Tampere University

Aino Heino

NEW PRODUCT DEMAND FORECASTING IN RETAIL

Applying Machine Learning Techniques to Forecast Demand for New Product Purchasing Decisions

Faculty of Management and Business
Master of Science Thesis
November 2021

ABSTRACT

Aino Heino: New Product Demand Forecasting in Retail: Applying Machine Learning Techniques to Forecast Demand for New Product Purchasing Decisions
Master of Science Thesis
Tampere University
Master's Degree Programme in Industrial Engineering and Management
Examiners: prof. Juho Kanniainen and prof. Jussi Heikkilä
November 2021

In retail, it's vital to minimize food spoilage and working capital, and to maximize product availability and sales. To make it possible, retailers must know what the proper order quantity is when making purchase decisions. Hence, demand forecasting plays a crucial role in retail industry. However, generating accurate forecasts is difficult, especially when it comes to new products for which there is no historical data available. In addition, the most common new product forecast methods are at least partly judgemental and thus inefficient. Consequently, there is a clear need for accurate and efficient method for forecasting new product demand.

This thesis addresses the phenomena by applying and evaluating five machine learning models for new product demand forecasting for purchase decisions in retail and selecting the model that performs best. The forecasting problem is formulated as a classification task in which the objective is to forecast the magnitude of sales for new products. The applied machine learning models are 1) logistic regression, 2) support vector classification, 3) nearest neighbors, 4) XGBoost, and 5) multi-layer perceptron. For applying the models, a machine learning workflow is designed. In the workflow, suitable features are formulated by converting raw data into desired variables and selecting the final features through a systematic process. The features are scaled using three different methods, and the best performing method is selected for each model. In addition, hyperparameters are tuned through cross-validation. The evaluation and model selection are based on accuracies, precision, recalls, and F1-scores of the models, and the sensitivity to random states is considered. The utilized data set is provided by a case company, which is an e-commerce retailer that focuses on surplus products.

The results show that all the five models perform better than the benchmark model, which predicts the major class among training samples for all the test samples. The performance metrics of the models don't depend significantly on different random states. The selected model for forecasting new product demand in the case company is the XGBoost model, which overperforms the other applied models in all the evaluation metrics. The XGBoost model is ready for implementation in the case company since the model is already optimized using the data from the company. The developed new method is robust and efficient, and it eliminates many problems related to the current method the case company uses. Even though the models must be optimized again if the models are used in other contexts, this study introduces a machine learning workflow that provides tools for that, and the workflow is easy to follow. This study proves that new product demand can be forecasted successfully using machine learning classification methods, which is an interesting alternative approach to more traditional regression methods. As a suggestion for future research, the models could be developed further and optimized separately for different product types. Moreover, in addition to demand forecasting, applying classification models to classify other product characteristics could be considered.

Keywords: retail, inventory management, procurement decisions, purchase decisions, demand forecasting, new product demand forecasting, machine learning, logistic regression, support vector classification, nearest neighbors, XGBoost, multi-layer perceptron

TIIVISTELMÄ

Aino Heino: Kysynnän ennustaminen vähittäiskaupassa: koneoppimismenetelmien soveltaminen kysynnän ennustamiseen ostopäätösten tueksi
Diplomityö
Tampereen yliopisto
Tuotantotalouden diplomi-insinöörin tutkinto-ohjelma
Tarkastajat: prof. Juho Kanniainen ja prof. Jussi Heikkilä
Marraskuu 2021

Vähittäiskaupassa on tärkeää sekä vähentää hävikin ja sitoutuneen pääoman määrää että edistää myyntiä huolehtimalla tuotteiden riittävästä saatavuudesta. Jotta tämä olisi mahdollista, hankinnassa on ymmärrettävä, kuinka paljon myytäviä tuotteita tulee tilata. Näin ollen kysynnän ennustaminen on erityisen kriittisessä roolissa vähittäiskaupan alalla, mutta samaan aikaan tarkkojen ennusteiden laatiminen on hyvin haastavaa. Erityisesti uusien tuotteiden kysynnän ennustaminen on ongelmallista, sillä tällaisista tuotteista ei ole lainkaan myyntidataa saatavilla ennusteita varten. Lisäksi yleisimmät uusien tuotteiden kysynnän ennustamisen menetelmät perustuvat ainakin osittain ammattilaisten subjektiivisiin näkemyksiin, minkä vuoksi ennustaminen on tehotonta. Siten on selvää, että vähittäiskaupan alan toimijat hyötyisivät menetelmästä, jolla uusien tuotteiden kysyntää voitaisiin ennustaa tarkasti ja tehokkaasti.

Tässä tutkimuksessa pyritään vastaamaan kyseiseen tarpeeseen soveltamalla viittä koneoppimismallia uusien tuotteiden kysynnän ennustamiseen ostopäätösten tueksi vähittäiskaupassa. Lopulta viidestä mallista valitaan yksi malli, joka suoriutuu ennustamisessa parhaiten. Ennustamisongelma muotoillaan luokittelutehtäväksi, jossa tavoitteena on ennustaa uusien tuotteiden myynnin suuruusluokkaa. Viisi sovellettua koneoppimismenetelmää ovat 1) logistinen regressio, 2) tukivektorikone, 3) lähimmän naapurin menetelmä, 4) XGBoost ja 5) monikerroksinen perseptroniverkko. Tutkimuksessa suunnitellaan prosessi koneoppimismallien optimointia varten. Prosessissa laaditaan selittävät muuttujat muuntamalla alkuperäistä dataa, ja lopulliset selittävät muuttujat valitaan systemaattisten vaiheiden kautta. Muuttujat skaalataan kolmella eri skaalausmenetelmällä, joista parhaiten suoriutuva menetelmä valitaan kullekin mallille. Lisäksi mallien hyperparametrit optimoidaan ristiinvalidoinnin avulla. Mallien arviointi ja lopullisen mallin valinta perustuu ennusteiden tarkkuuteen, precision- ja recall-arvoon sekä F-pisteytykseen. Lisäksi tarkastellaan mallien herkkyyttä erilaisille satunnaisille tiloille. Tutkimuksessa hyödynnetään poisto- ja jäännöseriin keskittyvän verkkokaupan dataa.

Tulokset osoittavat, että kaikki viisi koneoppimismallia suoriutuvat paremmin kuin vertailumalli, joka ennustaa kaikille testijoukon alkioille opetusjoukon suosituimman luokan. Mallien suorituskyky ei riipu merkittävästi satunnaisista tiloista. Lopullinen valittu malli on XGBoost-malli, joka menestyy kaikilla tarkastelluilla mittareilla paremmin kuin yksikään muu tutkimuksessa sovellettu malli. XGBoost-malli on valmis käyttöönotettavaksi kohdeyrityksessä, sillä mallin optimoinnissa hyödynnettiin kohdeyrityksen dataa. Kehitetty menetelmä on luotettava ja tehokas, ja se poistaa monia kohdeyrityksen nykyisen ennustamismenetelmän ongelmia. Vaikka mallit on optimoitava uudestaan, mikäli niitä hyödynnetään muissa ympäristöissä, tämä tutkimus tarjoaa selkeän prosessin ja työkalut optimointia varten. Tutkimuksen perusteella uusien tuotteiden kysyntää voidaan onnistuneesti ennustaa luokittelumallien avulla, mikä on mielenkiintoinen lähestymistapa perinteisempien regressiomallien rinnalle. Jatkossa aihepiirin tutkimusta voisi viedä pidemmälle kehittämällä koneoppimismalleja entisestään ja optimoimalla ne erikseen erilaisille tuotetyypeille. Lisäksi luokittelumallien soveltuvuutta tuotteiden muiden ominaisuuksien luokitteluun voisi tutkia tarkemmin.

Avainsanat: vähittäiskauppa, varastonhallinta, hankintapäätökset, ostopäätökset, kysynnän ennustaminen, uusien tuotteiden kysynnän ennustaminen, koneoppiminen, luokittelu, logistinen regressio, tukivektorikone, lähimmän naapurin menetelmä, XGBoost, monikerroksinen perseptroniverkko

PREFACE

I want to thank everyone who supported me when I worked on my master's thesis. First of all, I'm very thankful for my supervisors Juho Kanniainen and Jussi Heikkilä who both gave me valuable insights and advice during the process. Second, I want to thank the case company for making it possible to solve such an interesting real-life problem.

To be honest, I didn't have much time for this thesis as I have already transitioned to working on other projects full time. Sometimes the graduation felt almost impossible, and I'm very grateful to Ilari Ryyppö and my family for believing in me even when I couldn't. Finally, special thanks to my fellow students that motivated me not only during the thesis work but also during my studies from the very beginning. Even though the years in the Tampere University passed quickly, I was able to make a lot of memories I will remember for the rest of my life.

Helsinki, 27th November 2021

Aino Heino

CONTENTS

.

1. INTRODUCTION

1.1 Research background

In retail industry, many of the products are perishable, so spoilage is a major risk that can lead to significant decreases in profitability. Thus, efficient inventory management plays a crucial role. (Chen et al., 2014) To manage inventory efficiently, accurate demand forecasts are required. However, generating accurate demand forecasts is a difficult task, especially when it comes to new products for which historical data is not available. According to Fildes et al. (2019), the most common method for new product forecasting is an analogical approach, where new products are assumed to behave similarly to some comparable products for which there is historical data available.

Analogical approach is applied in the case company of this study, too. The case company is an e-commerce retailer that sells grocery surplus batches to consumers. Since business is based on surplus batches, the product range is constantly varying, and new products are often added to the selection. In addition, due to the nature of surplus batches, the shelf life of products is in most cases shorter than in regular grocery stores. Hence, the need for accurate demand forecasts for new products is emphasized, but at the same time the process shouldn't require a lot of effort as new products are purchased on a daily basis. However, currently the comparable products for forecasting are chosen manually by purchasers. This is problematic at least in three ways: 1) assuming that new products behave similarly as subjectively chosen comparable products can lead to inaccurate forecasts, 2) comparable products that are similar enough might not always be available, and 3) it's difficult to develop an effortless forecasting process if continuous human effort is needed. This raises a question if the company could use another approach to make the forecasting process more efficient and accurate.

In addition to traditional forecast methods, such as simple trend-based regressions, more advanced machine learning methods have been developed. According to Bajari et al. (2015) machine learning methods for demand estimation can significantly increase forecasting accuracy as compared to more traditional methods, and these methods combine the benefits of both parametric approaches with user-selected covariates and non-parametric approaches. In addition, use of direct comparable products in new product demand forecasting can be eliminated by using a set of product features (Fildes et al.,

2019), and machine learning models are one way to do so. Thus, utilizing a machine learning model instead of a traditional analogical approach is a potential alternative for new product demand forecasting in the case company.

1.2 Research objective and scope

The objective of this thesis is to create a method for forecasting new product demand for purchasing decisions in retail by applying and evaluating several machine learning models and selecting the one that performs best. Demand forecasting can be done in different levels, and in this study the selected levels are *day* in the time dimension, *stock keeping unit* in the product dimension, and *store* in the supply chain dimension. The decision on forecast levels is based on the nature of the business of the case company, but these levels are also important in retail demand forecasting in general. In the case company, the forecasts are generated for the needs of purchasing function, and purchase decisions require information about daily demand of the products (stock keeping units) to be purchased. In addition, the case company is not a retail chain but an online store, and thus the store level is a logical choice. The selected levels are in line with existing literature, as according to Fildes et al. (2019), in operational decisions, including purchasing decisions, stock keeping unit forecasts in a relatively high time granularity are necessary.

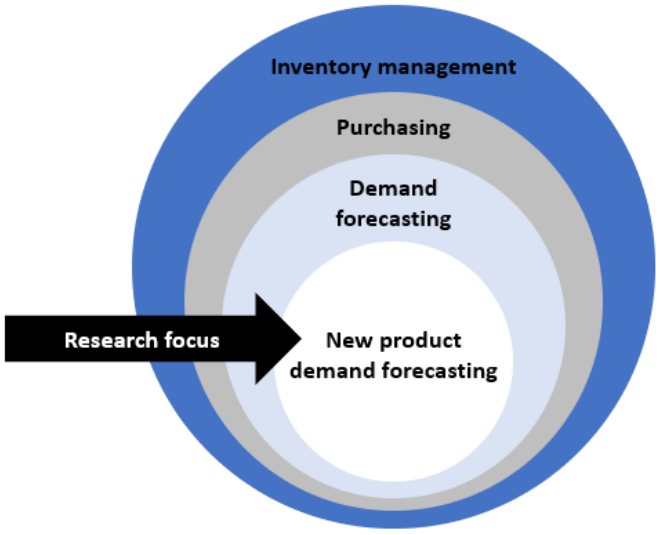


Figure 1. The focus of this thesis.

The study focuses on new product demand forecasting, and the demand forecasting of existing products is not addressed. New products are selected mainly for two reasons: 1) the case company already has a sufficient method for demand forecasting for existing products but the current method for new products is challenging, and 2) forecasting de-

mand for new and existing products are separate tasks that should be handled independently, since the availability of relevant data makes a significant difference between these two issues. Figure 1 illustrates the focus of this thesis.

1.3 Structure of the thesis

The next chapter of this thesis covers an overview of retail industry and demand forecasting in that context. The chapter starts with a short review on retail industry and inventory management in general, and after that the characteristics of demand forecasting are discussed, eventually narrowing the perspective to demand forecasting for new products. The third chapter focuses on theoretical aspects of machine learning workflow, including the theory behind the machine learning algorithms utilized later in this study.

After the two theory chapters, the research methodology of the study is described, covering justification for methodological choices, data set characteristics, the applied machine learning workflow, and the forecasting setup. Chapter 5 presents the results, and finally, the last chapter concludes the study by discussing the results, assessing the study, and reviewing directions for future research.

2. DEMAND FORECASTING IN RETAIL

2.1 Retail industry

Retailing is one of the largest industries in the world, and it can be viewed as a process of buying goods with the aim of reselling them to the end customer. In other words, retailers act as a linkage between manufacturers and consumers. Further, retailers perform several functions to add value to the distribution of merchandise. They 1) create an assortment by preselecting products from a wide selection that manufacturers offer, 2) reduce transportation costs and offer consumer friendly quantities by purchasing large batches of products and dividing them into smaller lot sizes, 3) eliminate geographic and temporal gaps between manufacturers and final customers by holding an inventory and offering the products in one place near the customers in the time of demand, 4) create demand by displaying products attractively, 5) reduce overall purchase transaction costs by standardizing ordering, picking and payment activities, and finally, 6) offer a variety of product-related services for final customers. (Zentes et al., 2012)

Recent technological advances have had a significant impact on retail industry (Shankar et al., 2021). For example, grocery retail has changed dramatically during the 21st century, as more and more consumers are buying groceries online (Kureshi & Thomas, 2019). Most recently, COVID-19 pandemic accelerated this change as many retailers and consumers preferred online shopping over physical stores to prevent the virus from spreading, which showed that technology enables the industry to adapt to unexpected circumstances (Shankar et al., 2021). Moreover, according to Renko and Ficko (2010), use of technologies is one of the most important competitive tools as retailers have to offer more than affordable prices to achieve competitive advantage nowadays. They emphasize that the role of the logistics is currently more critical than ever, and new technologies play a vital role among logistics activities. One crucial component of logistics activities is inventory management, which is discussed in more detail in the following section 2.2.

2.2 Inventory management

According to Williamson et al. (1990), inventory management is one of the primary functional groups of logistics. Other four groups are transportation, facility structure, material handling, and communication and information. Furthermore, they view inventory management as a group of activities, consisting of purchasing, raw material inventory, work-

in-progress inventory, finished goods inventory, parts and service support, and return goods handling. However, even though logistics can be divided into smaller functional groups and activities, it's vital that these activities work together: fragmented logistics can lead to serious productivity issues when different functions are pursuing their goals independently, leading to duplicated workload. (Renko & Ficko, 2010) Thus, inventory should be managed in a way that it contributes to the efficiency of the logistics functional groups as a whole.

When it comes to inventory management in the context of retail, some of the activities Williamson et al. (1990) address are not relevant. For example, raw material inventory is not needed if all the products are purchased as finished goods. According to Zentes et al. (2012), in retail inventory management, the main issue to address is how much stock should be held for different products. Efficient inventory management plays a crucial role as many of the products are perishable having a finite shelf life, and spoilage is a major risk that can lead to significant decreases in profitability (Chen et al., 2014). As a result, one of the key tasks in inventory management is to make sure that inventory turnover does not exceed the shelf life. However, obsolete and idle inventory are not the only factors to focus on. Retailers are also paying more and more attention to the costs of losing sales due to unavailability of products, so inventory management should be balanced regarding both the aspects (Rana, 2020).

Not only the retailer but also the whole supply chain benefits from efficient inventory management. According to Rana (2020), proper inventory management is a major driver for increasing the responsiveness of the whole supply chain. Nowadays retail supply chains are generally more integrated than before and the whole flow aims to be demand-oriented. As retailers have direct access to sales data, they are the gatekeepers for information flows in the supply chain. (Zentes et al., 2012) Holweg et al. (2005) agree that the retail demand controls the inventory and production control process in the supply chain, but they note that retailers often don't have appropriate demand forecasting processes for sharing the needed information. To manage inventory properly, retailers should know when and how much to purchase specific products to maintain feasible stock levels (Rana, 2020), but due to global competition and increases in the pace of product development, flexibility of manufacturing, and variation of products, it's very difficult for retailers to make accurate forecasts (Fisher et al., 1994). Despite of the challenges, demand forecasts are necessary for many operational decisions and activities (Huber & Stuckenschmidt, 2020).

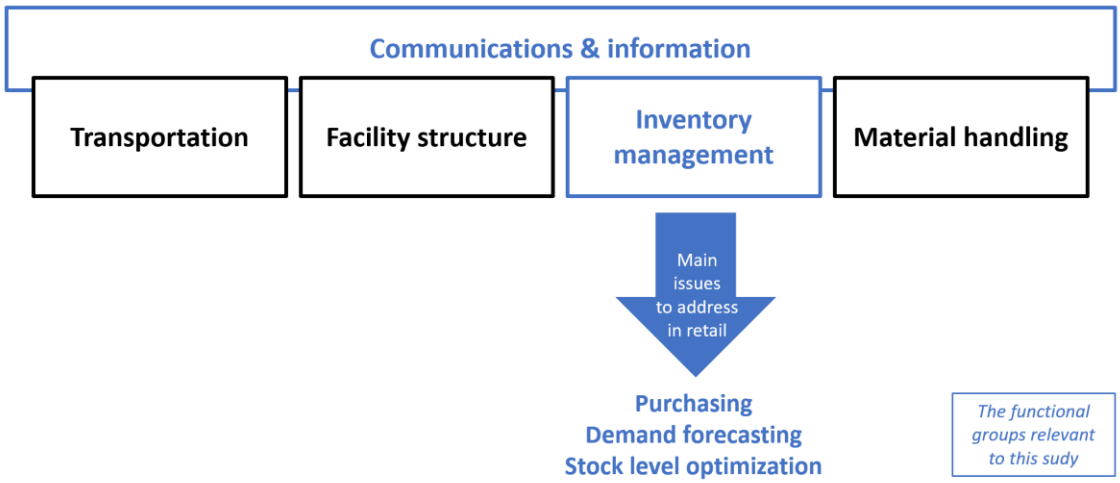


Figure 2. Logistics functional groups and the main issues to address in the context of retail inventory management (adapted from Williamson et al., 1990).

To conclude, inventory should not be managed separately from other logistics activities. In the broader view, inventory management plays a crucial role in the whole supply chain as it is a starting point for information flow. Thus, inventory management is strongly related to another previously presented logistics functional group, communications and information. In this thesis, demand forecasting is viewed as an activity of inventory management rather than an activity of separate functional group. This slightly differs from Williamson et al. (1990) definition, as they define communications and information as a separate functional group and demand forecasting is one of its activities. The logistics groups and the main issues in the context of inventory management in retail, as viewed in this study, are presented in Figure 2 above.

2.3 Retail demand forecasting

As a general definition, demand refers to the amount of a specific product or a service that a consumer is willing to buy with a specific price. According to Lewis (1997), demand can be classified into dependent or independent demand, where dependent demand is dependent of other related products or services, and independent is not. He also states that forecasting, defined as a process of estimating future using past data, differs from prediction, defined as a process of estimating future using subjective views. However, in this thesis both forecasting and prediction are equally interpreted as a process of estimating future, regardless of whether the estimation is based on data or subjective considerations.

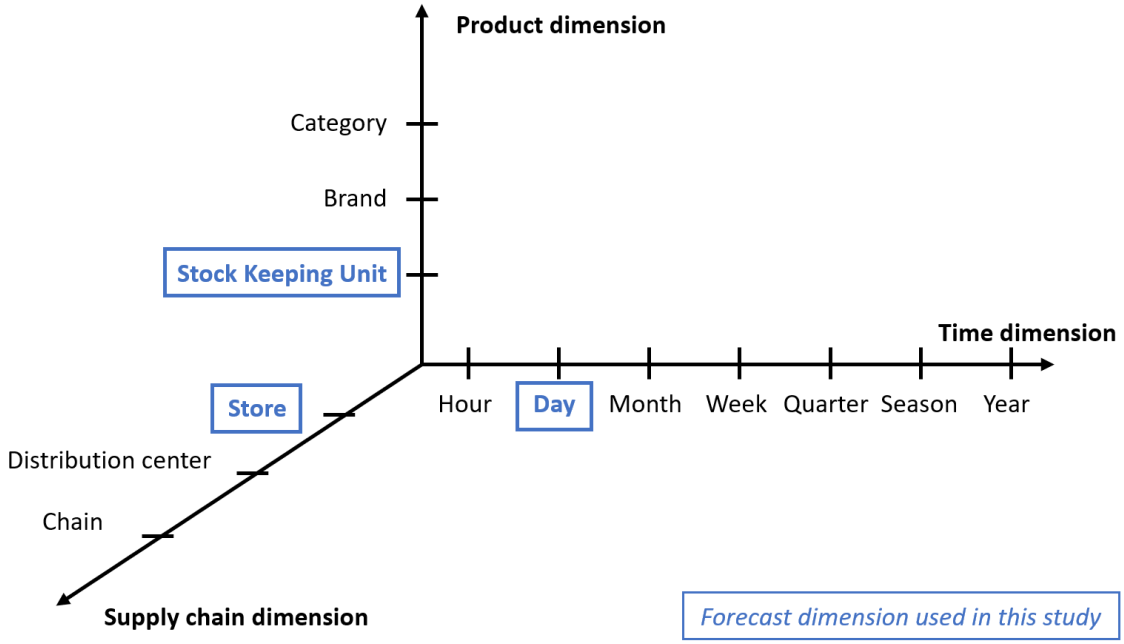


Figure 3. Dimensions of product-level forecasting in retail (adapted from Fildes et al., 2019).

Demand forecasting can be done in different levels, and in this thesis, we focus on product-level demand forecasting. Product-level demand forecasting is characterized by three dimensions, the time granularity, the level in the product hierarchy, and the position in the retail supply chain. Generally, the time granularity increases as we move from strategic decisions to operational decisions. The product hierarchy consists of three different levels that are SKU level, brand level and category level, where SKU (stock keeping unit) is the smallest level being necessary to both operational and tactical decisions. Finally, the level in the supply chain refers to store level, distribution center level, or chain level. (Fildes et al., 2019) The three different dimensions are summarized in Figure 3. In this study, the demand forecasting levels are day, SKU, and store. This is logical as we focus on developing a new method for estimating optimal quantities for purchasing new products. The case company is not a retail chain, but an online store and the decision is operational, so the chosen levels are in line with previous literature.

2.4 Demand forecasting methods

In this section, a general view of existing demand forecasting methods is introduced. Before that, it's important to understand that demand forecasting is often viewed as a separate process from sales forecasting, since demand might differ from sales, for example due to product unavailability. However, since the methods are often overlapping, and the data used in this thesis covers only sales data from the dates when the products were available, it's not necessary to distinguish between these two perspectives here.

In their study, Mentzer and Cox (1984) separate sales forecasting techniques into two board classes, subjective and objective techniques. In the study, subjective techniques cover, for example, opinions of customers and sales force. Objective techniques include, for example, moving average, exponential smoothing, regression. In contrast, Mahmoud (1984) classifies forecasting techniques into qualitative methods, such as judgmental forecasts of management and analysts, and quantitative methods, such as various time-series methods that are based on historical data. It seems that there is no universal terminology for different forecasting methods, but since the discussed subjective methods fall into class of qualitative methods and objective into quantitative methods, we can conclude that different methods can be separated into two broad categories, subjective qualitative techniques, and objective quantitative techniques. However, Mahmoud (1984) also addresses methods that are combinations of two or more techniques. As a result, one combination method can be both, quantitative and qualitative.

Several different quantitative techniques have been developed over the years, and the availability of large data sets has led to new more advanced methods (Bajari et al., 2015). Carbonneau et al. (2008) view simple time series methods as traditional methods, whereas a number of machine learning techniques are viewed as advanced methods. The examples of traditional methods they introduce are 1) naïve forecast which uses the latest value as a forecast for the future value, 2) moving average which uses the average of finite number of previous realizations, 3) trend-based forecast which is a simple regression model aiming to forecast demand as a function of time, and 4) multiple linear regression which is based on several past changes in demand as independent variables. Advanced methods include neural network and support vector machine methods. However, there exist also many other machine learning models that can be applied to forecast tasks, for example, Bajari et al. (2015) use LASSO, a penalized regression method, and random forests, an ensemble learning method based on decision tree algorithms. They found that, compared to more traditional methods, these machine learning methods for demand estimation can lead to significantly increased forecasting accuracies. A summary of the classification of demand forecasting methods is presented in the Figure 4 below.

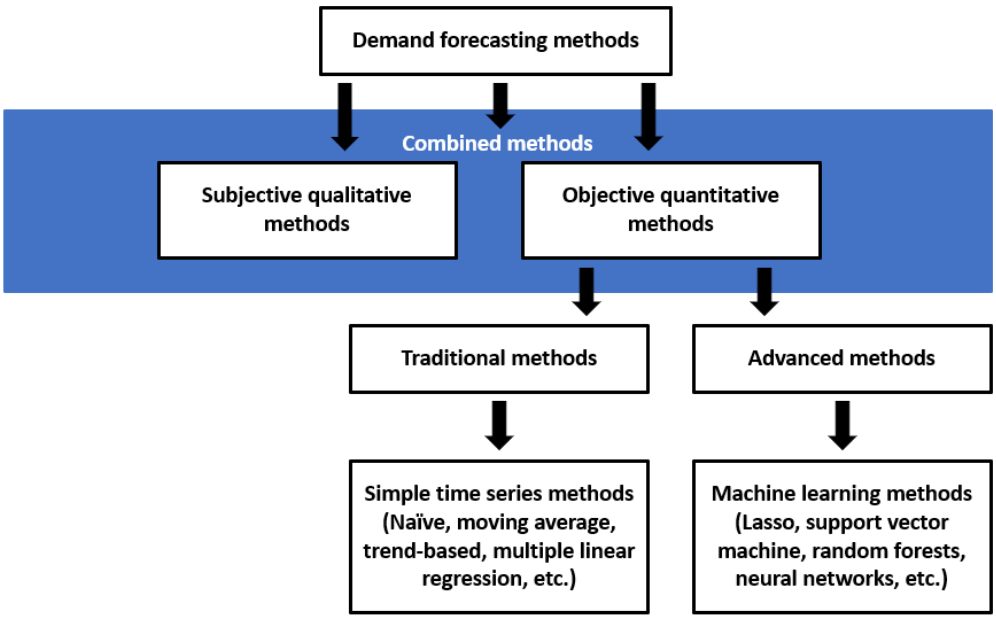


Figure 4. A summary of the classification of demand forecasting methods.

One topic related to demand forecasting methods, is classification of stock keeping units. According to Bajari et al. (2015) stock keeping units should be categorized based on the underlying demand structures, as those require different methods for demand forecasting and inventory management. However, they have noticed that very little research has been conducted in this area. Bacchetti and Saccani (2012) study the classification of stock keeping units, too, but from a broader view. In addition to demand structures, they review other classification criteria, such as costs and supply uncertainty. In line with Bajari et al. (2015), they state that classification has not received much academic attention. Classification of stock keeping units is a key topic in the context of this thesis, since typical time series forecasting methods were not successful, and as a result the whole modelling process is based on classification methods.

2.5 Demand forecasting for new products

New product demand forecasting is more difficult than forecasting demand for existing products since product-specific historical data is not available. Still, it's a critical issue to address as decisions are based on estimated demand in many different functions in retail. New product demand forecasting has some common characteristics that can be applied to the needs of many retailers and product types, but specific characteristics must be involved in most cases, too. Availability of previous research is limited, but there seems to be three broad categories for new product demand forecasting methods. Those are 1) the judgemental approach where management judgement is used, 2) the market research approach where market survey data are used, and 3) the analogical approach

where new products are assumed to perform similarly as some other products, and historical data of those are used. (Fildes et al., 2019)

In this study, we focus on the analogical approach, aiming to find a way to automate the forecasting process. Judgement or customer surveys require continuous human effort which makes the process inefficient, especially if the number of new products is high and forecasts needs to be made frequently. However, according to Fildes et al. (2019), also the analogical approach is often partly judgemental as the identification of comparable products might require judgement. This is the current situation in the case company addressed in this study, too. To make the forecasting process more automated and efficient, we aim to design new product forecasting methods that require little or no judgement. Syntetos et al. (2016) introduce some approaches for choosing the comparable products without judgement, including the use of product category and the use of machine learning techniques. However, in addition to the need for judgement, use of comparable products poses challenges in the case company since there is often no suitable comparable product available. According to Fildes et al. (2019), this challenge can be eliminated by using a set of features, such as brand, flavour, and price, rather than using one direct comparable product.

Finally, Syntetos et al. (2016) state that new product forecasts that are based on data of similar products only are very uncertain and sales data of the new product should be taken into account in forecasts as soon as it's available. Hence, new product demand forecasting is addressed as a separate problem from existing product forecasting in this thesis. Since the forecasts are likely to be highly uncertain, professional procurement judgement is essential in new product purchasing decisions even if the forecasting itself was based purely on data.

2.6 Synthesis of the literature review

The literature review on retail business and inventory management provided insights on the importance of demand forecasting in this context and gave guidelines on how new product demand forecasts could be generated. The amount of perishable goods increases the risk of profitability decreases due to spoilage (Chen et al., 2014), but at the same time retailers must avoid stock outs that lead to loss of sales (Rana, 2020). Thus, while evaluating which of the applied machine learning models for demand forecasting performs best in this study, the evaluation metrics should not only minimize risk of spoilage but also maximize days on sale. Several methods for demand forecasting were introduced in the literature, and machine learning methods were classified as more advanced methods. Carbonneau et al. (2008) found that machine learning methods for

demand estimation can significantly increase forecasting accuracy as compared to more traditional methods, which favors the choice of applying machine learning methods in this study.

In most cases, demand forecasting methods are not directly suitable for new product forecasting, and it's important to address new products independently from existing products. Typically, new product demand forecast methods require management judgement (Fildes et al., 2019), but the need of judgement can be replaced by using a set of features rather than one direct comparable product. Since we aim to make the new product demand process effortless and automated, it could be more beneficial to develop one machine learning model for all new products and eliminate the need of comparable products rather than developing different models for each comparable product. However, this kind of model is still based on data of different products only, and as Syntetos et al. (2016) state, this leads to highly uncertain forecasts. As a result, it might not be possible to forecast exact daily demand of new products accurately. An alternative way could be forecasting the magnitude of demand: according to Bajari et al. (2015) stock keeping units could be classified based on the underlying demand structures.

3. MACHINE LEARNING APPROACH

3.1 Machine learning basics

Machine learning, defined as building computers that improve automatically through experience, is one of the most rapidly growing technical fields. The three major machine learning paradigms are supervised learning, unsupervised learning, and reinforcement learning, but in recent studies these categories are often blended. (Jordan & Mitchell, 2015) In supervised learning, an algorithm uses both input values and output values to learn how to predict the output from the input. In unsupervised learning, the objective is to find a structure and only input values are known by the algorithm, so labelled data is not needed. Reinforcement learning is based on communication and exploration and here the algorithm uses feedback to learn. (Manco et al., 2021) The feedback indicates whether the action is correct or incorrect, so reinforcement learning can be viewed as an intermediate paradigm between supervised and unsupervised learning (Jordan & Mitchell, 2015)

The most widely used machine learning methods fall into the category of supervised learning (Manco et al., 2021). In this context, the inputs can be vectors or more complex objects, and the output type depends on the nature of the problem. In binary classification, outputs vary between two labels whereas in multiclass classification there are more possible outcomes. (Jordan & Mitchell, 2015) In addition to classification algorithms, supervised learning covers regressive algorithms where outputs take continuous values (Manco et al., 2021) As this thesis deals with supervised learning and classification problems, the methods and techniques introduced next focus on supervised learning and classification, too.

When it comes to the data set, the available data is splitted into three sets in most machine learning applications: train, validation, and test sets (Shalev-Shwartz & Ben-David, 2014). The train set is used to fit the parameters of the algorithm, the validation set is used for evaluating the model with an external data set while tuning hyperparameters, and finally the test set is used to estimate the actual performance of the final model with completely new data, aiming to guarantee an unbiased evaluation. (Manco et al., 2021)

3.2 Machine learning workflow

Machine learning workflow defines the phases involved in machine learning project. According to Quemy (2020), common machine learning workflow consists of two parts, data

pipeline and model building. Data pipeline is about finding and selecting suitable techniques for transforming the data set to be consumable by a machine learning algorithm, and model building is about selecting the machine learning algorithm and its hyperparameters in a way that the model achieves the desired performance given by the chosen evaluation metrics. He suggests that the typical steps that data pipeline covers are data pre-processing, feature extraction, and feature selection, and the steps that model building covers are algorithm selection and parameter tuning. Figure 5 illustrates the typical machine learning workflow.

Figure 5. The typical machine learning workflow (adapted from Quemy 2020).

Data pipeline and model building cannot be treated as independent parts since the choice of features affects the model and vice versa. In addition, good quality of the features plays a vital role. Proper features usually require much less complex model to achieve the desired performance than poor features do. (Zheng, 2018) However, finding proper features is itself a complex and time-consuming process, and one cannot define instructions that are suitable for every machine learning project. Examination of the previous literature showed that there exists quite a lot research focusing on designing workflow and features for specific machine learning tasks, for example Kaspi et al. (2021) study workflow for glass fragment analysis, but the results are not that relevant in other contexts. It's important to understand that the typical workflow Quemy (2020) introduces is not a final workflow for a specific machine learning task but a great base for designing a more detailed one.

3.3 Data pre-processing

The success of machine learning model depends heavily on representation and quality of data, but in most cases the quality of raw data is not sufficient as it might contain for example noisy data, redundant data, or missing data values. Thus, data pre-processing, which aims to eliminate these problems, can have a significant impact on performance of a supervised machine learning algorithm. In fact, data pre-processing is often the most time-consuming phase when developing machine learning models. (Kotsiantis et al., 2006) Usually, raw data consists of many different datatypes, such as numerical data, text data, categorical data, and time series data. The first three data types are relevant

in the context of this thesis, so some common data pre-processing techniques for these three data types are introduced in this chapter.

According to Galli (2020), many machine learning algorithms are affected by the scale of the numerical features. For example, in algorithms that perform distance calculations, features with bigger value ranges dominate over features with smaller ranges. Galli (2020) introduces several techniques for scaling features to similar ranges: one commonly used feature scaling technique is standardization, which transforms the features to have zero mean and unit variance. Feature standardization is defined by

$$\tilde{x} = \frac{x - mean(x)}{std(x)}.$$

(1)

However, standardization assumes that the original data fits a Gaussian distribution, and the method is sensitive to outliers. Another presented method is scaling the features to the maximum and minimum, where the scaled value is defined by

$$\tilde{x} = \frac{x - min(x)}{max(x) - \min(x)}.$$

(2)

This method is still sensitive to outliers. If the data contains outliers, Galli (2020) suggests a method called robust scaling, which is based on percentiles, defined as

$$\tilde{x} = \frac{x - median(x)}{75thQuantile(x) - 25thQuantile(x)}.$$

(3)

Pre-processing of categorical data is a crucial part as most machine learning models are capable of processing numerical features only. Categorical variables can be encoded in many different ways, and the most used coding scheme is one-hot encoding. (Dahouda & Joe, 2021) According to Hancock and Khoshgoftaar (2020), one-hot encoding is a technique where a discrete categorical variable x with n distinct values $x_1, x_2, \dots, x_n$ is transformed into n vectors. One-hot encoding of x_i is a vector where all but the i^{th} components are 0, when the i^{th} component is 1. The end result is a set of vectors where each vector represents one of the distinct values.

In addition to categorical data, text data must be transformed into numerical values. The simplest representation is bag-of-words, where a text is represented as a vector of word counts. Since bag-of-words representation contains information of word counts only, it eliminates the original textual structures. However, bag-of-words gives an equal weight

to all words, so the most meaningful and distinctive words might not stand out enough. A more sophisticated representation is term-frequency-inverse document frequency (tf-idf), which aims to emphasize meaningful words. In tf-idf, a pure word count is replaced with a normalized word count. There are many ways to normalize the word count, and one option is raw inverse document frequency, in which each word count is divided by the number of distinct documents the word appears N. Alternatively, one can use log normalization, in which each word count is multiplied by the log of N. (Zheng, 2018)

3.4 Feature extraction

Feature extraction means transforming the original set of features through mapping function into a new set of features, where the number of dimensions is reduced (Kotsiantis et al., 2006) Feature extraction techniques allows one to decrease the size of feature space without losing information, aiming to improve the performance of learning algorithms (Khalid et al., 2014)

One widely known feature extraction technique for text data is latent semantic analysis (LSA). Evangelopoulos (2013) introduces LSA as follows: LSA uses linear algebra to extract meaning of text while reducing the dimensionality of vector representation. Before applying LSA, term frequency matrix X, for example tf-idf matrix, is constructed. After that, X is decomposed into term eigenvectors U, document eigenvectors V, and singular values $\sum$ through singular value decomposition (SVD), defined as

$$X = U\sum V^T ,$$

(4)

where $X \in \mathbb{R}^{t\times d}, U \in \mathbb{R}^{t\times t}$, $\sum \in \mathbb{R}^{t\times d}$ and $V \in \mathbb{R}^{d\times d}$, when d is the number of documents in a space of t dictionary terms. The r elements of the diagonal matrix $\sum$, $r \leq \min(t, d)$, are called singular values and these values illustrate the relative importance of dimensions. The relative importance is defined by the capability to explain variability in term-document occurrences. The k most important dimensions with highest singular values are remained and the $r - k$ with lowest values are removed, resulting a matrix X_k, which is defined by

$$X_k = U_k\sum_k V_k^T,$$

(5)

where $X \in \mathbb{R}^{k\times d}, U \in \mathbb{R}^{t\times k}$, $\sum \in \mathbb{R}^{k\times k}$ and $V \in \mathbb{R}^{d\times k}$. Hence, LSA transforms X into matrix X_k, allowing one to represent the text features with reduced dimensionality. (Evangelopoulos, 2013)

3.5 Feature selection

Feature selection is a process where the number of features is reduced by selecting a subset of the original features, aiming to reduce the complexity and increase the effectiveness of the machine learning algorithm. When selecting the features, relevant, irrelevant, and redundant features are identified and only the relevant features are remained. Relevant features impact the output of the machine learning model, and that impact cannot be derived by other features, whereas irrelevant features don't impact the output and redundant features impact the output, but that impact can be derived by other features. (Kotsiantis et al., 2006)

According to Zheng (2018), feature selection techniques can be classified into three categories: filtering, wrapper methods, and embedded methods. In filtering, the unuseful features are identified and removed. Usefulness can be estimated in several ways, including computing correlation or mutual information statistics. In wrapper methods, different subsets of features are tested to find the best one. Finally, embedded methods select the features during model training.

In this thesis, filtering methods are applied, so two filtering techniques are introduced here in more detail, as presented by Batina et al. (2011). First of them is Pearson's correlation coefficient, which is a measure of linear dependence between two random variables X and Y, defined as

$$\rho(X,Y) = \frac{\mathrm{cov}(X,Y)}{\sigma_X \cdot \sigma_y} = \frac{\mathrm{E}[XY] - \mathrm{E}[X] \cdot \mathrm{E}[Y]}{\sigma_X \cdot \sigma_y},$$

(6)

where $\mathrm{cov}(X,Y)$ is the covariance between X and Y, $\mathrm{E}[X]$ is the expected value of X, and σ_X is the standard deviation of X. The correlation coefficient satisfies $0 \leq |\rho(X,Y)| \leq 1$.

Mutual information measures the dependence of any kind between two random variables by expressing the quantity of information obtained on X by observing Y. The mutual information of two random discrete variables X and Y is defined as

$$I(X;Y) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \Pr[X = x, Y = y] \cdot \log\left(\frac{\Pr[X=x, Y=y]}{\Pr[X=x] \cdot \Pr[Y=y]}\right).$$

(7)

It can be extended to continuous case, and the mutual information between a continuous random variable X and a discrete random variable Y is defined as

$$I(X;Y) = \sum_{y\in y}\int_{\mathcal{X}} \Pr[X=x, Y=y]\cdot\log\left(\frac{\Pr[X=x,Y=y]}{\Pr[X=x]\cdot\Pr[Y=y]}\right)dx.$$

(8)

Good features have a strong effect on the class labels but are not strongly correlated with each other, and hence machine learning models should be trained based on a subset of features, from which redundant and irrelevant features are removed (Li et al., 2018). Pearson's correlation can be used to identify the redundant features that are strongly correlated with another feature, and mutual information between a continuous variable and a discrete random variable can be used to identify the irrelevant features that don't have a strong effect on the class labels.

3.6 Machine learning algorithms for classification

Machine learning algorithms for classification are used to form a general hypothesis and based on that to assign a category to a set of predictor features. Since classification is a common task to perform, a number of different algorithms have been developed. (Kotsiantis, 2007) In this section, we go through some general classification algorithms that are applied later in this thesis. As previously stated, classification problems cover both binary and multiclass classification. In this thesis and hence in this section, we focus on multiclass classification, although many algorithms that can be applied to multiclass problems are extensions of algorithms that were originally designed for binary classification.

3.6.1 Logistic regression

Logistic regression is an algorithm for modelling the posterior probabilities of the K classes through linear functions in x. The probabilities take values between 0 and 1, and they sum to one. Logistic regression is defined by

$$\log\frac{\Pr(G=1|X=x)}{\Pr(G=K|X=x)} = \beta_{10} + \beta_1^T x$$

$$\log\frac{\Pr(G=2|X=x)}{\Pr(G=K|X=x)} = \beta_{20} + \beta_2^T x$$

...

$$\log\frac{\Pr(G=K-1|X=x)}{\Pr(G=K|X=x)} = \beta_{(K-1)0} + \beta_{K-1}^T x\,.$$

(9)

In most cases, logistic regression is fitted using maximum likelihood. The log-likelihood for N observations takes the form

$$l(\theta) = \sum_{i=1}^{N} \log p_{g_i}(x_i; \theta),$$

(10)

where $p_k(x_i; \theta) = \Pr(G = k | X = x_i; \theta)$. (Hastie et al., 2009)

3.6.2 Support vector classification

Support vector classifiers separate samples into classes based on linear or nonlinear boundaries, aiming to maximize the margin between classes. If the original input feature space cannot be separated to classes by a liner constraint, a linear boundary is constructed by using a large, transformed feature space instead. The transformed feature space is formed using expansions, which are known as kernel functions. Some popular kernel functions in support vector machine literature are linear ($(\kappa(x, x_i) = \langle x, x_i \rangle)$, dth-degree polynomial ($\kappa(x, x_i) = (1 + \langle x, x_i \rangle)^d$), radial basis ($\kappa(x, x_i) = \exp(-\gamma ||x - x_i||^2)$), and neural network ($\kappa(x, x_i) = \tanh(\kappa_1 \langle x, x_i \rangle + \kappa_2)$). For a binary classification, the decision function of a support vector machine for a given sample x is defined by

$$f(x) = \alpha_0 + \sum_{i=1}^{N} \alpha_i \kappa(x, x_i),$$

(11)

with nonzero coefficients α_i only for the observations for which the constraints are exactly met, and where κ is the kernel function. The parameters are chosen to minimize

$$\min_{\alpha_0, \alpha} \left\{ \sum_{i=1}^{N} [1 - y_i f(x_i)]_+ + \frac{\lambda}{2} \alpha^T K \alpha \right\},$$

(12)

where $y_i \in \{-1, 1\}$, $[z]_+$ is the positive part of z, λ is the regularization parameter, and where K is the $N \times N$ matrix of kernel evaluations for all pairs of training features.

Since this is a quadratic optimization problem with linear constraints, finding the solution requires quadratic programming. In addition to linear constraints, support vector machine can produce nonlinear constraints. The introduced binary case is a relatively simple version of support vector machines, as this technique can be generalized to multiclass cases. Multiclass problems are addressed by solving several binary problems or using the multinomial loss function with a suitable kernel. (Hastie et al., 2009)

3.6.3 Nearest neighbors

The k-nearest neighbors algorithm (kNN) is a simple technique that classifies an unknown example based on k training examples that are closest to it. The distance is defined by some distance metric, such as Euclidean distance, and the most common class among the k nearest neighbors is assigned to the unknown example. (Chandramouli, 2018)

3.6.4 XGBoost

XGBoost is a widely used scalable machine learning technique for tree boosting. A tree ensemble model uses K additive functions, and for a data set with n examples and m features $D = \{(x_i, y_i)\}$ ($|D| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}$), the prediction is the sum of classifiers

$$\hat{y_i} = \phi(x_i) = \sum_k^K f_k(x_i), f_k \in \mathcal{F},$$

(13)

where $\mathcal{F} = \{f(x) = w_{q(x)}\}(q : \mathbb{R}^m \to T, w \in \mathbb{R}^T)$ is the space of classification and regression trees, q refers to the structure of each tree, and T is the number of leaves in the tree. Each f_k corresponds to an independent q and leaf weights w.

The set of functions are learnt by minimizing the regularized objective

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k),$$

(14)

where $\Omega(f) = \gamma T + \frac{1}{2}\lambda \parallel w \parallel^2$, l is a loss function that measures the difference between $\hat{y}_i$ and y_i, and Ω penalizes the model complexity.

The model is trained iteratively, and at the t-th iteration, f_t is added to minimize the objective

$$\mathcal{L}^{(\mathrm{t})} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t),$$

(15)

where $\hat{y}_i^{(t)}$ is the prediction of the i-th instance.

The objective function can be simplified by using second-order approximation and removing the constants. The optimal weight w_j^* of leaf j for a fixed structure $q(x)$ is defined as

$$w_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h+\lambda},$$

(16)

where $g_i = \partial_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ thus, the optimal value is calculated by

$$\mathcal{L}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{\left(\sum_{i\in I_j} g_i\right)^2}{\sum_{i\in I_j} h + \lambda} + \gamma T.$$

(17)

Note that since it's not always possible to enumerate all the tree structures q, a greedy algorithm that adds iteratively branches to the tree can be used instead. The best split can be searched by enumerating over all possible splits using so-called exact greedy algorithm, which is a very powerful method. Alternatively, a more efficient method, called approximate algorithm, that enumerates over some of the candidates only, can be used. (Chen & Guestrin, 2016)

3.6.5 Multi-layer perceptron

Multi-layer perceptron (MLP) is a multiple feedforward artificial neural network that maps input vectors to output vectors. The MLP has multiple node layers, input layer, output layer and one or more non-linear hidden layers. The network is fully connected, which means that all nodes in one layer are connected to the neurons in the next layer. Usually, a backpropagation algorithm is used to train the MLP. Unlike single-layer perceptron, MLP is able to learn non-linearly separable decisions. (Wan et al., 2018) An example of multi-layer perceptron is presented in Figure 6 below.

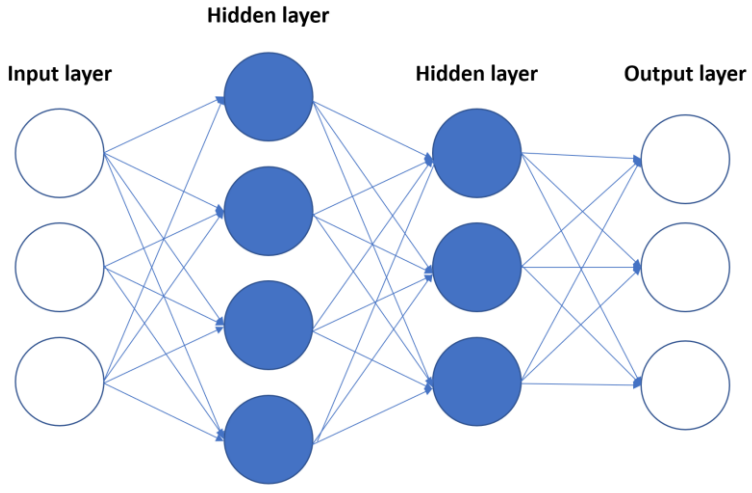


Figure 6. An example of multi-layer perceptron (adapted from Wan et al. 2018).

Except for the input nodes, each node represents a neuron with a nonlinear activation function, and sigmoid and tanh functions are traditionally employed for MLPs with less than five hidden layers, whereas relu or softplus are preferred for deeper MLPs (Wan et al., 2018). The equations for sigmoid, tanh, relu, and softplus, as defined by Ertugrul (2018), are respectively

$$g(x) = \frac{1}{1+e^{-(x)}}, \tag{18}$$

$$g(x) = \left(\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}\right), \tag{19}$$

$$g(x) = \frac{1}{1+e^{-(x)}}, \tag{20}$$

$$g(x) = \max(0, x), \text{ and} \tag{21}$$

$$g(x) = \log(1 + e^{x}). \tag{22}$$

3.7 Hyperparameter tuning

Machine learning models have parameters, called hyperparameters, that cannot be estimated from the data directly (Kuhn & Johnson, 2013). Hyperparameters can have a significant impact on the model performance, so the possible combinations of those parameters should be explored. This process is called hyperparameter tuning, and it is a crucial task when fitting a machine learning model. (Yang & Shami, 2020) There exist multiple approaches for hyperparameter tuning. A common approach is illustrated in the Figure 7 below.

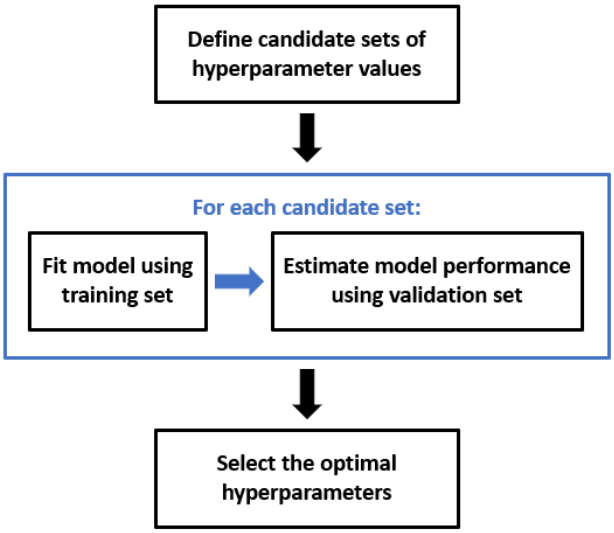


Figure 7. An approach for hyperparameter tuning (adapted from (Kuhn & Johnson, 2013)

As the figure suggests, the first step is to choose candidate sets of parameter values. After that, estimates of the model performance are generated, and finally, the optimal hyperparameters are selected. The estimates of the model performance must be reliable, so the estimates should be generated using samples that were not used for training. (Kuhn & Johnson, 2013) Hence, the remaining data set from which the final test set is excluded, is further divided into training and validation sets as previously discussed. In the next section, practices for performance evaluation are introduced, and those can be applied not only while evaluating the final performance but also while tuning hyperparameters.

3.8 Performance evaluation for classification

In most cases, the performance of a classification model is based on prediction accuracy. That is, the best model has the highest percentage of correct predictions divided by the total number of predictions. (Kotsiantis, 2007) Other widely used performance metrics include precision (number of samples correctly classified as i / number of samples classified as i), recall (number of samples correctly classified as i / number of samples for which the correct class is i), and F1-score ($2Precision \times Recall/(Precision + Recall)$) (Liu et al., 2014).

When calculating this accuracy, the data set can be sampled at least in three alternate ways: 1) the data set is divided into two sets, from which one is used to fit the model and the other to predict the accuracy, 2) using resampling technique called cross-validation, in which the data set is divided into k mutually exclusive and equal sized subsets, and

after that the model is repeatedly fitted and evaluated in a way that each subset is used for evaluating, and 3) using a special case of cross-validation, called leave-one-out-validation, in which all the subsets consists of a single instance. (Kotsiantis, 2007)

When applying sampling in real-world classification problems, a common challenge that has a significant effect on classification performance is class imbalance, meaning that some classes have a significantly higher number of samples than other classes. There are several methods to address this challenge, from which the most used is oversampling. Oversampling means that more samples are added to minority classes. This can be done simply by replicating samples in these classes, or by using more advanced methods. In contrast, another method that results having an equal number of samples in each class is undersampling. In undersampling, samples are removed randomly from dominating classes until the balance is achieved. (Buda et al., 2018)

4. RESEARCH METHODOLOGY

4.1 Research design and strategy

This study aims to apply machine learning models to forecast demand of new products in retail and evaluate the accuracy of the applied models to understand which of the models performs best. The research is conducted primarily for the benefit of the case company: if the machine learning models perform well, the traditional analogical approach, currently used for new product purchase decisions, can be replaced by the more efficient and robust machine learning method. However, new product demand forecasting is a crucial task in retail in general, but accurate forecasts are difficult to generate and research on the topic has been limited (Fildes et al., 2019). Hence, this study can give valuable insights not only for the case company but also for the retail industry.

The purpose is to develop an effortless and objective forecast method by applying machine learning models using the data provided by the case company. Thus, the study is an experimental case study that follows quantitative research design. Quantitative design is a logical choice, as according to Saunders (2019), using numerical and standardised data, examining relationships between variables, and deriving meanings from numbers, are all characteristics of quantitative research, and present in this study, too. The study is based on positivism research philosophy, as the insights of the study are purely based on numerical data and objective measurements, aiming to eliminate subjective views.

The research process is the following: First, the problem is defined based on the needs of the purchasing function of the case company. Second, the raw data for the defined problem, is retrieved from the database of the case company. After that, the data is cleaned and transformed for the applied machine learning models. Then, the hyperparameters of the machine learning models are optimized and the models are trained using the data. Finally, the models are evaluated, and the best performing model is identified based on evaluation metrics.

4.2 Problem definition

The forecast methods are designed for the needs of the purchase function of the case company. The purchased quantities must be optimized in a way that they meet the desired inventory turnover. Both falling below and exceeding the desired inventory turnover can lead to profitability issues, as in the former case the company is losing sales, and in

the latter case capital is tied for an unfavorable long time and the risk of spoilage might materialize. The desired inventory turnover is measured in days in the case company. Here, daily level demand forecasts are vital as purchasers must know how many products will be sold in the desired time frame. It's important to note that to define the optimal order quantity, purchasers don't need to know how many products will be sold each day, but how many products total will be sold during the time frame, which can be converted to average daily sales quantity.

Currently, the optimal demand (i.e. order quantity) for new products is forecasted using a traditional analogical approach if there are comparable products available. Often, there is no feasible comparable products, so the order quantity is simply determined in a way that the order amount measured in euros doesn't exceed a specific, rather small, limit. That minimizes the risk of spoilage and committing too much capital. However, this often leads to situations where the case company loses sales: if the sales of the new product turn out to be great, the product is sold out in few days and, as the company is focused on surplus batches, ordering more is not possible in most cases. This study aims to address this issue, balancing both sides of the equation.

At first, the exact average daily sales quantities for new products were tried to forecast using regression methods in this study, but the results were not feasible. As stated before, new product demand forecasts are very uncertain, so forecasting exact sales is often too ambitious. Thus, another approach was selected instead, aiming to forecast the magnitude of average daily sales. In addition, it was recognized that forecasting sales measured in euros resulted better performance than sales measured in quantities. As a result, the magnitude of average daily sales in euros is selected as a target variable. If the case company could classify the new products based on the magnitude of the average daily sales, the company could define the order limit to be lower for poorly performing products and higher for best performing products. In that way, both minimizing working capital and spoilage, and maximizing sales, are taken into account.

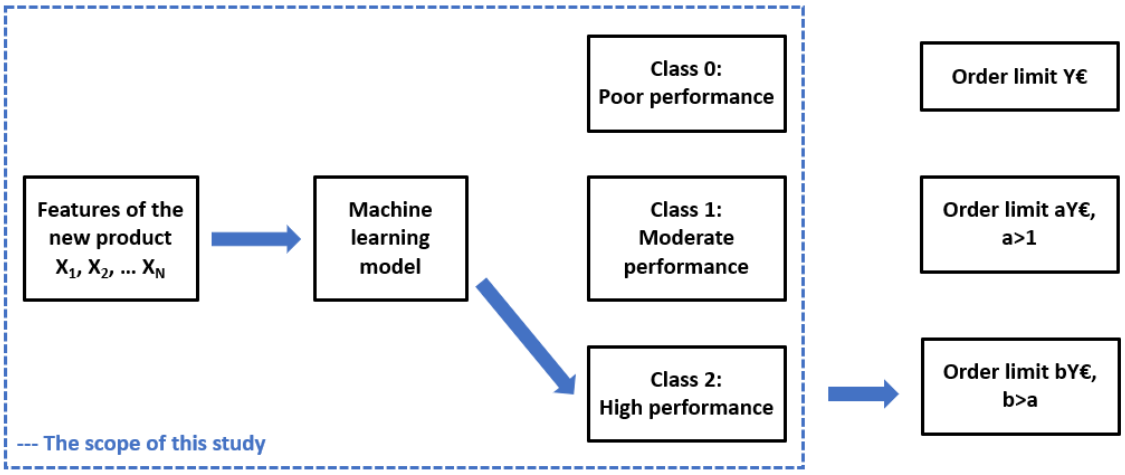


Figure 8. Forecasting process illustrated with an example of high performing product. The scope of the study is limited to the classification task, and the corresponding order limits are defined later by the case company.

The average daily sales vary greatly between products, so more than two classes are needed to define the magnitude of the sales. Thus, the problem is formulated as a multi-class classification problem. The selected number of classes is three, since this illustrates the current classification of products used in the case company in other contexts than forecasting. Each new product has a set of previously known features, and one of the three classes is assigned to the new product based on the features. The relationship between features and classes is derived based on the historical data of existing products, using machine learning algorithms. The case company can then define order limits for each class to guide the purchase decisions of new products. The forecast process is summarized in Figure 8, using high performing product as an example case.

4.3 Data collection and description

The data used in the study is retrieved from the database of the case company. It covers observations between 1st of January 2020 and 20th of August 2021. The rationale behind the selected time frame is the fact that the case company is a start-up, and the size of the business has grown significantly in recent years, so the data older than 2020 is not that relevant anymore. Of course, the nature of the business has developed since the beginning of 2020, but a smaller time frame would have produced an unnecessarily small data set.

The data is aggregated on product level, which is a crucial choice. There is only one sample for each product, and thus the data set doesn't consist of time series data. This is done because the purpose is not to forecast daily fluctuations in sales, but to forecast

the magnitude of average daily sales for a new product. Eliminating time dimension simplifies the modelling task remarkably. Since the data originally was time series data, some of the variables are averages of individual data points, and hence all the sales figures in the data set are average *daily* sales. The sales figures include data only from the dates when the product in question was available. On the whole, the data set retrieved from the database covers the variables introduced in the table Table 1 below.

Table 1. The variables of the data set.

Variable	Description	Type
Product ID	Product identification number	Categorical
Product name	Name of the product	Text
Main category ID	Identification number of the main category the product	Categorical
Main category name	Name of the main category the product belongs to	Text
Sub-category ID	Identification number of the sub-category the product	Categorical
Sub-category name	Name of the sub-category the product belongs to	Text
Brand ID	Identification number of the brand of the product	Categorical
Brand name	Name of the brand of the product	Text
Vat rate	Vat rate of the product: either 0.1, 0.14, or 0.24	Categorical
Keywords	Words that can be used for searching the product in the	Text
Days on sale	Number of days the product has been on sale during	Numerical
Average quantity	Average daily sales quantity of the product during the	Numerical
Total quantity	Total sales quantity of the product during the time frame	Numerical
Average product sales	Average daily sales of the product in euros during the	Numerical
Total product sales	Total sales of the product in euros during the time	Numerical
Average price	Average selling price of the product in the case	Numerical
Average reference price	Regular price of the product in other stores	Numerical
Average company sales	Average daily sales of the case company in euros during the days the product has been on sale	Numerical
Average days to BBD	Average days to best before date of the product during the time frame (negative number if the date is exceeded)	Numerical
Average main category sales	Average daily sales of the main category in euros the days the product has been on sale excluding the sales of the product	Numerical
Average sub-category sales	Average daily sales of the sub-category in euros during the days the product has been on sale excluding the sales of the product	Numerical
Average brand sales	Average daily sales of the brand in of the product in euros during the days the product has been on sale excluding the sales of the product	Numerical
Average brand quantity	Average daily sales quantity of the brand of the product in euros during the days the product has been on sale excluding the sales of the product	Numerical
Average discount	Average discount of the product (the relation between average price and average reference price)	Numerical

As the table shows, both *average price*, *average reference price,* and *average discount* are assigned to each product. That's because the case company sells all the products at a discount compared to regular grocery stores as the company focuses on surplus batches. Another important thing to note is that when the case company sets the dis-

counted prices, the prices are intended to be part of either the normal offers or the marketing offers. If the product is marketed, the demand increases significantly, and the sales data is no more relevant to normal demand forecasting. Also, the purchasing process for marketed products is different from the normal process, and thus demand forecasting for such products is not addressed in this thesis. As a result, the data set includes only sales data from the dates when a specific product hasn't been marketed. Lastly, only the products that have been on sale more than four days are included in the data set to make the data less sensitive to daily fluctuations in sales figures.

4.4 Data pre-processing, feature extraction, and feature selection

The data set introduced in the previous chapter needs to be processed to be suitable for the applied machine learning models. In other words, a target variable and a feature representation of the data set must be constructed, mostly using tools and methods presented in the literature review.

First, some new variables were created. The target variable *Class* was derived from average daily sales by addressing one of the three classes, defined by the equation (23), to each product.

$$Class = \begin{cases} 0, \text{if } average\ product\ sales < a \\ 1, \text{if } a \leq average\ product\ sales < b \\ 2, \text{if } average\ product\ sales \geq b \end{cases} \quad \textbf{(23)}$$

where a and b are predefined positive thresholds measured in euros, and $a < b$.

Another created new feature is *Name and keywords.* In the original data set, each product has two separate variables, *Name* and *Keywords,* but these were combined to one text feature to process text data more efficiently.

Second, irrelevant features were removed. Some variables were retrieved from the database even though they are not useful for the machine learning modelling because they might be useful if the same data set was used for further analysis later. These variables are *Product ID, Main category name, Sub-category name, Brand ID, Brand name, Days on sale, Average quantity, Total quantity,* and *Total product sales.* In addition, *Average product sales, Name,* and *Keywords* were removed since new variables were created based on these variables, so they are not used anymore.

Then, null values were handled. Three variables of the original data set included null values: *Main category ID, Sub-category ID,* and *Average days to BBD*. If *Main category*

ID or *Sub-category ID* were null, the sample in question was simply removed. If *Average days to BBD* was null, average value of all the BBDs was assigned to the sample.

For further processing, the data was splitted to train and test sets, where the test set covers randomly 20 % of the available samples. That's because most of the pre-processing tasks must be done based on the nature of the train set as test set is assumed to be unknown by the model. The train set is further divided into three folds for cross-validation as described later.

When it comes to text features, the text data was cleaned by removing special characters, numbers, and short words, and by transforming all the words to lowercase. Then, all the words were stemmed, meaning that they were transformed to their root form. After that, TF-IDF representation was created and LSA, defined in the equation (5), was performed to truncate the TF-IDF into 10 term features. Finally, categorical variables were one-hot encoded. Table 2 below summarizes the features before feature selection.

Table 2. Features constructed from the original data set.

Feature	Description	Type
Class (target variable)	Illustrates the magnitude of daily sales in euros of the product	Class (0, 1, or 2)
Main category ID	Main category the product belongs to	One-hot encoded
Sub-category ID	Sub-category the product belongs to	One-hot encoded
Vat rate	Vat rate of the product	One-hot encoded
Name and Keywords	Product name and keywords combined	LSA representation
Average price	Average selling price of the product in the case	Continuous number
Average reference price	Regular price of the product in other stores	Continuous number
Average company sales	Average daily sales of the case company in euros during the days the product has been on sale	Continuous number
Average days to BBD	Average days to best before date of the product during the time frame	Continuous number
Average main category sales	Average daily sales of the main category in euros the days the product has been on sale excluding the sales of the product	Continuous number
Average sub-category sales	Average daily sales of the sub-category in euros during the days the product has been on sale excluding the sales of the product	Continuous number
Average brand sales	Average daily sales of the brand in of the product in euros during the days the product has been on sale excluding the sales of the product	Continuous number
Average brand quantity	Average daily sales quantity of the brand of the product in euros during the days the product has been on sale excluding the sales of the product	Continuous number
Average discount	Average discount of the product	Continuous number

Feature selection was performed for numerical features using filtering methods Pearson's correlation and mutual information. If absolute value of correlation between two features was more than 0,60, one of the features were removed. Dependency between features and the target variable were measured using mutual information statistic, and

only the features with relatively high dependency on the target variable were remained. In addition, the feature removed based on Pearson's correlation is the one that has a smaller mutual information value.

After that, numerical features and LSA text features were scaled using all three introduced scaling methos, standardization, min-max scaling, and robust scaling, one at a time. As a consequence, the process for optimizing the machine learning models, described in the next section, was repeated four times: first, the models are optimized using features that are not scaled, and after that the models are optimized again using the three scaling methods. The best performing method, based on the prediction accuracy, was selected for further evaluation process.

4.5 Machine learning models

For the classification task, five multiclass classification algorithms were utilized: 1) logistic regression, 2) support vector classification, 3) nearest neighbors, 4) XGBoost, and 5) multi-layer perceptron (MLP). Theoretical background of the algorithms is introduced in the chapter 3.6. The models were trained using feature engineered training data. First, the training data was oversampled so that all the classes have equal number of samples. Then, hyperparameters of the algorithms were tuned through the process illustrated earlier in the Figure 7. For hyperparameter tuning, candidate sets of parameters were defined. In addition, previously introduced sampling technique called cross-validation was used so that the training set were divided randomly into three equal sized folds.

The best hyperparameters, i.e. the parameters that had the highest prediction accuracy, were chosen for the final models. The final hyperparameters are not introduced here as the parameters are optimized for the specific needs of the case company, and the parameters must be optimized separately for other use cases. After hyperparameter tuning, the models were trained using the selected parameters and the whole training data. Some models are initialized to a random state and thus the results depend on the used state. Hence, the models are trained 30 times using random state values from 1 to 30 to get a range of predictions. Different random states are not applied to the nearest neighbors model as it doesn't depend on the random state. In addition, with specific hyperparameters, some other applied models might be independent on the random state. Those models are still trained using different random states since, with some hyperparameters, the random state has effect on the model.

Finally, the models were evaluated, and the best final model was selected. Figure 9 below illustrates the evaluation and model selection process. For evaluation, the test set

was used to get reliable results of the model performance. As described in the previous section, the features were scaled using three different methods, so the model training process was repeated four times, one for each scaling method and one for the original data without scaling. Before more detailed model comparison, the scaling method was selected for each of the five models. The metric for evaluating the models with different scaling methods was prediction accuracy, introduced in the chapter 3.8. As mentioned above, the models were trained several times using different random states to get a range of accuracies. The final accuracy is the average of the accuracies for the model, and the lowest and the highest accuracies are also reported to understand how sensitive the models are to different random states. The best performing scaling method, based on the prediction accuracy, was selected for each of the models individually.

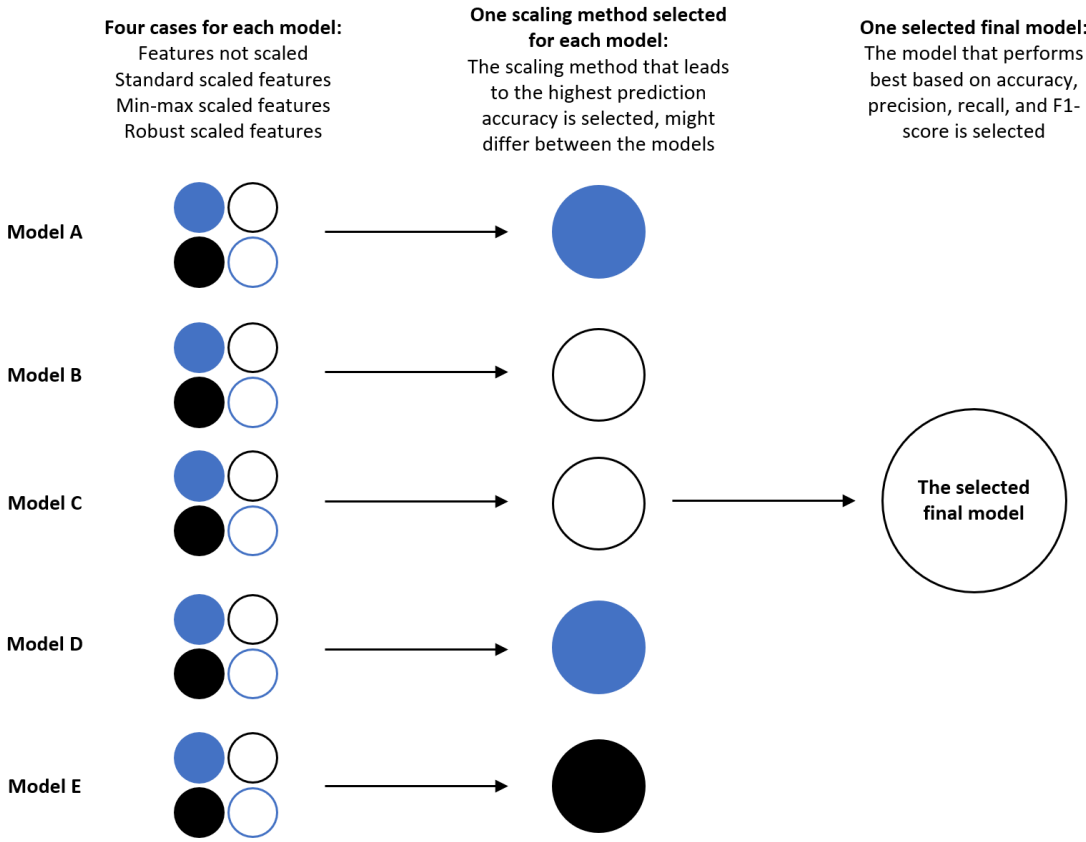


Figure 9. The evaluation and model selection process.

After that, the models with the selected scaling methods were compared, and the final model was selected. In addition to accuracy, the models were evaluated using performance metrics precision, recall, and F1-score. Since a multiclass classification problem is addressed, the precisions, recalls, and F1-scores were calculated separately for each class and the reported results are weighted averages of the class specific results. As a benchmark for all the models, accuracy, precision, recall, and F1-score of predicting the

most common class of the training set for each test sample was calculated. Naturally, the selected model should perform better than the benchmark. The model that achieved the best performance, measured by these metrics, was selected as the final model. Accuracies have the greatest weight if there is variation between the ranks of the models determined by the different metrics. The evaluation was based on the metrics for test data only, but also training accuracies are reported to detect the possible signs of overfitting.

4.6 Forecasting setup

The modelling was done using a programming language Python and a development environment called Spyder (Scientific Python Development Environment). In addition, Pandas, NumPy, SciPy, Scickit-learn, Matplotlib, Seaborn, Imblearn, NLTK, and XGBoost Python libraries were used.

4.7 Validity and reliability of the methodology

When developing the introduced research process, the aim was to ensure high quality by acknowledging potential threats to validity and reliability. Validity considers the accuracy of the analysis of the results, the appropriateness of the measures, and the generalisability of the findings. (Saunders, 2019) In more detail, there are three types of validity in quantitative research: content validity, construct validity, and criterion validity. Content validity refers to the ability of the research instrument to measure accurately all aspects of the construct, construct validity refers to the degree to which the research instrument measures the intended construct, and criterion validity refers to the degree to which the research instrument is related to other instruments measuring the same variables. (Heale & Twycross, 2015)

Content validity was taken into account in the research design, for example, by including a large number of features into the data set and removing through a systematic process the features that were not relevant for forecasting. Thus, only the relevant features should affect model performance. Further, used data set might have impact on construct validity: If the data set contained sales dates when the product in question was out of stock, would the model forecast sales rather than demand? Finally, criterion validity was considered for example by using cross-validation. However, it’s important to note that this is a case study, and the models were optimized using the data from the case company. Hence, the findings should be carefully analysed before applying them to other new product demand forecasting cases.

Reliability, in turn, refers to replication and consistency of the study (Saunders, 2019). Further, according to Heale and Twycross (2015), the attributes of reliability in quantitative research are internal consistency, stability, and equivalence. They define internal consistency as the extent to which all the items on a scale measure one construct, stability as the consistency of results using an instrument with repeated testing, and equivalence as the consistency among responses of many users of an instrument or among different forms of an instrument. In this study, reliability was considered by following a general and robust machine learning workflow when developing the models. For example, several random states were used to understand how reliable the achieved results are. Still, the models were optimized using data from a specific timeframe, so the results would likely be different if a data set from another timeframe was used. To conclude, it's clear that there are some limitations one needs to be aware of, even though the research process was designed carefully. The limitations of this study are discussed in more detail in the chapter 6.3.

5. RESULTS

5.1 Feature selection

Dependencies between the input variables were measured using Pearson's correlation, and Figure 10 presents the results. The absolute value is greater than 0,60 for two pairs of variables: *Average price – Average reference price*, and *Average brand sales – Average brand quantity*. To simplify the model and decrease amount of duplicate information, one variable from each pair is removed. The removed variable is the one that has smaller mutual dependency on the target variable, considered below.

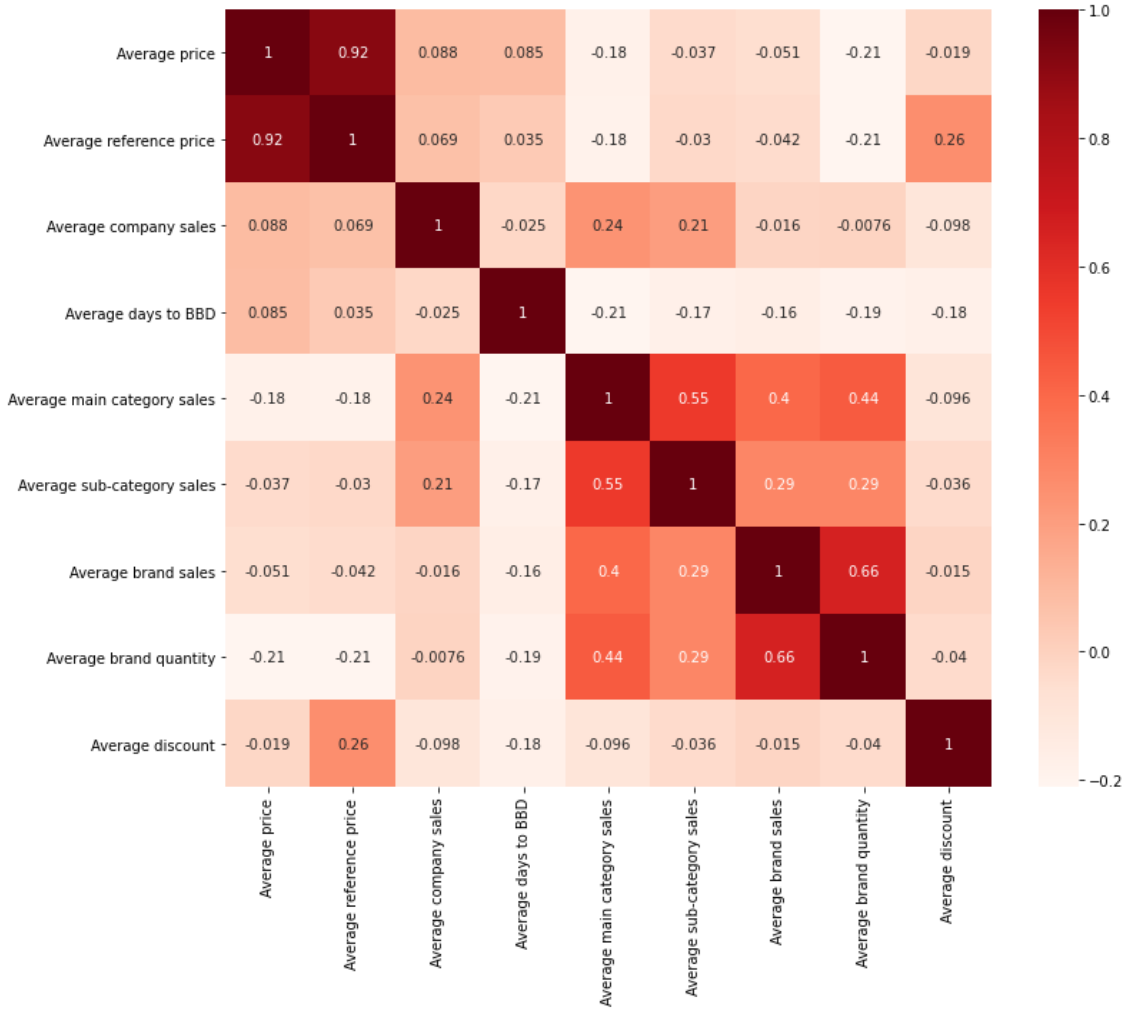


Figure 10. Pearson's correlation between input variables.

Dependencies between the input variables and the target variable were measured using mutual information statistic, and Figure 11 presents the results. As the figure shows, *Average discount* is unlikely to be as relevant as the other variables. Hence, it is removed. When it comes to the pairs identified using Pearson's correlation, *Average price*

and *Average brand quantity* are less relevant than *Average reference price* and *Average brand sales*, so *Average price* and *Average brand quantity* are removed.

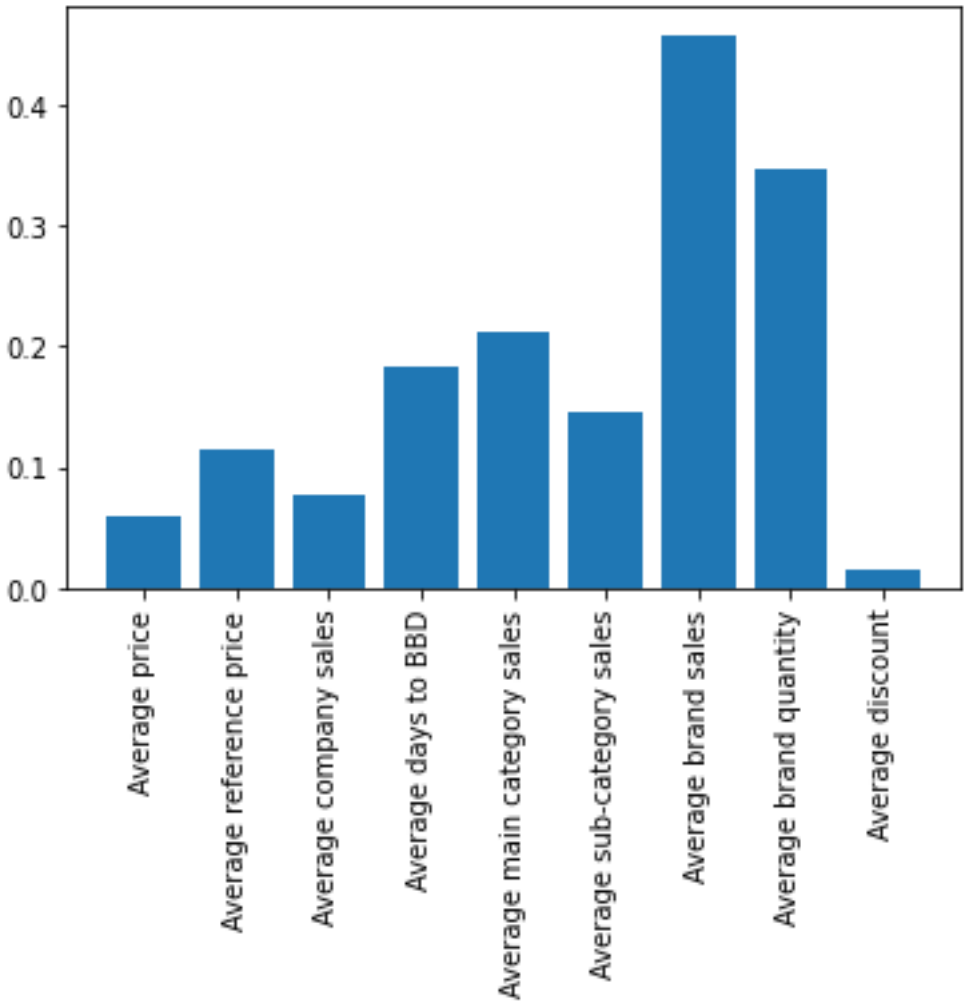


Figure 11. Mutual information of the target variable and the input variables.

To conclude, three features were removed, and the final set of features consists of six numerical features: three categorical features, and one LSA text feature presented as a 10-dimensional vector.

5.2 Scaling method selection

The set of figures below presents the accuracies for each model using the original features that are not scaled, standard scaled features, min-max scaled features, and robust scaled features. The blue marks indicate average accuracies, and grey marks minimum and maximum accuracies.

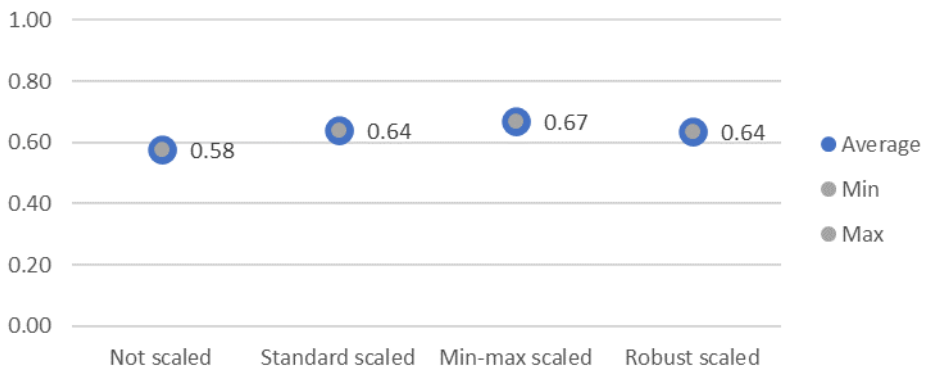


Figure 12. Accuracies for the support vector classification model using different scaling methods.

Figure 12 shows the accuracies for support vector classification. The model seems to be sensitive to features with different scales, so scaling must be implemented. The best performing method is min-max scaling, so it's selected for this model. Based on the results, the model is not sensitive to different random states as there is no variation between average, minimum, and maximum accuracies.

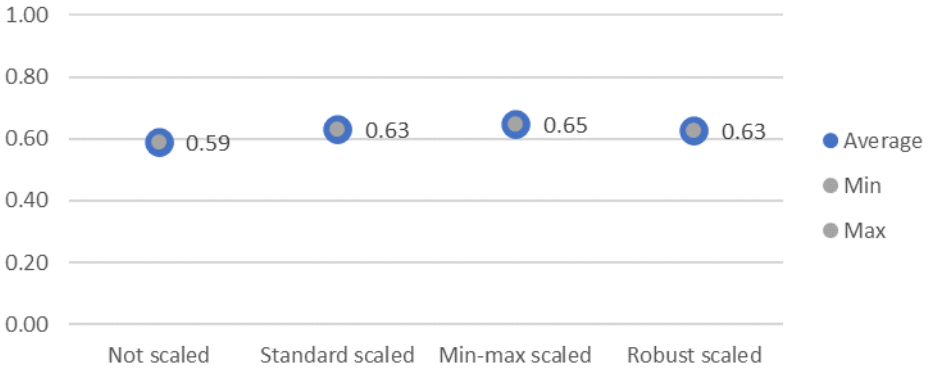


Figure 13. Accuracies for the nearest neighbour model using different scaling methods.

Figure 13 shows the accuracies for nearest neighbour. The model seems to be sensitive to features with different scales, so scaling must be implemented. The best performing method is min-max scaling, so it's selected for this model. Based on the results, the model is not sensitive to different random states as there is no variation between average, minimum, and maximum accuracies.

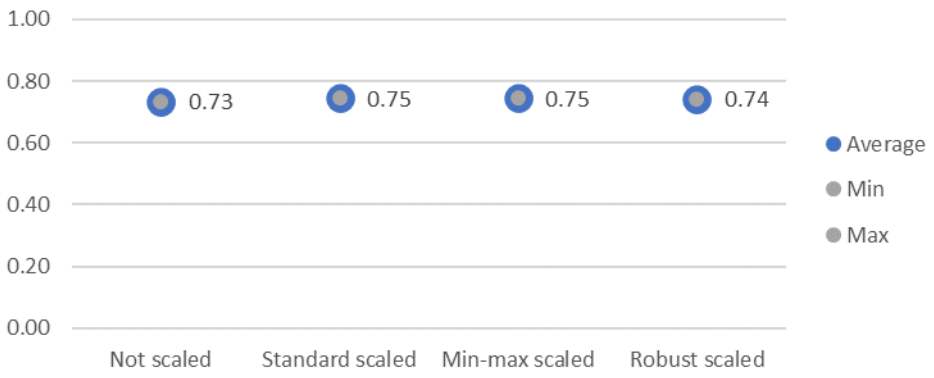


Figure 14. Accuracies for the XGBoost model using different scaling methods.

Figure 14 shows the accuracies for XGBoost. There are no significant differences between the original and the scaled features. Hence, there is no need for feature scaling, which simplifies the workflow. One could argue that scaling is preferred as the not scaled features lead to 0,02 accuracy decrease. However, the decrease is rather small, and as a decision-tree based method, XGBoost shouldn't be sensitive to the scale of the features. Thus, the difference is not directly explained by the scale of the features. In addition, based on the results, the model is not sensitive to different random states as there is no variation between average, minimum, and maximum accuracies.

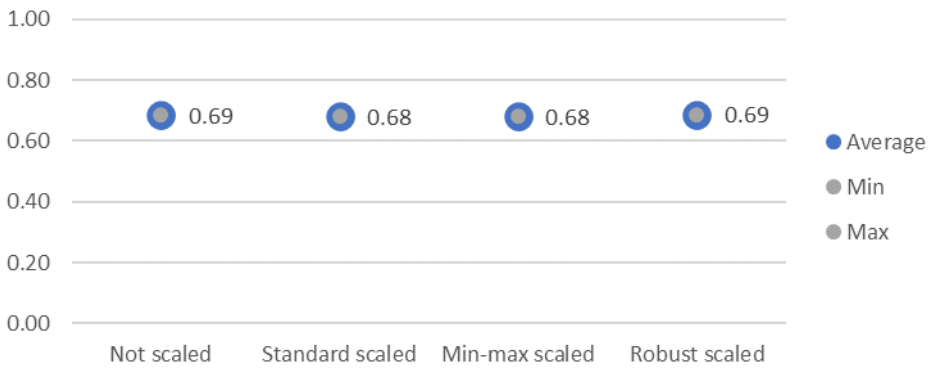


Figure 15. Accuracies for the logistic regression model using different scaling methods.

Figure 15 shows the accuracies for logistic regression. There are no significant differences between the original and the scaled features. Hence, there is no need for feature scaling, which simplifies the workflow. Based on the results, the model is not sensitive to different random states as there is no variation between average, minimum, and maximum accuracies.

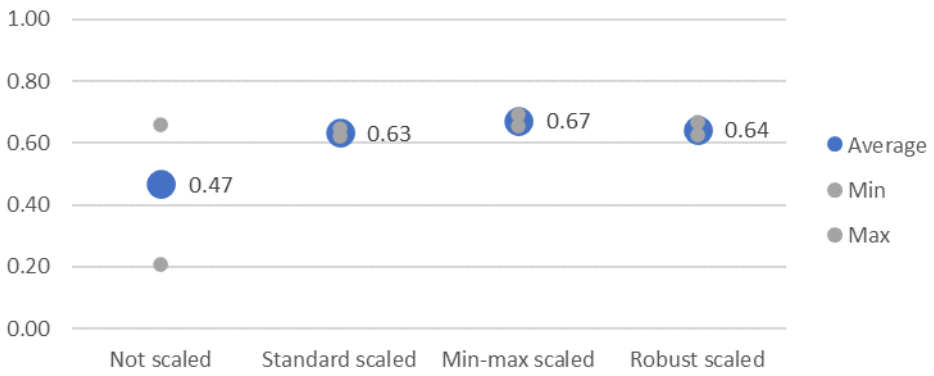


Figure 16. Accuracies for the multi-layer perceptron model using different scaling methods.

Figure 16 shows the accuracies for multi-layer perceptron. The model seems to be sensitive to features with different scales, so scaling must be implemented. The best performing method is min-max scaling, so it's selected for this model. Based on the results, the model is sensitive to different random states as there is variation between average, minimum, and maximum accuracies when the features are not scaled. However, with the suitably scaled features, the variation is not significant, and the model seems to be reliable regardless of the random state.

To conclude, support vector classification, nearest neighbours, and multi-layer perceptron models require feature scaling, and the best accuracies for each model were achieved using min-max scaling, so it's the selected scaling method for these models. On the other hand, XGBoost and logistic regression models are not sensitive to different scales, so the features are not scaled for these models. In addition, none of the models is significantly sensitive to different random states when the selected scaling method is implemented. Moreover, the only model random states have any effect on is the multi-layer perceptron. Still, it's important to note that this only applies with the chosen hyperparameters, and with different hyperparameters some other models might be affected by random states as well.

5.3 Final model selection

The results introduced in this section are based on the selected scaling methods, and for simplicity minimum and maximum metrics are not presented since it was shown that no significant variation exists. More detailed metrics, including minimum and maximum values, are presented in the Appendix A. As mentioned, the final hyperparameters are not introduced here. The parameters are optimized for the specific needs of the case company, and the parameters must be optimized separately for other use cases. For

transparency, the hyperparameters and the candidate sets are provided in the Appendix B.

Figure 17 presents the accuracies of the five machine learning models. The orange line is the benchmark accuracy 0,41, and the highest accuracy is highlighted in blue. Based on the test accuracies, XGBoost is the best model for forecasting new product demand in the case company with the accuracy of 0,73. Still, it's interesting to see that even the worst machine learning model, nearest neighbors, leads to 0,65 accuracy. The difference in accuracy between the best and the worst accuracies is notable, but surprisingly low. Furthermore, all the models perform significantly better than the benchmark model, which predicts the major class among training samples for all the test samples.

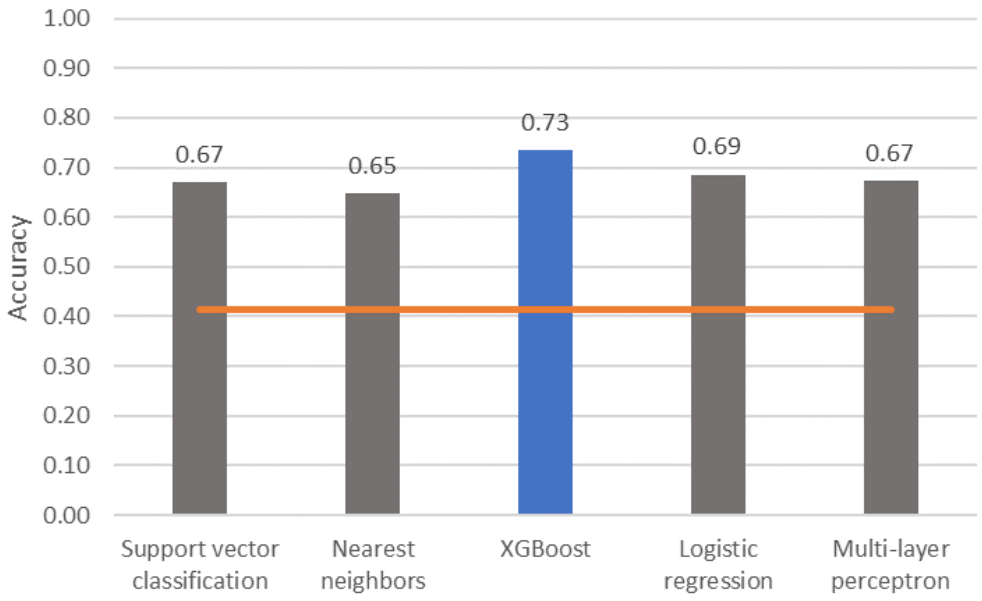


Figure 17. Accuracies of the machine learning models.

Weighted precisions, recalls, and F1-scores are presented below in Figure 18, Figure 19, and Figure 20, respectively. The figures support the previous result: XGBoost is the best performing model based on all the metrics, but the differences between the best and the worst values are relatively low. In line with the accuracies, all the models perform significantly better than the benchmark, for which precision is 0,17, recall 0,41, and F1-score 0,24.

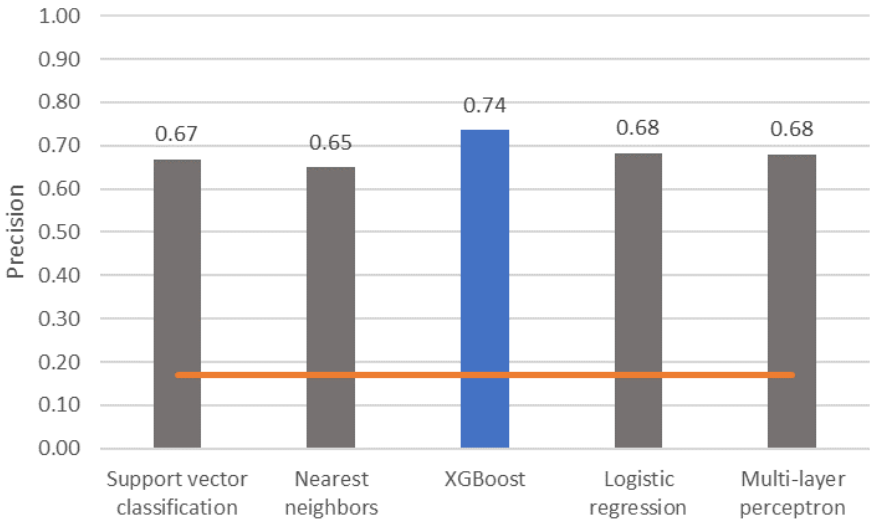


Figure 18. Precisions of the machine learning models.

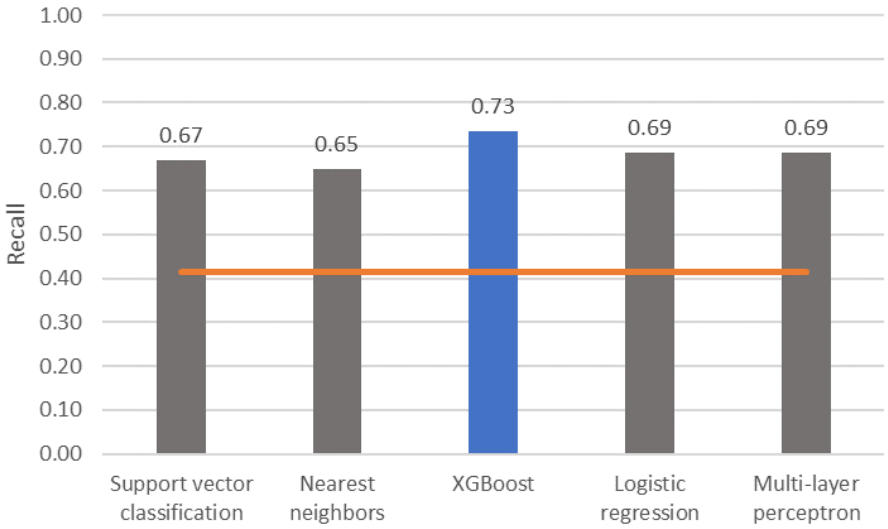


Figure 19. Recalls of the machine learning models.

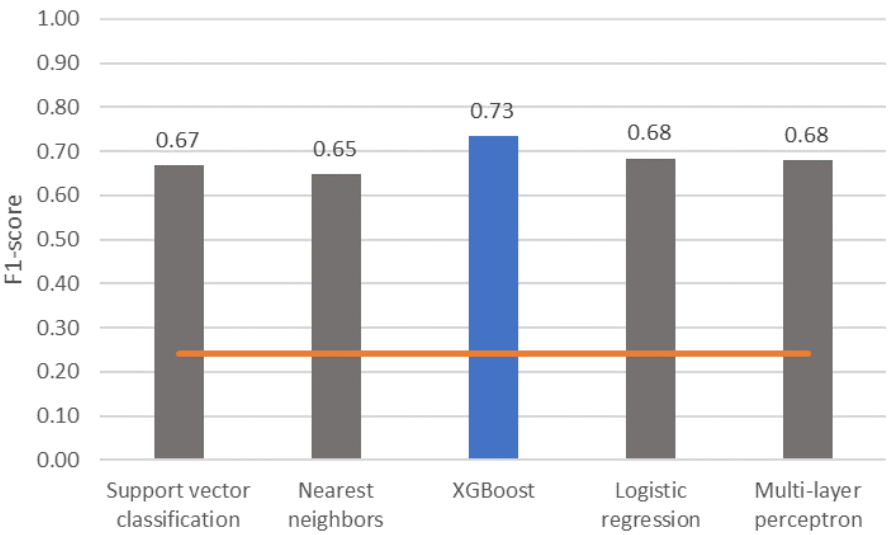


Figure 20. F1-scores of the machine learning models.

Finally, test and train accuracies are compared in the Figure 21. The figure shows that the accuracy for training samples is significantly better than the accuracy for test samples for support vector classification, nearest neighbors, and XGBoost. This is a possible sign of overfitting and quite poor generalization ability. One method for avoiding overfitting is early stopping, in which the training is stopped before training accuracy reaches the highest value. However, testing different early stopping rules didn't led to better test accuracies, so additional rules were not added to the selected hyperparameters. Thus, in the light of current knowledge, the best accuracies that can be achieved with the models are the presented test accuracies.

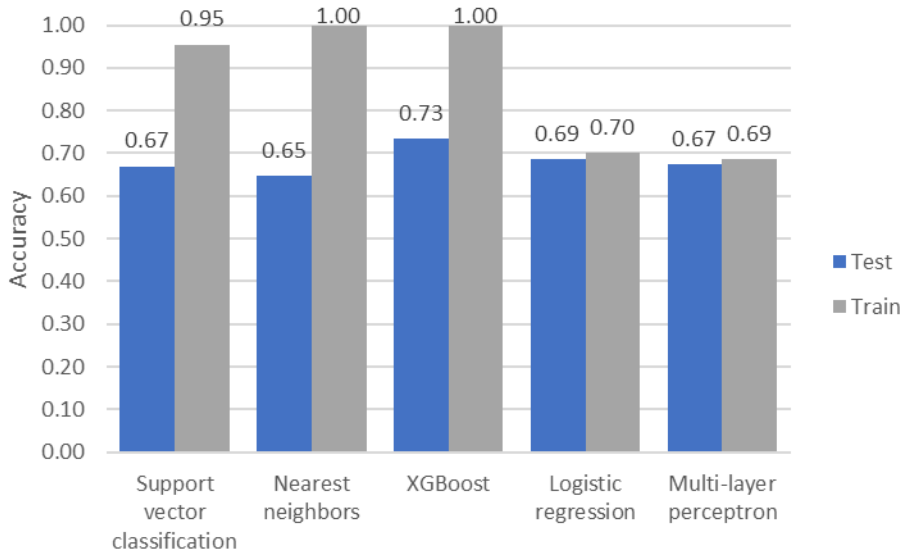


Figure 21. Test and train accuracies of the models.

Based on the reviewed metrics, the best and thus the selected model, is XGBoost. XGBoost overperforms the other four models and the benchmark model in all analysed metrics. Still, if it gives signs of even worse generalization when implemented in the case company or if the amount of available data increases, multi-layer perceptron or logistic regression could be considered. For these two models, there is no significant difference between training and test accuracies, and the models led to the second and the third highest accuracies among the applied models.

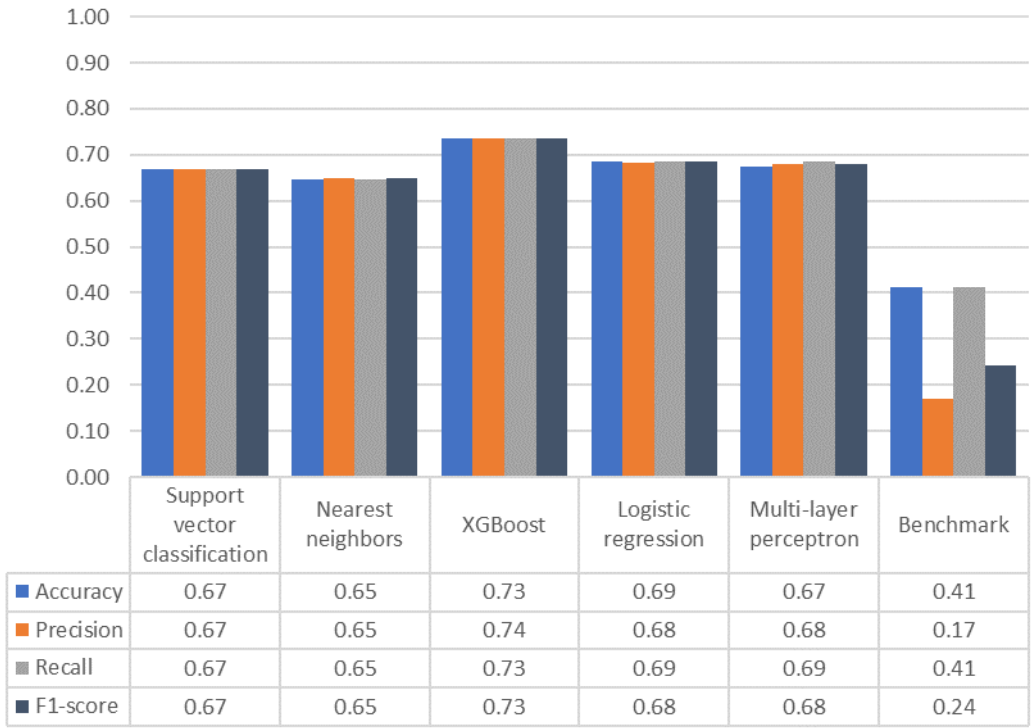
5.4 Summary of the results

Feature selection using Pearson's correlation and mutual information statistics led to removal of three features: *Average price, Average brand quantity,* and *Average discount*. When it comes to the feature scaling, the features must be scaled using min-max scaling

when applying support vector classification, nearest neighbors, or multi-layer perceptron. Logistic regression and XGBoost don't require scaled features. Based on the accuracy, precision, recall, and F1-score, XGBoost is the best and thus the selected model. However, there are signs of overfitting that must be acknowledged. The final features are presented in Table 3, and the final evaluation metrics in Figure 22 below.

Table 3. The final features.

Feature	Description	Type
Main category ID	Main category the product belongs to	One-hot encoded
Sub-category ID	Sub-category the product belongs to	One-hot encoded
Vat rate	Vat rate of the product	One-hot encoded
Name and Keywords	Product name and keywords combined	LSA representation (10 dimensional vector)
Average reference price	Regular price of the product in other stores	Continuous number
Average company sales	Average daily sales of the case company in euros during the days the product has been on sale	Continuous number
Average days to BBD	Average days to best before date of the product during the time frame	Continuous number
Average main category sales	Average daily sales of the main category in euros the days the product has been on sale excluding the sales of the product	Continuous number
Average sub-category sales	Average daily sales of the sub-category in euros during the days the product has been on sale excluding the sales of the product	Continuous number
Average brand sales	Average daily sales of the brand in of the product in euros during the days the product has been on sale excluding the sales of the product	Continuous number



	Support vector classification	Nearest neighbors	XGBoost	Logistic regression	Multi-layer perceptron	Benchmark
Accuracy	0.67	0.65	0.73	0.69	0.67	0.41
Precision	0.67	0.65	0.74	0.68	0.68	0.17
Recall	0.67	0.65	0.73	0.69	0.69	0.41
F1-score	0.67	0.65	0.73	0.68	0.68	0.24

Figure 22. The evaluation metrics of the machine learning models.

6. CONCLUSION

6.1 Key findings

The objective of this thesis is to create a method for forecasting new product demand for purchasing decisions in retail by applying and evaluating several machine learning models and selecting the one that performs best. It was found that regression methods didn't lead to feasible results, so the demand forecasting problem was formulated as a classification problem instead. Hence, five classification models were applied in this thesis to achieve the objective. Its' not necessary to forecast daily demand fluctuations for purchase decisions, so the classes were derived from average daily sales measured in euros. In other words, the three target classes indicate the magnitude of demand measured in euros.

The best model, based on the prediction accuracy, precision, recall, and F1-score is the XGBoost model. Using XGBoost, 0,73 accuracy can be achieved, which is 0,32 higher than the accuracy given by the benchmark model. For XGBoost, it's not necessary to scale the features, and the class is predicted using 10 selected features introduced earlier in Table 3. The achieved accuracy can be considered as a good result, since it's significantly better than the accuracy of the benchmark. Moreover, it's well known that new product demand forecasts are uncertain regardless of used forecast method, so it's not realistic to expect full accuracy. Consequently, we suggest the case company to apply the selected XGBoost model for forecasting new product demand for purchasing decisions.

However, it was surprising how small the performance differences are between the tested models: even the weakest model resulted 0,65 accuracy, which is only 0,08 below the selected XGBoost model. In addition, signs of overfitting appeared when comparing the train and test accuracies of the XGBoost model, but better results couldn't be achieved by using early stopping. In this thesis, the model selection was done based on evaluation metrics for test samples, so despite of possible overfitting, the XGBoost model was selected. Still, the case company should keep in mind the relatively small performance differences between the models and the signs of overfitting when applying the selected XGBoost model. The case company will have more data available over time and fitting the tested models again using a larger data set might lead to 1) different accuracies, precisions, recalls, and F1-scores, and thus to a different selected model, or 2) reduction of the signs of overfitting, and thus to a more robust XGBoost model.

In all, this study can be considered successful since from the five tested models we were able to select a method for forecasting new product demand with a favorable accuracy, and the method is designed especially for the needs of purchasing function in the case company. Thus, the objective was fully achieved. The implications of the findings are discussed in more detail in the next chapter.

6.2 Theoretical and practical implications

The findings yield to several theoretical and practical implications. First, as Fildes et al. (2019) states, the most common method for new product forecasting is an analogical approach, where new products are assumed to behave similarly to comparable products, and this method is typically partly judgemental as the identification of comparable products often require judgement. In this thesis, the demand forecast is derived from historical data of other products, but there is no need judgemental identification since the forecast is based on relationships between demand and product features rather than demand associated with few comparable products.

Second, even though machine learning methods are already applied for demand forecasting, the typical approach is to formulate the forecasting problem as a regression problem (Carbonneau et al., 2008). In this thesis, another approach was successfully implemented, when the forecasting problem was formulated as a classification problem. We found that regression methods were incapable of achieving feasible performance when applied to new product demand using a rather small data set. Aiming to forecast a magnitude of demand rather than an exact demand is an interesting approach since, in such an uncertain situation, the magnitude often gives an adequate level of information for decision making. On the contrary, aiming for an exact demand might be too ambitious.

The practical implications are significant especially from the case company's point of view. The case company gets a new product demand forecasting method for procurement needs that is ready for implementation. Since the selected model is already optimized for the case company, the only thing to do is to define the purchase limits for the three classes. The developed new method eliminates many problems related to the current method the case company uses. In the developed method, new products are not compared to subjectively chosen existing products, so the forecasts are not sensitive to subjective opinions, and the forecasting process is automated and effortless. Switching to the automated process is a huge advantage for the case company since new product demand forecasts are needed on a daily basis, and the current process is time-consuming, requiring human effort for choosing comparable products.

Another critical issue with the current analogical method is that often there are no suitable comparable products available. In that case, the ordered amount measured in euros must fall below a specific, rather small, limit. As mentioned, this often leads to situations where the case company orders too small quantities and loses sales when the supply doesn't meet the demand. The new model doesn't require a specific similar product for forecasting. It's enough that similar feature values a new product has are present in the data used for optimizing the model, so it's unlikely that the new model is unable to generate forecast for a new product. Suitable order quantities can be derived from the forecasts generated by the developed method, and that minimizes the risk for too large an order that would lead to spoilage and increased working capital, and the risk for too small an order that would lead to lost sales. This should significantly increase the sales and profitability of new products in the case company.

The case company has already planned to implement the model for demand forecasting for purchasing decisions. In addition to that, the company has noticed other use cases for the findings. For example, by analysing the categorization of new products the case company could identify characteristics that correlate with good (or weak) performance and understand better which product types should be kept in (or cut off from) the selection. Further, the developed machine learning workflow can be repeated in the future to achieve even better performance, when the company will have more data for optimizing the models.

Finally, suggesting an alternative approach to widely used partly judgemental analogical methods and regression methods is beneficial for new product forecasting in general, whether it was done for estimating demand, sales, or something else. Even though the models must be optimized again if they are used in other contexts, the provided machine learning workflow offers tools for that and it's easy to follow. This study proved that new product demand can be forecasted using machine learning classification methods, which contributes to increased accuracy, efficiency, sales, and profitability.

6.3 Limitations and quality assessment of the study

A few limitations of this study need to be acknowledged. First, the machine learning models were optimized for the needs of the case company only. Thus, the results can't be directly generalized to other retail stores. The models must be optimized again if used in different circumstances, and it's possible that some other algorithm performs better than XGBoost.

In addition, several assumptions and decisions were made when designing the machine learning models. To begin with, the three classes were derived based on classification used in other use cases in the case company. It's possible that different number of classes or different thresholds for converting average daily sales to classes might have led to better performance. Second, TF-IDF vector were truncated to 10 LSA features, and it's unknown if different number of LSA features performed better. Also, the candidate sets for hyperparameter tuning were selected at once. Even though a large number of different candidate sets were used, an alternative approach, in which sets with rough grids are selected first, and then the vicinity of the best set is divided to a new bunch of sets for another tuning round, might have resulted even better model performance.

Furthermore, the feature engineering process was essentially similar for all the applied models. However, the requirements for features is different for each model, and it would have been better to design a separate feature engineering process for each of the models. Feature engineering is a time consuming phase in machine learning workflow, so it wouldn't have been possible to design separate processes for five models in the scope of this thesis, and applying more models was prioritized. Still, some steps were done individually for each model, such as the selection of the scaling method, which increases the quality of the study.

Lastly, there are a couple concerns regarding the used data. First, there was limited amount of data available in the case company. The decision to use data from 2020 onwards is justified, but still it would have been possible to achieve more accurate results with larger amount of data. The data included one sample for each product only, and it's not known how dividing one product into several samples would have affected the results. Also, average daily sales figures were aggregated from daily sales, which include the dates when products were available for a part of the day only. It's naïve to assume that sales equal demand on such days. However, there are only a few partial sales dates and many of them are nearly complete, so there was no desire to remove those dates to further reduce the amount of data. In addition, the creation of train and test sets was performed randomly, and different split might have led to slightly different results.

When these limitations are kept in mind, the research and the results can be considered to have high quality. The study was designed in a way that contributes validity and reliability. For example, the sensitivity for different random states was reviewed, and it was shown that the models are robust and don't depend significantly on the random state. Further, the hyperparameters were tuned through cross-validation for all the models individually to ensure that the models are optimal for the case company. Even though the

hyperparameters weren't addressed in more detail in this paper, the hyperparameters are provided in the appendix for transparency.

6.4 Proposals for future research

This study reveals potential directions for future research. To begin with, the models designed in this study could be developed further. One could aim to eliminate the limitations presented in the previous chapter by 1) dividing the products into classes based on different criteria, 2) testing a wider range of candidate sets when tuning the hyperparameters, 3) designing a separate feature engineering process for each model, and 4) using a data set with different time frame and aggregation. At least the selected model could be critically examined and further developed to make sure that the assumptions and decisions behind its design are optimal. In addition, since only five machine learning models were tested in this study, it would be interesting to test more machine learning models or completely different forecasting methods.

To generalize the results of this study, the machine learning models could be applied using data from other retail stores. Moreover, the classification approach could be studied in other use cases than demand forecasting. As Bacchetti and Saccani (2012) state, classification of stock keeping units has not received much academic attention, even though it could be a powerful tool for identifying not only demand structures but also other characteristics, such as costs and supply uncertainty. In this study, new products were successfully classified to forecast demand, so it would be tempting to see if other product characteristics, for example price elasticity or contribution to the sales of the whole company, could be identified with classification methods.

In this study, it was recognized that regression algorithms were unsuccessful to forecast new product demand using the data of the case company, but it's not known why that happened. One potential reason could be the limited amount of data. As a consequence, regression algorithms could be examined further using a larger data set. Currently, it's difficult to increase the amount of data in the context of the case company. However, the amount of available data is continuously increasing with accelerating speed as the case company is growing rapidly. Hence, machine learning models based on regression algorithms may come into question in the case company in the future. Naturally, regression algorithms could be tested using larger data sets available from other sources already today.

Finally, larger amount of data would make it possible to successfully design separate models for different product types. For example, the case company sells both food and

non-food products, and these two product types might have fundamentally different behaviour patterns. Even though the used feature, *vat rate*, distinguishes the food and non-food samples from each other, it's not known how well one common model can identify the possible differences. Hence, it would be interesting to examine the model performance if separate models were designed for each product type.

REFERENCES

Bacchetti, A., & Saccani, N. (2012). Spare parts classification and demand forecasting for stock control: Investigating the gap between research and practice. *Omega (Oxford)*, *40*(6), 722–737. https://doi.org/10.1016/j.omega.2011.06.008.

Bajari, P., Nekipelov, D., Ryan, S. P., & Yang, M. (2015). Machine Learning Methods for Demand Estimation. *The American Economic Review*, *105*(5), 481–485. https://doi.org/10.1257/aer.p20151021.

Batina, L., Gierlichs, B., Prouff, E., Rivain, M., Standaert, F. X., & Veyrat-Charvillon, N. (2011). Mutual Information Analysis: a comprehensive study. *Journal of Cryptology*, *24*(2), 269–291. https://doi.org/10.1007/s00145-010-9084-8.

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, *106*, 249–259. https://doi.org/10.1016/j.neunet.2018.07.011.

Carbonneau, R., Laframboise, K., & Vahidov, R. (2008). Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research*, *184*(3), 1140–1154. https://doi.org/10.1016/j.ejor.2006.12.004.

Chandramouli, S. (2018). *Machine Learning* (S. Dutt & A. Das, Eds.; 1st edition). Pearson Education India.

Chen, T., & Guestrin, C. (2016). XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-*, 785–794. https://doi.org/10.1145/2939672.2939785.

Chen, X., Pang, Z., & Pan, L. (2014). Coordinating Inventory Control and Pricing Strategies for Perishable Products. *Operations Research*, *62*(2). https://doi.org/10.1287/opre.2014.1261.

Dahouda, M. K., & Joe, I. (2021). A Deep-Learned Embedding Technique for Categorical Features Encoding. *IEEE Access*, *9*, 114381–114391. https://doi.org/10.1109/ACCESS.2021.3104357.

Ertugrul, O. F. (2018). A novel type of activation function in artificial neural networks: Trained activation function. *Neural Networks*, *99*, 148–157. https://doi.org/10.1016/j.neunet.2018.01.007.

Evangelopoulos, N. E. (2013). Latent semantic analysis. *Wiley Interdisciplinary Reviews. Cognitive Science*, *4*(6), 683–692. https://doi.org/10.1002/wcs.1254.

Fildes, R., Ma, S., & Kolassa, S. (2019). Retail forecasting: Research and practice. *International Journal of Forecasting*. https://doi.org/10.1016/j.ijforecast.2019.06.004.

Fisher, M. L., Hammond, J. H., Obermeyer, W. R., & Raman, A. (1994). Making supply meet demand: in an uncertain world. *Harvard Business Review*, *72*(3), 83–94.

Galli, S. (2020). *Python feature engineering cookbook : over 70 recipes for creating, engineering, and transforming features to build machine learning models* (1st edition). Packt Publishing Ltd.

Hancock, J. T., & Khoshgoftaar, T. M. (2020). Survey on categorical data for neural networks. *Journal of Big Data*, *7*(1), 1–41. https://doi.org/10.1186/s40537-020-00305-w.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *Elements of Statistical Learning*. Springer.

Heale, R., & Twycross, A. (2015). Validity and reliability in quantitative studies. *Evidence-Based Nursing*, *18*(3), 66–67. https://doi.org/10.1136/eb-2015-102129.

Holweg, M., Disney, S., Holmström, J., & Småros, J. (2005). Supply chain collaboration: Making sense of the strategy continuum. *European Management Journal*, *23*(2), 170–181. https://doi.org/10.1016/j.emj.2005.02.008.

Huber, J., & Stuckenschmidt, H. (2020). Daily retail demand forecasting using machine learning with emphasis on calendric special days. *International Journal of Forecasting*, *36*(4). https://doi.org/10.1016/j.ijforecast.2020.02.005.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science (American Association for the Advancement of Science)*, *349*(6245), 255–260. https://doi.org/10.1126/science.aaa8415.

Kaspi, O., Girshevitz, O., & Senderowitz, H. (2021). PIXE based, Machine-Learning (PIXEL) supported workflow for glass fragments classification. *Talanta (Oxford)*, *234*, 122608–122608. https://doi.org/10.1016/j.talanta.2021.122608.

Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. *2014 Science and Information Conference*, 372–378. https://doi.org/10.1109/SAI.2014.6918213.

Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica (Ljubljana)*, *31*(3), 249–268.

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised leaning. *International Journal of Computer Science*, *1*.

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer New York. https://doi.org/10.1007/978-1-4614-6849-3.

Kureshi, S., & Thomas, S. (2019). Online grocery retailing – exploring local grocers beliefs. *International Journal of Retail & Distribution Management*, *47*(2). https://doi.org/10.1108/IJRDM-05-2018-0087.

Lewis, C. D. (Colin D. (1997). *Demand forecasting and inventory control : a computer aided learning approach*. Woodhead Pub. Ltd. in association with the Institute of Operations Management. https://doi.org/10.4324/9781856179898.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R., Tang, J., & Liu, H. (2018). Feature Selection. *ACM Computing Surveys*, *50*(6), 1–45. https://doi.org/10.1145/3136625.

Liu, Y., Zhou, Y., Wen, S., & Tang, C. (2014). A Strategy on Selecting Performance Metrics for Classifier Evaluation. *International Journal of Mobile Computing and Multimedia Communications*, *6*(4), 20–35. https://doi.org/10.4018/IJMCMC.2014100102.

Mahmoud, E. (1984). Accuracy in forecasting: A survey. *Journal of Forecasting*, *3*(2), 139–159. https://doi.org/10.1002/for.3980030203.

Manco, L., Maffei, N., Strolin, S., Vichi, S., Bottazzi, L., & Strigari, L. (2021). Basic of machine learning and deep learning in imaging for medical physicists. *Physica Medica*, *83*, 194–205. https://doi.org/10.1016/j.ejmp.2021.03.026..

Mentzer, J. T., & Cox JR, J. E. (1984). Familiarity, application, and performance of sales forecasting techniques. *Journal of Forecasting*, *3*(1), 27–36. https://doi.org/10.1002/for.3980030104.

Quemy, A. (2020). Two-stage optimization for machine learning workflow. *Information Systems (Oxford)*, *92*, 101483. https://doi.org/10.1016/j.is.2019.101483.

Rana, S. M. S. (2020). Supply chain drivers and retail supply chain responsiveness: strategy as moderator. *International Journal of Management Practice*, *13*(1). https://doi.org/10.1504/IJMP.2020.104066.

Renko, S., & Ficko, D. (2010). New logistics technologies in improving customer value in retailing service. *Journal of Retailing and Consumer Services*, *17*(3). https://doi.org/10.1016/j.jretconser.2010.03.012.

Saunders, M. N. K. (2019). *Research methods for business students* (P. Lewis & A. Thornhill, Eds.; Eighth edition.). Pearson Education.

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

Shankar, V., Kalyanam, K., Setia, P., Golmohammadi, A., Tirunillai, S., Douglass, T., Hennessey, J., Bull, J. S., & Waddoups, R. (2021). How Technology is Changing Retail. *Journal of Retailing*, *97*(1). https://doi.org/10.1016/j.jretai.2020.10.006.

Syntetos, A. A., Babai, Z., Boylan, J. E., Kolassa, S., & Nikolopoulos, K. (2016). Supply chain forecasting: Theory, practice, their gap and the future. *European Journal of Operational Research*, *252*(1), 1–26. https://doi.org/10.1016/j.ejor.2015.11.010.

Wan, S., Liang, Y., Zhang, Y., & Guizani, M. (2018). Deep Multi-Layer Perceptron Classifier for Behavior Analysis to Estimate Parkinson's Disease Severity Using Smartphones. *IEEE Access*, *6*, 36825–36833. https://doi.org/10.1109/ACCESS.2018.2851382.

Williamson, K. C., Spitzer, D. M., & Bloomberg, D. J. (1990). Modern Logistics Systems: Theory and Practice. *Journal of Business Logistics*, *11*(2), 65.

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing (Amsterdam)*, *415*, 295–316. https://doi.org/10.1016/j.neucom.2020.07.061.

Zentes, J., Morschett, D., & Schramm-Klein, H. (2012). *Strategic Retail Management*. Gabler Verlag. https://doi.org/10.1007/978-3-8349-6740-4.

Zheng, A. (2018). *Feature engineering for machine learning: principles and techniques for data scientists* (A. Casari, Ed.; First edition.). O'Reilly.

APPENDIX A: EVALUATION METRICS FOR THE MODELS

Data set	Model	Accuracy			Precision			Recall			F1-score		
		avg	min	max	avg	min	max	avg	min	max	avg	min	max
Test	Support vector classification (min-max scaled features)	0.670	0.670	0.670	0.669	0.669	0.669	0.670	0.670	0.670	0.669	0.669	0.669
	Nearest neighbors (min-max scaled features)	0.648	0.648	0.648	0.650	0.650	0.650	0.648	0.648	0.648	0.649	0.649	0.649
	XGBoost (features not scaled)	0.735	0.735	0.735	0.735	0.735	0.735	0.735	0.735	0.735	0.734	0.734	0.734
	Logistic regression (features not scaled)	0.686	0.686	0.686	0.683	0.683	0.683	0.686	0.686	0.686	0.684	0.684	0.684
	Multi-layer perceptron (min-max scaled features)	0.674	0.656	0.691	0.680	0.666	0.691	0.685	0.662	0.706	0.679	0.651	0.705
Train	Support vector classification (min-max scaled features)	0.955	0.955	0.955	0.955	0.955	0.955	0.955	0.955	0.955	0.955	0.955	0.955
	Nearest neighbors (min-max scaled features)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	XGBoost (features not scaled)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Logistic regression (features not scaled)	0.701	0.701	0.701	0.698	0.698	0.698	0.701	0.701	0.701	0.699	0.699	0.699
	Multi-layer perceptron (min-max scaled features)	0.685	0.662	0.706	0.690	0.681	0.704	0.685	0.662	0.706	0.679	0.651	0.705

APPENDIX B: SELECTED HYPERPARAMETERS AND CANDIDATE SETS

Model	Selected hyperparameters	Candidate sets (only the mentioned features were tuned)
Support Vector Classification	C=100 kernel='rbf' degree=3 gamma=2 coef0=0.0 shrinking=True probability=False tol=0.001 cache_size=200 class_weight=None verbose=False max_iter= 10000 decision_function_shape='ovr' break_ties=False random_state=None	kernel:['linear', 'poly', 'rbf', 'sigmoid'] C':[0.1, 1, 10, 100, 1000] gamma: [1, 0.1, 0.01, 0.001, 0.0001]
Nearest neighbors	n_neighbors=1 weights='uniform' algorithm='auto' leaf_size=1 p=2 metric='minkowski' metric_params=None n_jobs=None	leaf_size = list(range(1,42,3)) n_neighbors = list(range(1,21,3)) p=[1,2]
XGBoost	max_depth=8 learning_rate=0.3 n_estimators=140 silent=True objective='multi:softmax' booster='gbtree' n_jobs=10 nthread=None gamma=0 min_child_weight=1 max_delta_step=0 subsample=1 colsample_bytree=1 colsample_bylevel=1 reg_alpha=0 reg_lambda=1 scale_pos_weight=1 base_score=0.5 random_state=None missing=None	max_depth: range(2,10,2) n_estimators: range(60, 220, 40) learning_rate: [.3, .1, .01]
Logistic regression	penalty='none' dual=False tol=0.0001 C=1.0 fit_intercept=True intercept_scaling=1 class_weight=None random_state=None solver='lbfgs' max_iter=10000 multi_class='multinomial' verbose=0 warm_start=False n_jobs=None l1_ratio=None	solver:['newton-cg','sag','saga','lbfgs'] multi_class:['ovr','multinomial'] penalty:['none','l2']
Multi-layer perceptron	hidden_layer_sizes=(20) activation='tanh' solver='adam' alpha=0.05 batch_size='auto' learning_rate='adaptive' learning_rate_init=0.001 power_t=0.5 max_iter=10000 shuffle=True random_state=None tol=0.0001 verbose=False warm_start=False momentum=0.9 nesterovs_momentum=True early_stopping=False validation_fraction=0.1 beta_1=0.9 beta_2=0.999 epsilon=1e-08 n_iter_no_change=10 max_fun=15000	hidden_layer_sizes: [(10,30,10),(20,)] activation: ['tanh', 'relu'] solver: ['sgd', 'adam'] alpha: [0.0001, 0.05] learning_rate: ['constant','adaptive']