

VRIJE UNIVERSITEIT AMSTERDAM

MASTER THESIS

Understanding the Business of Mining Equipment

Author:
L.L. van Unen

Supervisor:
Dr. Mark Hoogendoorn

Second Supervisor:
Dr. Joost Berkhout

External Supervisor:
Ing. Xander Gerreman

Master Business Analytics
Faculty of Science

April 13, 2021

Understanding the Business of Mining Equipment

Author:
L.L. van Unen

Supervisor:
Dr. Mark Hoogendoorn

Second Supervisor:
Dr. Joost Berkhout

External Supervisor:
Ing. Xander Gerreman

*Submitted in partial fulfillment of the requirements
for the degree of Master Business Analytics*

Vrije Universiteit Amsterdam
Faculty of Science
Master Business Analytics
De Boelelaan 1085
1081 HV Amsterdam

Avanade
Analytics, Advanced Analytics
Orteliuslaan 1000
3528 BD Utrecht

Sandvik Group
Mining and Rock Technology
Barbara Strozziilaan 336
1083 HN Amsterdam

April 13, 2021

VRIJE UNIVERSITEIT AMSTERDAM
Faculty of Science
Master Business Analytics

Abstract

This thesis will help companies, and specifically Sandvik, with solving the issues on the aftermarket with the help of machine learning algorithms. Conducting business on the aftermarket namely poses new challenges for companies. The demand of products and work is more ad-hoc, while low waiting times are desired. Therefore, a trade-off should be found between the waiting time for customers and the associated costs for the company, as lower waiting times almost always lead to more costs. Both understanding your aftermarket business and composing efficient business strategies for these services are crucial when providing aftermarket services.

The main focus of this research is on the problem of forecasting the sales and associated invoice value of spare parts on the aftermarket. Furthermore, it focuses on the identification of important factors that influence the aftermarket sales. This thesis therefore helps Sandvik with gaining more business insight, which on its turn can improve the performance on the aftermarket. A good aftermarket performance leads to higher customer satisfaction and better customer relationships, which are essential for conducting business on the aftermarket.

Furthermore, this thesis provides an overview of multiple machine learning models that may be applied to the problem of revenue and sales forecasting. Their strengths and weaknesses are discussed as well as different application areas.

Also, multiple machine learning models have been deployed for this research. These models include machine learning models for regression and clustering problems, as well as appropriate models for time series analysis. Moreover, a combination of those models has been proposed in order to boost the overall model performance.

Keywords: XGBoost, Improved K-Prototypes, Random Forest, LSTM, Sales forecasting, Customer Segmentation, Feature Importance, Aftermarket

Acknowledgements

This thesis has been written in fulfilment of the requirements for the master Business Analytics at the Vrije Universiteit Amsterdam. During a six-month internship at Avanade this research has been conducted with the focus on the forecasting of the revenue and sales of Sandvik, one of Avanade's customers, as well as identifying the important factors that influence these forecasts the most. However, with the help of the people around me I would not be able to have done this research. Therefore I would like to take some time to thank some specific persons in particular in what follows.

First, I would like to thank Avanade for the opportunity and trust to conduct my research at one of their clients as a Data Scientist. This has made the experience of my internship even more valuable. However, this would also not be possible without the help of Sandvik and especially the time and support of Anna Loginovskaja. I would like to thank her very much for the opportunity. Moreover, I would like to thank the whole Fleet Application Team of the Mining and Construction department of Sandvik.

Secondly, a special thanks to Koen Zandvliet for the help with my project. The project presented would not be the same without his help. This also holds for the help of Sasha Sherstnev and Riccardo Pinosio.

Also, the help of my VU supervisor, Mark Hoogendoorn, is really appreciated. The feedback and guidance helped me to write this thesis.

But most of all, I would like to thank my supervisor Xander Gerreman for the support and guidance throughout my internship. Not only did you help me to get to know Avanade and its employees, you also made this online experience still feel very personal and connected. Thank you for this.

Furthermore, I would like to thank Robin van Ruitenbeek and Bart Maters for sharing the report template (Maters, 2020; Van Ruitenbeek, 2019).

Lastly, I would like to thank Joost Berkhout for the time and feedback he has given as second reader of my report.

Contents

Abstract	iii
Acknowledgements	v
1 Introduction	1
2 Background	3
2.1 About Avande	3
2.2 About Sandvik	3
2.3 The Aftermarket	4
2.4 Machine Learning Algorithms	5
2.4.1 Time Independent Models	7
2.4.2 Time Dependent Models	12
2.4.3 Overview Models	15
2.4.4 Clustering Methods	16
2.5 Evaluation Metrics	22
3 Problem Statement	24
3.1 Research Goal	24
3.2 Research outline	25
4 Literature Review	26
4.1 Revenue/Sales Forecasting	26
4.2 Time Series Analysis	27
4.3 Customer Segmentation	29
4.4 Feature Selection	30
4.5 Related Problems	33
4.5.1 Lifetime Estimation	33
Statistical Approaches	34
Artificial Intelligence (AI) Approaches	35
4.5.2 Asset Evaluation	35
Performance	36
5 Data	37
6 Methodology	38
7 Experimental Setup	39
8 Results	40
9 Conclusion	41
10 Discussion	42

Bibliography	43
A Customer Characteristics	48
B Heatmaps	49
C Activation Functions	50
D Results Revenue and Sales Model	52
E Time Series Model	53
F Customer Segmentation Models	54

Chapter 1

Introduction

In the mining and rock drill business, parts of all manufacturers are compatible with machines of all manufacturers. This means that a customer could buy their machine from one manufacturer and their replacement parts from another without any problem. Selling a machine to a customer does thus not necessarily mean that the customer will return for the aftermarket services and sales, like repairs of their machines or the sales of spare parts. Customers tend to go to the company that provides the best services for the best prices (Cohen et al., 2006). However, having a high customer satisfaction and strong customer relationship on the aftermarket, can help to assure that customers will return (Williams & Naumann, 2011). Having a high customer satisfaction and a strong relationship thus are key factors in the mining and rock drill aftermarket business; without those, a company would quickly go out of business.

Research shows that a strong relationship between a company and its customers can be achieved by a company providing good services at the aftermarket. However, providing good services at this market is not straightforward and can easily be inefficiently implemented (Cohen et al., 2006; Shokouhyar et al., 2020). Therefore, this thesis will try to help Sandvik, one of the companies active in the mining and rock drill business, with improving their aftermarket services. This will be done by forecasting their monthly revenue and sales and identifying the factors that influence these forecasts with the help of machine learning algorithms. The domain of machine learning is fast changing and relatively little recent research is available about implementing these techniques to solve the issues of the aftermarket. Therefore, this thesis also tries to investigate whether machine learning techniques may help to solve these aftermarket issues. The research question of this thesis will therefore be **"Which model performs the best for the prediction of the monthly potential revenue and sales per customer on the aftermarket and what are the factors that influence these potentials the most?"**. Not only may this increase the performance of Sandvik's of aftermarket services, but it may also give them an insight in their business.

Several algorithms will be reviewed in this thesis, all of which have been proposed in related literature as suitable for the problem on hand. Furthermore, algorithms that can be applied for time series analysis will be discussed as multiple researches have proposed that sales and revenue data can be seen and handled as time series. Doing so may improve the predictive power of the applied models (Venishetty, 2019). Customer segmentation models have also been applied, to identify similar customers in Sandvik's customer pool. Clustering the customers before sales prediction may boost the predictive power of the model according to literature (Chen & Lu, 2017). Furthermore, the identified clusters of customers also increase the business insight of Sandvik and these clusters can be used for other business purposes as well (Caruso et al., 2019; Hjort et al., 2013).

Not only will this thesis present potential business value for Sandvik, it will also provide an overview of various algorithms and their performance discussed in recent literature. Therefore, this thesis may also be seen as an overview and review of machine learning methods and their different applications. The methods discussed include forecasting, time series and clustering methods as well as feature selection methods. Furthermore, this thesis will contribute to the literature on solving the aftermarket issues with these machine learning techniques.

In the following chapter, a brief background on Avande, Sandvik and the aftermarket will be given. Afterwards, an overview of several algorithms and evaluation metrics will be discussed as background for the literature in Chapter 4 and implemented models in Chapter 6. Then, in Chapter 3 the problem discussed in this thesis will be presented and the structure for the remainder of this thesis will be given. Related literature will be discussed in Chapter 4 and afterward the Data and Methodology will be presented in Chapters 5 and 6 respectively. Results will be shown in 8 and this thesis will finish with a conclusion (Chapter 9) and discussion including recommendations for further research in Chapter 10.

Chapter 2

Background

This chapter provides a brief background about Avanade and Sandvik. Thereafter an introduction of the aftermarket will be given in Section 2.3. To conclude, this chapter will discuss some (machine learning) algorithms and evaluation metrics in Section 2.4 and 2.5. This introduction into the algorithms will be given in order to concisely discuss the literature in Chapter 4.

2.1 About Avanade

Avanade is an international services company, founded in 2000 based on a joint venture between Microsoft and Accenture. They help their customers with (digital) innovation and business solutions and try to prepare them for the fast-changing digitizing landscape of the world. They offer various services in multiple areas such as, security, data analytics, user experience and ERP systems. Avanade's goal is to use the Microsoft-ecosystem and the power of their people to build strong long-term relationships with their customers and deliver innovative solutions. Their customer base ranges from middle large to large companies and governments, working in the areas including but not limited to, banking, health care and retail. Today, Avanade is located in 24 countries and has around 30,000 employees (Avanade, [n.d.](#)).

2.2 About Sandvik

One of Avanade's clients is the Sandvik Group. The Sandvik group is a Swedish engineering company, founded in 1862. On this day, Sandvik has approximately 40,000 employees and sales in 160 countries.

Sandvik has three core businesses, namely: Sandvik Machining Solutions, Sandvik Materials Technology and Sandvik Mining and Rock Technology (Sandvik, [n.d.-a](#)).

1. Sandvik Machining Solutions

Also called Sandvik Coromant. This part of Sandvik is dedicated to manufacturing metal-cutting tools and tooling systems and provides services in these areas. These services include tool recycling, logistic solutions and software for tool data management (Sandvik, [n.d.-b](#)).

2. Sandvik Materials Technology

Offers a broad range of corrosion-resistant alloys. This branch of the Sandvik group also includes furnace products, which are products for thermal processing equipment and heating systems (Sandvik, [n.d.-d](#)).

3. Sandvik Mining and Rock Technology

Sandvik Mining and Rock Technology focuses on manufacturing and selling heavy machinery and mining equipment. Further, they offer a wide range of services for maintenance and monitoring tools for all machines. This thesis will focus on this branch of the Sandvik group (Sandvik, [n.d.-c](#)).

2.3 The Aftermarket

The aftermarket is a special market for offering spare parts for already manufactured machines (University, 2020), but also to provide services like repairs, maintenance and installing upgrades. (Cohen et al., 2006) The importance of the aftermarket is growing since the competition in most business areas has increased. Which, in its turn, resulted in a decrease in demand and profit on the regular market. Providing good after-sales services and goods on the aftermarket can generate a steady stream of revenue since it leads to a higher customer satisfaction. Those satisfied customers are more loyal to the company and are more likely to repurchase at this company (Aksoy et al., 2008; Fornell et al., 2006; Gruca & Rego, 2005; Williams & Naumann, 2011). Further, offering after-sales services gives the company a special insight in the customer's business, leading to a better understanding of the customer's needs and therefore a competitive advantage to other aftermarket suppliers (Cohen et al., 2006).

Although all of the above sounds promising, implementing after-sales services in a efficient and cost-effective way is difficult. Suboptimally implementing those services leads to more costs with minor revenue increases, which resulted in a large group of companies that do not see the relevance of providing after-sales services.

Most after-sales services are sporadically and unexpectedly requested since machine failures and breakdowns are unpredictable. Therefore, it is not known in advance when a mechanic should be at a certain customer and which parts should be in stock at the different warehouses. The scheduling of the workforce and stock management in this market thus require a different strategy than that of the original parts and work scheduling of the company. This may result in different strategies, for example: a company produces and stores the aftermarket parts in advance, instead of on demand and last minute which is common practice for parts for the regular market. This strategy will add extra holding costs, but results in lower waiting time for the customer when a machine of a customer breaks down.

Furthermore, the scheduling of workforce should allow for ad-hoc repair jobs. The extra time for ad-hoc job may result in more over time if a lot of job requests are made, or in more idle time in case of a few repair requests are made.

Lastly, extra costs are associated with providing services for many parts on the aftermarket. Companies that still provide services for products produced in the past require extra knowledge and more materials that should be kept by the company, leading to higher costs.

There are different theories on how to provide good after-sales services (Durgbo, 2020). In Cohen et al. (2006) an overall strategy and widely used theory is presented. Cohen et al. (2006) argue that operating on the aftermarket in an efficient way requires the following six-step approach:

1. **"Identify which products to cover"**: a company can offer after-sales services for different groups of products. For example, a firm can provide services only for products that they are currently manufacturing, or only a subset of those products. However, they may also choose to additionally provide after-sales

services for products they have manufactured in the past, or products manufactured by their competitors. The best set of products may be identified by looking at those products that distinguish you from the competition, but also those which generate the most profit. Considerations should also include the company's values and which cover policy fits those the best, e.g. a company that produces products that are eco-friendly may choose to not provide services for products from competitors that are not eco-friendly.

2. **"Create a portfolio of service products"**: companies need to take into account their own interests, as well as those of their customers. Customers are only willing to pay a certain amount for a certain service performance, while the company has to invest to meet this level of performance. The investment should be lower than the average amount that customers are willing to pay to make providing these services profitable. The willingness to pay and needs may differ for all clients, however a small set of service contracts should be made.
3. **"Select business models to support service products"**: the service products offered by the company may require different business models. A performance-based model (pay for level of service) and an ad-hoc model (pay per use) are examples of such business models. There also exist business models that differ in product ownership, e.g. models in which a customer leases the product (company owns the product) or a customer buys the product (customer owns the product). The right model can be chosen based on for example the lifetime of the product or in order to avoid conflicts of interest. A conflict of interest can occur when a company is both the service provider and manufacturer and the business model is pay per use.
4. **"Modify after-sales organizational structures"**: it is important to determine which department is in charge of which step in the after-sales service (holding stock, selling products, etc.), but also to determine how the different departments can work together optimally. It may be the case that outsourcing some work is the best option in order to reduce costs.
5. **"Design and manage an after-sales service supply chain"**: different services and parts are delivered in the after-market, preferably at the lowest cost and with highest service performance. Distributing the resources over the available places (e.g. warehouses, customer's sites and factories) may therefore be a challenging tasks and should get the proper attention. Further, it should be decided whether products are stored as a whole or as separate parts.
6. **"Monitor performance continuously"**: evaluate the performance of the company's after-sales services by monitoring the delivered services, the corresponding costs as well as technologies and rivals.

2.4 Machine Learning Algorithms

In this thesis the performance of multiple (machine learning) algorithms and their applications will be discussed in Chapter 4. In order to concisely discuss the literature, an overview of some algorithms will first be presented in this section. Note that not all the discussed methods will be implemented for the problem on hand in this thesis. The models that will be used and their implementation will be presented

in more depth in Chapter 6.

Machine Learning algorithms can be applied to several different tasks in a variety of application areas. One can roughly divide the different tasks into the following categories (Killada, 2017):

- **Classification Problems.** Given an observation, the task is to predict to which class this observation belongs, i.e. the output is a categorical label. Examples are: predicting whether a customer is likely to repay their loan, spam detection and categorize images.
- **Regression Problems.** The task of regression is to map a given input into a continuous output variable, i.e the output is of a quantitative nature. Examples are: sales prediction, prediction of size and height and asset evaluation.
- **Clustering Problems.** Given a set of observations, similar groups of observations should be found, i.e. a categorical output is expected of the cluster label. Examples are: targeted marketing (similar customers will see similar advertisements), document analysis (cluster documents based on similar subjects) and identifying fraudulent transactions (clustering transactions based on similarities)

Note that clustering and classification problems seem to have similar objectives. However, clustering problems are so called unsupervised learning problem in contrast to classification problems, which are supervised learning problems. This means that the difference lies in the fact that in classification problems the true label or category is known and contained in the dataset. In case of clustering, these true labels or categories are not known.

In this section regression models as well as clustering models will be discussed, as these models fit the objectives of this thesis (more detailed explanation of the objectives may be found in the next chapter Problem Statement). As discussed before, regression and clustering problems may be categorized as a different categories of machine learning problems. Hence, these problems require different methods. Moreover, as time series data and appropriate models to analyse this kind of data will be discussed, the regression models will also be split in two section. Time series data is defined as: data that is an ordered sequence of observations with equally spaced time intervals. Some models inherently handle these time dependencies within the data, i.e. they are designed to handle sequential data (Brockwell et al., 2016; de Gunst, 2013; Hochreiter & Schmidhuber, 1997). These models will be discussed in Section 2.4.2. But first, in Section 2.4.1, some models will be presented that are suitable for regression problems that do not assume sequential data, i.e. the time independent models. These models however may also be made appropriate to analyse time dependent data via feature engineering. By adding features describing changes over time or adding so called lag variables, the time independent models may learn time dependencies within the historical data. The discussed regression models will be summarized in Section 2.4.3, providing an overview of the main advantages and disadvantages of the regression models.

To finish the overview of algorithms, several clustering methods will be presented in Section 2.4.4.

In the following sections, $\mathbf{x}_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,m}\}$ represents an input variable of the models with m features and $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, the $n \times m$ matrix of all n observations. Further, y_i denotes the i^{th} target variable (with $\mathbf{y} = \{y_1, \dots, y_n\}$) and \hat{y}_i the prediction of the model for the i^{th} variable.

2.4.1 Time Independent Models

As discussed, in machine learning, some models are designed especially to handle sequential data, i.e. data that is an ordered sequence of observations with equally spaced time intervals. The models discussed in this section do not inherently handle time dependencies, hence they are time independent models.

First, a time independent model discussed in relevant literature for asset evaluation will be presented, namely the Principal Components Regression (PCR) method. This methods summarize the data into a few linear combinations, reducing noise and dimension.

Then, Neural Network (NN) will be discussed, as NN are widely used for many different regression problems. To end this section, two types of tree based models will be discussed, namely: Boosting Trees and Random Forests.

Principal Components Regression (PCR)

Principal Components Regression discards specific features based on the Singular Value Decomposition (SVD) of the input data \mathbf{X} . The SVD is the dot product of three matrices, namely \mathbf{UDV}^T . Here \mathbf{D} is a rectangular diagonal matrix with on the diagonal the so called singular values (the square root of the eigenvalues) of \mathbf{X} , \mathbf{U} the orthogonal unit vectors, i.e. the left singular values of \mathbf{X} , and \mathbf{V} the eigenvectors of the covariance matrix $\mathbf{X}^T\mathbf{X}$ of the input data.

Now, the matrices are arranged in such a way that the diagonal values of \mathbf{D} are ordered in descending order. Therefore, selecting the first A columns of \mathbf{V} results in the highest eigenvectors of $\mathbf{X}^T\mathbf{X}$ and these columns are argued to explain the data the most. Thus, these values will be used by the PCR algorithm (Artigue & Smith, 2019; Bagheri, 2020).

The main drawback of PCR compared to PLS is the fact that PCR does not take into account the goal of the model (i.e. predicting the response variable), and the dimension reduction is done independently of the problem. However, one could incorporate their expected ability to predict the response variable into the selection of variables (Artigue & Smith, 2019; Reiss & Ogden, 2007).

The dimension reduction power of this model have been applied to the problem of reducing the dimensionality of data. This power is for example often applied to the unsupervised learning problem clustering via the Principal Component Analysis (PCA) method. However, one should keep in mind that the original features are transformed and cannot be straightforwardly use in identifying the most important features (Hancer et al., 2020).

Neural Network (NN)

The previous algorithms could only model linear dependencies in the data. However, assuming only linear dependencies limits the performance of the model as most real-world problems have some degree of non-linearity. An example of a model that can handle non-linearity within the data is a Neural Network (NN). Neural Networks have been the subject of many researches the last few decades and are currently implemented for a broad range of tasks (Schmidhuber, 2015). The application to different problems has lead to a variety of different Neural Networks.

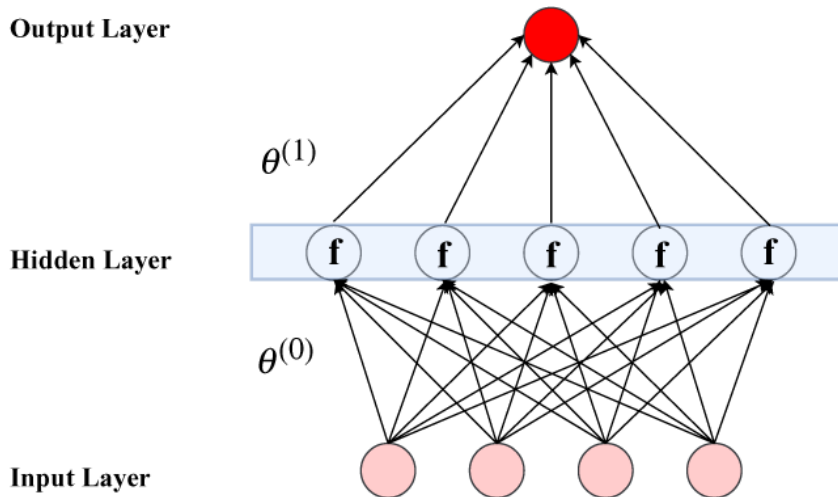


FIGURE 2.1: A shallow Neural Network with one hidden layer. The light red circles represent the input layer, i.e. the input of the model. The red circles represent the output layer, i.e. the response variable. All the arrows have a corresponding weight. In the NN with one hidden layer, the function f transforms the data before passing it onto the network. (Gu et al., 2019)

Examples are: Recurrent Neural Network (RNN), Convolutional Neural Networks (CNN) and feedforward (Deep) Neural Networks, which differ in the way the models learn and therefore differ in the type of problems they are best suited to. Note that Deep Neural Networks and Neural Networks can be categorized as one type as they are in principle the same but differ in the number of layers (Schmidhuber, 2015). These feedforward (Deep) Neural Networks are mostly considered for regression problems and will be explained in what follows. In the discussion of the literature, these models will be referred to as Neural Network (NN). In Section 2.4.2 Recurrent Neural Network (RNN) will be presented since those inherently use time dependencies and will be discussed in both Time Series Analysis (4.2) and Lifetime Estimation (4.5.1).

(Deep) Neural Networks

In figure 2.1 a representation of a shallow feedforward NN is shown. A feedforward NN consist of an input layer, one or more hidden layers and an output layer. The input layer takes the data as input and has one node for each dimension (i.e. number of features) of the data. The input nodes are linearly transformed by some weights and forwarded into the next layer. In figure 2.1 these weights are represented by θ , a $m + 1$ parameter vector where m is the number of input features and the additional parameter is the bias. $\theta^{(0)}$ represents the weights into the first hidden layer and $\theta^{(1)}$ the weights into the output layer. In the example, the NN is so called fully connected. This means that all input variables are connected to all nodes in the hidden layers.

For the NN with one hidden layer, the linearly transformed input data is fed into an activation function f . Common activation functions include the sigmoid function, the rectified linear unit (ReLU) function and the hyperbolic tangent (tanH) function. Selecting the right activation function should be a part of the model optimization and an overview of the different activation functions and their (dis)advantages is given in Appendix C. Note, that multiple different activation functions may be used in a NN with multiple hidden layers.

As said before, there are multiple different types of Neural Networks and they

differ in the way they learn. A feedforward NN learns via back-propagation. The weights are initially assigned randomly and then iteratively adjusted via back-propagation to minimize the cost function and improve the model performance. Back-propagation is the process of calculating the gradient (i.e. multivariable derivative) of the cost function (which has as input all the weights and biases) and adjusting the weights accordingly to minimize the cost function at every iteration. By repeating this process multiple times, values of the weights yielding the minimum cost function can be found. But, as with many optimization problems, this minimum is not necessarily the global minimum, it can also be a local minimum. Moreover, the result of the Neural Network may be influenced by the initialization of the weights, converging to different local minimums.

(Dis)advantages

NNs have the advantage that they are flexible in number of layers, nodes per layer and the activation function, which can all be determined and changed by the programmer. However, this flexibility also makes them highly parameterized and therefore time consuming to optimize. Even more, it causes the model to be prone to overfit. Regularization methods and parameter tuning should thus be carefully done. Further, NN are considered as least transparent machine learning methods. Leading them to be called a 'black-box', as very few people truly understand what goes on inside the method and how the results are achieved. This lack of understanding may result in poor performance and difficulties with fixing bugs. Along with the problem of the vanishing and exploding gradients, they can be tricky to use (Gu et al., 2019; Krauss et al., 2017; Schmidhuber, 2015).

But, Neural Networks have outperformed other methods in many data science competitions in the last decades, especially in the field of speech and language recognition. Examples of competitions are Kaggle competitions ¹ or conference competitions like the ones of TRECVID ² in which many data scientists compete (Schmidhuber, 2015). Therefore, it may be concluded that when NNs are tuned well, they have high explanatory power.

Tree-Based Models

Tree-based models use a whole different approach than the methods discussed above. These algorithms recursively split the data into groups that have similar characteristics. Each of those groups is represented by a node and if possible, every node is again split up. The tree is built until a certain threshold (size of the group, number of nodes, etc.) is reached. These trees can be applied to either classification and regression problems. In classification problems the dominating class in the end node will be the forecast output, whereas in regression problems the average outcome of the observations in the node will be the forecast outcome. An example of a shallow classification tree is given in Figure 2.2.

These trees are also called Classification And Regression Trees (CARTs). They associate a real score with each of the leaves. This score gives a richer interpretation of not only the classification output, but an overall fitness of the model.

The drawback of regular regression trees is that they are very likely to overfit the data when they are not regularized, leading to for example, multiple leaf nodes with only one observation (Gu et al., 2019). Therefore, methods have been developed to

¹<https://www.kaggle.com/competitions>

²<https://trecvid.nist.gov/>

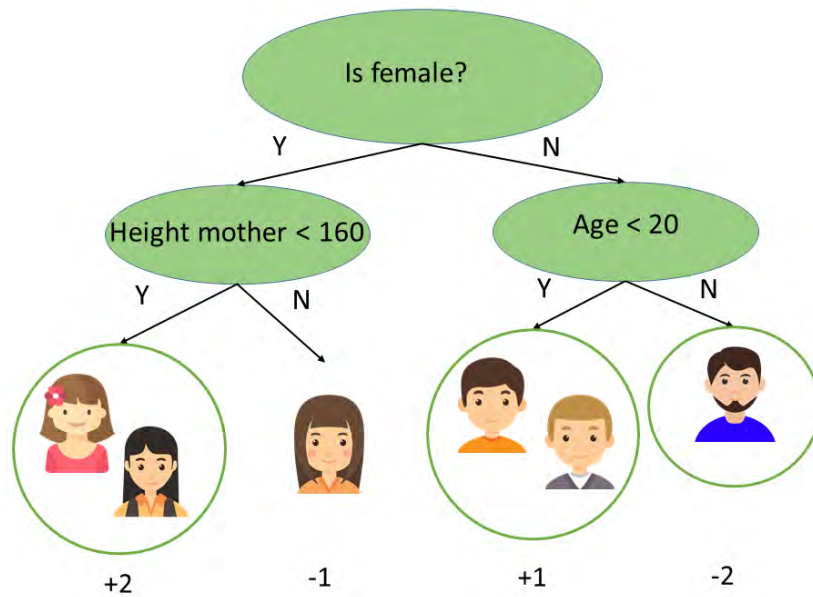


FIGURE 2.2: Example of a shallow classification tree. The numbers below the leaves are the scores associated with each leaf.

improve their generalization. Examples of such methods are boosting trees and random forests. Both methods combine multiple so called shallow trees, i.e. trees with only a few nodes. These trees on their own have weak prediction power, but through combining the results of multiple trees, the predictive power increases significantly. Boosting and random forest methods differ in the way they combine these shallow trees, but they can be applied to similar problems. The real score given to each leaf by the CART allows for optimization of the resulting overall model. Both models can be written in the form:

$$\hat{y} = \sum_{k=1}^K f_k(\mathbf{x}_i) \quad f_k \in \mathcal{F} \quad (2.1)$$

With K the number of trees and f a function in the given functional space \mathcal{F} . Now the objective function becomes:

$$\mathcal{L}(\theta) = \sum_i^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (2.2)$$

Where Ω is the regularization term and l the loss function of the model (Chen & Guestrin, 2016; xgboost developers, 2020).

Boosting Trees

The framework for boosting algorithms have been presented by Friedman (2001). If a model based on this framework is applied to regression problems, the models are often referred to as gradient boosting algorithms. Boosting trees boost the performance of shallow trees by building a new tree on the residuals of the previous trees, i.e. it learns from the errors made so far by the model (Natekin & Knoll, 2013). This can be more formally written by:

Let $\hat{y}_i^{(t)}$ be the predicted value of the i^{th} observation of tree t . Then,

$$\hat{y}_i^{(t)} = \sum_{k=1}^t f_k(\mathbf{x}_i) = \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i) \quad (2.3)$$

Further it holds that:

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t) \quad (2.4)$$

As one can see, the objective function of tree t now depends on the prediction made in tree $t - 1$, which allows the model to learn from the residuals of the previous trees (Chen & Guestrin, 2016).

Probably the most famous version of gradient boosting algorithms is Extreme Gradient Boosting (XGBoost), which was developed during a research project at the University of Washington (Yuan, 2019). The model was build upon the gradient boosting framework presented by Friedman (2001) and XGBoost was first described in Chen and He (2015). The XGBoost differentiates from other boosting trees algorithms in two ways, namely: it uses a novel sparsity aware algorithm that handles sparse data more efficiently; further, it learns via a weighted quantile sketch algorithm.

This leads to one of the main advantages of this model, namely its scalability (Chen & Guestrin, 2016). However, as boosting trees are build upon the errors made in earlier trees, the model is more likely to overfit than for example bagging algorithms.

Random Forests

In contrast to the boosting trees, random forests as introduced by Breiman (2001) combine the predictive power of numerous of independent trees (making them member of the group of bagging algorithms, i.e. bootstrap aggregating algorithms). For each tree, only a subset of the training sample is considered and at every node only a subset of the input variables is considered for the split. Further, each of these individual trees are shallow trees. Introducing the randomness through the subsets of samples and features and the use of only shallow trees, the algorithm is less likely to overfit than the original CARTs. However, by choosing N , i.e. the number of trees, large enough, they will converge to a model with high predictive power. The output of these trees are then combined by taking the average (or in case of classification, the class that is predicted the most).

Drawbacks of random forests are, that they are, like Neural Networks, so called black-boxes. Further, they are computationally and memory intensive and, in case of regression, they do not predict beyond the ranges of the training data as the prediction is the average of all the outcomes in the final node (and each individual tree does not predict beyond those ranges).

However, they do naturally reduce the dimensionality of the data as only the best splits are made. They can handle missing data without *a priori* data handling and they are not affected by transformations of the predictors. Along with those advantages, random forests can handle variable interaction and are able to handle non linearity's (Breiman, 2001; Gu et al., 2019).

2.4.2 Time Dependent Models

The models discussed in this section do assume time dependency within the considered data. The architecture of these models is made in such a manner that they automatically take previous time steps into account, i.e. sequenced data. They are very suitable for time series analysis, but Recurrent Neural Networks (RNN) may be applied to many other time dependent problems, like language processing problems. The Autoregressive Integrated Moving Average (ARIMA) model (and its variations) and the RNN model are the most discussed time dependent models for sales forecasting. Therefore, these models will be reviewed in this section, for the literature on time series analysis see Chapter 4.

Autoregressive Integrated Moving Average (ARIMA)

Autoregressive Integrated Moving Average (ARIMA) is a generalization of the Autoregressive Moving Average (ARMA) model. As the name suggests these models contain an Autoregressive (RA) and a Moving Average (MA) part. First, these two processes followed by the ARMA model will be explained. Afterwards, the ARIMA model will be explained.

A Moving Average process is a process defined as follows:

Suppose that $\{Z_t\}$ is a white noise process with σ^2 as the variance of Z_t . Let q be a positive number and let $\beta_0, \beta_1, \dots, \beta_q$ be constants. Then, the process $\{X_t\}$ defined by

$$X_t = \beta_0 Z_t + \beta_1 Z_{t-1} + \dots + \beta_q Z_{t-q} \quad (2.5)$$

is a Moving Average process of order q (MA(q)). A MA process thus predicts the observation X_t based on weighted past errors Z_t .

Second, an Autoregressive Process is defined in the following way:

Let $\{Z_t\}$ be a white noise process with σ^2 as the variance of Z_t . Let p be a positive number and let $\alpha_0, \alpha_1, \dots, \alpha_p$ be constants. Then, if $\{X_t\}$ is stationary it satisfies:

$$X_t = \alpha_1 X_{t-1} + \dots + \alpha_p X_{t-p} + Z_t \quad (2.6)$$

and $\{X_t\}$ is an Autoregressive process of order p (AR(p)). This process thus fully depends on the weighted previous observations.

An ARMA models combines the equations 2.5 and 2.6 into

$$X_t = \alpha_1 X_{t-1} + \dots + \alpha_p X_{t-p} + Z_t + \beta_1 Z_{t-1} + \dots + \beta_q Z_{t-q} \quad (2.7)$$

(de Gunst, 2013, p. 72-74)

Due to the definition of MA or AR processes, ARMA models can only handle stationary time series. A stationary time series is defined as follows: "A time series is (weakly) stationary if, for any value k , EX_t and $EX_t EX_{t+k}$ exist and do not depend on t " (de Gunst, 2013, p. 58). Or less formally, a time series is stationary if trend and seasonality are removed from the data and if the variance of the remaining residuals is constant over time (Hoogendoorn & Funk, 2018). Stationary processes are more easy to analyze, therefore it may be preferable to first extract a stationary time series from a non-stationary one. To do this, one should define and extract the trend and seasonality within the data.

However, removing the trend and season may not be straightforward and can be tricky. Therefore, another model has been proposed in order to evaluate non-stationary data (as most real-world datasets are non-stationary), the ARIMA model. The variable X_t in the ARMA model will be replaced by the difference operator $\Delta^d X_t$,

where

$$\Delta X_t = X_t - X_{t-1} \quad (2.8)$$

and $\Delta^d X_t$ the d^{th} order of ΔX_t . If $\{X_t\}$ has a linear trend $m_t = a + bt$, then $\{Y_t\} = \{\Delta X_t\}$ will have no trend. Or in general, if $\{X_t\}$ has a polynomial trend with degree d (i.e. $m_t = a_0 + a_1 t + \dots + a_d t^d$), then $\{Y_t\} = \{\Delta^d X_t\}$ has no trend. This d value will also be a model parameter, resulting in three parameters to optimize: q from the MA process, p from the AR process and d for the integrated part.

Finding the right model parameters for AR(I)MA models may be hard as explicit efficient estimators cannot be found. Therefore, these parameters should be found by numerical experiments and finding the right initial parameters is found difficult (de Gunst, 2013, p.78). Moreover, AR(I)MA models can only handle linear dependencies, which makes them less applicable to many problems.

However, they have been proven to have high explanatory power and they have robust performance (Hewamalage et al., 2021). Therefore, they are still used in many researches, which will be discussed in more depth in Chapter 4.

Recurrent Neural Network (RNN)

Another type of Neural Networks as discussed in Section 2.4.1 are Recurrent Neural Network (RNN). These Neural Networks are designed in such a way that they may contain information about previous input values. This ability makes them very suitable for time series analysis problems, but also for language processing problems like speech recognition and semantic analysis or other problems with (possible) dependencies between input variables (Hewamalage et al., 2021). A simple RNN with one hidden layer, can be defined as follows:

Given input variables x_1, x_2, \dots, x_t for every time $t = 1, 2, \dots, T - 1$, the prediction of the next time step z_{t+1} will be

$$\begin{aligned} h_s &= f_\alpha(\mathbf{h}_{s-1}, \mathbf{x}_s) \quad \forall s = 1, \dots, t \\ z_{t+1} &= g_\beta(\mathbf{h}_t) \end{aligned} \quad (2.9)$$

where f_α and g_β some nonlinear functions which are parameterized by α and β . They can generally given by:

$$\begin{aligned} f_\alpha(\mathbf{h}_{s-1}, \mathbf{x}_s) &= \mathcal{H}_f(\mathbf{W}\mathbf{x}_s + \mathbf{U}\mathbf{h}_{s-1} + \mathbf{b}) \quad \forall s = 1, \dots, t \\ g_\beta(\mathbf{h}_t) &= \mathcal{H}_g(\mathbf{V}\mathbf{h}_t + \mathbf{c}) \end{aligned} \quad (2.10)$$

where $\alpha = (\mathbf{W}, \mathbf{U}, \mathbf{b})$ and $\beta = (\mathbf{V}, \mathbf{c})$ for some weight matrices $\mathbf{W}, \mathbf{U}, \mathbf{V}$ and bias vectors \mathbf{b} and \mathbf{c} , and \mathcal{H} a nonlinear activation function (Choe et al., 2017).

The most popular units used in RNN are Elman cells, Long Short Term Memory (LSTM) cells, and the Gated Recurrent units. These units are shown in Figure 2.3. In all representations \mathbf{h}_{t-1} represents the hidden cell from the previous timestep, \mathbf{x}_t the input vector of cell t and z_t the output of the cell. The Elman cell is, as one can see in the figure, the most simple cell. In this cell the only memory passed along the network is that of the hidden cell of the timestep, i.e. \mathbf{h}_{t-1} . Its set of equations is given by:

$$\mathbf{h}_t = \sigma(\mathbf{W}_i \cdot \mathbf{h}_{t-1} + \mathbf{V}_i \cdot \mathbf{x}_t + \mathbf{b}_i) \quad (2.11a)$$

$$\mathbf{z}_t = \tanh(\mathbf{W}_o \cdot \mathbf{h}_t + \mathbf{b}_o) \quad (2.11b)$$

In the Gated Recurrent Unit (GRU), the information from the previous hidden

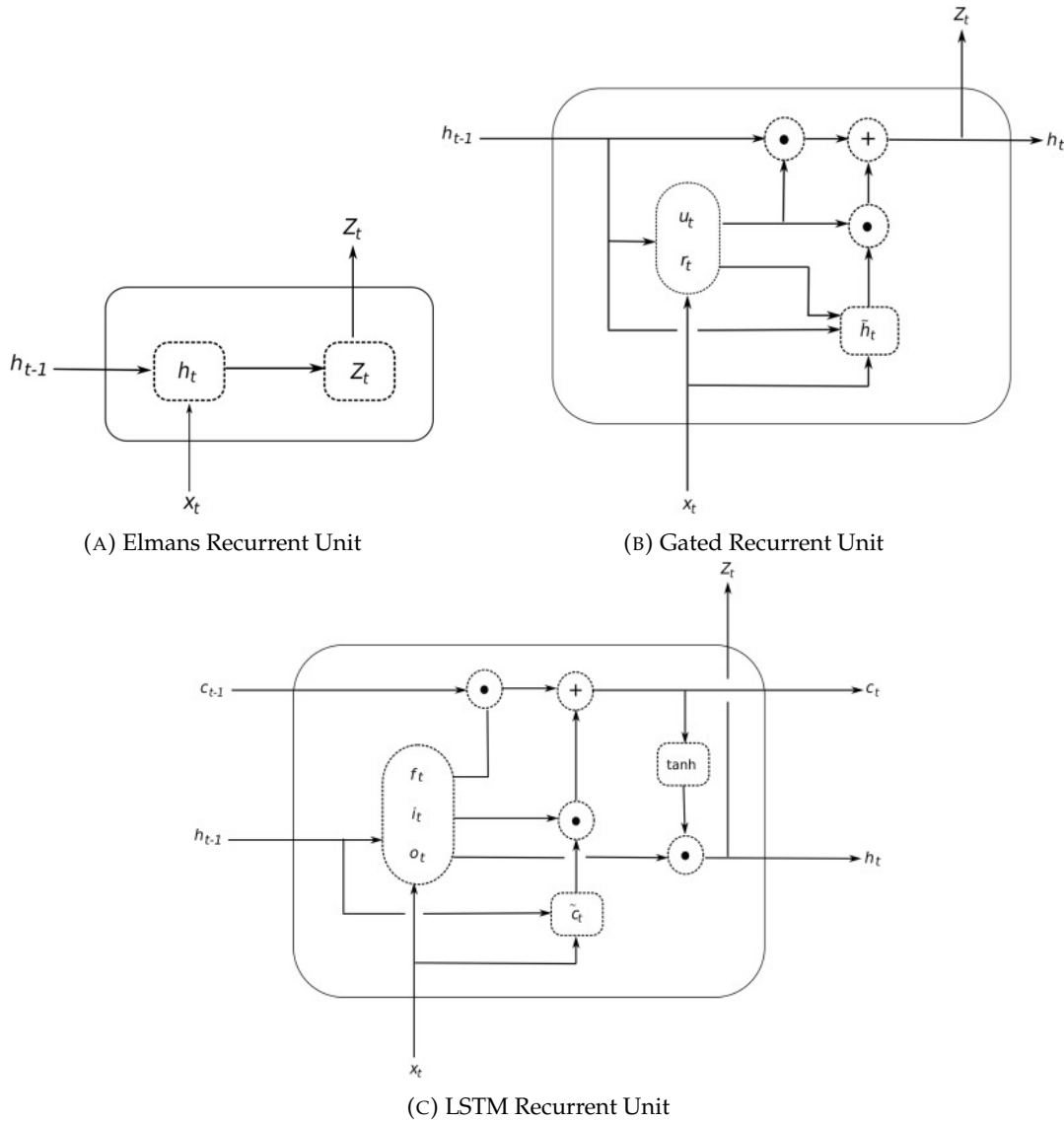


FIGURE 2.3: Representations of different recurrent units of RNNs (Hewamalage et al., 2021)

cell and the input variable are combined via so called update and reset gates (i.e. \mathbf{u}_t and \mathbf{r}_t respectively). Both gates take the input and previous hidden cell information and determine the proportion in which they should contribute to the output. The reset gate, resets a proportion of the previous hidden state and combines it with the input into the candidate hidden state $\tilde{\mathbf{h}}_t$. This candidate hidden state and previous hidden state are element-wise multiplied (denoted with the \odot) and added together to form the output \mathbf{z}_t and the new hidden state \mathbf{h}_t . Formally this can be written as:

$$\mathbf{u}_t = \sigma(\mathbf{W}_u \cdot \mathbf{h}_{t-1} + \mathbf{V}_u \cdot \mathbf{x}_t + \mathbf{b}_u) \quad (2.12a)$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \cdot \mathbf{h}_{t-1} + \mathbf{V}_r \cdot \mathbf{x}_t + \mathbf{b}_r) \quad (2.12b)$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h \cdot \mathbf{r}_t + \mathbf{V}_h \cdot \mathbf{x}_t + \mathbf{b}_h) \quad (2.12c)$$

$$\mathbf{h}_t = \mathbf{u}_t \odot \tilde{\mathbf{h}}_t + (1 - \mathbf{u}_t) \odot \mathbf{h}_{t-1} \quad (2.12d)$$

$$\mathbf{z}_t = \mathbf{h}_t \quad (2.12e)$$

The LSTM unit does not only have a hidden state, but has an internal state as well. This internal state allows the unit to capture long-term dependencies, while the hidden state acts as short term memory, hence the name. The forget gate \mathbf{f}_t determines the proportion of the long-term memory to be passed along, while the input gate \mathbf{i}_t determines the proportion of the input and previous hidden cell information. The output gate \mathbf{o}_t , in combination with the hyperbolic tangent function, transforms the new long-term memory into the output of the cell. Note that the output of the LSTM \mathbf{z}_t is equal to the hidden state of that cell \mathbf{h}_t like in the GRU. The set of equations for the LSTM cell are given by:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot \mathbf{h}_{t-1} + \mathbf{V}_i \cdot \mathbf{x}_t + \mathbf{b}_i) \quad (2.13a)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot \mathbf{h}_{t-1} + \mathbf{V}_o \cdot \mathbf{x}_t + \mathbf{b}_o) \quad (2.13b)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot \mathbf{h}_{t-1} + \mathbf{V}_f \cdot \mathbf{x}_t + \mathbf{b}_f) \quad (2.13c)$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_c \cdot \mathbf{h}_{t-1} + \mathbf{V}_c \cdot \mathbf{x}_t + \mathbf{b}_c) \quad (2.13d)$$

$$\mathbf{C}_t = \mathbf{i}_t \odot \tilde{\mathbf{C}}_t + \mathbf{f}_t \odot \mathbf{C}_{t-1} \quad (2.13e)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \quad (2.13f)$$

$$\mathbf{z}_t = \mathbf{h}_t \quad (2.13g)$$

The LSTM and GRU models have the benefit that they tackle the vanishing and exploding gradient problem via their gates. The GRU is gaining popularity over the LSTM unit as it is simpler (less parameters) and more efficient.

The main drawback for a Recurrent Neural Network (RNN) is that they are, like Neural Network (NN), highly parameterized which makes optimizing these models hard and it is argued that only experts can reach the best performance (Hewamalage et al., 2021).

However, they do have the advantage over the Autoregressive Integrated Moving Average (ARIMA) models that the lag parameters (p, q) can be learned automatically from the data and they can capture non-linear dependencies.

2.4.3 Overview Models

Model	Advantages	Disadvantages
Principal Components Regression (PCR)	<ol style="list-style-type: none"> 1. Dimension reduction 2. keeps the original components 	<ol style="list-style-type: none"> 1. Only captures linear relations 2. does not take into account the goal of the model
Neural Network (NN)	<ol style="list-style-type: none"> 1. Very powerful in capturing non-linear relationships within the data 	<ol style="list-style-type: none"> 1. Highly parameterized, which makes them hard to understand and time consuming to optimize
Boosting Trees	<ol style="list-style-type: none"> 1. Dimension reduction 2. Scalability 	<ol style="list-style-type: none"> 1. More likely to overfit compared to other models
Random Forest (RF)	<ol style="list-style-type: none"> 1. Dimension reduction 2. Handling missing data without <i>a priori</i> data handling 3. Can handle non linear and non transformed data 	<ol style="list-style-type: none"> 1. Computationally and memory intensive 2. Unable to predict beyond the ranges of the training data 3. May perform poorly in case of correlated features
Autoregressive Integrated Moving Average (ARIMA)	<ol style="list-style-type: none"> 1. Able to capture (long-term) time dependencies 1. Robust performance 	<ol style="list-style-type: none"> 1. Can only capture linear dependencies 2. Finding the model parameters may be difficult
Recurrent Neural Network (RNN)	<ol style="list-style-type: none"> 1. Able to capture (long-term) time dependencies 	<ol style="list-style-type: none"> 1. Even more parameterized than NN

TABLE 2.1: Overview of the disadvantages and advantages of the different linear and non-linear models.

2.4.4 Clustering Methods

Cluster analysis is used to identify natural groups or clusters in the data based on (dis)similarities between the data points. Cluster analysis can be used as a semi-supervised learning technique of classification, or as unsupervised learning technique to identify classes within the data (Frades & Matthiesen, 2010; Omran et al., 2007).

In Omran et al. (2007) some guidelines for setting up a clustering model are presented. These guidelines will be followed to clearly describe the different aspects of a clustering model. Afterwards, some clustering algorithms will be presented.

The formal definition of a clustering problem is:

Given a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p, \dots, \mathbf{x}_n\}$ where \mathbf{x}_p is a pattern in the m -dimensional feature space, and n is the number of patterns in \mathbf{X} , then the clustering of \mathbf{X} is the partitioning of \mathbf{X} into K clusters $\{C_1, C_2, \dots, C_K\}$ satisfying the following conditions:

- Each pattern should be assigned to a cluster, i.e.

$$\bigcup_{k=1}^K C_k = \mathbf{X}$$
- Each cluster has at least one pattern assigned to it, i.e.

$$C_k \neq \emptyset, k = 1, \dots, K$$

- Each pattern is assigned to one and only one cluster, i.e.
 $C_k \cap C_l = \emptyset$ where $k \neq l$

(Omran et al., 2007)

This last condition holds for hard clustering. There also exist methods, so called fuzzy or probabilistic methods, that do not require a pattern to be present in only one cluster. A pattern may be part of multiple clusters in these methods, represented by some degree of belonging to a certain cluster (Inza et al., 2010). For hard clustering it holds that the clustering problem tries to maximize the within cluster similarity while at the same time minimize the similarity between clusters.

1. Pattern Representation This step of cluster analysis identifies the features that should be included into the model. The features should include as much information as possible. But, noisy or informative features can negatively impact the results of the clustering. Therefore, features with no meaning should be excluded, while on the other hand extra features should be made which include important information of the data. Examples are: **1.** In a dataset with both age and date of birth, one of the two may be included as they represent the same data. In case date of birth is kept, it may be split into year, month and day to fully capture the information included in the feature. **2.** A feature which does not represent any useful information, like shirt color when clustering on whether or not somebody will repay their loans.

The feature selection procedure will select the subset with the most optimal and robust performance. It should be stressed that feature selection does not alter the original features, but merely selects a subset of the features. As the number of features can be very large (the number of possible subsets is 2^m , where m is the number of features in the original dataset), heuristics may be used to identify the best subset. Techniques for selecting features include: filter techniques, wrapper techniques and embedded approaches. Filter techniques are often based on an univariate feature relevance score, e.g. the correlation between each feature and the target feature. These techniques are simple in implementation, but have a major drawback because they do not take into account the dependencies between features. Moreover, the filter technique is independent of the chosen model and thus does not take into account the model performance with the selected features.

Wrapper techniques do take the chosen model into account. These techniques incorporate the model into the evaluation of every subset of features. Methods can be focused on the individual features (univariate) or they can also take into account correlation and dependencies (multivariate). (Frades & Matthiesen, 2010; Inza et al., 2010)

2. Similarity measure

The similarity measure has to be chosen to identify the optimal cluster for each of the patterns. The measure should fit the nature of the data point, e.g. the distance between numerical points and categorical point should be measured with different measures. Here, some of the most commonly used distance measure will be discussed. First, Gower's distance metric will be discussed, afterwards the Euclidean, Manhattan and Minkowski metric for numerical features will be presented. To conclude the pairwise agreement and .. will be discussed (Hancer et al., 2020; Hoogenboom & Funk, 2018).

Gower's metric is defined as follows: Let x_i and x_j be two patterns with m variables denoted by $v = 1, \dots, m$, then the difference between the two patterns is defined

by

$$S_{ij} = \frac{\sum_{v=1}^m w_{ijv} s_{ijv}}{\sum_{v=1}^m w_{ijv}} \quad (2.14)$$

where w_{ijv} is the weight for variable v between observations x_i and x_j , and s_{ijv} : the difference between x_{iv} and x_{jv} .

Two examples of difference measures that can be implemented as s_{ijv} are:

The simple mismatching measure defines the distance between categorical variables as:

$$s_{ijv} = \begin{cases} 0 & \text{if } x_{iv} = x_{jv} \\ 1 & \text{if } x_{iv} \neq x_{jv} \end{cases} \quad (2.15)$$

For numerical variables the difference is defined by

$$s_{ijv} = \frac{|x_{iv} - x_{jv}|}{r_v} \quad (2.16)$$

where r_k is the range of variable v , i.e. $\max(x_{.v}) - \min(x_{.v})$ (van den Hoven, 2015). But, other similarity measures can be used, making the measure very suitable for a wide range of applications and datasets (Huang, 1998).

Other widely used distance measures are:

Euclidean distance is defined as follows:

$$S_{ij} = \sqrt{\sum_{v=1}^m (x_{iv} - x_{jv})^2} \quad (2.17)$$

with S_{ij} the distance between numerical patterns x_i and x_j .

The Euclidean distance corresponds to the typical definition of distance, i.e. it gives the shortest distance between two points without any restrictions in the directionality of the distance vector. However, it might be preferable to do have some restrictions in the directionality. The most common used measure that does have restrictions is the Manhattan distance, which assumes that one can only move horizontally or vertically (like moving over a grid), defining the distance by:

$$S_{ij} = \sum_{v=1}^m |x_{iv} - x_{jv}| \quad (2.18)$$

The Minkowski metric is a generalization of the Euclidean and Manhattan distance measures. The distance is computed by

$$S_{ij} = \left(\sum_{v=1}^m |x_{iv} - x_{jv}|^{\frac{1}{q}} \right) \quad (2.19)$$

As one can see, $q = 1$ equals the Manhattan distance, while $q = 2$ equals the Euclidean distance (Hoogendoorn & Funk, 2018, p.75).

It may be important that the data is scaled before the similarity between patterns is calculated. Non-scaled data may cause for certain features with a high spread or magnitude will become dominant over the other features in the calculations (Hoogendoorn & Funk, 2018).

For clustering time series, one can use an adjusted version of the Euclidean distance. The distance will be calculated between two time series with equal number

of observations (N), per variable (v):

$$\text{euclidean_distance_per_variable}(x_{q_i}^v, x_{q_j}^v) = \sqrt{\sum_{z=1}^N (x_{z,q_i}^v - x_{z,q_j}^v)^2} \quad (2.20)$$

With x_{z,q_i} the z^{th} observation of the i^{th} time series.

Then the overall Euclidean distance can be calculated by the sum over all distances per attribute (Hoogendoorn & Funk, 2018).

3. Clustering Algorithms

Most clustering algorithms are based on the principles of two techniques: hierarchical and partitional clustering techniques.

Partitional Clustering Techniques

Partitional clustering techniques divide the data into k clusters, where k is a predefined number. As said before, the models try to minimize some measure, e.g. the within-group sum of squares, they can be seen as optimization problems. Therefore these problems are considered NP-hard and combinatorial. Note, that the methods can get stuck in local optima, and finding the global optimal is challenging.

Probably the most widely known partitional clustering technique is the K-means method. This method minimizes the within-groups sum of squared errors using the Euclidean distance. The algorithm works as described in the following pseudo code:

Algorithm 1: K-means

```

1 randomly initialize K cluster centroids  $c_1, c_2, \dots, c_k$ ;
2 while not converged and number of iterations < max_iterations do
3   for each pattern  $x_i$  in dataset do
4     determine membership by finding the nearest centroid based on
       distance measure ;
5     assign the pattern to that cluster ;
6   end
7   for each cluster  $C_k$  do
8     calculate the new centroids of the cluster, i.e. the mean of all patterns
       assigned to that cluster
9   end
10 end

```

The number of clusters can be determined by using for example the so called *elbow method*. Although the elbow method is commonly used, the problem of determining the optimal number of clusters remains open (Inza et al., 2010). Another method of determining the number of clusters is the use of the silhouette score. The silhouette score provides insight in how tight the clusters are relative to the distance to the closest cluster. The score is calculated as follows:

Given the average distance of a point to the other points in the cluster $a(x_i)$:

$$a(x_i) = \frac{\sum_{x_j \in C_k} \text{distance}(x_i, x_j)}{|C_k|}, \text{ where } x_i \in C_k \quad (2.21)$$

And the average distance to the points in the closest cluster

$$b(x_i) = \min_{\forall C_l \neq C_k} \frac{\sum_{\forall x_j \in C_l} \text{distance}(x_i, x_j)}{|C_l|}, \text{ where } x_i \in C_k \quad (2.22)$$

The silhouette score is:

$$\text{silhouette} = \frac{\sum_{i=1}^n \frac{b(x_i) - a(x_i)}{\max(a(x_i), b(x_i))}}{n} \quad (2.23)$$

A higher silhouette score, indicates that the within cluster distance is relatively low compared to the points in the closest cluster (Hoogendoorn & Funk, 2018).

Advantages of the K-means algorithms are that it is a very easy to implement algorithm with a time complexity of $O(N)$, making it applicable for large datasets as the computational time will increase linearly in relation with the number of observations. However, the algorithm is data-dependent and greedy. The initial conditions may as a result cause the algorithm to converge to a suboptimal solution and different runs to yield different clusters. Next to that, the number of clusters needs to be specified in advance. This can be an advantage in certain applications where it is desirable to have a certain amount of clusters, but more often it is a drawback (Frades & Matthiesen, 2010; Huang, 1998; Omran et al., 2007).

Note that the K-means algorithm only takes numeric input variables, as the means are involved in minimizing the cost function. Transforming categorical variables without order into numeric values does not necessarily produce meaningful results. Furthermore, the method of transforming the categories into binary columns proposed by Ralambondrainy (1995) also does not yield meaningful results; the values 0 and 1 do not represent the characteristic of the clusters (Huang, 1998; Madhuri et al., 2014). Multiple extensions of the K-means are therefore developed.

One of these extensions is the K-modes algorithm. This algorithm uses a different similarity measure instead of the sum of squared error, making it applicable for categorical features. The measure is based on simple matching, working as follows. Let $\mathbf{x}_i, \mathbf{x}_j$ be two patterns with solely categorical variables. The dissimilarity measure between \mathbf{x}_i and \mathbf{x}_j can be defined by the total mismatches of the corresponding attribute categories of the two patterns. A smaller number of mismatches indicates two more similar patterns. (Huang, 1998). Formally this measure is defined as:

$$d_1(\mathbf{x}_i, \mathbf{x}_j) = \sum_{v=1}^m s_{ijv}(x_{iv}, x_{jv}) \quad (2.24)$$

where s_{ijv} as defined in Equation 2.15. Now, assign an object to the cluster whose mode is the nearest according to Equation 2.24. Then, update the mode of that cluster immediately, and continue with the next object. After all objects have been assigned to a cluster, reevaluate all objects again and relocate objects to clusters with the nearest node if applicable. Repeat this step, until no objects have been moved between any clusters for a whole cycle.

Another extension is the K-prototypes algorithm. This algorithm has a similar procedure as the K-means and the K-modes algorithm, but differs in the dissimilarity measure. The K-prototypes algorithm can handle both numerical and categorical features by taking the squared Euclidean distance measure (Equation 2.17) for the

numerical features and calculating the mismatches (Equation 2.15) between categorical features (Huang, 1998; Madhuri et al., 2014).

Hierarchical Clustering Techniques

As the name suggests, these techniques obtain a cluster by construction a hierarchy of clusters. Algorithms in this category construct a cluster tree, which can be represented by a so called dendrogram. Each node in this tree represent a partition of the tree. An example of a dendrogram with 10 observations is shown in Figure 2.4. The y-axis represents the distance measure and the x-axis represents the observations. The figure also shows how a cut would be made if the threshold of the distance between the clusters would be h , resulting in 4 clusters: $\{2,1,3\}$, $\{4,9\}$, $\{7,6,8\}$ and $\{5,10\}$.

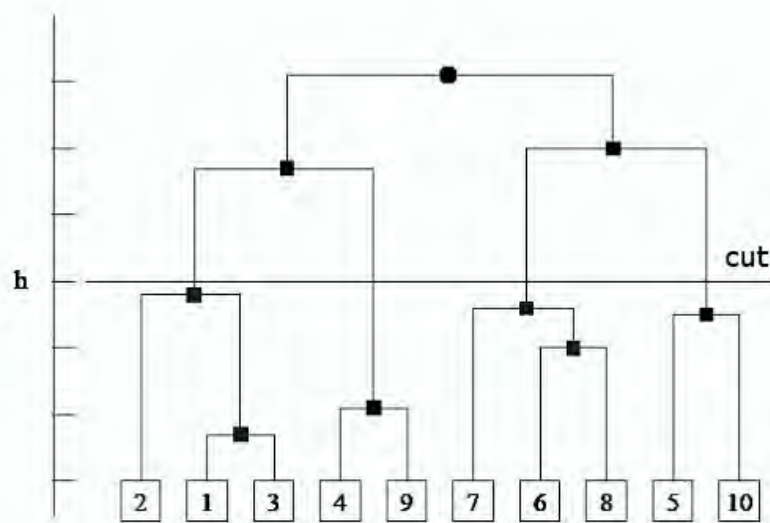


FIGURE 2.4: Representation of a dendrogram with an example of a cut.

There are two types of hierarchical clustering techniques namely, agglomerative and divisive techniques. The methods differ in how they construct the tree; agglomerative hierarchical methods build a tree so called bottom up. This means that in the beginning every data point is treated as separate cluster. The algorithm then iteratively searches for clusters that may be merged based on their similarities, i.e. clusters that are close based on their similarity measure.

In contrast to these methods are the divisive clustering methods, or top-down methods, in which the data points are seen as one single cluster and in every step this cluster is split into separate clusters based on their dissimilarities, i.e. observations that are far away based on their similarity measure.

A major drawback of hierarchical clustering techniques is that they are computationally expensive ($O(N_p^2 \log N_p)$ in terms of time complexity) and therefore not suitable for large data sets.

Examples of hierarchical clustering algorithms may be found in (Omran et al., 2007; Swarndeep Saket & Pandya, 2016).

2.5 Evaluation Metrics

Determining the accuracy of the predictions of implemented models is necessary to compare them, but also to interpret the quality of the predictions. Therefore, in what follows evaluation metrics will be discussed. Only the evaluation metrics suitable for regression models will be presented, as the problem discussed in this thesis is a regression problem.

The most widely used metrics for regression models are: Mean Square Error (MSE), Mean Absolute Error (MAE) and R Squared (R^2), or an adjusted version of those (Handelman et al., 2019; Killada, 2017; Wu, 2020). Therefore, those metrics will be discussed in the remaining of this section.

Note, in all discussed methods y_i represents the i^{th} value of the responsive variable whereas \hat{y}_i stand for the prediction of this i^{th} variable. Further, N is the total number of predicted variables.

Mean Square Error (MSE)

The Mean Squared Error is a useful metric when it is desirable to penalize all errors proportionally. It can be seen from its formula:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.25)$$

As the errors are squared, even small errors will contribute to the value of the MSE.

The MSE can be adjusted to the Rooted Mean Square Error (RMSE), which is calculated by taking the square root of the MSE (\sqrt{MSE}). In other words, the square root is taken of the average error made.

The contribution of large errors to the RMSE and MSE will be disproportionately higher than the contribution of small errors. Hence, these models is sensitive to outliers and allow for minor errors.

The outcome of the MSE and RMSE cannot easily be compared to the outcome of models applied to different datasets. The outcomes are highly influenced by the scale of the numbers used in the model. However, normalization of the RMSE is possible, which makes the metric suitable for comparison. Further, they can be used for comparison between different models applied to the same data. A smaller MSE or RMSE would indicate a better performance of the model.

Mean Absolute Error (MAE)

The Mean Absolute Error is argued to be the more intuitively interpretable than the (R)MSE. This metric uses the absolute errors between the predicted value and the actual value. The MAE is calculated as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (2.26)$$

As it takes the absolute value, all errors influence the MAE proportionally without penalizing certain errors in particular. It logically follows that a low value of the MAE is desirable.

An adjusted version of the MAE is the Mean Absolute Percentage Error or MAPE.

As the name suggests, this metric calculates the mean absolute percentage of the errors made, ie:

$$MAPE = \left(\frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{y_i} \right) * 100 \quad (2.27)$$

R Squared (R^2)

R Squared, also called the coefficient of determination, gives a percentage of the output variability, i.e. "how much of the variability in [the] dependent variable can be explained by the model" (Wu, 2020). As this is an universal measure, this metric can be used to compare models applied to different datasets.

R^2 is calculated as follows:

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (2.28)$$

Where $\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i$, i.e. the mean target value.

The best possible score of this metric is 1.0, in contrast to the other metrics discussed before which aim for a score close to 0.0.

However, the metric does not take into account the chance of overfitting. Therefore, an adjusted version (adjusted R square) has been proposed. This version penalizes the model for using too much predictive variables.

Let p be the number of predictive variables and n be the number of datapoints, then the adjusted R square (denoted by \bar{R}^2) is given by:

$$\bar{R}^2 = 1 - (1 - R^2) \frac{n - 1}{n - p - 1} \quad (2.29)$$

The results of the adjusted R square will always be lower than the results of R square. When the results of both versions give similar results, one may conclude that the model is robust.

One should be cautious when using the R^2 measure for time series models as it represents the overall fitness of the model (of the past observations) instead of the predictive power.

Chapter 3

Problem Statement

This chapter will present the research question, including the sub research questions. Furthermore, the assumptions made in this thesis will be stated as well as the research outline.

3.1 Research Goal

In order to help Sandvik with gaining more insight in their after-sales business, the following research question has been formulated:

Which model performs the best for the prediction of the monthly potential revenue and sales per customer on the aftermarket and what are the factors that influence these potentials the most?

This question will be the main focus of this thesis as it helps Sandvik with understanding and forecasting their sales and the associated revenue per month, while it is still academically interesting to investigate the wide range of possible models. Some model typologies will be discussed in Chapter 2 and the performance of these models will be compared in different application areas in Chapter 4.

As this thesis will work with a lot of different customers and it tries to predict their behaviour, the use of clustering methods seems appropriate. Different clustering methods and their performance in customer segmentation will be considered and discussed in 4.

Furthermore, historical sales data can be interpreted as a time series, which may improve the predictive performance of the models. This will be further justified in the literature review in Chapter 4.

This resulted in the following sub-research questions:

1. **Can the yearly potential revenue and number of sales be estimated by implementing time independent forecast models?**
2. **Can the yearly potential revenue and number of sales estimate be improved by implementing time dependent forecast models?**
3. **Can the yearly potential revenue and number of sales estimate be improved by combining clustering methods with forecast models?**
4. **which features influence the forecasts and does the importance of features change when different models are applied?**

As implied all models considered will be optimized as best as reasonable in order to yield the best results.

In this research, the focus will be on one specific type of product sold by Sandvik, namely rock drill parts. The choice for this product will be explained in Chapter 5. Furthermore, some assumption will be made about the data and behaviour of the customers, namely:

- **Customers do not buy any products in advance, i.e. they do not have any stock on their sites.** From this assumption it follows that estimating the number of sales per client will reveal their use and an indication of the lifetime of the products. Further, this assumption is reasonable as the holding costs for the stock are now paid by Sandvik instead of their customers, which will be preferred by the customers.
- **All rock drill parts are of the same type.** Rock drill consists of a lot of small parts. As Sandvik sells a number of different rock drills, the number of those separate parts quickly increases. Therefore, this category could be split up in a lot of small subcategories. In order to have sufficient data for the analysis, all the types of rock drill parts will be considered as if they were the same.
- **Customer characteristics do not change over time.** As the changes of these characteristics are not saved consistently or saved at all, this assumption will be made in order to be able to analyse the data.

3.2 Research outline

The remaining of this thesis will be divided in six sections. First, an overview of literature relating to the topic of this research will be presented in chapter 4. This chapter will also contain some aspects and methods that will not be used in this research. However, these aspects could be used by Sandvik for future projects, once the right data is gathered and it gives a good overview of possible methods. Therefore, these methods are included in the literature overview.

Second, in chapter 5 the data used in this thesis will be presented and discussed. This includes some first insights and statistics of the data, but will not contain feature engineering.

Then, the methodology will be discussed and the choice of methods will be reasoned in chapter 6. The results of the research will be presented in the chapter that follows. This thesis will conclude with a conclusion of the research and a discussion of recommendations and limitations of the conducted research in the chapters 9 and 10.

Chapter 4

Literature Review

This chapter will give an overview of the different fields of research related to gaining business insight of aftermarket services. As stated before, this will include some fields that are not used in this thesis, namely lifetime estimation and asset evaluation. These two fields will be discussed at the end of this chapter.

First, methods for sales forecasting will be discussed as this is the main subject of this thesis. Secondly, literature about time series analysis and customer clustering (i.e. customer segmentation) will be presented. Then, feature selection methods for both regression and clustering algorithms will be presented.

Note that the area of machine learning is a fast changing field, resulting in researches and state-of-the art algorithms to be quickly outdated. Therefore, the focus of the reviewed literature in this chapter will be on the most recent studies.

4.1 Revenue/Sales Forecasting

As discussed in Chapter 2, sales forecasting can, among other things, help by planning the logistics of the supply chain and improve the business' performance (Chen & Lu, 2017; Ji et al., 2019). A balance should be found in meeting customer's demand and holding costs for the company. Therefore, forecasting sales has been a returning problem for many companies. Which also led to a variety of ever improving forecast methods (Venishetty, 2019).

In their research Bajari et al. (2015) compare standard economic models for predicting demand (i.e. predicting sales) with machine learning algorithms. For this comparison they use grocery store data of a store chain for a period of six years. More specifically, their focus is on a specific product type, namely salty snacks. The data consists of over 1.5 million data records of more than 3,000 unique products. Not only information about price and sold quantity is used, also information about promotions, brand, flavour, package size, etc. is included in the model. They find that the Random Forest (RF) outperformed the other models including, Support Vector Machines, Bagging and LASSO.

However, later in 2015 a new algorithm gained popularity amongst data scientist. As stated in the paper by Chen and Guestrin (2016), Extreme Gradient Boosting (XGBoost) has outperformed many other algorithms in multiple Kaggle competitions in 2015. 17 out of the 29 competitions, under which also sales prediction competitions, recorded a XGBoost algorithm in their top three best performing algorithms. Since then, the popularity and predictive power of this algorithm has not declined (Mello, 2020; Pavlyshenko, 2019).

Since a lot of researches about revenue and sales forecasting include time series analysis, more research on the topic of revenue and sales forecasting will be discussed in the next section. These researches also include performance comparison of the XGBoost algorithm.

4.2 Time Series Analysis

The number of sales of a company is often influenced by time of the year (e.g. summer or winter) and special days (e.g. Christmas time would be more busy for toy stores). These are regular fluctuations, as they return every year around the same time and are called seasonal effects. As discussed in Chapter 2, time series models try to capture this seasonality along with for example the trend in the data. As sales data has these seasonality and trend, it can be treated as time series.

However, one should be aware of some limitations of time series analysis for sales data. Three of those limitations are stated by Pavlyshenko (2019):

- Data that has been captured for a long period of time has to be available in order to capture seasonality
- "Sales data can have a lot of outliers and missing data."
- There are a lot of exogenous factors that influence the sales

These limitations should be kept in mind when implementing such models for sales data.

Note that time series analysis over a period of time can be done in two different ways. Either in a so called one-step or multi-step fashion (Tyrallis & Papacharalampous, 2017). The one-step approach forecasts one time step at a time. The one-step ahead method, iteratively forecasts the desired number of steps. In each iteration including the prediction of the previously time steps as additional input for the model. The multi-step approach forecasts the values all at once (Tyrallis & Papacharalampous, 2017).

In the paper of Pavlyshenko (2019) the data from the Kaggle competition "Rossmann Store Sales" is used to predict the future sales. This data consists of the historical sales for 1,115 Rossmann drugs stores. In the paper multiple models are considered and applied to different feature sets. The first case uses a Random Forest algorithm as described in Chapter 2 with the following features: promo (whether or not there was a promotion running), day of week, day of month and month. Where categorical features were one-hot-encoded. A constant under- or over-valuation of the sales in the validation set was observed, as is common to the use of machine learning methods for time series. This bias can be corrected with the use of linear regression on the validation set (only in case of a small trend).

The paper also discusses another approach, namely stacking models. The stacking can be done by giving the results of the validation set of the model from the previous stacking level as input for the models of the next stacking level. Most of the first level models considered in the paper are based on the XGBoost algorithm. The second level models are an Extra Tree, Linear and Neural Network model, which are combined into the final output. The features engineered for this stacking model are mostly aggregated features of the target variable and lags of the target variable based on grouping by different features. It is shown that the stacking increased the accuracy of the predictions in this case study, suggesting that stacking models may

improve the overall model performance.

In their paper Tyralis and Papacharalampous (2017) compare an extended version of the ARMA model and a Random Forest (RF) model in an one-step ahead approach for time series forecasting. In the case study 16,000 simulated time series with added white noise¹ are used. The simulated data is derived from a variety of models of an extended version of ARIMA namely, the Autoregressive Fractionally Integrated Moving Average (ARIFMA) model. Next to the simulated time series, 135 time series of annual instrumental temperatures are used. All of these datasets consisted of stationary data. Each of the time series has a length of 101 of which 100 observations are used for fitting the model and the last observation is used as the observation to be predicted.

They conclude that the RF models performed better in case of a smaller number of lagged input variables (e.g. moving average variables), and performed better on the non-simulated data. The performance of the extended ARMA model was slightly better on the simulated data. Note that this data was simulated by a corresponding AR(FI)MA model, resulting in a higher expected performance of those prediction models. The authors further conclude that the performance of an optimized RF² is comparable with the performance of other algorithms and they can be considered appropriate to use for time series analysis. However, they stress the fact that it is preferred to use a variety of models Tyralis and Papacharalampous (2017).

Another research discussing the forecast of sales, now applied to the furniture industry, is presented by Yucesan et al. (2018). They investigate sales data of a large furniture company based in Turkey from January 2009 to December 2015. In the data a distinction is made between products for dining rooms, bedrooms, teen rooms, sitting groups and armchairs. For all categories a prediction of the sales for the coming month are made using the following models: ARIMAX, NN (single models) and a combination of those two (hybrid model). The ARIMAX model is a generalization of the ARIMA model. It allows incorporation of an external input variable. As discussed in 4.5.1 ARIMA models are linear and are unable to fit nonlinear data. But, they are very flexible and are suitable for non-stationary data.

NN also have limitations, like dicey results, as stated by the researches, but they are able to model non-linearity. Therefore, the authors propose a hybrid model to combine the advantages of both models and reduce their limitations. In this hybrid model, the ARIMAX model is fitted to the data. The resulting residuals are then used as input for a NN, along with some independent variables, like month, number of vacation days. The worst performing model was the single ARIMAX model and the best performance were achieved by the hybrid ARIMAX-ANN model.

In a recent study Hewamalage et al. (2021) compare a wide range of Recurrent Neural Network (RNN) with more traditional time series forecasting models, like an ARIMA model. The data used in their research is that of multiple forecasting competitions, all univariate, multi-step-ahead single seasonality datasets. The Neural Networks are used on data with and without removed seasonality. However, all

¹White noise is a time series of independent random variables with mean 0 and variance σ^2 (de Gunst, 2013)

²Note that the optimized Random Forest is one with a small number of input variables, a Random Forest with many input variables is found to perform worse than the other methods considered in their research.

datasets differ in the number of time series, seasonality strength and length of the time series.

They conclude that out of the three gated units GRU, LSTM and ER, the Long Short Term Memory (LSTM) cell performed better. However, not statistically significantly. In all but one models, removing seasonality improved the performance of the models. In the single dataset in which removing the seasonality did not improve the performance, little seasonality is present in the dataset beforehand. Furthermore, the traditional models (ARIMA and ETS) performed better in most datasets. Although it should be noted that the RNN "with LSTM cells and peephole connections optimized by COCOB was a competitive model combination for forecasting" (Hewamalage et al., 2021, p.417).

4.3 Customer Segmentation

As said by Huang (1998), partitioning objects into clusters may reveal interesting groups. With clustering methods, the customers can be grouped into different clusters with the same behaviour which may improve the performance of the overall models looked into in this thesis. The act of clustering customers is also known as customer segmentation and will be discussed in what follows. Furthermore, the different clusters can be used for more specific marketing and supply chain strategies (Caruso et al., 2019; Hjort et al., 2013).

In this research the focus will be on the unsupervised application of clusters, as there are no predefined classes of customers. Note that the goal is to identify these classes. This information can then be used as input for the model to predict the potential revenue as similar customers can be more likely to have similar purchase behaviour (Swarndeeep Saket & Pandya, 2016). As discussed by Chen and Lu (2017), identifying the clusters before forecasting the sales can significantly boost the performance of the sales forecast. Furthermore, it reduces the computational time and reduces the amount of irrelevant data.

In the paper of Caruso et al. (2019) multiple cluster algorithms are applied to mixed-type data. They stress that most existing clustering approaches are limited to numerical or categorical data only, which is not common to find in real-world datasets. Applying these clustering algorithms may therefore cause a loss of information, as they need to discard or summarize certain features. The three clustering algorithms that were described are the *K*-Means and *K*-prototype algorithms as well as an iterative clustering algorithm presented by Cheung and Jia (2013).

The real-world data that is used by Caruso et al. (2019), is that of departure and arrival delays of US domestic flights from the Bureau of Transportation Statistics database. The data from the first of August 2017 is used, which consists of 1426 flights. The nine features include both numerical and categorical data and are all used in the *K*-prototypes and iterative clustering algorithms. The *K*-means algorithm is only applied to the five numerical features.

They conclude that the *K*-prototype and the iterative clustering algorithm allow for non-numerical features while still retaining the efficiency of the *K*-Means algorithm. The *K*-prototype algorithm is found to work better with numerical features, while the iterative clustering algorithm relied more on the categorical features in the data.

More customer segmentation algorithms will be discussed in the following section as these algorithms combine the feature selection procedure and the clustering of the customers.

4.4 Feature Selection

Another important part of building an appropriate model is selecting the best (number of) features. In case of machine learning, the more the better is not always true. One of the main drawbacks of too many features is overfitting, especially if some of the features are correlated with each other or not correlated at all with the response variable. Choosing only one of the correlated features can improve the performance of the algorithm significantly. Further, too many features can lead to computational inefficient algorithms. Therefore, feature selection is recommended (Gregorutti et al., 2017; Hancer et al., 2020).

As said in Section 2.4.4 there are multiple different feature selection methods, which can be divided in multiple classes, like filter, embedded and wrapper methods. Filter methods choose the features independent of the model, i.e. they do not take into account the model outcome. This limits their performance, but makes them computationally less intensive. The embedded approach does take the learning algorithm into account and selects the features during the learning process. Examples are regulation methods like lasso or decision trees. Lastly, the wrapper methods, which incorporate the learning method within the feature selection. The model output is used in the determination of the best subset of features. This makes them computationally more intensive than the filter methods, but they yield better results in general (Gregorutti et al., 2017; Hancer et al., 2020). Therefore, this section will be focused on wrapper methods.

Wrapper methods for tree-based models

A common issue regarding feature selection methods is their instability. This instability increases if the variables are correlated (Gregorutti et al., 2017). Therefore, multiple methods have been proposed to handle datasets with correlated features. For explanation of these methods, the permutation importance measure will be used. The permutation importance measure can be derived by Procedure Permutation Importance Measure. Another importance measure is, for example the gini-importance measure (also called impurity-based importance). This importance measure calculates the (normalized) decrease in the loss function after an feature is used to make a split. However, this importance measure is more likely to favor features with a high number of unique values (e.g. continuous variables) over features with a small number of unique values (e.g. binary features) (scikit-learn developers (BSD License), 2020).

Procedure Permutation Importance Measure (Breiman, 2001)

- 1 Construct a classification model ;
 - 2 Permute the values of the m^{th} variable in the out-of-bag example randomly ;
 - 3 Run the out-of-bag data through this model ;
 - 4 Count the number of errors made using these sets of models and call this the misclassification rate ;
 - 5 Consider the variable with the largest misclassification rate as most important ;
-

Note that the Permutation Importance Measure can also be applied to regression problems by determining the misclassification rate by subtracting the original error from the permuted error ($e^{perm} - e^{orig}$), or by dividing the permuted error by the original error (e^{perm} / e^{orig}).

This permutation importance measure is used in the paper of Gregorutti et al. (2017). In this paper, the authors propose a method which uses the permutation importance measure in a setting with correlated variables and a random forests as predictor model. The idea is to rank the input variables with the permutation importance measure and choosing the appropriate features for the model while reducing the computational cost of the algorithm.

One precondition to their calculations of the permutation importance is that the regression function should have an additive structure. Further, an observation (\mathbf{x}, y) is assumed to be a normal vector. These conditions can easily be derived by using a random variable ϵ such that: $y = f(\mathbf{x}) + \epsilon$.

As one can image, having two variables that have a positive correlation influence the importance measure. When permuting one of the two, the other variable would still be able to cover similar information. This then leads to less misclassifications and therefore a lower importance measure. If the set of positive correlated variables becomes bigger, the importance measure quickly decreases to zero.

In case of anti-correlated variables the importance measure increases if one of the variables is randomly permuted. This logically follows from the fact that the model needs both variables since they explain \mathbf{Y} in different directions. When one of the two is left out of the model, the model is not able to predicts \mathbf{Y} correctly.

Gregorutti et al. (2017) propose two approaches for feature selecting. The first approach is the Non Recursive Feature Elimination (NRFE) as developed by Svetnik et al. (2004) and Diaz-Uriarte and De Andres (2006). The NRFE works as follows:

Procedure NRFE

- 1 Use an Importance Measure to rank the variables and discard the ones with the smallest importance ;
 - 2 **while** *Not all variables are eliminated* **do**
 - 3 | Build a model ;
 - 4 | Eliminate the less relevant variables
 - 5 **end**
-

The second approach they considered is Recursive Feature Elimination (RFE), which does update the importance measure after reconstruction a random forest. The algorithm consists of the following steps as explained in Procedure RFE and was first implemented by Jiang et al. (2004).

Procedure RFE

- 1 **while** *Not all variables are eliminated* **do**
 - 2 | Build a model ;
 - 3 | Rank the variables using an importance measure ;
 - 4 | Eliminate the less relevant variables
 - 5 **end**
-

Comparing the two algorithms RFE and NRFE in a dataset with correlated variables leads to a preference to the RFE, as this algorithm calculates the permuted importance measure every iteration. By recalculating the importance measure, the

algorithm allows the remaining variables to be re-evaluated after the correlated variables are eliminated. This leads to better performance of the overall model.

Wrapper methods for clustering

In Hancer et al. (2020) and de Amorim (2016) a wide range of feature selection methods for clustering are discussed. Some of these approaches will be discussed in what follows. The discussion limits itself by methods that can be applied to either the K -means or the K -prototypes algorithm.

In the paper by Hancer et al. (2020), no experiments are performed. The authors choose for a discussion of multiple papers which on their turn experimented with multiple approaches. This resulted in an overview of more and less used algorithms and their applications. In the overview for the K -means algorithm, the focus is on weighting the input features in such way that irrelevant and redundant features are weighted less than relevant features. This allows for a sophisticated features selection method, which in general can be represented by the following formula:

$$W(Z, C, W) = \sum_{k=1}^K \sum_{z_p \in C_k} \sum_{f_i \in F} w_{f_i} (z_{pi} - c_{ki})^2 \quad (4.1)$$

where w_{f_i} is the weight of feature f_i in subset F , z_{pi} the i^{th} feature of the p^{th} instance of the data and K the number of clusters.

Hancer et al. (2020) mostly empathized the weighted feature selection methods also discussed in the paper of de Amorim (2016). This paper gives a comparative overview of multiple weighted feature selection procedures and the methods discussed in these papers will therefore be discussed later on in this section. One should keep in mind that most of these algorithms have the challenge of predefining the control parameter which influences the outcome of the algorithm.

In de Amorim (2016) 13 real-world data sets and 20 generated datasets are used to deploy multiple weighted based k -means algorithms. These algorithms include, but are not limited to: Attribute weighting clustering algorithm (AWK) (Chan et al., 2004), Weighted K -means (WK-Means) (Huang et al., 2005), Improved K -Prototypes (IK-P) (Ji et al., 2013) and Intelligent Minkowski Weighted K -Means (iMWK) (De Amorim & Mirkin, 2012).

In all considered datasets the number of clusters are known in advance, which allows for comparison of the performance of the algorithms. The datasets consist of solely numerical, solely categorical or a mixture of numerical and categorical features. The numerical features are standardised and the categorical features are transformed into numerical features except in the experiments with the Attribute weighting and Improved K -Prototypes (IK-P).

The experiments are compared in terms of the adjusted Rand Index (ARI). Note that parameter estimation was not part of the research and the optimal parameters were found by trying the values between 1.0 and 5.0 in steps of 0.1. As all but one (the iMWK-Means algorithm) are non-deterministic, all algorithms are run 100 times at each parameter after which the parameter with the highest average ARI is selected. They conclude that the iMWK-means reaches the highest ARI in 9 of the 13 real-world datasets, 8 times in the real-world datasets with noise added, and in all 20 generated datasets (with and without noise³).

³The noise is a variable composed entirely of uniform random values

Amongst the algorithms that can handle categorical features, the IK-P algorithm reaches the best performances in all but one of the real-world datasets. However, in case of noise, the Attribute Weighting algorithm performed better than the IK-P algorithm 6 out of the 13 times. Furthermore, it should be mentioned that the IK-P algorithm was unable to detect the correct amount of clusters in one of the noisy datasets. However, as stated by the authors this may be related to the data spread and would not happen if the surely irrelevant features were removed beforehand (de Amorim, 2016).

4.5 Related Problems

In what follows two related problems to the problem discussed in this thesis will be presented. These problems will not be considered in this thesis as the appropriate data is not yet available at Sandvik. But, Sandvik has the goal to eventually consider these problems as well. Therefore, some related literature will be presented in order to give a brief overview of possibilities for those models. The first of those problems is estimating the remaining useful lifetime of Sandvik's components in the machines of their customers, so called Remaining Useful Lifetime (RUL). Using these RUL estimations in the sales forecast model proposed in this thesis, the predictions can become even more accurate. Furthermore, those RUL estimations can be used in the second related problem, that of asset evaluation. Asset evaluation focuses on estimating the worth per machine at customers' sites. This estimation gives an indication of the revenue forecast per machine. The revenue forecast model proposed in this thesis can thus help by the asset evaluation. With asset evaluation, the focus is on how much revenue an asset (here, machine) possibly yields. This can be only based on the replacement parts, but can be elaborated by including maintenance into the evaluation too.

4.5.1 Lifetime Estimation

In the area of lifetime estimation of assets, multiple articles propose methods for estimating the Remaining Useful Lifetime, also called RUL. The remaining useful lifetime of a product indicated the expected time until failure of that product. The measure is widely used for lifetime estimation (when to replace the product) and applied in scheduled maintenance (when to repair/perform maintenance on the product). This section will present some of the different methods of estimating this RUL discussed in the literature.

The RUL of an object is, as said before, the expected time until failure. There are multiple different types failures that can occur. For example, a part can break down due to rust, a part can deform due to pressure or a technical error can occur. The main purpose of estimating the RUL of a part is to determine which failure is most likely to occur and to perform maintenance timely. Other useful knowledge achieved from RUL estimation are the inspection interval and the forecasting of ordering spare parts' quantity (Changhua et al., 2018; Si et al., 2011; Zio & Compare, 2013). For the problem discussed in this research, the application of the RUL to forecast the order quantity of spare parts is the most interesting.

Machines nowadays are more complex than machines a few decades ago, and there is an increasing number of different strings of events that can lead to failure. Besides that, machines are becoming more reliable, which makes the number of experienced engineers lower. Therefore, human decision making is found insufficient

when handling these machines. This has led to an increase in the number of researches to the different topics of determining failure in machines, like predictive maintenance (preventing the machine to fall out) and lifetime estimation (estimating when a machine will fail). This increase of researches about the topic leads to a variety of methods and algorithms, each with specific assumptions and requirements. Selecting the right methods is thus not straightforward and some level of mathematical and business insight is needed (Sikorska et al., 2011).

As stated by Hu et al. (2015) the majority of research done on the topic of life prediction of equipment is proprietary of companies as it examines their equipment, and therefore not publicly available. To overcome this problem, they state that using a reliability prediction model with a threshold can be seen as substitute. But, the literature reviewed in this research will be limited to the RUL estimations.

Following the classification of RUL estimation methods presented by Abid et al. (2018), the methods can be divided into multiple subgroups, first: experience approaches and degradation model approaches. Experience based approaches use the data of inspections and maintenance feedback of a long period of time. This data will be fitted to a statistical failure distribution from which the RUL of other parts will be estimated. The degradation model approaches aim to model the degradation process and try to estimate the failure time. Then the Remaining Useful Lifetime is the time from degradation detection until failure.

The degradation model approaches can be divided in physical and data-driven approaches. The physical approaches use physical and mathematical relations to determine when a certain part will wear down, while the data-driven approaches use the data retrieved from sensors in order to predict the time between degradation detection and failure. The latter can again be divided in two groups (statistical and Artificial Intelligence (AI)) accordingly to the literature (Abid et al., 2018; Changhua et al., 2018; Si et al., 2011).

Statistical Approaches

Model based approaches can give a prediction with high precision if they are adapted on component level, instead of system level. The models should be verified and require extensive experimentation. Furthermore, the models are only valid if the systems are not upgraded or changed in any other way (Abid et al., 2018; Saha et al., 2009).

According to Si et al. (2011) the data used in RUL estimations can be categorised into two main types: event data, i.e. historical data about failures, and condition monitoring (CM) data. For CM of mining machinery think about transducers, like "accelerometer, acoustic emission sensors, tachometers, thermocouples, etc." which collect the environmental, operational and performance information of the machines (Chaulya & Prasad, 2016).

The availability of the different types of data are crucial for choosing the right statistical model. As presented by multiple papers, (e.g. Abid et al. (2018), Sikorska et al. (2011) Saha et al. (2009)) the data used in lifetime prediction can be seen as time series. As there are multiple points in time that a part is diagnosed and those diagnoses will be used for the estimation. Autoregressive Integrated Moving Average (ARIMA) or Autoregressive Moving Average (ARMA) are widely used to analyse the diagnoses data and predict the RUL. However, as stated in 2.4, ARIMA models can only find linear dependencies within the data. Furthermore, finding the right model parameters can be hard, as no efficient estimator is known.

Artificial Intelligence (AI) Approaches

In contrast to the statistical approaches, the AI approaches can more easily adapt to upgrades or changes in the system. Further, there is no knowledge necessary about the model's degradation mechanism, which is a huge advantage especially in complex systems. Recurrent Neural Network (RNN) are commonly used for the problem of predicting the RUL (Abid et al., 2018). RNN has the advantages and disadvantages as presented in 2. To sum up the most important ones: RNNs have the ability to incorporate past events into the future predictions, and are very powerful in revealing structures and patterns in the data. However, they suffer from vanishing or exploding gradients and are hard to tune.

4.5.2 Asset Evaluation

Many of the methods described in the section of Sales and Revenue Forecast are also applicable for revenue or sales forecasting models, as both areas require regression models. Although the amount and kind of data would be different. Therefore, some literature will be presented here to discuss the results of these researches. As stated by Weigand (2019), machine learning is still relatively little investigated in the area of asset pricing, traditional methods still dominate the field. Further, asset evaluation research usually focuses on the valuation of stocks and bonds. Whereas one can imagine, the value illiquid assets behaves differently than that of liquid assets (Aubry et al., 2019). Therefore, even though the assets of interest for Sandvik are illiquid, some literature about liquid asset pricing is included in the literature review.

As computers increase their compute power at a lower cost and more data becomes available, machine learning is becoming more popular in more business areas. Further, the quality of the predictions of automated valuation mechanisms is higher than those of traditional methods, and can make a market more liquid. Since better evaluations lead to an increased availability of information and therefore, reduces the need for a trading intermediary, which on its turn will increase the number of trades. Furthermore, better valuation mechanisms lead to a higher productivity and risk-reduction for all parties (Aubry et al., 2019; Gu et al., 2019; Weigand, 2019).

An overview of multiple machine learning techniques in the area of asset evaluation is given by Gu et al. (2019). Then, Weigand (2019) build upon that paper and presented another overview about machine learning in asset pricing.

As said, Gu et al. (2019) gives an overview of multiple machine learning approaches and uses the task of measuring asset risk premiums. They focus on the two problems of "predicting returns in the cross-section and time series."

They state a few benefits that can be achieved by using machine learning methods. First, they find that "machine learning methods as a whole has the potential to improve our empirical understanding of asset returns", which would be desirable for Sandvik. Secondly, a large feature space can be handled by machine learning methods, as long as penalization or dimension reduction is preformed. The best performing algorithms were Neural Networks and regression trees. The performance of Neural Networks is, as it seems, slightly better, but the difference is not statistically significant.

The machine learning methods that are discussed in the paper include, Partial Least Squares (PLS), Principal Components Regression (PCR), Generalized Linear Models (GLM), regression trees and Neural Network (NN). These machine learning

methods are compared to an Ordinary Least Square (OLS) method and other linear models (Gu et al., 2019).

Performance

As mentioned before, Gu et al. (2019) implemented these models and applied them to data of almost 30,000 stocks over the period of 1957 to 2016. The data per stock include 74 dummy variables to represent the industry sector, time series variables and characteristics of each stock, like size, growth in shareholder equity and book-to-market ratio (i.e. the difference between the company's book value and market value). In this study, "[The researchers] find that trees and neural networks unambiguously improve return prediction". Note, that they find that the neural networks perform better for larger and more liquid stocks. This indicated that neural networks do require larger amount of data to perform well. Further, they state that the generalized linear model does not consistently outperform the linear models. The lack of predictor interactions may be the reason for this result as the predictors in this research are highly correlated.

Krauss et al. (2017) also looks into (deep) neural networks with one and three hidden layers, gradient-boosting trees, random forests and combinations of the models. They conclude that in their application, one with a noisy feature space, the random forest algorithm performs best. However, they note that the performance of deep neural networks might still improve when the tuning is improved. But, the best results are achieved when combining the individual models into an ensemble. Suggesting that the models supplement each other. The methods of ensemble the models can be looked into in further research.

The data used in the Krauss et al. (2017) research is data of the S&P 500, an index containing the leading 500 companies in the US stock market, from January 1990 to October 2015. The survival bias, only taking into account stocks that done well instead of also looking at stocks that did not survive, is omitted by a binary matrix. The binary matrix contains a one if the stock was also present in the subsequent month, and zero otherwise.

Aubry et al. (2019) evaluates the valuation of illiquid, heterogeneous assets: paintings. These asset's values are linked in a complex way with hard to quantify characteristics. The data consists of paintings that have been auctioned between 2008 and 2015, resulting in a dataset with over 1.1 million paintings. The neural network they propose works significantly better than the standard hedonic pricing model often used in these settings. They find that the machine learning model can help to correct the biases in expectations formation by human experts.

Another study on evaluated illiquid assets is conducted by Erel et al. (2018), who try to evaluate nominated directors using machine learning. They conclude that the machine learning algorithms accurately predict whether or not a director would perform poorly, especially the Boosted tree algorithm XGBoost. The data used for the research is that of directors appointed between 2000 and 2011 for the training set and two years of appointed directors for the test set, resulting in a test and train set of over 41,000 directors. Note that this data set is significantly smaller than the data set used in the three above mentioned researches. This may be the reason for the NN to be outperformed by the XGBoost algorithm.

Chapter 5

Data

<Confidential.>

Chapter 6

Methodology

<Confidential.>

Chapter 7

Experimental Setup

<Confidential.>

Chapter 8

Results

<Confidential.>

Chapter 9

Conclusion

<Confidential.>

Chapter 10

Discussion

<Confidential.>

Bibliography

- Abid, K., Mouchaweh, M. S., & Cornez, L. (2018). Fault prognostics for the predictive maintenance of wind turbines: State of the art. *Joint european conference on machine learning and knowledge discovery in databases*, 113–125.
- Aksoy, L., Cooil, B., Groening, C., Keiningham, T. L., & Yalçın, A. (2008). The long-term stock market valuation of customer satisfaction. *Journal of Marketing*, 72(4), 105–122.
- Artigue, H., & Smith, G. (2019). The principal problem with principal components regression. *Cogent Mathematics & Statistics*, 6(1), 1622190.
- Aubry, M., Kräussl, R., Manso, G., & Spaenjers, C. (2019). Machine learning, human experts, and the valuation of real assets. *HEC Paris Research Paper No. FIN-2019-1332*.
- Avanade. (n.d.). *Our story*. Retrieved September 14, 2020, from <https://www.avanade.com/en/about-avanade>
- Bagheri, R. (2020). Retrieved November 12, 2020, from <https://towardsdatascience.com/understanding-singular-value-decomposition-and-its-application-in-data-science-388a54be95d>
- Bajari, P., Nekipelov, D., Ryan, S. P., & Yang, M. (2015). Machine learning methods for demand estimation. *American Economic Review*, 105(5), 481–85.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5–32.
- Brockwell, P. J., Brockwell, P. J., Davis, R. A., & Davis, R. A. (2016). *Introduction to time series and forecasting*. Springer.
- Caruso, G., Gattone, S. A., Balzanella, A., & Di Battista, T. (2019). Cluster analysis: An application to a real mixed-type data set. *Models and theories in social systems* (pp. 525–533). Springer.
- Chan, E. Y., Ching, W. K., Ng, M. K., & Huang, J. Z. (2004). An optimization algorithm for clustering using weighted dissimilarity measures. *Pattern recognition*, 37(5), 943–952.
- Changhua, H., Hong, P., Zhaoqiang, W., Xiaosheng, S., & Zhang, Z. (2018). A new remaining useful life estimation method for equipment subjected to intervention of imperfect maintenance activities. *Chinese Journal of Aeronautics*, 31(3), 514–528.
- Chaulya, S., & Prasad, G. (2016). Chapter 6 - formation of digital mine using the internet of things. *Sensing and monitoring technologies for mines and hazardous areas* (pp. 279–350).
- Chen, I.-F., & Lu, C.-J. (2017). Sales forecasting by combining clustering and machine-learning techniques for computer retailing. *Neural Computing and Applications*, 28(9), 2633–2647.
- Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.
- Chen, T., & He, T. (2015). Higgs boson discovery with boosted trees. *NIPS 2014 workshop on high-energy physics and machine learning*, 69–80.

- Cheung, Y.-m., & Jia, H. (2013). Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number. *Pattern Recognition*, 46(8), 2228–2238.
- Choe, Y. J., Shin, J., & Spencer, N. (2017). Probabilistic interpretations of recurrent neural networks. *Probabilistic Graphical Models*.
- Cohen, M. A., Agrawal, N., & Agrawal, V. (2006). Winning in the aftermarket. *Harvard business review*, 84(5), 129.
- De Amorim, R. C., & Mirkin, B. (2012). Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering. *Pattern Recognition*, 45(3), 1061–1075.
- de Amorim, R. C. (2016). A survey on feature weighting based k-means algorithms. *Journal of Classification*, 33(2), 210–242.
- de Gunst, M. (2013). *Statistical models*.
- Diaz-Uriarte, R., & De Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1), 3.
- Durugbo, C. M. (2020). After-sales services and aftermarket support: A systematic review, theory and future research directions. *International Journal of Production Research*, 58(6), 1857–1892.
- Erel, I., Stern, L. H., Tan, C., & Weisbach, M. S. (2018). *Selecting directors using machine learning* (tech. rep.). National Bureau of Economic Research.
- Fornell, C., Mithas, S., Morgeson III, F. V., & Krishnan, M. S. (2006). Customer satisfaction and stock prices: High returns, low risk. *Journal of marketing*, 70(1), 3–14.
- Frades, I., & Matthiesen, R. (2010). Overview on techniques in cluster analysis. *Bioinformatics methods in clinical research* (pp. 81–107). Springer.
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of statistics*, 1189–1232.
- Gregorutti, B., Michel, B., & Saint-Pierre, P. (2017). Correlation and variable importance in random forests. *Statistics and Computing*, 27(3), 659–678.
- Gruca, T. S., & Rego, L. L. (2005). Customer satisfaction, cash flow, and shareholder value. *Journal of marketing*, 69(3), 115–130.
- Gu, S., Kelly, B. T., & Xiu, D. (2019). Empirical asset pricing via machine learning. *Chicago Booth Research Paper*, (18-04), 2018–09.
- Hancer, E., Xue, B., & Zhang, M. (2020). A survey on feature selection approaches for clustering. *Artificial Intelligence Review*, 53(6), 4519–4545.
- Handelman, G. S., Kok, H. K., Chandra, R. V., Razavi, A. H., Huang, S., Brooks, M., Lee, M. J., & Asadi, H. (2019). Peering into the black box of artificial intelligence: Evaluation metrics of machine learning methods. *American Journal of Roentgenology*, 212(1), 38–43.
- Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1), 388–427.
- Hjort, K., Lantz, B., Ericsson, D., & Gattorna, J. (2013). Customer segmentation based on buying and returning behaviour. *International Journal of Physical Distribution & Logistics Management*.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.
- Hoogendoorn, M., & Funk, B. (2018). Machine learning for the quantified self. *On the art of learning from sensory data*.
- Hu, C., Zhou, Z., Zhang, J., & Si, X. (2015). A survey on life prediction of equipment. *Chinese Journal of Aeronautics*, 28(1), 25–33.

- Huang, J. Z., Ng, M. K., Rong, H., & Li, Z. (2005). Automated variable weighting in k-means type clustering. *IEEE transactions on pattern analysis and machine intelligence*, 27(5), 657–668.
- Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3), 283–304.
- Inza, I., Calvo, B., Armañanzas, R., Bengoetxea, E., Larranaga, P., & Lozano, J. A. (2010). Machine learning: An indispensable tool in bioinformatics. *Bioinformatics methods in clinical research* (pp. 25–48). Springer.
- Ji, J., Bai, T., Zhou, C., Ma, C., & Wang, Z. (2013). An improved k-prototypes clustering algorithm for mixed numeric and categorical data. *Neurocomputing*, 120, 590–596.
- Ji, S., Wang, X., Zhao, W., & Guo, D. (2019). An application of a three-stage xgboost-based model to sales forecasting of a cross-border e-commerce enterprise. *Mathematical Problems in Engineering*, 2019.
- Jiang, H., Deng, Y., Chen, H.-S., Tao, L., Sha, Q., Chen, J., Tsai, C.-J., & Zhang, S. (2004). Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC bioinformatics*, 5(1), 81.
- Killada, P. (2017). *Data analytics using regression models for health insurance market place data* (Doctoral dissertation). University of Toledo.
- Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500. *European Journal of Operational Research*, 259(2), 689–702.
- Madhuri, R., Murty, M. R., Murthy, J., Reddy, P. P., & Satapathy, S. C. (2014). Cluster analysis on different data sets using k-modes and k-prototype algorithms. *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol II*, 137–144.
- Maters, B. (2020). *Passenger punctuality computation using smart card and vehicle location data in a multimodal public transport network* (Master's thesis). Retrieved April 11, 2021, from <https://science.vu.nl/en/education/internship-office-for-mathematics-and-computer-science/master-project-ba/internship-papers-online/index.aspx>
- Mello, A. (2020). *Xgboost: Theory and practice*. Retrieved March 1, 2021, from <https://towardsdatascience.com/xgboost-theory-and-practice-fb8912930ad6>
- Missinglink.ai. (n.d.). *7 types of neural network activation functions: How to choose?* Retrieved November 16, 2020, from <https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/>
- Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7, 21.
- Omran, M. G., Engelbrecht, A. P., & Salman, A. (2007). An overview of clustering methods. *Intelligent Data Analysis*, 11(6), 583–605.
- Pavlyshenko, B. M. (2019). Machine-learning models for sales time series forecasting. *Data*, 4(1), 15.
- Ralambondrainy, H. (1995). A conceptual version of the k-means algorithm. *Pattern Recognition Letters*, 16(11), 1147–1157.
- Reiss, P. T., & Ogden, R. T. (2007). Functional principal component regression and functional partial least squares. *Journal of the American Statistical Association*, 102(479), 984–996.
- Saha, B., Goebel, K., & Christophersen, J. (2009). Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Transactions of the Institute of Measurement and Control*, 31(3-4), 293–308.

- Sandvik. (n.d.-a). *About us*. Retrieved September 14, 2020, from <https://www.home.sandvik/en/about-us/>
- Sandvik. (n.d.-b). *Metal-cutting and digital manufacturing solutions*. Retrieved September 14, 2020, from <https://www.home.sandvik/en/products-services/stainless-steels-and-special-alloys/>
- Sandvik. (n.d.-c). *Mining and construction equipment and tools*. Retrieved September 14, 2020, from <https://www.home.sandvik/en/products-services/mining-and-construction-equipment-and-tools/>
- Sandvik. (n.d.-d). *Stainless steels, special alloys and titanium*. Retrieved September 14, 2020, from <https://www.home.sandvik/en/products-services/metal-cutting-and-digital-manufacturing-solutions/>
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85–117.
- scikit-learn developers (BSD License). (2020). *Permutation feature importance*. Retrieved April 11, 2021, from https://scikit-learn.org/stable/modules/permutation_importance.html
- Shokouhyar, S., Shokoohyar, S., & Safari, S. (2020). Research on the influence of after-sales service quality factors on customer satisfaction. *Journal of Retailing and Consumer Services*, 56, 102139.
- Si, X.-S., Wang, W., Hu, C.-H., & Zhou, D.-H. (2011). Remaining useful life estimation—a review on the statistical data driven approaches. *European journal of operational research*, 213(1), 1–14.
- Sikorska, J., Hodkiewicz, M., & Ma, L. (2011). Prognostic modelling options for remaining useful life estimation by industry. *Mechanical systems and signal processing*, 25(5), 1803–1836.
- Svetnik, V., Liaw, A., Tong, C., & Wang, T. (2004). Application of breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules. *International Workshop on Multiple Classifier Systems*, 334–343.
- Swarndeeep Saket, J., & Pandya, D. S. (2016). An overview of partitioning algorithms in clustering techniques. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 5(6), 1943–1946.
- Tyralis, H., & Papacharalampous, G. (2017). Variable selection in time series forecasting using random forests. *Algorithms*, 10(4), 114.
- University, C. (2020). *Cambridge online dictionary*. Retrieved September 14, 2020, from <https://dictionary.cambridge.org/dictionary/english/aftermarket>
- van den Hoven, J. (2015). *Clustering with optimised weights for gower's metric* (Master's thesis). Retrieved February 3, 2021, from https://science.vu.nl/en/Images/stageverslag-hoven_tcm296-777817.pdf
- Van Ruitenbeek, R. (2019). *Vehicle damage detection using deep convolutional neural network* (Master's thesis). Retrieved April 11, 2021, from <https://beta.vu.nl/nl/onderwijs/project-en-stage/stagebureau-wiskunde-informatica/master-project-ba/stageverslagen-online/index.aspx>
- Venishetty, S. V. (2019). *Machine learning approach for forecasting the sales of truck components* (Master's thesis). Retrieved February 25, 2021, from <https://www.diva-portal.org/smash/record.jsf?pid=diva2%5C%3A1366957&dswid=8164>
- Weigand, A. (2019). Machine learning in empirical asset pricing. *Financial Markets and Portfolio Management*, 33(1), 93–104.
- Williams, P., & Naumann, E. (2011). Customer satisfaction and business performance: A firm-level analysis. *Journal of services marketing*.

- Wu, S. (2020). *3 best metrics to evaluate regression model?* Retrieved January 7, 2021, from <https://towardsdatascience.com/what-are-the-best-metrics-to-evaluate-your-regression-model-418ca481755b>
- xgboost developers. (2020). *Introduction to boosted trees*. Retrieved March 19, 2021, from <https://xgboost.readthedocs.io/en/latest/tutorials/model.html>
- Yuan, J. (2019). *Extreme gradient boosting*. Retrieved February 22, 2021, from <https://github.com/dmlc/xgboost>
- Yucesan, M., Gul, M., & Celik, E. (2018). Performance comparison between arimax, ann and arimax-ann hybridization in sales forecasting for furniture industry. *Drona industrija: Znanstveni časopis za pitanja drone tehnologije*, 69(4), 357–370.
- Zio, E., & Compare, M. (2013). Evaluating maintenance policies by quantitative modeling and analysis. *Reliability engineering & system safety*, 109, 53–65.

Appendix A

Customer Characteristics

<Confidential.>

Appendix B

Heatmaps

<Confidential.>

Appendix C

Activation Functions

The choice of the activation function should be thought through. The different functions will yield different result. Therefore, an overview of the most common used activation functions will be presented. In Table C.1, the plots and formulas of each of the discussed functions can be found.

Logistic Sigmoid

Pros:

- Smooth gradient
- The range of the function lies between 0 and 1, normalizing the input variables
- The function makes a clear distinction between everything above 2 and every thing below -2 , resulting in clear classification predictions.

Cons:

- Vanishing Gradient
- Outputs are not centered around zero
- Computationally expensive

Hyberbolic tangent (tanH)

Very similar to the logistic sigmoid function, however this function ranges from -1 to 1, making it more useful if negative output values are appropriate

Rectified Linear Unit (ReLU)

Pros:

- Computationally efficient, network quickly converges
- The function is non-linear

Cons:

- If the input data is negative or close to zero, the ReLU function will become 0, resulting in a model that cannot learn

Leaky ReLU

Pros:

- Similar to the ReLU function, but prevents the function to become zero too quickly

Cons:

- The addition makes the results inconsistent for negative input values

(Missinglink.ai, n.d.)

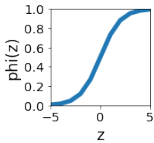
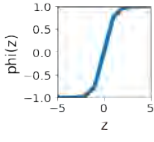
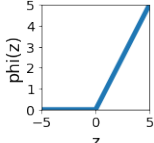
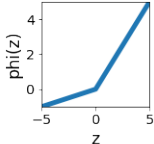
Name	Function	Derivative	Plot
Sigmoid	$\phi(z) = \sigma(z) = \frac{1}{1+e^{-z}}$	$\phi'(z) = \sigma(z)(1 - \sigma(z))$	
TanH	$\phi(z) = \text{TanH}(z) = \frac{e^{2z}-1}{e^{2z}+1}$	$\phi'(z) = 1 - \text{TanH}^2(z)$	
ReLU	$\phi(z) = \text{ReLU}(z) = \max(0, z)$	$\phi'(z) = \begin{cases} 1 & z > 0 \\ 0 & z \leq 0 \end{cases}$	
L-ReLU	$\phi(z) = \text{ReLU}(z, \alpha) = \begin{cases} \alpha z & z < 0 \\ z & z \geq 0 \end{cases}$	$\phi'(z, \alpha) = \begin{cases} -\alpha & z < 0 \\ 1 & z \geq 0 \end{cases}$	

TABLE C.1: Overview of the different activation functions and their characteristics.

Appendix D

Results Revenue and Sales Model

<Confidential.>

Appendix E

Time Series Model

<Confidential.>

Appendix F

Customer Segmentation Models

<Confidential.>

