DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING,
SECOND CYCLE, 30 CREDITS
*STOCKHOLM, SWEDEN 2021*

# Machine learning based inventory optimization respecting supplier order line fees

**MINDAUGAS VARKALYS**

**KTH ROYAL INSTITUTE OF TECHNOLOGY**
**SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE**

# Machine learning based inventory optimization respecting supplier order line fees

Mindaugas Varkalys

2021-10-31

Master’s Thesis

**Examiner**
Anders Västberg

**Academic adviser**
Jan Markendahl

**Industrial adviser**
Ričardas Varkalys

KTH Royal Institute of Technology
School of Electrical Engineering and Computer Science (EECS)
Department of Computer Science
SE-100 44 Stockholm, Sweden

# Abstract

This thesis addresses an inventory management problem of what, when and how many products should be ordered from the supplier applying order line fees. Order line fee is a fixed fee which the company pays to the supplier per every ordered product not depending on the ordered quantity. Even though there are various inventory management methods and variety of research done in the field, there was no research found related to inventory management when supplier order line fees are applied. The described problem is real and currently exists at the company ASWO Baltic.

The problem is solved by using experimental research method and CRISP-DM process. The historical company's data of customer and supplier orders is used for the project. Data is analyzed and prepared for model creation by using feature engineering, data transformation and data normalization methods.

Min/Max inventory management method is used as a base for model creation. The improvement proposed by the thesis is to use machine learning algorithms to predict Min and Max stock levels. Support Vector Regression, k-nearest neighbors, Random Forest, Artificial Neural Network, ARIMA, and Prophet machine learning algorithms are tested both for Min and Max level prediction. It was found out that the best results for Min stock level prediction were achieved by k-nearest neighbors algorithm with the average sMAPE measure of 7.0079%. The best predictions for Max stock level were done by Random Forest algorithm with the average sMAPE of 15.0303%. After the hyperparameter optimization sMAPE was improved to 6.8730% and 14.6813% accordingly.

The simulation was run to evaluate if the proposed algorithm outperforms the current system. It showed that for the items which have more than 200 orders the algorithm decreased the number of supplier orders by 35,83% and the number of backorders by 49,29% while keeping almost the same inventory turnover. If the same results are achieved with the all products, it is expected that the company would save around 60K euros per annum on supplier order line fees and the lower number of backorders would increase sales by 24%.

**Keywords**

Machine learning; inventory management; order line fee; supply chain; Min/Max inventory management method

# Sammanfattning

Detta examensarbete tar upp ett lagerhanteringsproblem om vad, när och hur många produkter som ska beställas från leverantören som tillämpar orderradsavgifter. Orderradsavgift är en fast avgift som företaget betalar till leverantören för varje beställd produkt, inte beroende på beställd kvantitet. Även om det finns olika lagerhanteringsmetoder och olika undersökningar som gjorts inom området, hittades ingen forskning relaterad till lagerhantering när orderradsavgifter tillämpas. Det beskrivna problemet existerar idag på företaget ASWO Baltic.

Problemet löstes genom att använda experimentell forskningsmetod och CRISP-DM-process. Företagets historiska data om kund- och leverantörsbeställningar har används för projektet. Data har analyseras och förbereds och modellerats med hjälp av funktionsteknik, datatransformation och datanormaliseringsmetoder.

Min/Max lagerhanteringsmetoden används som bas för att skapa en modell. Förbättringen som föreslås i avhandlingen är att använda maskininlärningsalgoritmer för att förutsäga Min och Max lagernivåer. Stöd för vektorregression, k-närmaste grannar, Random Forest, Artificiellt neuralt nätverk, ARIMA och Prophet maskininlärningsalgoritmer testas både för förutsägelse av min- och maxnivå. Det visade sig att de bästa resultaten för förutsägelse av Min lagernivå uppnåddes med algoritmen "k-nearest neighbors" med det genomsnittliga sMAPE-måttet på 7,0079%. De bästa förutsägelserna för Max lagernivå gjordes av Random Forest-algoritmen med den genomsnittliga sMAPE på 15,0303%. Efter hyperparameteroptimeringen förbättrades sMAPE till 6,8730 % och 14,6813 % i enlighet därmed.

Simuleringen kördes för att utvärdera om den föreslagna algoritmen överträffar det nuvarande systemet. Den visade att för de artiklar som har mer än 200 beställningar minskade algoritmen antalet leverantörsbeställningar med 35,83 % och antalet restorder med 49,29 % samtidigt som det bibehöll nästan samma lageromsättning. Om samma resultat uppnås med alla produkter, förväntas företaget spara cirka 60 000 euro per år på leverantörsorderavgifter och det lägre antalet restorder skulle öka försäljningen med 24 %.

# Acknowledgments

*I would like to express my special thanks to my supervisor Jan Markendahl and examiner Anders Västberg for helping and guiding me through the whole research and writing process of this theses.*

*I am especially graceful to my girlfriend Patricija, mother Vida and sister Laura for the constant encouragement, support, and motivation, which helped me a lot to finish this research.*

*Finally, I would like to give a huge thank you to my father Ričardas, director of ASWO Baltic, for giving me the expertise and the industrial knowledge as well as sharing with me the private company sales data and allowing to use it for this project. Without it this project would not be possible.*

Stockholm, June 2021
Mindaugas Varkalys

# Table of contents

# List of Figures

# List of Tables

# 1 Introduction

This thesis discusses the inventory management problem in the supply chain. A direct supply chain consists of a company, a supplier, and a customer involved in the upstream and/or down-stream flows of products, services, finances, and/or information [1]. In reality, the flow of products between a company, a supplier and a customer is not instant due to the shipping time. Thus, the company might keep some products in his warehouse to eliminate the shipping time from the supplier that way minimizing the overall product shipping time to the final customer. Keeping the right products and the right amount in the warehouse is crucial for the company since the long delivery times is usually a deal breaker for the customer buying a product. To do that the company needs to know what to order from the supplier beforehand to keep up with the customer demand in the future.

Such problem exists for decades in the supply chain and there is a wide variety of inventory management policies and algorithms [2]–[4]. The algorithms try to calculate what to order from the supplier based on the order history of the customers. However, such algorithms are generic and do not consider the case if the supplier applies order line fee. Order line fee is a fixed fee which the company pays to the supplier per every order line. For example, if the order line fee is 1€, no matter if the company orders 1 or 1000 units of the product, it has to pay that fee to the supplier for ordering the product. This fee makes the whole problem way more complex because the number of supplier orders has to be minimized.

## 1.1 Problem

The problem of the project is inefficient supplier order and warehouse inventory management in a supply chain with a supplier applying order line fees. This problem can be divided in several subproblems presented as research questions:

- What products should be ordered from the supplier applying order line fees?
- How many units of each product should be ordered?
- When the supplier order should be executed?
- What products should be kept in stock?

The cause of such the problem and its subproblems is inefficient warehouse inventory management process. To solve the problem a new machine learning based inventory management algorithm is created which solves the issues of the current process.

There is a little research done in machine learning application to the warehouse inventory management and there is no found research for the case if the supplier applies order line fees. Therefore, solving the problem would be a large contribution to the research field as a whole.

## 1.2 Background of the problem

The problem currently exists at the company ASWO Baltic. The company is selling spare parts of various electronics and home appliances [5]. The company has a warehouse where it stores the most frequently bought products to make them available for the customers instantly. However, the company cannot store all the supplier products in stock due to the limited warehouse size and storing expenses. Therefore, the unpopular products are ordered directly from the supplier and have delivery time of few days. If the popular products run out-of-stock they are ordered from the supplier as well.

ASWO Baltic has a supplier ASWO, which applies order line fees - the company pays a fixed fee of few euros for every ordered product independently from the ordered quantity or item value.

Optimizing supplier orders and products kept in stock is crucial to run a profitable business. If stock and orders are managed badly, the company can lose money or even go into bankruptcy. This can be illustrated in the simple example. Let us say there is a customer who wants to buy a cable from the company. The company does not have it in its warehouse so orders it from the supplier at the price of 1€ and sells it to the customer for 2€. It looks that the company made 1€ profit. However, it is only the case if there are no supplier fees. If the supplier order line fee is 4€, the company pays to the supplier 1+4=5€ and receives from the customer only 2€, so the company loses 3€ by pursuing this deal. Such deal is only profitable if the company orders 5 units or more and stores the others in its warehouse for the future sales.

Efficient stock and order management reduces not only the fees, but also the storage expenses. If the company orders the products without predicting the future customer demand, it could cause long inventory turnover – there will be many products staying in the warehouse for the long period of time. Because of long inventory turnover, the company would need larger warehouse to store all the product. That would increase expenses spent on rent, utility bills and staff, since the management of the larger warehouse would require more human labor.

Moreover, inefficient order management is bad for the environment. Usually, every order is packed separately. As a result, reducing the number of orders by ordering more in a single batch could reduce the materials used for packaging. In addition, lowering the number of orders could also reduce the shipping costs, fuel consumption as well as carbon emissions, since the orders are delivered less often.

Most importantly, having the right items at the right time is crucial for the high customer satisfaction and sales. One study [6] showed that if the product is out-of-stock, around 33% of customers buy the item in the other store or do not buy it at all. Another study [7] found that 69% of customers are less likely to shop with a retailer for future purchases if an item is not delivered within two days of the date promised. That shows the importance of keeping the items in stock as well as delivering them fast and on time.

Finally, in the current company's system, most of the decisions what to order from the supplier are done manually. It is far from optimum since the company sells a few millions different products and receives hundreds of orders per day. Such large amount of data is impossible to be processed by a human efficiently.

As ASWO Baltic is a fast-growing business, the consequences of the inefficient order and inventory management problem will amplify in the future. Thus, the problem should be solved now to allow the company scale efficiently by minimizing the fees paid to the supplier, storage expenses, environmental impact as well as increasing customer satisfaction and sales.

## 1.3 Purpose

The purpose of the thesis is to lower fees paid by the company to the supplier and reduce product delivery times to the final customer. As a result, this would increase the overall customer satisfaction, sales, and profits. It would also lower the expenses and human resources needed for the order and warehouse management. In addition, it would lower the company's environmental impact by reducing packaging waste and fuel consumption.

## 1.4 Goals

The main goal of this project is to create an algorithm which can predict when and which products should be ordered from the supplier based on the customer order history. To achieve the thesis purpose the algorithm should outperform the current company's system.

## 1.5 Research Methodology

This section describes the high-level research methodology concepts used in the project. The concepts described in this section are implemented by applying research methods and techniques which are explained in Chapter 3 in more detail.

### 1.5.1 Method

Quantitative experimental research method is used in the project [8]. This research method is chosen because an algorithm which optimizes the supplier orders depending on the historical order data is being developed. The performance of various algorithms is measured in metrics extracted from the resulting data. All the metrics are depended on each other, so finding an optimal solution just looking at the problem is almost an impossible task to do. Thus, the best way to solve such problem is to test many different algorithms and models, experiment with their various parameters and that way find an optimal solution.

Apart from that, the experimental research method is selected because the project involves the machine learning. A large amount of machine learning algorithms has hyperparameters, which optimal values are usually impossible to calculate for the specific data. Therefore, the only option is to experiment and test which hyperparameters' values are the optimal.

### 1.5.2 Approach

Deductive research approach is used in the project [8]. This approach is selected because the company has a large historical dataset containing more than 360K received orders in the period of more than 5 years. Running an algorithm using the historical data and extracting the certain metrics from the results allow to validate the system and expect that it will still be valid in the future.

### 1.5.3 Strategy

Experimental research strategy is used in the project [8]. This research strategy is chosen because during the testing of the algorithm all the factors and variables are controlled. This is because the results of the algorithm depend only on the historical data and hyperparameters which can be changed. Such environment allows to analyze the causalities between variables and results as well as draw conclusions about such behavior.

### 1.5.4 Process

In this thesis the research method, approach and strategy are applied by following CRISP-DM process (**CR**oss-**I**ndustry **S**tandard **P**rocess for **D**ata **M**ining) [9]. CRISP-DM process consists of a cycle that comprises six stages:

1. **Business understanding.** This initial phase focuses on understanding the project objectives and requirements from a business perspective.

2. **Data understanding.** The data understanding phase starts with the data collection and proceeds with activities in order to get familiar with the data, to identify data quality problems, to discover first insights into the data or to detect interesting subsets.
3. **Data preparation.** The data preparation phase covers all activities to construct the final dataset from the initial raw data.
4. **Modeling.** In this phase, various modeling techniques are selected and applied, and their parameters are calibrated to optimal values.
5. **Evaluation.** At this stage, the model obtained are more thoroughly evaluated and the steps executed to construct the model are reviewed to be certain it properly achieves the business objectives.
6. **Deployment.** At this stage, the knowledge gained is organized and presented in a way that the customer can use it [10].

The process is illustrated in Figure 1.

This process is chosen because it is an industry standard used in many various scale projects and companies. According to the survey [11] CRISP-DM was the most popular methodology for analytics, data mining, or data science projects both in 2007 and 2014. Using widely known and approved research process helps to avoid common mistakes and prove research truthiness and legitimacy.

The whole project is conducted by following this process. This is also illustrated in the thesis structure - Chapters 3 and 4 of the thesis are written by following the steps of this process. Business Understanding, Data Understanding, Data Preparation, and Modeling stages are detailly described in Chapter 3. Evaluation is described in Chapter 4. Deployment is out of scope of this thesis and is left for the future research.

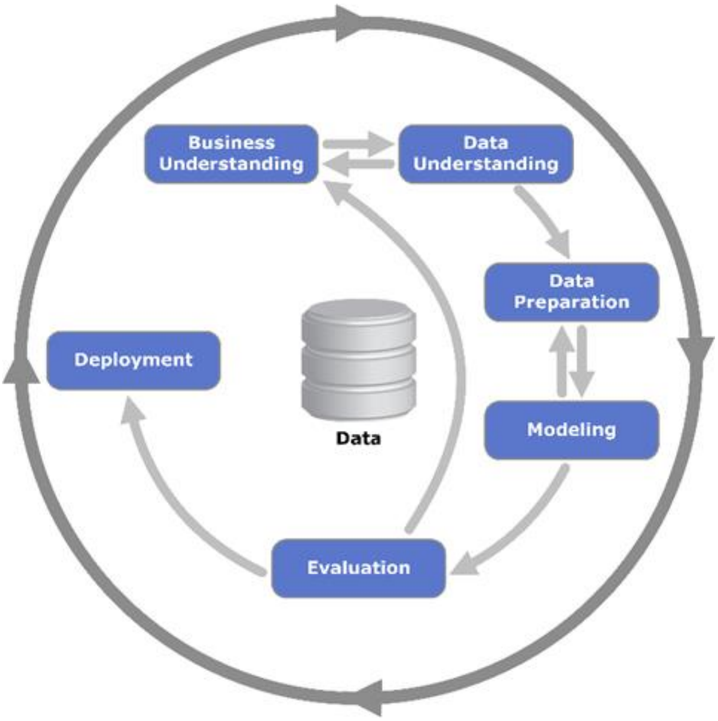

**Figure 1: CRISP-DM lifecycle [10]**

## 1.6 Delimitations

The algorithm described in the thesis focuses only on the quantities of the items without considering other properties such as dimensions, value, availability, shipping or storing conditions and others. As a result, some actions proposed by the algorithm might be impossible, impractical, or unprofitable. Due to this limitation the algorithm should work as a recommendation system rather than be self-functioning since the actions recommended by the algorithm should be reviewed before execution.

Challenges such as supply chain disruptions and bullwhip effect are not specifically addressed by the algorithm. However, these concepts might still be learnt and addressed by the machine learning algorithms used in the project.

Chosen methods and results of the project are highly dependent on the data. Even though it is likely, there is no guarantee that the similar results are achieved if the different data is used. In order to reproduce the results with the different data, it is recommended to focus on the process described in thesis rather than the machine learning algorithms or their hyperparameters chosen in this project.

Simulation done for the project evaluation does not consider all the real-world scenarios. Supply chain disruptions, supplier being out of stock, and warehouse size are not considered in the simulation. Thus, if the system is used in the reality the improvement found by the simulation might be lower.

The thesis does not cover the deployment, visualization, or integration of the algorithm to the current company's infrastructure. Such work is left for the future.

## 1.7 Structure of the thesis

Chapter 2 describes the relevant context and background in the supply chain area. In the end of the chapter, the related works are reviewed.

Chapter 3 detailly presents the entire algorithm creation process.

In Chapter 4 the project is evaluated by running a simulation and doing execution time as well as reliability analysis. The chapter ends with the discussion.

In Chapter 5 the conclusions are stated, and the future research opportunities presented.

# 2 Background

This chapter provides the background information about the supply chain. In the end of the section the other related works are analyzed which are related to this thesis.

## 2.1 Supply chain

A supply chain is a network of organizations that are involved in the different processes and activities that produce value in the form of products and services in the hands of the ultimate consumer [12].

Three degrees of supply chain complexity can be identified: a "direct supply chain," an "extended supply chain," and an "ultimate supply chain." A direct sup-ply chain consists of a company, a supplier, and a customer involved in the upstream and/or down-stream flows of products, services, finances, and/or information. An extended supply chain includes suppliers of the immediate supplier and customers of the immediate customer, all involved in the upstream and/or downstream flows of products, services, finances, and/or information. An ultimate supply chain includes all the organizations involved in all the upstream and downstream flows of products, services, finances, and information from the ultimate supplier to the ultimate customer [1]. This is illustrated in Figure 2.

The work done in this thesis considers a direct supply chain only.

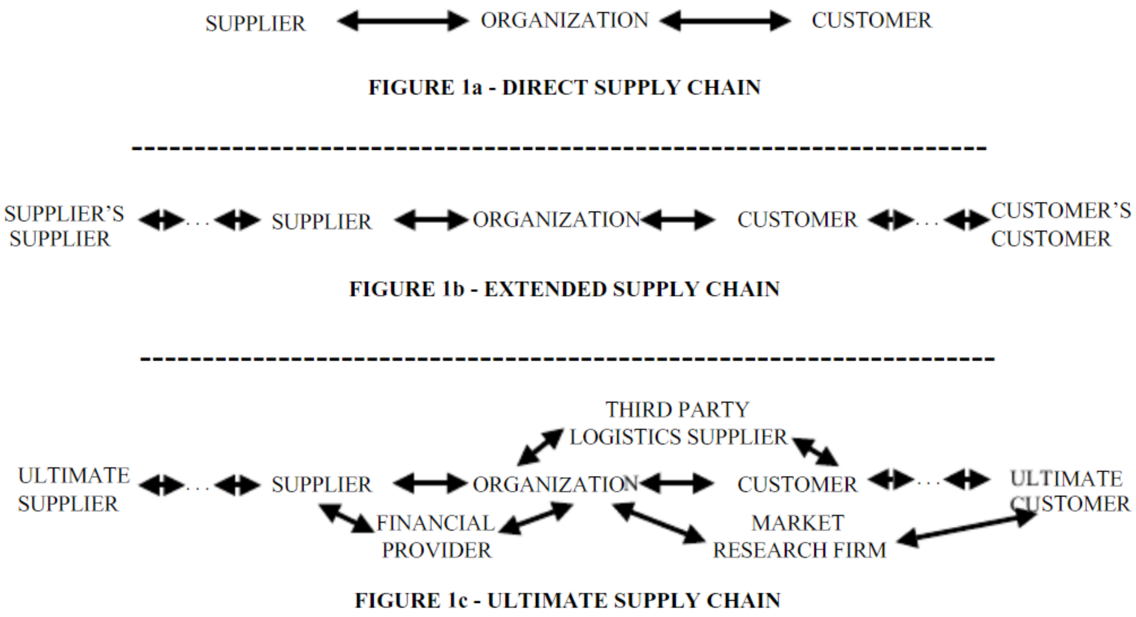

**Figure 2: Different degrees of supply chain [1]**

## 2.2 Supply chain management

There are many different definitions of supply chain management [1]. Generally, it can be seen as the management of upstream and downstream relationships with suppliers and customers in order to deliver superior customer value at less cost to the supply chain as a whole [12]. Supply chain management is a wide concept which includes many areas such as logistics, inventory management, demand management, manufacturing, procurement and many more.

In today’s global economy efficient supply chain management is extremely important, yet very difficult task. Today a product available for purchase in the retail store might be manufactured on the other side of the world and sourced through multiple distributors and wholesalers. Furthermore, the parts and materials needed to manufacture the product might also come from multiple continents. Moreover, the customer demand might also vary depending on the weather, season, or social trends. Lastly, globally distributed supply chain requires a lot of time for shipping forcing managers to predict the demand long in the future that way making it less accurate. All in all, such complex, global and hardly predictable structure is very difficult to manage. Thus, the supply chain is excessively sensitive to various kinds of disruptions.

A good example of such situation could be a shortage of personal protective equipment during the global pandemic of COVID-19 in 2020. The high number of infected people increased the public demand for protective masks and gloves making them quickly run out of stock. Even the hospitals did not have enough protective equipment that way putting many medical workers lives at huge risk of being infected [13]. Neither the retailers, nor the manufacturers could have predicted such outbreak and adjust their decisions early enough in a way that the consequences could have been avoided.

Another situation which largely disrupted the global supply chain was the blockage of Suez Canal in 2021. While sailing through the canal, a massive container ship Ever Given lost control due to the strong wind and became stuck sideways blocking the entire passage. Around 12% of total global trade goes through Suez Canal and it was estimated that each hour while the canal was blocked cost 400 million dollars to the global economy. The ship was freed in 6 days and during that period the blockage caused price changes and shipping delays across various industries [14].

The aforementioned examples were major supply chain disruption events in the recent years. However, there might be various other types of events which effect supply chains such as strikes [15], weather [16], political decisions [17] and others. Such events are usual, yet they cannot be predicted. Thus, the needed adjustments must be made as fast as possible to reduce their impact to the final customers.

## 2.3 Inventory management methods

In the long history of the inventory management there has been a wide variety of methods, which were defined and used across the industry. This section describes two popular inventory management methods – Periodic Review System and Min/Max.

### 2.3.1 Periodic Review System

One of the simplest inventory management methods is a Periodic Review System. It works as follows:

- Inventory levels start at some restocking level, *RS*.
- At the regular time intervals (ex. – 3 days, two weeks, etc.), the inventory level is reviewed. This new inventory level is called *I*.
- Some amount, *Q*, is added to bring the inventory level back up to *RS* [3].

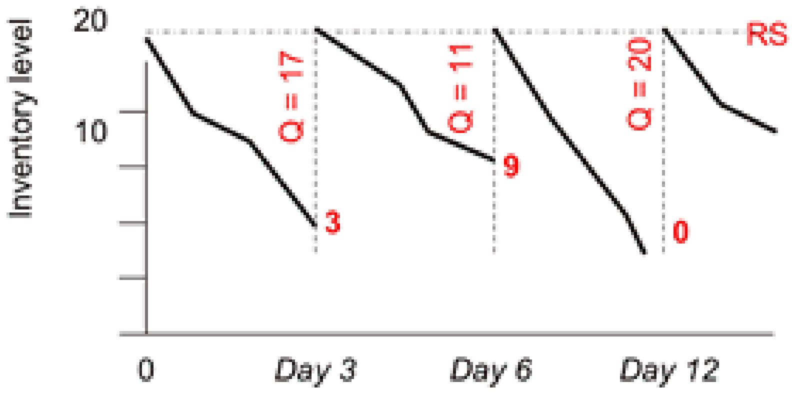

Figure 3: Periodic Review System [3]

Restocking level can be variable and be calculated using the formula:

$$RS = D + SS$$

*D* - average demand during the reorder period plus the replenishment lead time (if there is a delay getting new products in).

*SS* - safety stock. This is a “cushion” of inventory held to mitigate the uncertainties of forecasts and lead times. Higher safety stock levels increase the likelihood that goods are available, but also drives up inventory levels and costs [3].

The main advantage of Period Review System is its simplicity. However, it has many disadvantages. Firstly, in this method the supplier orders do not depend on the demand. For example, if the demand increases and the item runs out of stock, the supplier order is not executed until the next review time comes. Secondly, such model creates many small supplier orders. This is because every item, which is below the restocking level *RS* (at least by 1), is ordered from the supplier, even though the current inventory would be enough until the next review. For our case this is not acceptable, since every line in the supplier order has an additional fee and such model would increase the total fees paid to the supplier tremendously.

### 2.3.2 Min/Max method

Some drawbacks of Period Review System are overcome by Min/Max method. It is an improved version of Periodic Review System, which has two stock levels (Min and Max) instead of one (Restocking level). In addition, instead of ordering the products from the supplier at the fixed interval, in Min/Max the items are ordered only when the inventory becomes low.

This method tracks the current total stock level, which is typically the sum of the stock-on-hand plus the stock-on-order for every single item. When total stock reaches the Min value, a reorder is triggered. The reorder quantity targets the Max value for the new total stock level; hence the reorder quantity is the difference between Max and Min [2].

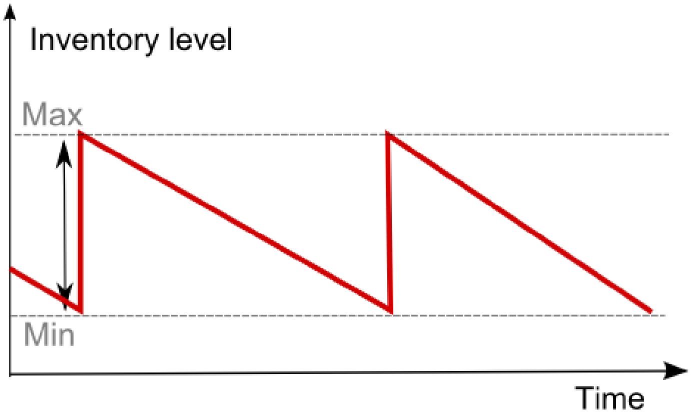

Figure 4: Min/Max inventory model [2]

Min/Max solves the problem of the unnecessary supplier orders. Orders also depend on the supply since the order is executed when Min level is reached not waiting until the next review cycle. However, the biggest challenge for Min/Max method is calculating Min and Max levels.

For any Min/Max system, you should set the Max and Min levels high enough to avoid stockouts, yet low enough so you do not increase the risk of expiration or damage. You must set a Min level high enough to ensure that the facility never completely runs out of stock. At the same time, you must still set the Max low enough to ensure that space in the storeroom is adequate and that the stock does not expire before it can be used [18].

Setting the correct Min/Max levels is a very difficult task which is crucial for the business success. The levels are very hard to calculate correctly because the future demand needs to be predicted. For that case machine learning algorithms are employed in this thesis.

## 2.4 Supplier order line fee

One of the major areas discussed by this thesis is supplier order line fee. This section analyzes the background of such fee and why it is applied by the supplier.

Every supplier having a warehouse has the certain warehouse processes. One of the main processes used in warehouse is a picking process. During this process, the worker processes the order by picking the items, packing them, and shipping. If the discrete picking process [19] is used, the warehouse operator has to take the order, walk to the first item's location, pick the needed amount of items and repeat this until all the order items are collected. After that, the order is packed and shipped to the customer. Such process is illustrated in Figure 5.

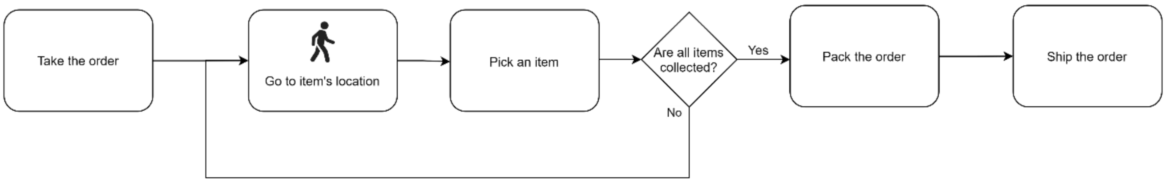

**Figure 5: Discrete picking warehouse process**

A warehouse might use other picking processes [19] or add additional steps to this process like barcode scanning or label printing but the general idea remains the same – most of the time while executing the process is spend on going from one item's location to the other's and picking the items. For example, a regular worker in Amazon can walk around 17 miles during a shift by picking items [20]. As a consequence, the amount of human labor needed to fulfil the order is proportional to the number of different items in the order, since the worker has to walk from item to item.

Human labor is expensive and most importantly limited resource in the warehouse. If there is a large number of orders of different items, the warehouse management cannot just simply employ enormous number of people to fulfil them, since there is a limited amount of working space. Therefore, each warehouse has a certain limit of orders it can process, and that threshold cannot be increased just by employing additional staff.

Some warehouses increase that threshold by using robots and automations. A good example of this could be Amazon. In some warehouses walking workers were replaced by robots. Instead of walking to the shelf and picking the item, the robots bring the entire shelf to the worker who picks the item. That way the number of items taken by each worker increased 3-4 times and in the same way the employees no longer needed to walk long distances during their workday [21]. However, even though the total number of orders which can be handled by the warehouse increased, it still remained limited. This is due to the limited number of robots and employees who can be fitted in the warehouse. Thus, the number of orders should be controlled to not overcome the warehouse limits.

The unique items order control discussed in this thesis is an order line fee. As earlier described in Introduction, order line fee is a fixed fee which the company pays to the supplier per every order line, not depending on the order size. For instance, if the company buys 3 products at unit price 2€ and order line fee is 3€, the company pays to the supplier in total 3 x 2€ + 3€ = 9€. Such fee allows the supplier to control the number of unique items orders by initiating its customers to order more units of the item in a single order.

## 2.5 Bullwhip effect

Bullwhip effect or whiplash effect is a common phenomenon where the orders to the supplier tend to have larger variance than sales to the buyer and the distortion propagates upstream in the supply chain in an amplified form [22]. There are many causes of the bullwhip effect but the major one is demand signaling which can be solved by sharing stock and order information across various supply chain members and reducing lead time [22] – the time period between order initiation and receiving of the items [23]. The bullwhip effect can be observed in a real supply chains [24] or in “Beer Distribution Game” [25].

“Beer Distribution Game” is a boardgame simulating a supply chain. During the game, every player (or a group of players) plays a role of a retailer, wholesaler, distributor, or a factory. During every round, the retailer draws a card from the pile with the number representing the end customer order. If the retailer has enough bottles of beer in stock, he/she fulfils the customer order and places the order for his/her neighbor - the wholesaler. The wholesaler tries to fulfil the retailer’s order and places the order for the distributor. Distributor does the same and creates the order for the factory. Manufacturer tries to fulfil the distributor’s order and orders the materials for future production. Between every order and delivery there is a certain delay representing the shipping time. The goal of each player is to fulfil as many orders received from the neighbor while keeping as less items in stock as possible [25].

During the play, the factory's orders fluctuate by a huge margin, even though the retailer's orders change a little. This is an example of the bullwhip effect. The changes in orders can be seen in Figure 6.

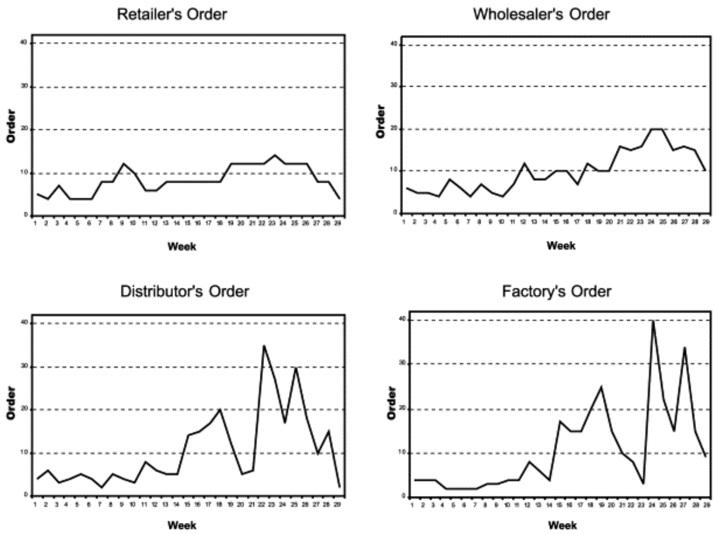

**Figure 6: Changes in order size of each role while playing "Beer Distribution Game" [26]**

However, the bullwhip effect can be eliminated if "Beer Distribution Game" is played by the computers. A study showed that by empowering a genetic machine learning algorithm the computer managed to track demand, eliminate the bullwhip effect, discover the optimal policies under complex scenarios [26].

## 2.6 Related work

This section describes the related research to the future demand forecasting which is done in the thesis. The described works analyze the usage and performance of various machine learning and statistical models for the future demand forecasting task.

### 2.6.1 Machine learning demand forecasting techniques

Real Carbonneau, Kevin Laframboise, Rustam Vahidov paper "Application of machine learning techniques for supply chain demand forecasting" [27] describes and tests various machine learning and statistical forecasting techniques to predict customer demand. The research analyses Naïve Forecast, Average, Moving Average, Trend, Multiple Linear Regression, Neural Networks, Recurrent Neural Networks and Support Vector Machines methods. The project uses two datasets: first data is generated through the simulation and the second one is gathered from the real-world data. The paper found out that the best performing algorithms in their study were Recurrent Neural Networks and Support Vector Machines. However, only the case when the different supply chain entities (company, supplier, customer) do not share any information about their stock levels and orders is considered and the impact of information sharing is left for the future research.

### 2.6.2 Inventory optimization using genetic algorithm

The paper “Inventory Optimization in Efficient Supply Chain Management” [28] written by S.R. Singh and Tarun Kumar dicusses the genetic algorithm used for calculation of optimal excess and shortage stock levels. The algorithm is implement using MATHLAB and proved to be sufficient by comparing the results with the data from the past.

### 2.6.3 Demand forecasting using neural network

The research “Demand Forecasting using Neural Network for Supply Chain Management“ [29] conducted by Ashvin Kochak and Suman Sharma observes the performance of Artificial Neural Network algorithm implemented in MATHLAB. The study used 4 years sales data from the fuel filter distributor. Different backpropagation training methods are tested: Batch Gradient Descent, Variable Learning Rate, Conjugate Gradient Algorithms and Levenberg-Marquardt algorithm. The research found that the performance of Levenberg-Marquardt algorithm was more effective and reliable compared to the others tested for that case.

### 2.6.4 Literature study of demand forecasting methods

The literature survey done by Agnieszka Lasek, Nick Cercone and Jim Saunders in the paper “Restaurant Sales and Customer Demand Forecasting: Literature Survey and Categorization of Methods” [30] describes various methods used for restaurant sales and demand forecasting as well as revenue management. The methods of Multiple regression, Poisson regression, Box-Jenkins model (AR, MA, ARIMA), Exponential smoothing and Holt-Winters models, Artificial neural networks, Bayesian network model, Hybrid model, and Association rules are described clearly providing references and usage examples as well as advantages and disadvantages of each of them.

## 2.7 Summary

To sum up, a supply chain is a complex structure of organizations, which management is extremely hard but very important task. This is due to the global expansion of supply chains which causes long delivery times and creates various challenges and risks. All of this leads to supply chain vulnerability to changes in demand and various disruptions which are almost impossible to predict and hard to manage. Furthermore, even if the demand varies a little and there are no disruptive events, the supply chain can be imbalanced due to the bullwhip effect.

Supplier order line fee discussed in the thesis is a fixed fee paid to the supplier per every order line, not depending on the order size or number of units bought. The reason of this fee lies in the warehouse processes.

Inventory management methods Period Review and Min/Max can help to manage the inventory and solve some supply chain vulnerability problems. However, the methods by themselves do not solve the problem since their parameters have to be optimally chosen for the methods to work efficiently.

Various research has been done in the field of the future demand forecasting, which showed that the best performing algorithms for such task are Support Vector Machine, Artificial Neural Network, and genetic algorithms.

# 3 Building an algorithm

This chapter describes the research process stages, activities, methods and tools used to build an inventory management algorithm. Firstly, the hardware and technologies are described. Later, Business Understanding, Data Understanding, Data Preparation, and Modeling stages of CRISP-DM process are detailly described in the appropriate sections.

## 3.1 Hardware and Technologies

Python [31] programming language versions 3.8 and 3.9 are used for the project implementation. This language is selected due to the large number of libraries available for machine learning, data processing and visualization. The following libraries are used in the project:

- Pandas [32] version 1.2.4
- NumPy [33] version 1.20.0
- scikit-learn [34] version 0.24.2
- sktime [35] version 0.6.1
- Matplotlib [36] version 3.4.2

Jupyter Notebook [37] and JetBrains PyCharm [38] were used for code editing and execution. Pip [39] and Anaconda [40] tools were used for dependency management and Git [41] was used for the version control.

For the development and testing of the project ASUS N551VW laptop running Windows 10 Pro was used. Its specifications are listed below:

- CPU: Intel Core i7-6700HQ 2.6 GHz 4 cores
- RAM: 16 GB DDR4
- SSD: Samsung 850 PRO 512GB

## 3.2 Business Understanding

During the business understanding stage the business needs, problem and background are analyzed. In this phase the existing company processes and systems are reviewed and their drawbacks understood. In this project the business understanding stage has been done by doing interviews with the company’s employees by asking questions about the problem and its background. Later, the existing company’s systems and data has been analyzed. The exact questions asked in the interviews, or the methods used for system analysis are not mentioned, since there are out of scope of this thesis. Chapter 1 and 2 provide the relevant information which has been found out during this phase and they can be considered as the results of the completed business understanding stage of CRISP-DM process.

## 3.3 Data Understanding

During the business understanding phase it was found out that the company has the following data relevant for this project: the history of customer orders, the history of supplier orders and the history of inventory changes.

Every customer order is received from the ecommerce system in a form of automatically generated email to the specific mailbox. The order information is stored in the attached text file in CSV format [42]. In the current company's system when a new email arrives, the CSV file is automatically imported into the internal company's Microsoft SQL database for further processing. The internal database also stores the history of the stock quantity changes for each item and supplier orders history. The data from the internal database was exported and used for the project. The data has more than 360K rows and 28 columns. However, not all data columns are relevant for the project. The relevant values and their types are described in Table 1.

**Table 1: Customer orders and inventory changes data types use in the project**

| **Name** | **Type** | **Description** |
|---|---|---|
| *sku* | string | Stock Keeping Unit – the item identifier |
| *customer_number* | string | Customer identifier |
| *price* | decimal | Unit price |
| *quantity* | decimal | Ordered quantity |
| *date* | date | Date and time when the order was placed |
| *stock* | decimal | Stock left in the warehouse after executing this order |
| *stock_yesterday* | decimal | Stock which was available yesterday |
| *stock_supplier* | decimal | Stock at the supplier's warehouse (including reserved) |
| *stock_supplier_available* | decimal | Stock at the supplier's warehouse available for purchase |

The exploratory data analysis [43] are used in order to understand the data, its structure and patterns. The primary aim with exploratory analysis is to examine the data for distribution, outliers, and anomalies to direct specific testing of your hypothesis. It also provides tools for understanding the data usually through graphical representation [43]. That knowledge helps to decide on which machine learning algorithms to use for the project and their parameters. It also allows to identify missing or corrupted data, which helps to decide what exact data preparation steps are needed.

The exploratory data analysis is performed by drawing plots and charts which allow to understand the data from the different perspectives and identify its patterns. The following subsections present the data distribution from the various angles.

### 3.3.1 Orders distribution throughout the year

By counting the number of orders made in each day and plotting the data throughout the whole period (Figure 7), no correlation between different months or seasons are found. There is an overall increase in orders, which happens due to the company expansion. The plot illustrates the usual drops, which happen during the weekends or national holidays.

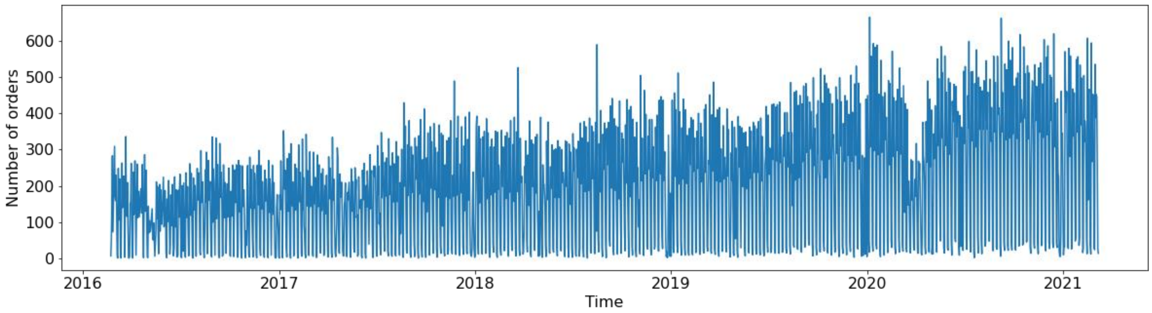

Figure 7: Order distribution throughout the history

The same patterns can be observed in the data of year 2020 shown in Figure 8. It is observed that there was a drop of orders in March and April in 2020. This can be explained by the lockdown which happened due to the pandemic of Covid-19.

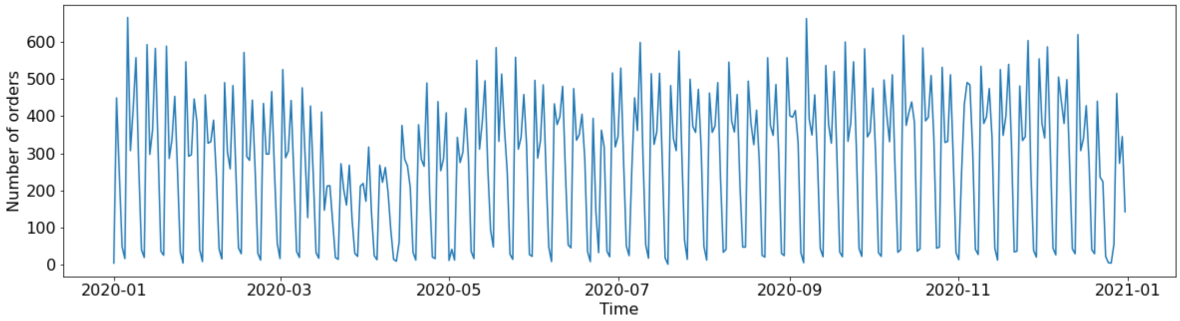

**Figure 8: Order distribution throughout the year 2020**

### 3.3.2 Order distribution depending on the weekday

Order distribution by the weekday is illustrated in Figure 9. A high decrease in orders can be seen during the weekend – Sunday has ~27 times less orders compared to Monday. The difference is also visible throughout the workdays Monday and Thursday have more orders compared to other workdays. This can be explained by the fact that on these days the products are shipped from the supplier. As a result, customers do more orders on these days to receive the goods as fast as possible.

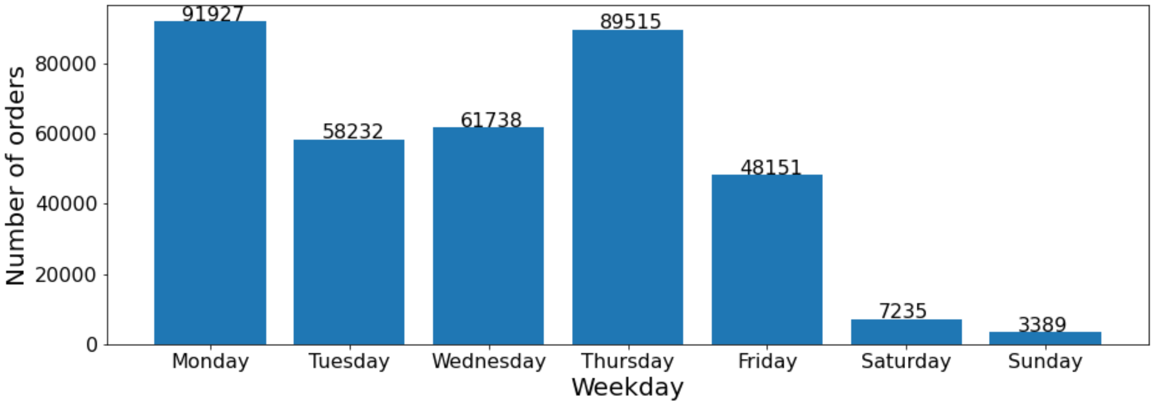

**Figure 9: Order distribution depending on the weekday**

### 3.3.3 Orders distribution between items

Figure 10 plots the distribution of number of items having certain amount of orders on the logarithmic scale. From the plot it can be observed that there is a high number of items which have a low number of orders and a low number of items which have a lot of orders. The number of items having certain amount of orders decreases in a double exponential way.

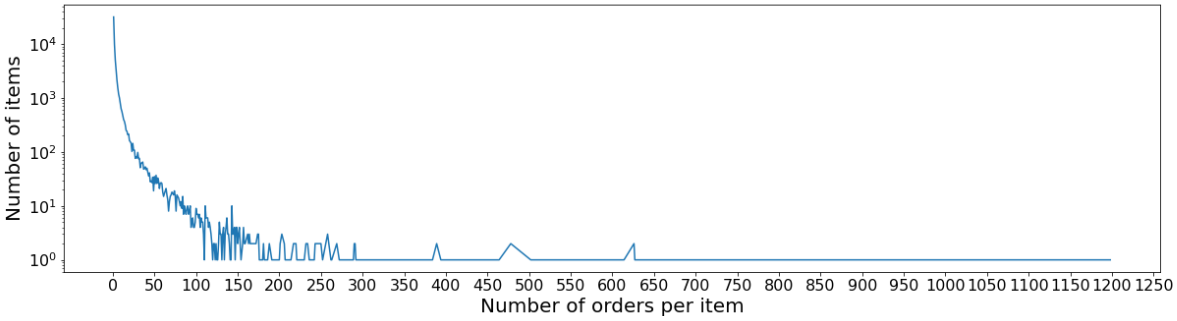

**Figure 10: Distribution of number of items having certain amount of orders**

Such behavior can be seen in Figure 11 more clearly. In the figure all the items are divided in the groups depending on the number of orders they have. Around half of items have a single order, around a quarter of the items have 2 or 3 orders and so on. Even the group size is increased exponentially, the number of items in the group decreases exponentially. Thus, the distribution of orders between items is double exponential.

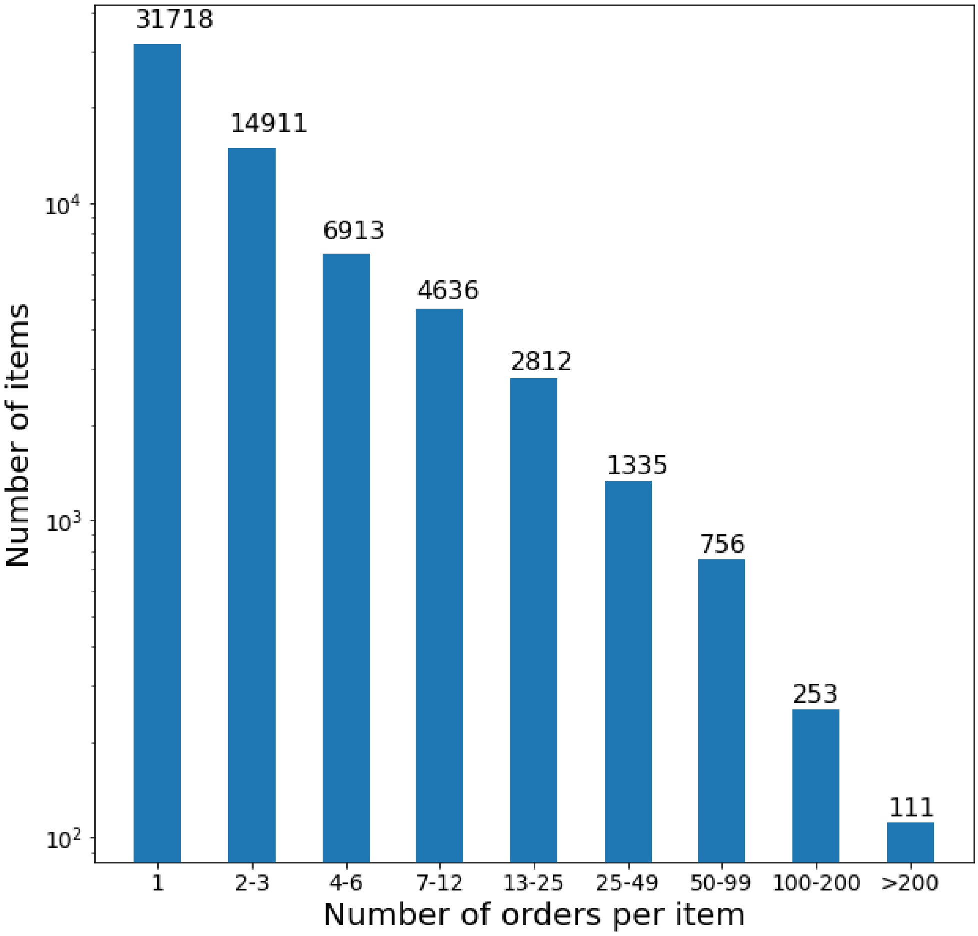

**Figure 11: Distribution of number of items between item groups having certain amount of orders**

However, as Figure 12 illustrates, the number of total orders between the items which have certain amount of orders first decreases in the double exponential way and later changes to the linear increase. This is due to the fact that usually there is just a single item which has a certain high number of orders. This is also illustrated in Figure 13. If compared to Figure 11, it shows more stable distribution.

To conclude, most of the items have a low number of orders, however most of the orders come from the small amount of items. For example, around 26% of all items have 4 or more orders and these items produce 81% of all orders. This applies for a widely known Pareto principle [44] or 80/20 rule which is observed in many real world scenarios.

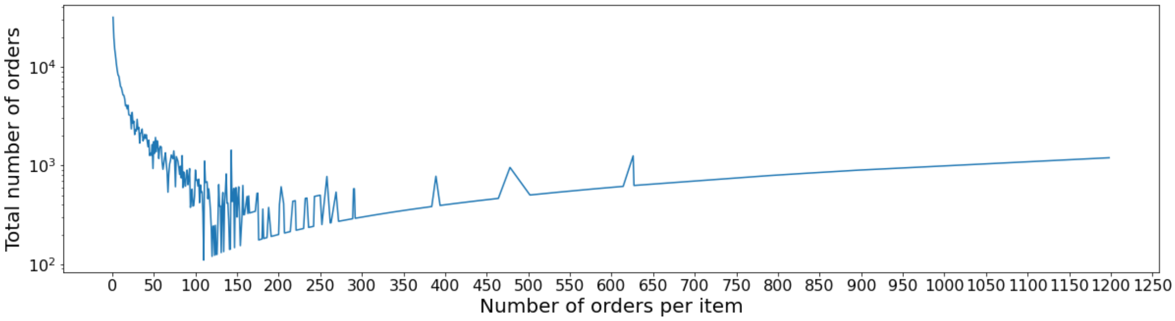

**Figure 12: Distribution of number of orders between items having certain amount of orders**

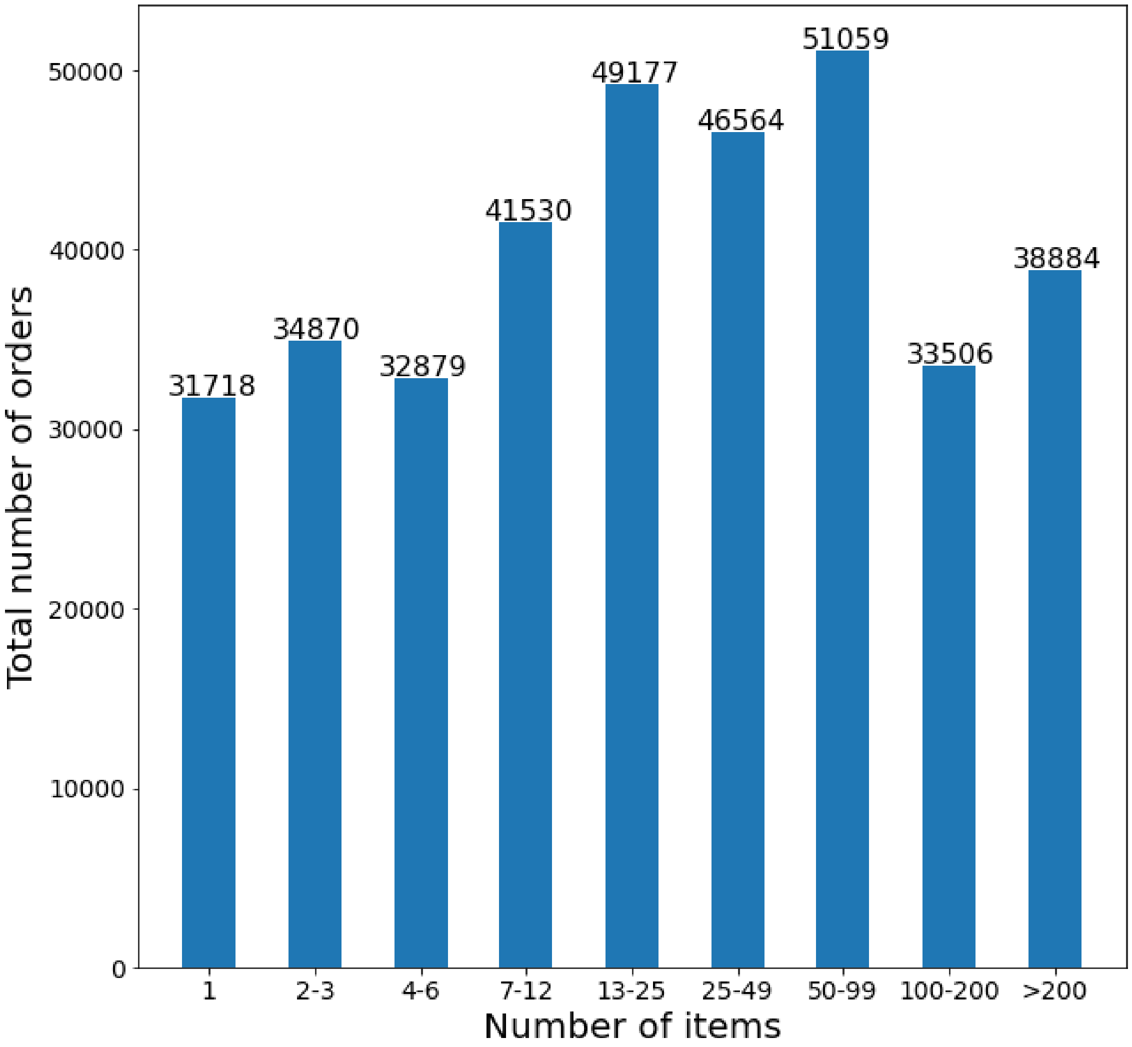

**Figure 13: Distribution of number of orders between item groups having certain amount of orders**

### 3.3.4 Distribution of orders between customers

Figure 14 represents the distribution of orders between customers. There are many customers having a single or a couple of orders, but this number quickly decreases by increasing the number of orders per customer and remains almost stable.

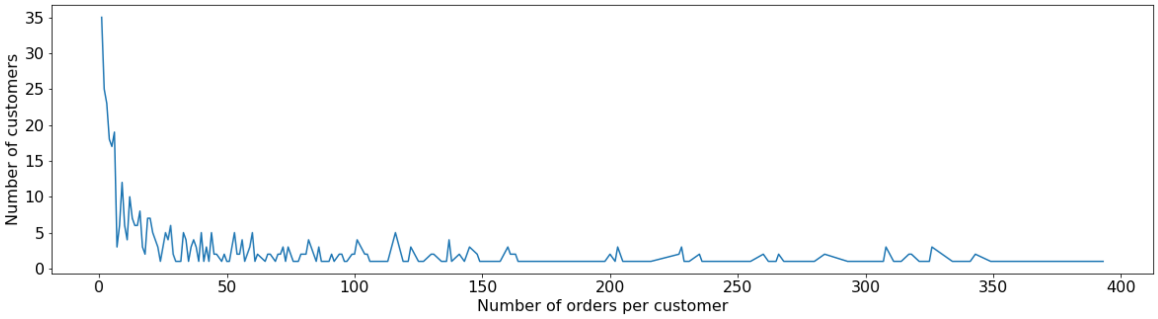

**Figure 14: Distribution of number of customers having certain amount of orders (truncated to 400)**

This behavior can also be observed in Figure 15 where the customers are divided into groups depending on the number of orders made. Even though the group sizes change exponentially, the number of customers per group varies a little.

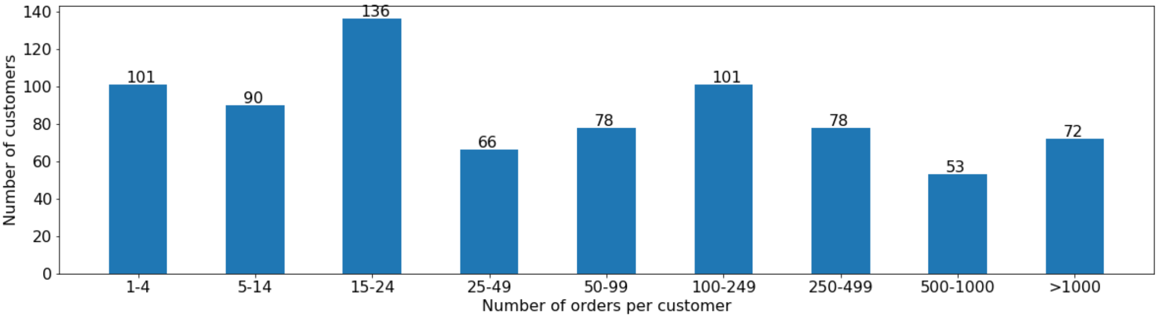

**Figure 15: Distribution of number of customer between customers' groups having certain amount of orders**

However, the total number of orders made by each group increases exponentially. This is illustrated in Figure 16. If the customers having more than 500 orders are taken, Pareto principle [44] can be applied since around 16% of customers produce 85% of orders.

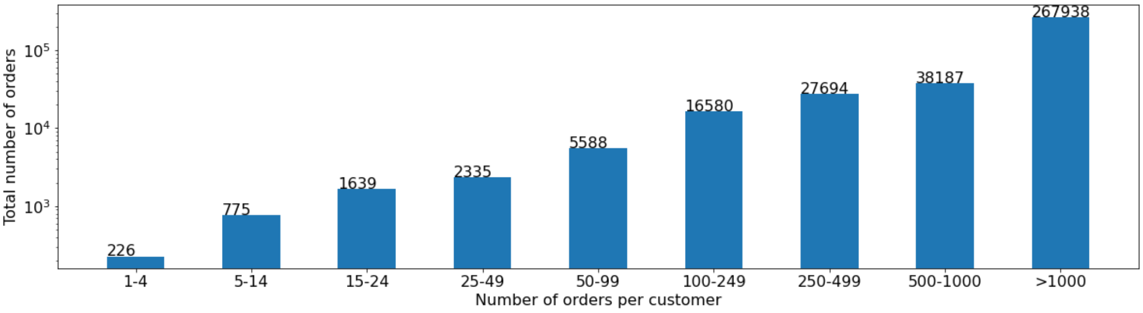

**Figure 16: Distribution of number of orders between customers' groups having certain amount of orders**

## 3.4 Data Preparation

Data preparation is divided into three main activities: missing values cleaning, feature engineering and data normalization.

### 3.4.1 Missing values cleaning

Missing values cleaning depending on the data type is done using one of two methods: deletion or previous data imputation.

Deletion method is used for the data where restoring corrupted information is not possible. Using the deletion method, the entire data row is deleted if it has empty values in one of the following columns: *sku, customer_number, price, quantity, date.* In total 625 rows were deleted.

Previous data imputation method is used to insert the previous known value. Previous value imputation was used for columns: *stock, stock_yesterday, stock_supplier, stock_supplier_available.* 233K previous values were inserted for *stock* column and 274K for *stock_yesterday* column. *stock_supplier* and *stock_supplier_available* columns have no missing values.

### 3.4.2 Feature engineering

Feature engineering is the science of extracting more information from the existing data. Feature engineering itself can be divided in 2 steps: variable transformation and variable (feature) creation. These two techniques are vital in data exploration and have a remarkable impact on the power of prediction [45].

#### 3.4.2.1 Feature creation

Variable (feature) creation is a process to generate a new variable (features) based on the existing variable(s). For example, from the data variable new variables day, month, year, week, weekday can be generated that may have better relationship with the target variable [45].

Feature creation is done for *date* variable. Year, month, day, day of the year, week of the year, weekday, quarter, month end and start indicators are extracted as separate variables. After that the categorical variables year, month, weekday, quarter are converted to indicators to the separate columns. All the orders are grouped by the product and additional variables *total_ordered_quantity, number_of_orders, unique_customers* are generated for each item. Also, for each item and day in the observed history, the perfect Min and Max stock levels are calculated. Min level is taken as the sum of ordered quantity in the following 4 days and Max level as the sum of ordered quantity in 90 days.

#### 3.4.2.2 Variable transformation

Transformation refers to the replacement of a variable by a function. For instance, replacing a variable x by the square root or logarithm x is a transformation [45].

As of now the data structure consists of 5-year time series data of each item. However, such data structure cannot be used for machine learning because every data point in the time series represents only a single day. This is because all the machine algorithms selected for this project (except ARIMA and Prophet) consider each data point separately and cannot build historical relationship with the previously provided data. As a workaround to such problem, additional columns are added to every data point with the data of the previous 90 days. In such way the algorithms can build the relation with the historical data since it is provided in the same data row. Additionally, data transformation is done by calculating difference, sum, mean, median, standard deviation, variance of each attribute in the interval range 2-90 days and adding them as additional columns. The final dataset contains 6992 columns for every item and day.

Finally, data normalization is done which attempts to give all attributes an equal weight. Normalization is particularly useful for classification algorithms involving neural networks or distance measurements such as nearest-neighbor classification and clustering. Data normalization is done by using MinMax scaling method. It performs a linear transformation on the original data so that the range of features scales to the range of [0, 1] [46].

## 3.5 Modeling

The inventory management model will be based on Min/Max inventory management method described earlier in the thesis. This method was selected because it is currently used in the company, and it fits the company needs quite well. In addition, using Min/Max model makes it possible to integrate the project with the current company system to be easy to use and understand by the company's employees.

The interpretation of Min/Max inventory levels would be the following:

- If the current stock of the item is below Min level, the supplier order needs to be executed to fill the stock up to the Max level.
- If the predicted Min level is more than 0 and the item is not kept in stock, the item should be put in stock.
- If the predicted Max level is 0 and the item is in stock, the item should no longer be kept in stock.

As described earlier in the thesis, the biggest challenge for Min/Max method is calculating Min and Max levels. Currently, in the company these levels are static and changed manually in rare occasions. Such system is inefficient since it does not follow trends in the customer demand. The improvement proposed by the thesis is to use the dynamic Min/Max levels which are predicted based on the historical data using the machine learning algorithms.

Having the historical data makes it possible to calculate the most optimal Min and Max levels on the certain day based on the orders which happened in the following days. The calculated levels are provided to the machine algorithms as target values for prediction. The other data generated in the data preparation phase is provided as inputs.

It was decided to build a different prediction model for every product. Such decision was made because building a single model which fits all products without sacrificing performance is much more difficult task since every product follows different sale trends. The performance of machine learning models deeply relates to the number of data points used for learning. Therefore, building a model for items with the low number of orders would result in a low model performance. Additionally, machine learning is a computationally expensive task, so building and testing the models for high number of items would take very long time. As a consequence, only the items having more than 200 orders are used for building and evaluating the models. As observed in Data understanding section, there are 111 of these items and they produce around 10% of all orders.

Min and Max stock levels prediction are two different tasks, and the performance of the specific algorithm might vary depending on whether it is a Min or Max level prediction. Thus, all the algorithms are separately tested for predicting both Min and Max level.

During the project Support Vector Regression, Random Forest, Artificial Neural Network, k-nearest neighbors, Prophet and ARIMA algorithms are tested. All the algorithms are tested with the initial hyperparameters. The performance of the algorithms are calculated by using cross validation and symmetric mean absolute percentage error metric [47].

The models for every item are evaluated by using 3 step cross-validation on a rolling basis. The cross-validation works as follows: start with a small subset of data for training purpose, forecast for the later data points and then checking the accuracy for the forecasted data points. The same forecasted data points are then included as part of the next training dataset and subsequent data points are forecasted [48]. The process of cross-validation data splits in each step is visualized in Figure 17.

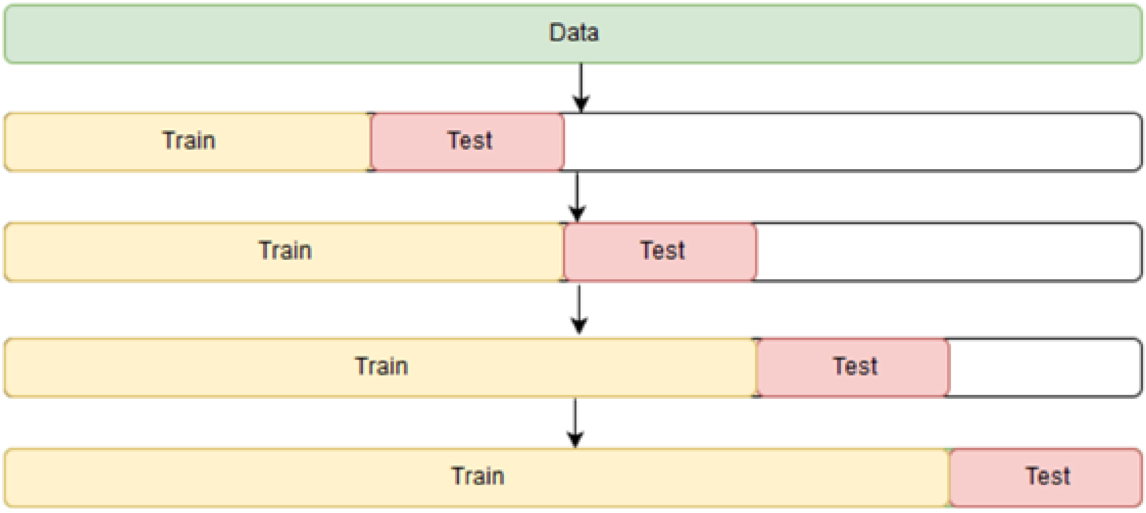

**Figure 17: Time series data cross-validation [48]**

The metric used for model evaluation is symmetric mean absolute percentage error (sMAPE). One of the downsides of this metric is that the error increases tremendously if the true value is zero. To overcome this limitation the true scaled Min and Max values are increased by 1 to be in the range [1, 2].

The average Min/Max stock levels prediction errors of items having more than 200 orders are presented in Table 2. Figure 18 and Figure 19 present error ranges between different runs.

**Table 2: Average Min/Max stock levels prediction errors of each algorithm**

| **Algorithm** | **Symmetric mean absolute percentage error, %** | |
|---|---|---|
| | **Min level prediction** | **Max level prediction** |
| Support Vector Regression (SVR) | 9.0663 | 15.5678 |
| k-nearest neighbors (KNN) | **7.0079** | 15.1890 |
| Random Forest (RF) | 10.1979 | **15.0303** |

| | | |
|---|---|---|
| Artificial Neural Network (ANN) | 22.1115 | 27.6492 |
| ARIMA | 8.0817 | 15.6104 |
| Prophet | 8.4024 | 25.1340 |

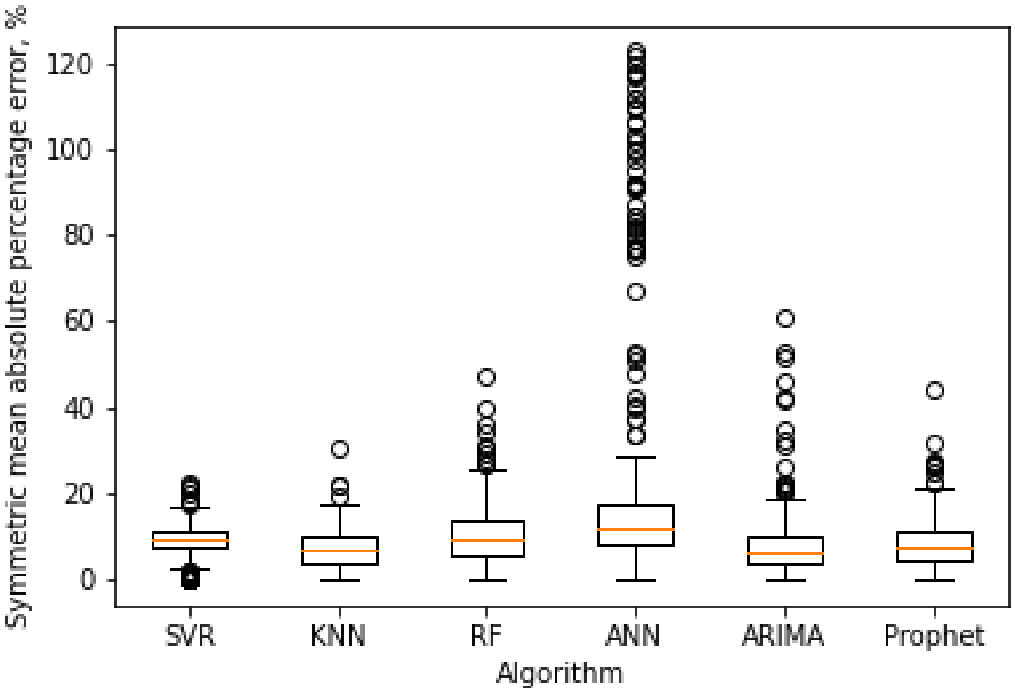

**Figure 18: Min level prediction errors of each algorithm**

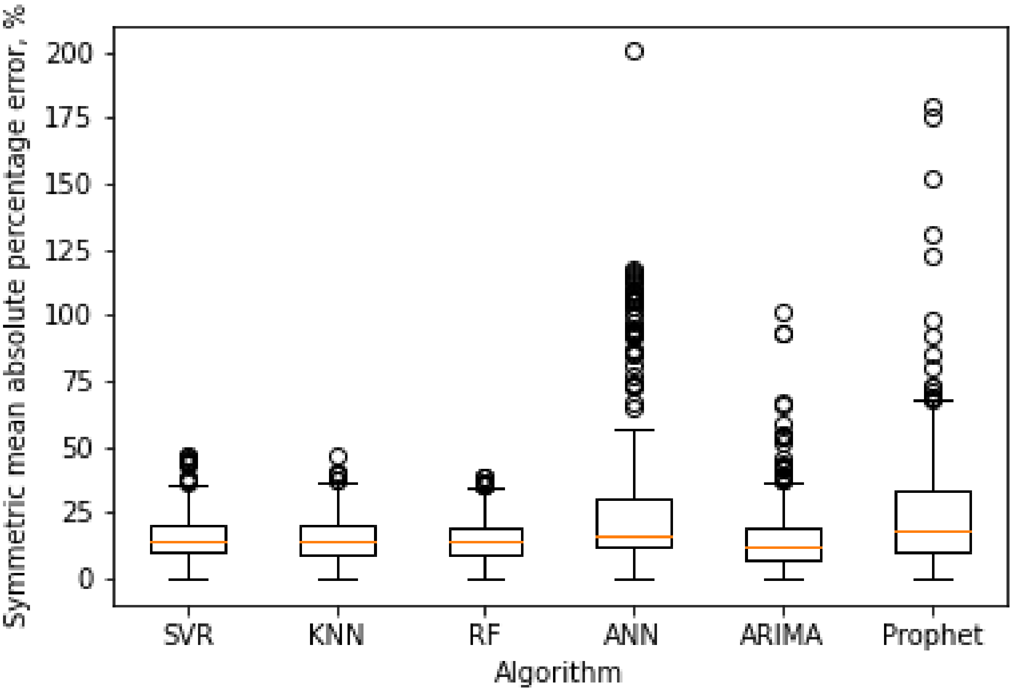

**Figure 19: Max level prediction errors of each algorithm**

The best performing algorithm for Min level prediction is k-nearest neighbors with the average sMAPE around 7%. The best results for Max level prediction achieved Random Forest algorithm with sMAPE around 15%.

### 3.5.1 Hyperparameter optimization

Hyperparameters are used to configure various aspects of the learning algorithm and can have wildly varying effects on the resulting model and its performance. Hyperparameter search is commonly performed manually, via rules-of-thumb or by testing sets of hyperparameters on a predefined grid [49]. Hyperparameter for this project optimization was done for k-nearest neighbors and Random Forest algorithms using grid search method. Grid search method brute forces all the combinations of hyperparameters in the given range and finds the best performing parameters.

Tested values for each algorithm are shown in Table 3 and Table 4. Every hyperparameter of k-nearest neighbors algorithm is explained in more detail in [50] and hyperparameters of Random Forest algorithm are explained in [51].

**Table 3: Tested hyperparameters of k-nearest neighbors algorithm**

| **Parameter** | **Tested values** |
|---|---|
| *n_neighbors* | 3, **5**, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51 |

| | |
|---|---|
| *weights* | Uniform, **distance** |
| *p* | **1**, 2 |

Table 4: Tested hyperparameters of Random Forest algorithm

| **Parameter** | **Tested values** |
|---|---|
| *n_estimators* | **10**, 20, 40, 75, 100, 200, 400, 500, 1000 |
| *criterion* | **mse**, mae |
| *max_features* | auto, **sqrt**, log2 |

The pipeline of data preparation, model creation and cross-validation for a single item takes around 16 seconds for k-nearest neighbors algorithm and around 17 seconds for Random Forest algorithm. From Table 3 and Table 4 it can be seen that there are 100 (25 x 2 x 2) hyperparameters' combinations for k-nearest neighbors algorithm and 60 (10 x 2 x 3) combinations for Random Forest algorithm. As a result, completing hyperparameter optimization for 111 products would take around 111 x 100 x 16 = 177 600 seconds ~ 49 hours for k-nearest neighbors algorithm and around 111 x 60 x 17 = 113 220 seconds ~ 31 hours for Random Forest algorithm. As a workaround to such computationally complex problem, hyperparameter optimization was done for a single product rather than all. The optimization for a single item takes around 100 x 16 = 1600 seconds ~ 27 minutes for k-nearest neighbors algorithm and 60 x 17 = 1020 seconds = 17 minutes for Random Forest algorithm. The item used for hyperparameter optimization was selected as a median item in the list of products which have more than 200 orders sorted by the total number of orders. Optimizing the algorithms for the median item in terms of the number of orders should result in the optimal performance both for items which have high and low number of orders.

The best performing values found during hyperparameter optimization are marked in bold in Table 3 and Table 4. Performance optimization of k-nearest neighbors algorithm improved performance of Min stock level prediction for the median item from 5.8809% to 5.6970% sMAPE. The average sMAPE of Min stock level prediction for all items which have more than 200 orders decreased from 7.0079% to 6.8730%. The optimization of Random Forest algorithm improved the performance of Max stock level prediction for the median item from 13.5160% to 10.9788% sMAPE. The average sMAPE of Max stock level prediction for all items which have more than 200 orders decreased from 15.0303% to 14.6813%.

# 4 Results and Analysis

This chapter describes the evaluation methods and techniques used in the project as well as results. The chapter starts with the description of the simulation. Later execution time and reliability analysis are presented. The chapter ends with the discussion of the achieved results.

## 4.1 Simulation

Evaluation is done by simulating a real usage of the algorithm throughout the year of 2020 for items which have more than 200 orders. The simulation goes through every Tuesday and Thursday (the days the company executes supplier orders), takes the known customer order history until that day and creates two prediction models for every item: one for Min stock level prediction and one for Max stock level prediction. After that, the models predict the next Min and Max stock level values. If the current stock is below Min level, a supplier order is created to fill the stock up to the Max level. The simulation algorithm is presented in Figure 20.

| | Current system | Proposed algorithm | Savings per year |
|---|---|---|---|
| Supplier orders for items with more than 200 orders | 1150 | 738 | 2060€ |
| Supplier orders for items with more than 50 orders | 32K | 20K | 60K € |
| Backorders for items with more than 200 orders | 769 | 390 | 24% increase in sales of the items |
| CO2 emissions per year | 393 tons | 262 tons | 131 tons ~ 131 trees planted |

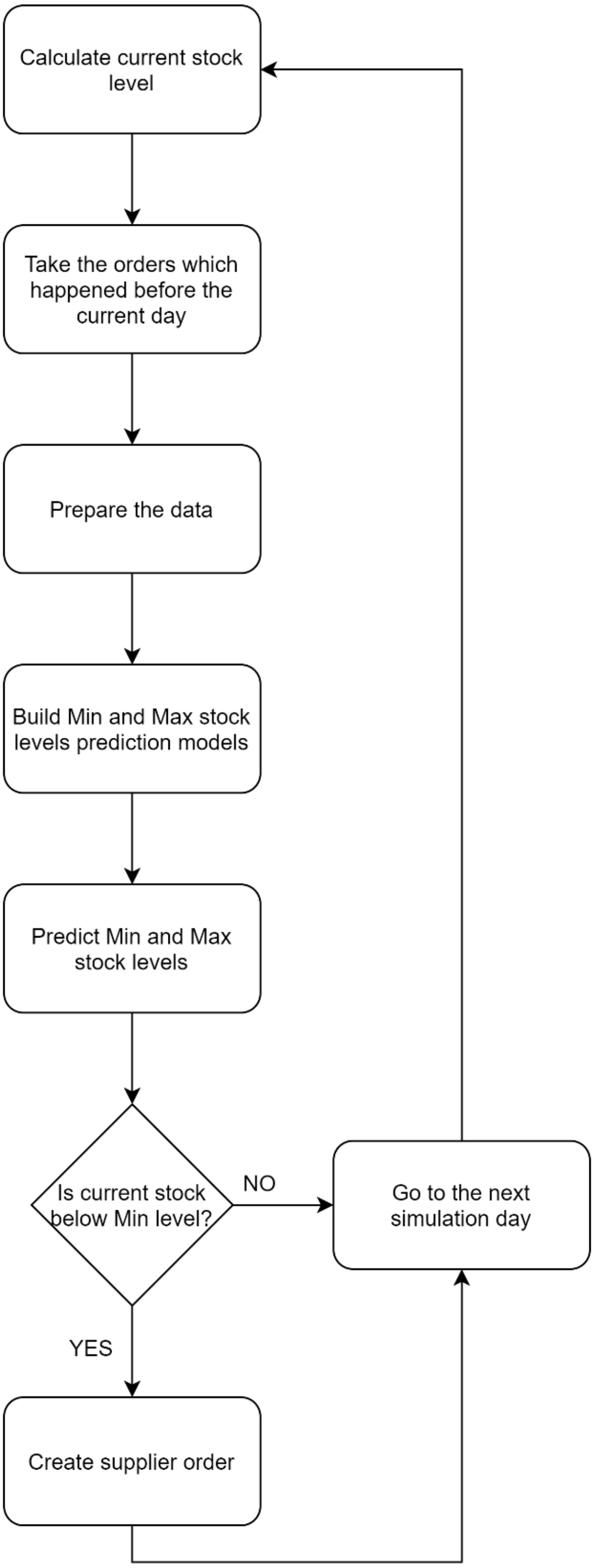
Calculate current stock level
Take the orders which happened before the current day
Prepare the data
Build Min and Max stock levels prediction models
Predict Min and Max stock levels
Is current stock below Min level?
NO
Go to the next simulation day
YES
Create supplier order

**Figure 20: Simulation algorithm**

During the simulation all the supplier orders and stock level changes for each item are saved for analysis. The gathered data is used to calculate the performance metrics.

### 4.1.1 Performance metrics

To prove that the proposed algorithm performs better than the current company's system the following evaluation metrics are defined based on which the comparison is done:

- the number of backorders in the given period (the lower, the better)
- the total number of supplier orders in the given period (the lower, the better)
- inventory turnover in the given period (the shorter, the better)

These performance metrics represent the business expenses – backorders (orders which cannot be executed immediately due to the shortage in stock) reduce the customer satisfaction; the supplier orders cause additional fees; the long inventory turnover increases storage expenses. The selected metrics are dependent on each other. It is easy to optimize the results for a single metric, but it is very hard to do that for all at once. For example, to have a low number of backorders the large number of products can be kept in the inventory. However, that would result in very long inventory turnover. To change that, the low number of products can be kept in the warehouse and ordered from the supplier more frequently, but that would increase the number of supplier orders and fees paid to the supplier and so on.

The metrics of the current system are calculated from the historical supplier orders and stock data. The metrics of the suggested algorithm are calculated from the stored simulation data of supplier orders and stock changes.

The simulation treats all the items equally not depending on their price or value. Therefore, the value of the products is not included in the inventory turnover calculation, and it is considered that all the items have equal value of 1€. Due to this simplification the inventory turnover of one year period is calculated by the following formula:

$$Inventory\ turover = \frac{sum\ of\ items'\ average\ inventory}{total\ number\ of\ items\ sold} * 365$$

Backorders are calculated by counting the number of orders which were not fully fulfilled due to the shortage in stock.

### 4.1.2 Simulation results

The first simulation was run with the model built from data with the perfect Max stock level calculated as sum of sold quantity in the next 90 days. However, such Max level calculation resulted in the average inventory turnover of 47.1 days. It is way larger compared to the current company's inventory turnover of the items having more than 200 orders which is 33.2 days. Since all the metrics are dependent on each other such large difference in inventory turnover cannot prove that the proposed algorithm improves the current solution because it is unknown how much supplier orders and backorders would decrease if inventory turnover were increased in the current system. Furthermore, increasing inventory turnover in practice is a difficult task because it means the company needs to store more goods for the longer period of time and that might require warehouse expansion.

To prove that the algorithm improves the current company system, inventory turnover should match the current inventory turnover. To do that the second simulation was run with the model built from the data with the perfect Max stock level calculated as sum of sold quantity in the next 60 days. This resulted in inventory turnover of 33.7 days which differs from the current system only by 1.5%. However, the supplier orders decreased by around 36% from 1150 to 738. The number of backorders decreased by around 49% from 769 to 390. The results of the simulations are also presented in Table 5.

**Table 5: Simulation metrics compared to the current system**

| **Metric** | **Current system** | **Proposed algorithm (max level - 90 days sales)** | **Proposed algorithm (max level - 60 days sales)** |
|---|---|---|---|
| Inventory turnover | 33.2 days | 47.1 days (+41.86%) | 33.7 days (+1,5%) |
| Supplier orders | 1150 | 557 (-51,57%) | 738 (-35,83%) |
| Backorders | 769 | 270 (-64,89%) | 390 (-49,29%) |

## 4.2 Execution Time Analysis

To predict the next Min and Max stock levels for each item the algorithm takes the following steps:

- Historical data preparation.
- Building Min level prediction model using k-nearest neighbors algorithm.

- Building Max level prediction model using Random Forest algorithm.
- Predicting Min stock level.
- Predicting Max stock level.

The average execution time of each step for a single item is displayed in Table 6.

**Table 6: Execution time of each algorithm step**

| **Step** | **Execution time (milliseconds)** |
|---|---|
| Historical data preparation | 12 210 |
| Building Min level prediction model | 45 |
| Building Max level prediction model | 155 |
| Predicting Min stock level | 157 |
| Predicting Max stock level | 44 |
| **Total** | **12 611** |

The data shows that the longest step is historical data preparation which takes more than 12 seconds. All the other steps (building the models and predicting Min and Max stock levels) combined take around 400 milliseconds. In total the execution of the algorithm for a single item takes around 12.6 seconds. Running a simulation for 111 items for the period of one year took around 40 hours.

Data preparation takes relatively long time because of not optimized feature creation operations. This shows a lot of room for improvement because code optimization could be done to reduce the computational complexity of this step. Other steps of the algorithm are harder to optimize. This is because the project uses the popular open-source implementations of the machine learning algorithms which are already optimized.

## 4.3 Reliability Analysis

Reliability analysis of the algorithm is done by testing if the algorithm manages to find patterns for the items with increasing, decreasing, or fluctuating demand. To do that, 3 different items are tested representing each case.

### 4.3.1 Increasing demand

Figure 21 presents the predicted Min level (presented as solid blue line) and the perfect Min level (presented as dashed orange line) of the item with the increasing demand throughout the year of 2020. Even though the data is noisy, the algorithm still manages correctly predict some dumps in the demand.

Figure 22 compares the predicted and the perfect Max levels of the same item throughout the year of 2020. It can be easily recognized that the demand of the item is increasing, and the algorithm correctly predicts the overall direction.

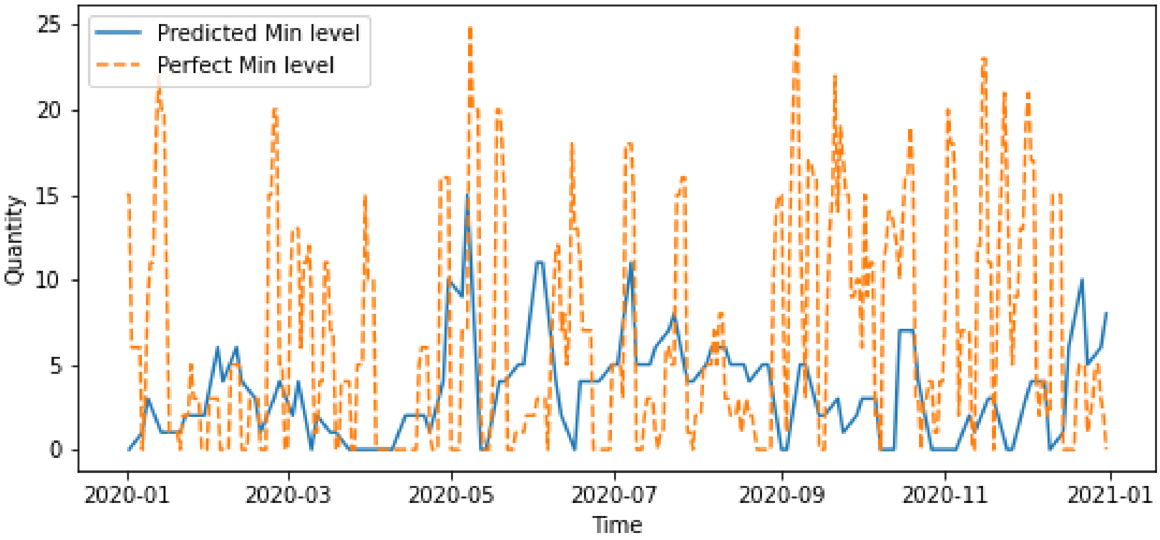

**Figure 21: Min level prediction of the item with the increasing demand**

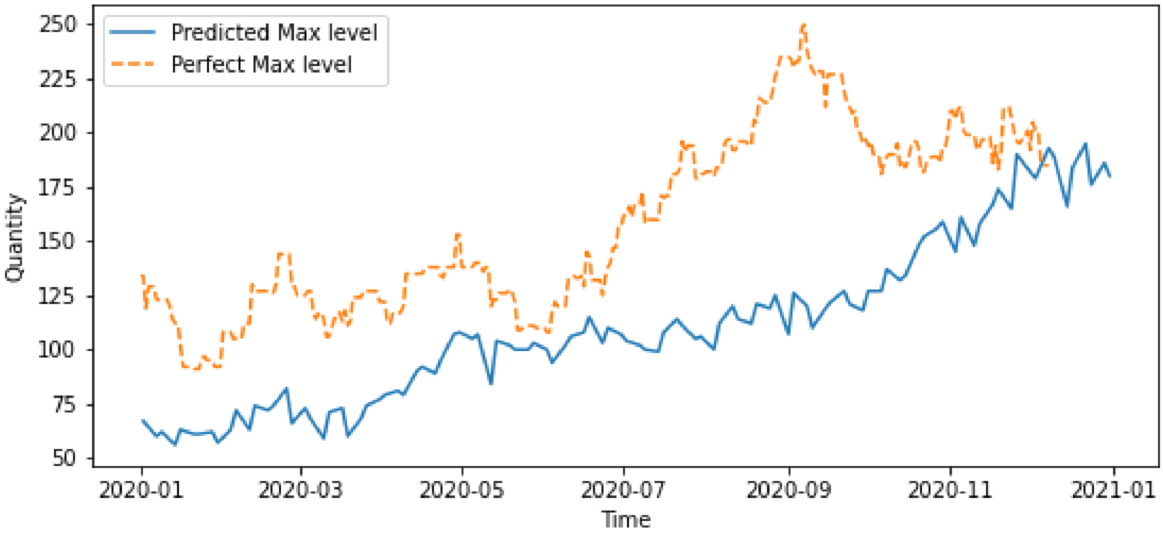

**Figure 22: Max level prediction of the item with the increasing demand**

### 4.3.2 Decreasing demand

Figure 23 displays the perfect and the predicted Min level of the item with the decreasing demand throughout the year of 2020. Even though the item has small number of orders, the algorithm predicted most of the sudden increases.

Figure 24 plots the perfect and predicted Max level of the same item throughout the year of 2020. Despite that the algorithm misses to predict the correct values almost throughout the entire period, the overall direction of decreasing demand is predicted correctly.

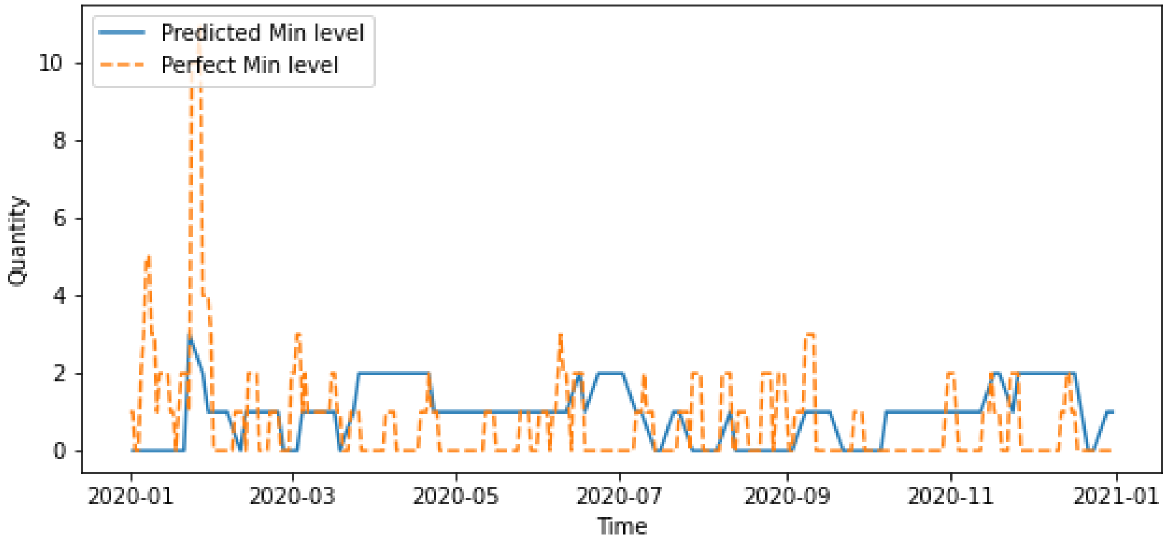

Figure 23: Min level prediction of the item with the decreasing demand

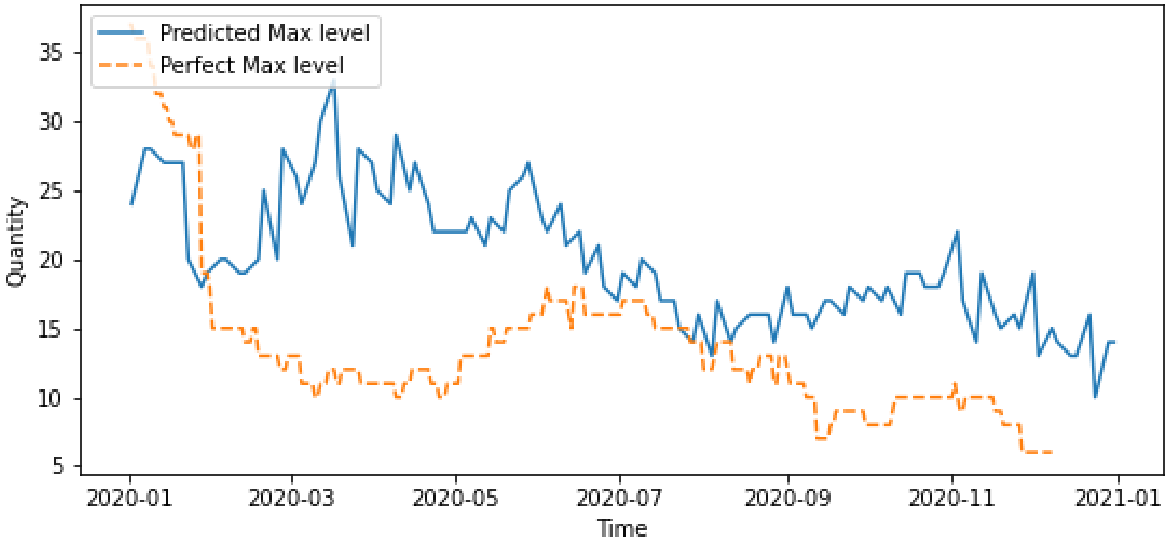

Figure 24: Max level prediction of the item with the decreasing demand

### 4.3.3 Fluctuating demand

Figure 25 shows the perfect and the predicted Min level of the item with the fluctuating demand. Most of the sudden increases predicted by the algorithm are either too early or too late. However, the predicted values are in the same range as the real values.

The perfect and predicted Max levels of the same item are presented in Figure 26. It can be seen that the algorithm with the small relative error correctly predicted most of the demand increases and decreases.

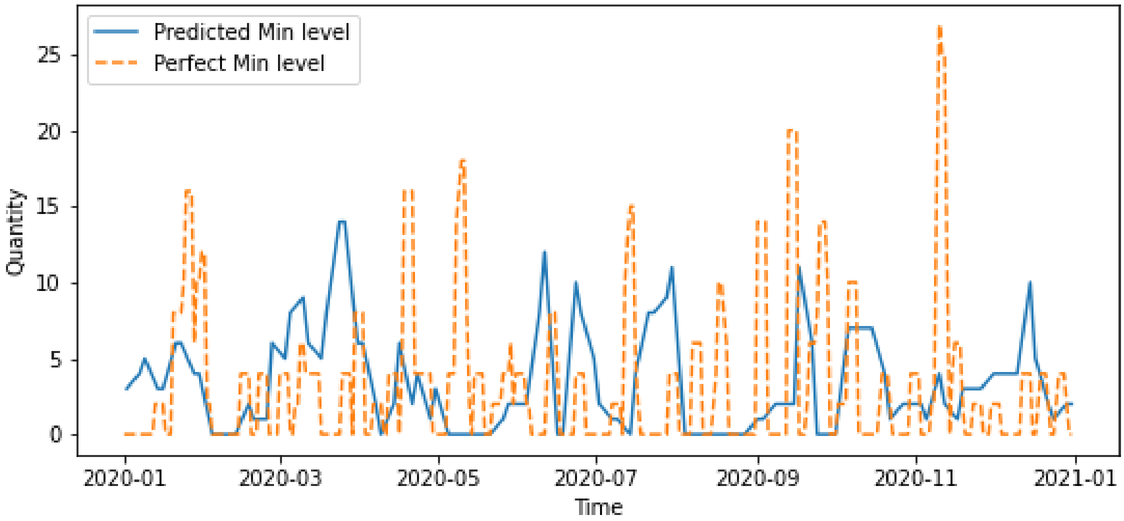

**Figure 25: Min level prediction of the item with the fluctuating demand**

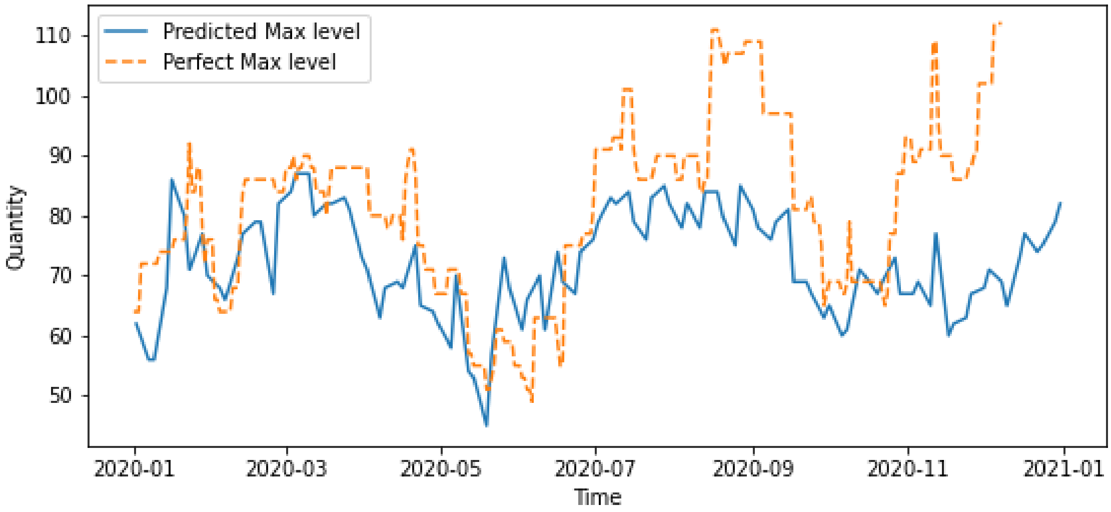

**Figure 26: Max level prediction of the item with the fluctuating demand**

## 4.4 Discussion

Min and Max level prediction using machine learning algorithms improved the current company's system. The simulation showed that the algorithm lowered the number of supplier orders by around 36% (from 1150 to 738) and the number of backorders by around 49% (from 769 to 390) with the minimal increase in inventory turnover. If the company had used the algorithm throughout the year of 2020, it would have saved (1150 – 738) x 5€ = 2060€ on supplier order line fees considering the supplier order line fee of 5€.

However, the simulation was run only for the top-selling 111 items which have more than 200 orders and receives around 10% of all customer orders. Such limitation was done due to the computational complexity of the algorithm and simulation – simulating a single item through a single day takes around 12 seconds. As a consequence, simulating 1120 items having more than 50 orders through the period of one year would take around 16 days. Nevertheless, if the algorithm produces the same results for the items which have more than 50 orders, the saving on supplier fee could be much higher. Throughout 2020 the company made around 32K supplier orders of items which have more than 50 customer orders in total. If the same improvement is made by the algorithm as it is for items which have more than 200 orders, the supplier orders could have been reduced by around 12K, which would have resulted in around 60K Euros of annual savings.

As much as the lower number of supplier orders, the reduced number of backorders is also important for company's income and sales. A study [7] showed that 69% of customers are less likely to shop with a retailer for future purchases if an item is not delivered within two days of the date promised. Knowing that each backorder usually results in a late delivery, the reduction of backorders by 49% across 111 top-selling items should result in increase in sales of such items by 24%.

Moreover, the number of shipments from the supplier might also be reduced. This is because the simulation simulated the delivery of goods from the supplier twice a week and still outperformed the current system where the products are delivered 3 times a week. If additional shipment is eliminated, it would result in huge saving on shipping expenses and would lower carbon emission. A supplier warehouse is around 1400 km away and each roundtrip burns around 1000 liters of fuel and emits around 2520 kg of CO2 [52]. If such weekly shipment is removed, it would result in 131 tons of CO2 saved annually, which is equal to around 131 trees planted annually and not cut for 100 years [53].

# 5 Conclusions and Future work

This chapter presents the conclusions of the project and future work opportunities.

## 5.1 Conclusions

The project addressed an inefficient inventory management problem in the supply chain with the supplier applying order line fees. The problem was divided into subproblems: when the supplier order should be executed, what products and how many units should be ordered and what products should be kept in stock.

To tackle this problem, the experimental research method and CRISP-DM process was chosen. The process fit the project needs well and following the process steps helped to avoid common mistakes and pitfalls.

During the business understanding phase it was found out that the order distribution between items in the dataset is double exponential – the high number of orders come from the small number of items and the majority of items have a small number of orders. Thus, only the data of the items which have more than 200 orders were used for model creation and evaluation.

The data preparation was done by doing feature engineering and data transformation. Even though the step was completed, the number of generated additional features could be reduced and the whole data preparation process optimized.

The inventory management algorithm proposed by the thesis is based on Min/Max inventory management method. The project proposes to use machine learning algorithms for Min and Max stock level prediction. Support Vector Regression, k-nearest neighbors, Random Forest, Artificial Neural Network, ARIMA, and Prophet algorithms were tested both for Min and Max level prediction. It was found out that the best results for Min stock level prediction were achieved by k-nearest neighbors algorithm with the average sMAPE of 7.0079%. The best predictions for Max stock level were done by Random Forest algorithm with the average sMAPE of 15.0303%. After the hyperparameter optimization sMAPE was improved to 6.8730% and 14.6813% accordingly.

The simulation was run to compare the proposed algorithm with the current company's system. It showed that for the items which have more than 200 orders the algorithm decreased the number of supplier orders by 35,83% and the number of backorders by 49,29% while keeping almost the same inventory turnover. If the same results are achieved with all products, it is expected that the company would save around 60K euros per annum on supplier order line fees and the lower number of backorders would increase sales by 24%.

The project outperformed the current company’s system and the purpose as well as all goals were achieved. The system has not been deployed and it is not yet used by the company employees, but it will be done in the near future.

## 5.2 Future work

The project opened a lot of opportunities for the future research. All the future work possibilities are presented in the bullet points below.

- The execution time of the data preparation step could be improved. This would lower the overall computational complexity of the algorithm and would speed up the predictions and testing of the algorithm.
- The algorithm could be tested with the different inventory management methods (e.g., Periodic Review System) instead of Min/Max and the results compared between different solutions.
- Additional machine learning algorithms could be tested. Relevant literature study showed that similar projects achieved great results while using a genetic algorithm. However, due to its complexity, the algorithm was not included in this project.
- Instead of using the same algorithms and hyperparameters for all items, the machine learning algorithm and its hyperparameters could be chosen for every item individually rather. That should increase the overall system performance.
- More item attributes such as value, dimensions, availability at the supplier, shipping or storing conditions could be considered when predicting the supplier orders.
- The algorithm could be improved to react to the supply chain disruptions and avoid the bullwhip effect. The information from the social media or news could be gathered and used by the algorithm to adjust the prediction based on the supply chain disruption events happening around the globe.
- The data available at the company could be merged with the similar open-sourced data. The higher number of datapoints gathered from open data and provided to the algorithm should lead to the overall better performance.
- The system should be deployed and integrated with the current company’s system so that the employees can use it and benefit from it.

## 5.3 Reflections

The project has a large impact for the research field, the company, and the environment.

The research field is highly impacted due to the lack of research of the inventory management algorithms when the supplier applies order line fees. Since there are no found research for such case, this project would be the main guideline for the future researchers solving similar problems. Even if the supplier order line fee is irrelevant, the process and the results of the thesis will help researchers choose or develop the correct machine learning algorithm for the inventory management or customer demand forecasting tasks.

The project a has a large impact for the company ASWO Baltic. The developed algorithm should dramatically reduce company’s expenses on supplier order line fees, reduce shipping time to the final customer, increase sales and lower human resources needed for the inventory and order management. Overall, this would increase company’s income and profits by a huge margin.

Finally, the project impacts the environment by lowering the packaging waste and carbon emissions. This is caused by the reduced number of orders because combining many items in a single order might require less packaging material and fuel spent on shipping.

## References

[1] J. T. Mentzer *et al.*, “DEFINING SUPPLY CHAIN MANAGEMENT,” *J. Bus. Logist.*, vol. 22, no. 2, pp. 1–25, Sep. 2001, doi: 10.1002/j.2158-1592.2001.tb00001.x. [Online]. Available: http://doi.wiley.com/10.1002/j.2158-1592.2001.tb00001.x. [Accessed: 06-May-2021]

[2] “Min/Max Inventory Method.” [Online]. Available: https://www.lokad.com/min-max-inventory-planning-definition. [Accessed: 08-May-2021]

[3] “PERIODIC REVIEW SYSTEM: Inventory Management Models : A Tutorial | SCM | Supply Chain Resource Cooperative (SCRC) | North Carolina State University,” *Supply Chain Resource Cooperative*, 28-Jan-2011. [Online]. Available: https://scm.ncsu.edu/scm-articles/article/periodic-review-system-inventory-management-models-a-tutorial. [Accessed: 08-May-2021]

[4] D. Erlenkotter, “Ford Whitman Harris and the Economic Order Quantity Model,” *Oper. Res.*, vol. 38, no. 6, pp. 937–946, Dec. 1990, doi: 10.1287/opre.38.6.937. [Online]. Available: http://pubsonline.informs.org/doi/abs/10.1287/opre.38.6.937. [Accessed: 08-May-2021]

[5] “ASWO - Lithuania.” [Online]. Available: https://lt.aswo.com/. [Accessed: 08-May-2021]

[6] A. Grubor, N. Milicevic, N. Djokic, A. Grubor, N. Milicevic, and N. Djokic, “The impact of store satisfaction on consumer responses in out-of-stock situations,” *Rev. Bras. Gest. Neg.*, vol. 19, no. 66, pp. 520–537, Dec. 2017, doi: 10.7819/rbgn.v0i0.2436. [Online]. Available: http://www.scielo.br/scielo.php?script=sci_abstract&pid=S1806-48922017000400520&lng=en&nrm=iso&tlng=en. [Accessed: 09-May-2021]

[7] Voxware, “Voxware Research Indicates that Consumers Have Little Patience for Incorrect or Late Deliveries, Especially at the Holidays.” 08-Dec-2014 [Online]. Available: https://www.voxware.com/wp-content/uploads/2016/12/consumer-Survey-December-2014.pdf. [Accessed: 09-May-2021]

[8] A. Håkansson, “Portal of Research Methods and Methodologies for Research Projects and Degree Projects,” *undefined*, 2013 [Online]. Available: /paper/Portal-of-Research-Methods-and-Methodologies-for-H%C3%A5kansson/e591fa1db4a633cd956cf06e204f82cffdc02e3e. [Accessed: 09-May-2021]

[9] R. Wirth and J. Hipp, “CRISP-DM: Towards a standard process model for data mining,” in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, 2000, vol. 1.

[10] A. Azevedo and M. Santos, *KDD, semma and CRISP-DM: A parallel overview*. 2008, p. 185.

[11] “What main methodology are you using for your analytics, data mining, or data science projects? Poll.” [Online]. Available: https://www.kdnuggets.com/polls/2014/analytics-data-mining-data-science-methodology.html. [Accessed: 09-May-2021]

[12] M. Christopher, *Logistics & supply chain management*, 4. ed. Harlow: Financial Times Prentice Hall, 2011.

[13] “Shortage of personal protective equipment endangering health workers worldwide.” [Online]. Available: https://www.who.int/news/item/03-03-2020-shortage-of-personal-protective-equipment-endangering-health-workers-worldwide. [Accessed: 12-May-2021]

[14] “The cost of the Suez Canal blockage,” *BBC News*, 29-Mar-2021 [Online]. Available: https://www.bbc.com/news/business-56559073. [Accessed: 13-May-2021]

[15] S. G. October 2019, “GM strike brings widespread disruption to US supply chain,” *Automotive Logistics*. [Online]. Available: https://www.automotivelogistics.media/gm-strike-brings-widespread-disruption-to-us-supply-chain/39375.article. [Accessed: 13-May-2021]

[16] “The Impact of Winter Weather the Cold Supply Chain,” *Redwood Logistics*, 28-Dec-2018. [Online]. Available: https://www.redwoodlogistics.com/the-impact-of-winter-weather-the-cold-supply-chain/. [Accessed: 13-May-2021]

[17] J. Gerstein, “The Trump administration is revoking the licenses of companies that supply to Huawei, as a final blow to the Chinese tech giant,” *Business Insider*. [Online]. Available: https://www.businessinsider.com/trump-administration-halts-huawei-supply-shipments-2021-1. [Accessed: 13-May-2021]

[18] “Logistics Handbook.pdf.” [Online]. Available: https://ghsupplychain.org/sites/default/files/2019-07/Logistics%20Handbook.pdf. [Accessed: 16-May-2021]

[19] gmiles, “Warehouse Management Basics-Discrete Picking | iCepts,” 03-Aug-2016. [Online]. Available: https://www.icepts.com/warehouse-management-basics-discrete-picking/. [Accessed: 13-May-2021]

[20] “Amazon fight with workers: ‘You’re a cog in the system,’” *BBC News*, 10-Feb-2021 [Online]. Available: https://www.bbc.com/news/business-55927024. [Accessed: 14-May-2021]

[21] J. D. Rey, “How robots are transforming Amazon warehouse jobs — for better and worse,” *Vox*, 11-Dec-2019. [Online]. Available: https://www.vox.com/recode/2019/12/11/20982652/robots-amazon-warehouse-jobs-automation. [Accessed: 14-May-2021]

[22] H. L. Lee, V. Padmanabhan, and S. Whang, “Information Distortion in a Supply Chain: The Bullwhip Effect,” *Manag. Sci.*, vol. 43, no. 4, pp. 546–558, 1997 [Online]. Available: https://www.jstor.org/stable/2634565. [Accessed: 16-May-2021]

[23] W. Kenton, “What Is Lead Time?,” *Investopedia*. [Online]. Available: https://www.investopedia.com/terms/l/leadtime.asp. [Accessed: 16-May-2021]

[24] “Chapter 13: The Bullwhip Effect - Fundamentals of Supply Chain Theory, 2nd Edition [Book].” [Online]. Available: https://www.oreilly.com/library/view/fundamentals-of-supply/9781119024842/c13.xhtml. [Accessed: 16-May-2021]

[25] J. D. Sterman, “Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment,” *Manag. Sci.*, vol. 35, no. 3, pp. 321–339, 1989 [Online]. Available: https://www.jstor.org/stable/2631975. [Accessed: 10-May-2021]

[26] “Computers play the beer game: can artificial agents manage supply chains?,” *Decis. Support Syst.*, vol. 33, no. 3, pp. 323–333, Jul. 2002, doi: 10.1016/S0167-9236(02)00019-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923602000192. [Accessed: 16-May-2021]

[27] R. Carbonneau, K. Laframboise, and R. Vahidov, “Application of machine learning techniques for supply chain demand forecasting,” *Eur. J. Oper. Res.*, vol. 184, no. 3, pp. 1140–1154, Feb. 2008, doi: 10.1016/j.ejor.2006.12.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221706012057. [Accessed: 18-May-2021]

[28] S. R. Singh and T. Kumar, “Inventory Optimization in Efficient Supply Chain Management,” May 2012.

[29] A. Kochak and S. Sharma, “Demand forecasting using neural network for supply chain management,” *Int. J. Mech. Eng. Robot. Res.*, vol. 4, no. 1, pp. 96–104, 2015.

[30] A. Lasek, N. Cercone, and J. Saunders, “Restaurant Sales and Customer Demand Forecasting: Literature Survey and Categorization of Methods,” in *Smart City 360°*, Cham, 2016, pp. 479–491, doi: 10.1007/978-3-319-33681-7_40.

[31] “Welcome to Python.org,” *Python.org*. [Online]. Available: https://www.python.org/. [Accessed: 23-May-2021]

[32] “pandas - Python Data Analysis Library.” [Online]. Available: https://pandas.pydata.org/. [Accessed: 23-May-2021]

[33] “NumPy.” [Online]. Available: https://numpy.org/. [Accessed: 23-May-2021]

[34] “scikit-learn: machine learning in Python — scikit-learn 0.24.2 documentation.” [Online]. Available: https://scikit-learn.org/stable/. [Accessed: 23-May-2021]

[35] M. Löning *et al.*, *alan-turing-institute/sktime: v0.6.1*. Zenodo, 2021 [Online]. Available: https://zenodo.org/record/3749000. [Accessed: 23-May-2021]

[36] “Matplotlib: Python plotting — Matplotlib 3.4.2 documentation.” [Online]. Available: https://matplotlib.org/. [Accessed: 23-May-2021]

[37] “Project Jupyter.” [Online]. Available: https://www.jupyter.org. [Accessed: 23-May-2021]

[38] “PyCharm: the Python IDE for Professional Developers by JetBrains,” *JetBrains*. [Online]. Available: https://www.jetbrains.com/pycharm/. [Accessed: 23-May-2021]

[39] T. pip developers, *pip: The PyPA recommended tool for installing Python packages*. [Online]. Available: https://pip.pypa.io/. [Accessed: 23-May-2021]

[40] “Anaconda | The World’s Most Popular Data Science Platform,” *Anaconda*. [Online]. Available: https://www.anaconda.com/. [Accessed: 23-May-2021]

[41] “Git.” [Online]. Available: https://git-scm.com/. [Accessed: 23-May-2021]

[42] Y. Shafranovich, “Common Format and MIME Type for Comma-Separated Values (CSV) Files,” RFC Editor, RFC4180, Oct. 2005 [Online]. Available: https://www.rfc-editor.org/info/rfc4180. [Accessed: 20-May-2021]

[43] “Exploratory Data Analysis | SpringerLink.” [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-43742-2_15. [Accessed: 20-May-2021]

[44] R. Dunford, Q. Su, and E. Tamang, “The Pareto Principle,” 2014 [Online]. Available: https://pearl.plymouth.ac.uk/handle/10026.1/14054. [Accessed: 23-May-2021]

[45] “A Complete Tutorial which teaches Data Exploration in detail,” *Analytics Vidhya*, 10-Jan-2016. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/. [Accessed: 20-May-2021]

[46] J. Han and M. Kamber, *Data mining: concepts and techniques*, 3rd ed. Burlington, MA: Elsevier, 2012 [Online]. Available: http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf

[47] C. Tofallis, “A better measure of relative prediction accuracy for model selection and model estimation,” *J. Oper. Res. Soc.*, vol. 66, Mar. 2015, doi: 10.1057/jors.2014.124.

[48] S. Shrivastava, “Cross Validation in Time Series,” *Medium*, 17-Jan-2020. [Online]. Available: https://medium.com/@soumyachess1496/cross-validation-in-time-series-566ae4981ce4. [Accessed: 20-May-2021]

[49] M. Claesen and B. De Moor, “Hyperparameter Search in Machine Learning,” *ArXiv150202127 Cs Stat*, Apr. 2015 [Online]. Available: http://arxiv.org/abs/1502.02127. [Accessed: 24-Sep-2021]

[50] “sklearn.neighbors.KNeighborsRegressor — scikit-learn 0.24.2 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html. [Accessed: 05-Jun-2021]

[51] “sklearn.ensemble.RandomForestRegressor — scikit-learn 0.24.2 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html. [Accessed: 05-Jun-2021]

[52] “2015 09 TE Briefing Truck CO2 Too big to ignore_FINAL.pdf.” [Online]. Available: https://www.transportenvironment.org/sites/te/files/publications/2015%2009%20TE%20Briefing%20Truck%20CO2%20Too%20big%20to%20ignore_FINAL.pdf. [Accessed: 08-Jun-2021]

[53] “How much CO2 does a tree absorb?” [Online]. Available: https://www.viessmann.co.uk/heating-advice/how-much-co2-does-tree-absorb. [Accessed: 09-Jun-2021]

TRITA-EECS-EX-2021:811