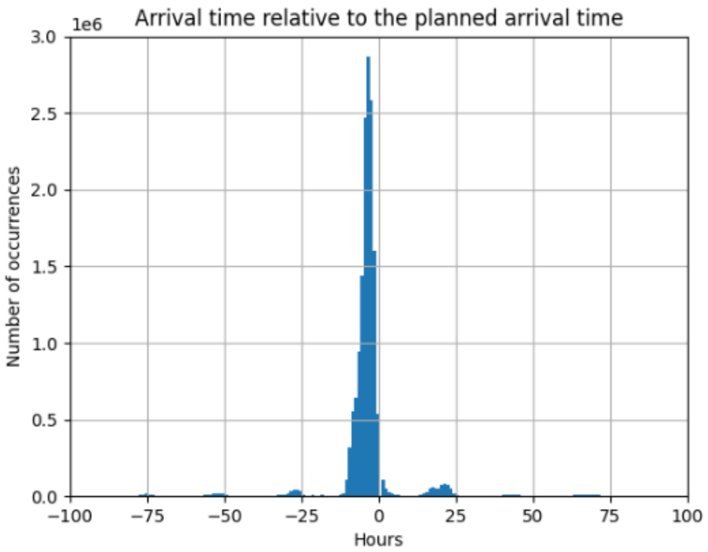


Figure C.4: The distribution of actual arrival time relative to the planned arrival time. The number 0 on the x-axis represents the planned arrival time.

C.2 Feature selection

In order to find features that are able to correctly predict on-time as well as not on-time entries, a balanced dataset was opted for. For both classifications 500.000 samples were randomly selected from the existing pool. For the filter methods only the categorical variables will be tested, which are 20 in total. Afterwards the filtered categorical variables together with the numerical variables will be tested through the use of a wrapper method. Besides that, the empty or largely incomplete variables were not taken into consideration. The total list of variables that were considered can be found in table 6.1 in appendix D. These variables were selected based on the expert interviews, literature and analysis of the system.

C.2.1 Chi-square test

After some initial experimentation with the Chi square test the results were very skewed towards a single variable, as can be seen in figure C.5. The variable to_location_id is a very spread out variable with many different categories, since it is unique to every location used in the database. Many of these categories are used only a few times. Namely out of the one million records there 86.778 unique ones. McHugh (2013) notes that the value of a specific category should be at least 5 in more than 80% of the cells. This is not the case for this variable and some others as well.

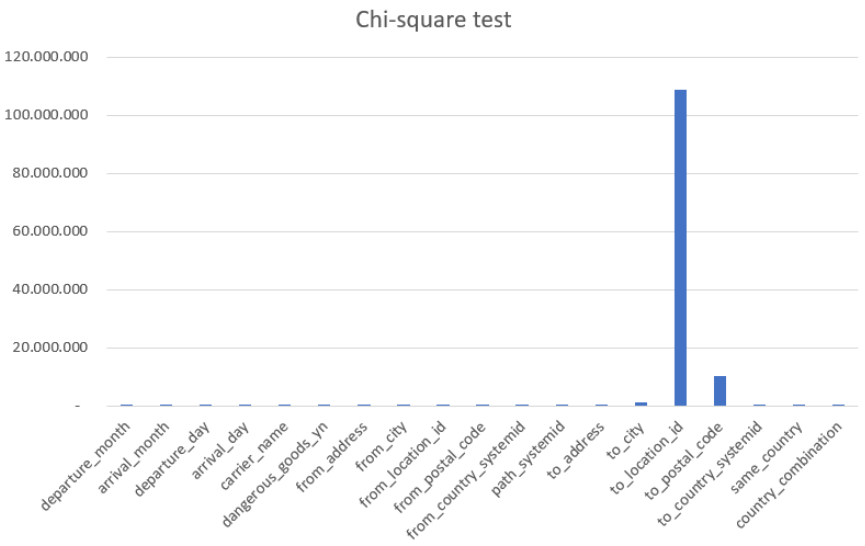


Figure C.5: Chart containing the chi-square test values for the variables.

Since these variables drowned out the relevance of other variables these will no longer be considered valid. Table C.5 contains the unique counts per variable. It can clearly be seen that there is a large difference between the variables. Some have only a few dozen unique categorizations whilst others have in the tens of thousands. Because these variables are so distributed their usefulness is most likely limited. Therefore the variables to_address, to_city, to_location_id and to_postal_code will no longer be used.

Name	Count
departure_month	12
arrival_month	12
departure_day	7
arrival_day	6
carrier_name	39
dangerous_goods_yn	2
from_address	15
from_city	11
from_location_id	11
from_postal_code	15
from_country_systemid	10
path_systemid	188
to_address	63.346
to_city	32.391
to_location_id	86.143
to_postal_code	33.620
to_country_systemid	33
same_country	2
country_combination	87

Table C.5: Variables with their respective number of unique categorizations

The resulting values for the chi-square test to the variables can be found in table C.6. These results are illustrated in figure C.6. The variable that has the most dependency with the target variable according to the chi-squared test is the path_systemid along with country_combination. For the from variables only from_address and from_city give a lot of information relative to the others. However, these are strongly correlated since there are only fifteen locations in eleven cities. They all contain similar information and therefore a combination of these or a single one might capture enough detail. Furthermore, to_country_systemid and arrival_day also have some dependency. According to the chi-square test all the other variables are relatively independent of the target variable.

Name	Score
departure_month	372
arrival_month	196
departure_day	1.077
arrival_day	1.870
carrier_name	157
dangerous_goods_yn	619
from_address	44.239
from_city	27.534
from_location_id	3.446
from_postal_code	5.007
from_country_systemid	15.738
path_systemid	107.681
to_country_systemid	4.185
same_country	953
country_combination	65.171

Table C.6: Variables with their respective chi-square test values.

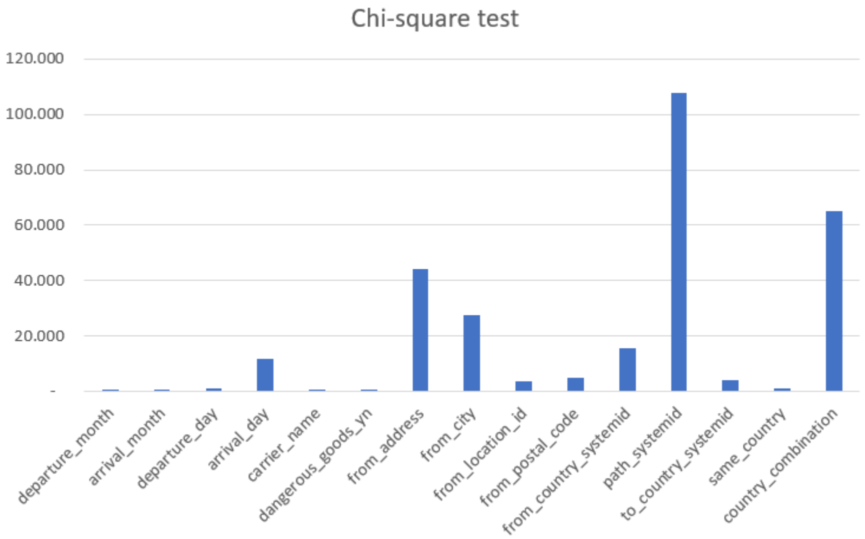


Figure C.6: Chi-square test values for the variables.

C.2.2 Mutual information test

In figure C.7 the results are shown for the mutual information test. The exact values are shown in table C.7. The best performers according to the mutual information test are path_system_id, carrier_name and country_combination. The different from variables all perform similarly and are around the average score. On the other hand, dangerous_goods, departure_month and arrival_month perform relatively poorly.

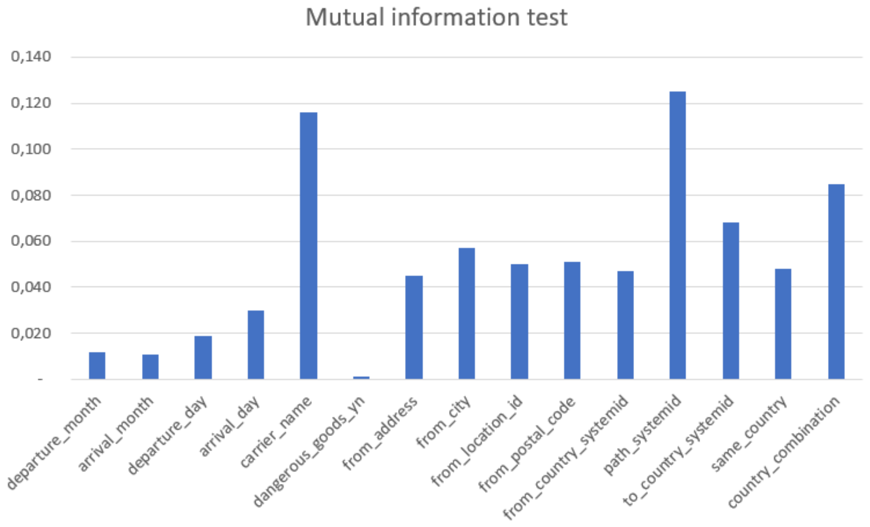


Figure C.7: Mutual information test values for the variables.

Name	Score
departure_month	0,012
arrival_month	0,011
departure_day	0,019
arrival_day	0,030
carrier_name	0,116
dangerous_goods_yn	0,001
from_address	0,045
from_city	0,057
from_location_id	0,050
from_postal_code	0,051
from_country_systemid	0,047
path_systemid	0,125
to_country_systemid	0,068
same_country	0,048
country_combination	0,085

Table C.7: Variables with their respective mutual information test values.

C.2.3 Filtering conclusion

By combining the knowledge obtained from the mutual information and chi-squared tests, a selection of variables can be made that will be used for the recursive feature elimination (RFE) in the next subsection. The variables that performed poorly in both tests were departure_month, arrival_month and dangerous_goods. These will therefore not be used. Furthermore, the from variables all convey similar information since there are fifteen locations in eleven cities not all of them will most likely be necessary. In the mutual information test these variables performed similarly but in the chi square test from_city and from_address outperformed the others. Therefore these will be selected in favour of the others. This leaves nine categorical variables from the initial twenty.

C.2.4 Recursive feature elimination

As concluded in the previous section nine categorical variables will be considered along with two numerical. In table C.8 the variables that will be tested are listed. Due to the complexity of recursive feature elimination algorithm a small subset of the data will have to be used in order to keep the computation times feasible. Therefore a balanced dataset containing a 10.000 on-time as well as on on-time samples was opted for. The evaluation algorithm used for the wrapper is a decision tree, as discussed previously in section 5.3.

Name	Type
departure_day	**Categorical**
arrival_day	**Categorical**
carrier_name	**Categorical**
from_address	**Categorical**
from_city	**Categorical**
path_systemid	**Categorical**
to_country_systemid	**Categorical**
same_country	**Categorical**
country_combination	**Categorical**
total_volume_m3	**Numerical**
total_weight_kg	**Numerical**

Table C.8: Variables considered for recursive feature elimination along with their category.

The results of random feature elimination algorithm can be found in table C.9. The table starts with the top row showing the accuracy of the entire set of considered variables which is 62.89%. Afterwards in every iteration one variable is removed in every iteration according to its ranking. The first variable to be removed is same_country and the last one total_weight_kg. The last variable to be removed is the variable that according to the RFE algorithm has the most impact on the performance once removed and is therefore the most important feature. The difference in accuracy is shown in the rightmost column. When looking at this column from top to bottom (as per the order of removal by the algorithm), it can clearly be seen that the reduction in accuracy is relatively low until the removal of departure_day, which is the seventh ranked variable. When removing this variable the accuracy drops by 1.65%. For most variables that are removed afterwards the accuracy also lowers by 1% or more, with the exception of carrier_name. The variable departure_day, therefore, seems like a cut-off point where the drop in accuracy per variable left behind increases by a lot. As discussed in 5.3 accuracy is of the utmost importance, a loss of more than 1% for the removal of a single feature is not a worthwhile exchange. Therefore the set when from_city is removed will be the features used for the rest of this research. Interesting to note is that by removing same_country and carrier_name the accuracy slightly increased. Since same_country was the first variable to be removed it is most likely that although slight statistical correlations were found during filtering this was not actually information that could be used for predictions. On the other hand, carrier_name is a highly ranked variable although removing it resulted in an increase in accuracy. In this case it is probable that the carrier_name provided information together with another variable. The removal of that other variable resulted in carrier_name also becoming an unuseful variable.

Set	Removed variable	Ranking	Accuracy	Difference
{same_country, to_country_systemid, from_address, from_city, departure_day, total_volume_m3, country_combination, carrier_name, arrival_day, path_systemid, total_weight_kg}	-	-	62.89%	-
{to_country_systemid, from_address, from_city, departure_day, total_volume_m3, country_combination, carrier_name, arrival_day, path_systemid, total_weight_kg}	same_country	11	63.01%	0.12%
{from_address, from_city, departure_day, total_volume_m3, country_combination, carrier_name, arrival_day, path_systemid, total_weight_kg}	to_country_systemid	10	62.94%	-0.07%
{from_city, departure_day, total_volume_m3, country_combination, carrier_name, arrival_day, path_systemid, total_weight_kg}	from_address	9	62.78%	-0.16%
{departure_day, total_volume_m3, country_combination, carrier_name, arrival_day, path_systemid, total_weight_kg}	from_city	8	62.76%	-0.02%
{total_volume_m3, country_combination, carrier_name, arrival_day, path_systemid, total_weight_kg}	departure_day	7	61.11%	-1.65%
{country_combination, carrier_name, arrival_day, path_systemid, total_weight_kg}	total_volume_m3	6	60.06%	-1.05%
{carrier_name, arrival_day, path_systemid, total_weight_kg}	country_combination	5	59.35%	-0.71%
{arrival_day, path_systemid, total_weight_kg}	carrier_name	4	59.45%	0.10%
{path_systemid, total_weight_kg}	arrival_day	3	54.17%	-5.28%
{total_weight_kg}	path_systemid	2	47.43%	-6.74%
{}	total_weight_kg	1	-	-

Table C.9: Rankings of the random feature elimination algorithm with the accompanied accuracy.

Appendix D

Figures

D.1 SCM grid

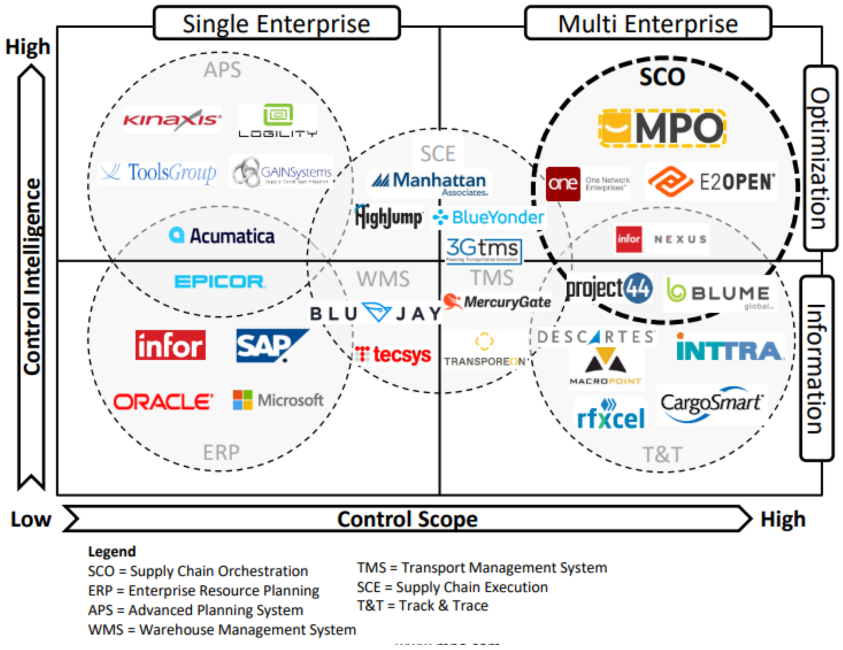


Figure D.1: Grid that shows different elements related to SCM and the companies that operate in that space.

D.2 Confusion matrices of classification results

Actual versus predicted	Not on-time	On-time
Not on-time	True negative: 6221	False negative: 2058
On-time	False positive: 2364	True positive: 5249

Table D.1: Average confusion matrix of the baseline evaluation for random forest.

Actual versus predicted	Not on-time	On-time
Not on-time	True negative: 6581	False negative: 1630
On-time	False positive: 2824	True positive: 4869

Table D.2: Average confusion matrix of the baseline evaluation for neural network.

D.3 Percentiles of regression results

	10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Random forest	0,31	0,62	0,94	1,30	1,73	2,22	2,93	4,40	9,92
Neural network	0,06	0,61	0,94	1,28	1,68	2,18	2,87	4,15	9,05

Table D.3: Showing the number of hours for a time-frame to capture a certain percentile. These values need to be doubled, since the predictions are both positive and negative absolute values were used.

References

Aamer, A., Eka Yani, L., & Alan Priyatna, I. (2020). Data analytics in the supply chain management: Review of machine learning applications in demand forecasting. *Operations and Supply Chain Management: An International Journal*, *14*(1), 1–13.

Aamer, A., Yani, L. P. E., & Priyatna, I. M. (2021, 01). Data analytics in the supply chain management: Review of machine learning applications in demand forecasting. *Operations and Supply Chain Management An International Journal*, *14*, 1-13. doi: 10.31387/oscm0440281

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, *4*(11), e00938.

Babaee Tirkolaee, E., Sadeghi Darvazeh, S., Mooseloo, F., Rezaei Vandchali, H., & Aeini, S. (2021, 06). Application of machine learning in supply chain management: A comprehensive overview of the main areas. *Mathematical Problems in Engineering*, *2021*, 1-14. doi: 10.1155/2021/1476043

Barbour, W., Samal, C., Kuppa, S., Dubey, A., & Work, D. B. (2018). On the data-driven prediction of arrival times for freight trains on us railroads. In *2018 21st international conference on intelligent transportation systems (itsc)* (pp. 2289–2296).

Basturk, O., & Cetek, C. (2021). Prediction of aircraft estimated time of arrival using machine learning methods. *The Aeronautical Journal*, *125*(1289), 1245–1259.

Ben Naylor, J., Naim, M. M., & Berry, D. (1999). Leagility: Integrating the lean and agile manufacturing paradigms in the total supply chain. *International Journal of Production Economics*, *62*(1), 107-118. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0925527398002230` doi: https://doi.org/10.1016/S0925-5273(98)00223-0

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade* (pp. 437–478). Springer.

Berry, M. A., & Linoff, G. S. (2000). Mastering data mining: The art and science of customer relationship management. *Industrial Management & Data Systems*.

Bertolini, M., Mezzogori, D., Neroni, M., & Zammori, F. (2021). Machine learning for industrial applications: A comprehensive literature review. *Expert Systems with Applications*, *175*, 114820. Retrieved from `https://www.sciencedirect.com/science/article/pii/S095741742100261X` doi: https://doi.org/10.1016/j.eswa.2021.114820

Blake, M. (2021). Moody's warns of 'dark clouds ahead' for the global supply chain as 77% of the world's largest ports face backlogs. Business Insider. Retrieved from `https://www.businessinsider.com/global-supply-chain-crisis-ports-face-record-delays-2021-10?r=US&IR=T`

Bogaars, G. (1955). The effect of the opening of the suez canal on the trade and development of singpore. *Journal of the Malayan Branch of the Royal Asiatic Society*, *28*(1 (169)), 99–143. Retrieved 2022-05-12, from `http://www.jstor.org/stable/41503171`

Botchkarev, A. (2018). Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. *arXiv preprint arXiv:1809.03006*.

Brownlee, J. (2019). How to choose a feature selection method for machine learning. *Machine Learning Mastery*, *10*.

Bullinger, H.-J., Warschat, J., & Fischer, D. (2000). Rapid product development — an overview. *Computers in Industry*, *42*(2), 99-108. Retrieved from https://www.sciencedirect.com/science/article/pii/S0166361599000640 doi: https://doi.org/10.1016/S0166-3615(99)00064-0

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16–28.

Cloud Education, I. (2019). Neural networks.

Croxton, K. L., García-Dastugue, S. J., Lambert, D. M., & Rogers, D. S. (2001, Jan 01). The supply chain management processes. *The International Journal of Logistics Management*, *12*(2), 13-36. Retrieved from https://doi.org/10.1108/09574090110806271 doi: 10.1108/09574090110806271

Davenport, T. H. (1993). *Process innovation: reengineering work through information technology*. Harvard Business Press.

de Brito, M., & Dekker, R. (2003, 06). A framework for reverse logistics. *Reverse logistics*. doi: 10.1007/978-3-540-24803-3_1

Duffy, K. (2021). Pay for truckers is soaring — one said his salary shot up to $70,000 from $40,000. but it's not enough to fill thousands of driver vacancies. Business Insider. Retrieved from https://www.businessinsider.com/truckers-pay-hike-salary-truck-driver-shortage-2021-6?international=true&r=US&IR=T

El Naqa, I., & Murphy, M. J. (2015). What is machine learning? In I. El Naqa, R. Li, & M. J. Murphy (Eds.), *Machine learning in radiation oncology: Theory and applications* (pp. 3–11). Cham: Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-319-18305-3_1 doi: 10.1007/978-3-319-18305-3_1

Fernandez Garcia, A. J., Iribarne, L., Corral, A., Criado, J., & Wang, J. (2018, 11). A recommender system for component-based applications using machine learning techniques. *Knowledge-Based Systems*, *164*. doi: 10.1016/j.knosys.2018.10.019

Frey, D. D., Engelhardt, F., & Greitzer, E. M. (2003). A role for" one-factor-at-a-time" experimentation in parameter design. *Research in Engineering design*, *14*(2), 65–74.

Goldsby, T. J., & García-Dastugue, S. J. (2003). The manufacturing flow management process. *The International Journal of Logistics Management*, *14*(2), 33–52.

Granitto, P. M., Furlanello, C., Biasioli, F., & Gasperi, F. (2006). Recursive feature elimination with random forest for ptr-ms analysis of agroindustrial products. *Chemometrics and intelligent laboratory systems*, *83*(2), 83–90.

H2O. (2022). H2o ai.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Harrison, T. P. (2001). Global supply chain design. *Information Systems Frontiers*, *3*(4), 413–416.

Helms, M. M., Ettkin, L. P., & Chapman, S. (2000). Supply chain forecasting–collaborative forecasting supports supply chain management. *Business process management journal*.

Hildebrandt, F. D., & Ulmer, M. W. (2022). Supervised learning for arrival time estimations in restaurant meal delivery. *Transportation Science*, *56*(4), 1058–1084.

History. (2021). Silk road.. Retrieved from https://www.history.com/topics/ancient-middle-east/silk-road#section_2

Hoekstra, S., Romme, J., & Argelo, S. (1992). *Integral logistic structures: developing customer-oriented goods flow*. McGraw-Hill Book Company Limited.

Huber, J., & Stuckenschmidt, H. (2020). Daily retail demand forecasting using machine learning with emphasis on calendric special days. *International Journal of Forecasting*, *36*(4), 1420–1438.

Jantawan, B., & Tsai, C.-F. (2014). 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*, *2*(6), 4501–4508.

Jović, A., Brkić, K., & Bogunović, N. (2015). A review of feature selection methods with applications. In *2015*

38th international convention on information and communication technology, electronics and microelectronics (mipro) (pp. 1200–1205).

Kersten, W., See, B. v., Lodemann, S., & Grotemeier, C. (2020). Trends und strategien in logistik und supply chain management-entwicklungen und perspektiven einer nachhaltigen und digitalen transformation.

Kotsiantis, S. B., Zaharakis, I., Pintelas, P., et al. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*(1), 3–24.

Lee, H., Malik, W., & Jung, Y. C. (2016). Taxi-out time prediction for departures at charlotte airport using machine learning techniques. In *16th aiaa aviation technology, integration, and operations conference* (p. 3910).

Lickert, H., Wewer, A., Dittmann, S., Bilge, P., & Dietrich, F. (2021). Selection of suitable machine learning algorithms for classification tasks in reverse logistics. *Procedia CIRP*, *96*, 272–277.

Lin, F.-R., & Shaw, M. J. (1998). Reengineering the order fulfillment process in supply chain networks. *International Journal of Flexible Manufacturing Systems*, *10*(3), 197–229.

Lin, Y., Yang, X., Zou, N., & Jia, L. (2013). Real-time bus arrival time prediction: case study for jinan, china. *Journal of Transportation Engineering*, *139*(11), 1133–1140.

Lin, Y.-K., & Yeh, C.-T. (2010). Optimal carrier selection based on network reliability criterion for stochastic logistics networks. *International Journal of Production Economics*, *128*(2), 510–517.

Mahajan, P. C., Kiwelekar, A. W., Netak, L. D., & Ghodake, A. B. (2021). Predicting expected time of arrival of shipments through multiple linear regression. In *Icdsmla 2020: Proceedings of the 2nd international conference on data science, machine learning and applications* (Vol. 783, p. 343).

Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, *9*, 381–386.

Mahraz, M.-I., Benabbou, L., & Berrado, A. (2022, 03). Machine learning in supply chain management: A systematic literature review. *International Journal of Supply and Operations Management*, xx-xx. doi: 10.22034/ijsom.2021.109189.2279

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, *405*(2), 442–451.

McHugh, M. L. (2013). The chi-square test of independence. *Biochemia medica*, *23*(2), 143–149.

Mentzer, J. T., DeWitt, W., Keebler, J. S., Min, S., Nix, N. W., Smith, C. D., & Zacharia, Z. G. (2001). Defining supply chain management. *Journal of Business logistics*, *22*(2), 1–25.

Mohammed, M., Khan, M. B., & Bashier, E. B. M. (2016). *Machine learning: algorithms and applications*. Crc Press.

Naumov, M. (2017). Feedforward and recurrent neural networks backward propagation and hessian in matrix form. *arXiv preprint arXiv:1709.06080*.

Ni, D., Xiao, Z., & Lim, M. K. (2020, 07). A systematic review of the research trends of machine learning in supply chain management. *International Journal of Machine Learning and Cybernetics*. doi: 10.1007/s13042-019-01050-0

Nolinske, T. (2022). Customer service drives financial performance. National Business Research Institute. Retrieved from `https://www.nbrii.com/customer-survey-white-papers/customer-service-drives-financial-performance/`

NOS. (2021). Postnl bezorgde recordaantal pakketten in 2021.. Retrieved from `https://nos.nl/artikel/2419249-postnl-bezorgde-recordaantal-pakketten-in-2021`

Oshiro, T. M., Perez, P. S., & Baranauskas, J. A. (2012). How many trees in a random forest? In *International workshop on machine learning and data mining in pattern recognition* (pp. 154–168).

Oxford, D. (2022). Metric.. Retrieved from `https://www.oxfordlearnersdictionaries.com/definition/english/metric_2`

Pang, J., Huang, J., Du, Y., Yu, H., Huang, Q., & Yin, B. (2018). Learning to predict bus arrival time from heterogeneous measurements via recurrent neural network. *IEEE Transactions on Intelligent Transportation Systems*, *20*(9), 3283–3293.

Park, K., Sim, S., & Bae, H. (2021). Vessel estimated time of arrival prediction system based on a path-finding algorithm. *Maritime Transport Research*, *2*, 100012.

Patel, N., & Upadhyay, S. (2012). Study of various decision tree pruning methods with their empirical comparison in weka. *International journal of computer applications*, *60*(12).

Poona, N. K., & Ismail, R. (2013). Reducing hyperspectral data dimensionality using random forest based wrappers. In *2013 ieee international geoscience and remote sensing symposium-igarss* (pp. 1470–1473).

Prajapati, H., Kant, R., & Shankar, R. (2019). Bequeath life to death: State-of-art review on reverse logistics. *Journal of cleaner production*, *211*, 503–520.

Premack, R. (2020). Retailers are scrambling to find trucks amid the pandemic, generating record-smashing pay for drivers. Business Insider. Retrieved from `https://www.businessinsider.com/truck-driver-pay-record-high-coronavirus-2020-9?international=true&r=US&IR=T`

Puthiyamadam, T., & Reyes, J. (2018). Experience is everything. get it right. PWC. Retrieved from `https://www.pwc.com/us/en/services/consulting/library/consumer-intelligence-series/future-of-customer-experience.html#:~:text=In%20the%20U.S.%2C%20even%20when,loved%20after%20one%20bad%20experience.`

Qi, Y. (2012). Random forest for bioinformatics. In *Ensemble machine learning* (pp. 307–323). Springer.

Randles, W. G. L. (1988). *Bartolomeu dias and the discovery of the south-east passage linking the atlantic to the indian ocean (1488)* (Vol. 188). UC Biblioteca Geral 1.

Reinhoudt, E. M., & Velastin, S. (1997). A dynamic predicting algorithm for estimating bus arrival time. *IFAC Proceedings Volumes*, *30*(8), 1225–1228.

Richter, B., Knichel, D., & Moradi, A. (2019). A comparison of $$$
backslash$chiˆ 2$$-test and mutual information as distinguisher for side-channel analysis. In *International conference on smart card research and advanced applications* (pp. 237–251).

Rubio, S., & Jiménez-Parra, B. (2014). Reverse logistics: Overview and challenges for supply chain management. *International Journal of Engineering Business Management*, *6*, 12.

Sarker, I. (2021). Machine learning: Algorithms, real-world applications and research directions. In *Advances in computational approaches for artificial intelligence, image processing, iot and cloud applications.* Springer International Publishing.

Sarker, I. H., Kayes, A., Badsha, S., Alqahtani, H., Watters, P., & Ng, A. (2020). Cybersecurity data science: an overview from machine learning perspective. *Journal of Big data*, *7*(1), 1–29.

Schroeder, M., & Lodemann, S. (2021). A systematic investigation of the integration of machine learning into supply chain risk management. *Logistics*, *5*(3), 62.

Smith, J., Berger, P., & O'Neal, L. (2021). Shipping and logistics costs are expected to keep rising in 2022. The Wall Street Journal. Retrieved from `https://www.wsj.com/articles/shipping-and-logistics-costs-are-expected-to-keep-rising-in-2022-11639918804`

Song, Y.-Y., & Ying, L. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, *27*(2), 130.

Suganthi, L., & Samuel, A. A. (2012). Energy models for demand forecasting—a review. *Renewable and Sustainable Energy Reviews*, *16*(2), 1223-1240. Retrieved from `https://www.sciencedirect.com/science/article/pii/S1364032111004242` doi: https://doi.org/10.1016/j.rser.2011.08.014

Tanizaki, T., Hoshino, T., Shimmura, T., & Takenaka, T. (2019). Demand forecasting in restaurants using machine learning and statistical analysis. *Procedia CIRP*, *79*, 679–683.

Tharwat, A. (2020). Classification assessment methods. *Applied Computing and Informatics*.

Thierry, M., Salomon, M., Van Nunen, J., & Van Wassenhove, L. (1995). Strategic issues in product recovery management. *California management review*, *37*(2), 114–136.

Towill, D. R., & McCullen, P. (1999). The impact of agile manufacturing on supply chain dynamics. *The international journal of Logistics Management*, *10*(1), 83–96.

van der Spoel, S., Amrit, C., & van Hillegersberg, J. (2017). Predictive analytics for truck arrival time estimation: a field study at a european distribution centre. *International journal of production research*, *55*(17), 5062–5078.

Wah, Y. B., Ibrahim, N., Hamid, H. A., Abdul-Rahman, S., & Fong, S. (2018). Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy. *Pertanika Journal of Science &*

Technology, *26*(1).

Waters, J. (2021). What is otif, how to calculate and how did it come about?. Retrieved from `https://www.tive.com/blog/on-time-in-full-otif-what-is-otif-and-how-to-improve-metrics-with-technology`

Wei, Q., & Dunbrack Jr, R. L. (2013). The role of balanced training and testing data sets for binary classifiers in bioinformatics. *PloS one*, *8*(7), e67863.

Wenzel, H., Smit, D., & Sardesai, S. (2019). A literature review on machine learning in supply chain management. In *Artificial intelligence and digital transformation in supply chain management: Innovative approaches for supply chains. proceedings of the hamburg international conference of logistics (hicl), vol. 27* (p. 413-441). Berlin: epubli GmbH. Retrieved from `http://hdl.handle.net/10419/209380` (urn:nbn:de:gbv:830-882.054345; 10419/209196; https://econpapers.repec.org/RePEc:zbw:hiclpr:27) doi: 10.15480/882.2478

Wesely, D., Churchill, A., Slough, J., & Coupe, W. J. (2021). A machine learning approach to predict aircraft landing times using mediated predictions from existing systems. In *Aiaa aviation 2021 forum* (p. 2402).

Xiong, H., Pandey, G., Steinbach, M., & Kumar, V. (2006). Enhancing data analysis with noise removal. *IEEE Transactions on Knowledge and Data Engineering*, *18*(3), 304–319.

Xu, Y., Yen, D. C., Lin, B., & Chou, D. C. (2002). Adopting customer relationship management technology. *Industrial management & data systems*.

Yang, M., Chen, C., Wang, L., Yan, X., & Zhou, L. (2016). Bus arrival time prediction using support vector machine with genetic algorithm. *Neural Network World*, *26*(3), 205.

Yiu, T. (2019). Understanding random forest.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zeng, A. Z. (2000). A synthetic study of sourcing strategies. *Industrial Management & Data Systems*.

Zibran, M. F. (2007). Chi-squared test of independence. *Department of Computer Science, University of Calgary, Alberta, Canada*, *1*(1), 1–7.