# The Conditions of Reliable Demand Forecasting in a High-mix Low-volume Environment

den Boer, Tim J.

*Award date:*
2022



# Eindhoven University of Technology

Department of Industrial Engineering and Innovation Sciences
MSc. Operations Management and Logistics

## 1BM96

## Master Thesis Project

---

# The Conditions of Reliable Demand Forecasting in a High-mix Low-volume Environment

April 14, 2022

---

**Author:**
T. J. (Tim) den Boer

**Student Number:**
0957346

**Supervisors:**

| | |
|---|---|
| R.M. Dijkman | Eindhoven University of Technology |
| L. Bliek | Eindhoven University of Technology |
| B. Aysolmaz | Eindhoven University of Technology |
| A. van Hout | KMWE Eindhoven |

**Final Version**

# Abstract

KMWE operates in a high-mix low-volume industry, which makes holding inventory expensive and risky. Therefore, KMWE resorts to reactive manufacturing, thereby exposing them to supply chain disturbances, which impacts delivery performance. Demand forecasting can mitigate the effects by providing accurate predictions of future demand. In this research we investigated under what conditions KMWE can achieve a reliable demand forecast. It was found that, according to existing evaluation metrics, adding customer forecast data as an additional predictor can improve predictive performance. Likewise, it was found that aggregating demand data quarterly improved MAAPE, but undermined $R^2$ and SIME, compared to monthly aggregation. Moreover, it was found that, components produced better MAAPE scores than assemblies, DUV products produced better $R^2$ scores than EUV products, and service products better $R^2$ scores than non-service products. Also, average price and setup date were negatively correlated with performance, which was supported by studying the best performing products. However, according to an inventory simulation metric, which measures the practical consequences of a forecast, it is likely that demand forecasting will not deliver the desired results, as the improvement over simple baseline predictions proved to be small.

# Preface

This master thesis reports on the findings of a 7 months graduation project at KMWE as part of the Master of Science Operations Management and Logistics (OML) with the specialization "Data intensive industries" at the Technical University Eindhoven. During this project I had the opportunity to get a unique insight into the activities of the high-tech industry on which KMWE operates. It has been an educational and eventful experience, despite the restrictions put upon us by the pandemic.

I would like to thank my supervisors for their guidance and support. I would like to thank Remco Dijkman, my TU/e supervisor, for his input regarding the direction of the project and for his clear and concise instructions for improving my work. I would like to thank Arthur van Hout, my company supervisor, for guiding me to the correct people and providing me with useful insights. I would like to thank Laurens Bliek, my second TU/e supervisor, for providing me with fair and effective feedback. And finally, I would like to thank Quinten van Alphen, a KMWE employee, for having fruitful discussions with me that improved the quality of my work.

I hope you enjoy reading my work.

Tim den Boer

Eindhoven, April, 2022

# Management Summary

## Introduction

Demand forecasting is an important aspect of the inventory management process, especially when keeping stock is undesirable. Companies that operate in a high-mix low-volume industry often find that keeping stock is expensive and risky, since products can be costly and customer specific. As a result, these companies resort to reactive manufacturing, which comes with strict delivery agreements and unbalanced production load. Disturbances in the supply chain are a big threat, as they can impact the delivery performance. Demand forecasting can help overcome these challenges by providing accurate predictions of future demand, which will allow the company to proactively manufacture products. This research takes place at KMWE, a high-tech company, specialized in precision engineering and machining of assemblies and components. The goal of this research is to investigate under what conditions KMWE can generate a reliable demand forecast, which will help them to address these challenges.

## Problem Statement

KMWE carries a great responsibility towards it's customers when it comes to complying to expected due date. Customers mostly order their products according to the just-in-time principle. As a result, customers often shift their desired delivery date, which has an impact on workload distribution and delivery performance of KMWE. Customers try to support KMWE by providing forecast information about future orders. However, due to uncertainty in this information, KMWE chooses to only produce products based on confirmed orders. Disturbances, changes in the expected due data, or lack of production capacity can increase late deliveries. Ideally, KMWE would like to proactively manufacture products, based on reliable predictions. Prior internal research revealed the difficulties of producing reliable forecasts.

# Research and Results

In this research, we have built product level forecasting models for 114 products, that were used for three statistical analyses.

In the first analysis, we studied how using customer forecast data affected the predictive performance of demand forecasting models. Firstly, we compared the performance of forecasting models that were built without customer forecast data, to models that were built with customer forecast data. Secondly, we compared the performance of models with customer forecast, to the direct use of customer forecast data. We found that adding customer forecast data to a model improves the model, in terms of $R^2$ and MAAPE. We also found that, using the customer forecast data directly produces better or equal results compared to building a forecasting model. Surprisingly, for SIME, the size of the improvement, compared to MAAPE and R2 was much smaller under all situations. This led us to conclude that adding customer forecast data, or using it directly would improve the situation of KMWE only slightly, compared to a simple baseline. Even though the customer forecast was able to explain more of the variability in the demand data, we argued that for inventory management, the precision of the forecast is more important.

In the second analysis, we investigated how aggregating demand data from monthly to quarterly would impact the average performance of product forecasting models. Firstly, we found that the $R^2$ decreased significantly for quarterly aggregated models, which seemed to be caused by the major increase in the variance in performance for the 114 products. We argued that the increase in performance variance could be caused by a decreased test set length, which is more sensitive to extreme prediction errors. Additionally, we found that MAAPE decreased significantly for quarterly forecasts, which was explained by a decrease in intermittent time series data. Subsequently, we found that SIME is significantly lower for monthly forecasts. We argued that his was caused by the timing of orders during the time periods. We concluded that, in terms of SIME, a higher precision for a longer time period is not always better than a lower precision for a shorter period of time. Finally, we compared our findings to the baseline and found that the quarterly aggregated forecast produces better $R^2$ and MAAPE scores, yet for SIME, the difference in performance was small which made us question the practical use of demand forecasting.

In the third analysis, we studied the relation between product characteristics and the predictive performance of forecasting models. We found that assemblies produced higher MAAPE scores compared to components, which we explained by how the customer perceives the importance of a product. Next, we found that DUV, and related machine categories NXT and XT, are superior in terms of $R^2$, compared to EUV, and related machine categories NXE and EXE. We showed that EUV demand patterns are more often intermittent due to less mature products, which could be why $R^2$ favors

DUV. Additionally, we found that the underrepresented service products produced higher $R^2$ than non-service products, which we explained by higher sales volumes for service products. Furthermore, we found that the setup date was negatively correlated to $R^2$, which supported our findings for DUV and EUV. Surprisingly, the setup is positively correlated to the $R^2$, for customer forecast data, which could suggest that the customer can better anticipate it's behaviour for newer products. For the average price and number of complaints we found that the data was skewed, which raised questions about the validity of the results. When studying the best performing models, it was found that DUV is more dominant than EUV, which supports earlier findings. We concluded that, the correlations that were found during this analysis, can also be the result of unknown, unavailable and external factors.

# Conclusions and Recommendations

We concluded that adding customer forecast data improves the $R^2$ and MAAPE, yet direct use of customer forecast still produces better results in terms of $R^2$ and MAAPE, which makes building forecasting models hard to justify under these business conditions. Also, compared to a simple baseline prediction, the improvement of the SIME metric was small, which makes us question the practical usefulness of demand forecasting.

Next, we concluded that aggregating from monthly to quarterly, decreases $R^2$ due to a decrease of training data. It also decreases MAAPE, due to an increase in stability of the underlying demand. Furthermore, it increases SIME, which is caused by the nature of the practical assumptions, made by SIME. And overall, also in the quarterly aggregated situation, we found a small performance difference with a simple baseline, which leads us to conclude that, overall, building forecasting models, is lacking in the improvement of practical usefulness.

Finally, we concluded that components produce better forecasting models than assemblies, in terms of MAAPE, which was explained by a decreased demand stability for assemblies. Also, DUV, and related machine categories are better than EUV, and related machines, in terms of $R^2$, which is explained by less intermittent demand data. Additionally, the underrepresented service products produced higher performance, in terms of $R^2$. Subsequently, we found some weak evidence that the average price is negatively correlated with MAAPE, which was increased in strength by finding among the best performing products. Finally, we found that DUV was dominant among the best performing products, which supported earlier findings.

We recommended that, in most situations demand forecasting is lacking improvement in practical usefulness. Nonetheless, we suggested improving customer forecast quality, or finding additional predictor that could potentially improve the predictive

performance of forecasting models. Alternatively, we proposed that KMWE should consider keeping more inventory, or add more products to the collaborative SMI project. Furthermore, we suggested that improving efficiency in the manufacturing process could help improve how fast KMWE can react to changes, which in turn, could lead to a more stable production schedule.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Demand forecasting is an important aspect of the inventory management process. It is typically used for support in deciding how much stock to manufacture/order. It can also be used to anticipate customer behaviour with the goal of avoiding or minimizing stock. For some companies keeping stock of their products can be very undesirable. This is especially true for companies that operate in a high-mix/low-volume (HMLV) industry. High-mix indicates that the company sells a wide variety of products, low-volume suggests that the products are sold in small quantities. High-mix usually implies that products are custom build, specifically for a single customer. A symptom of this type of manufacturing is that keeping stock can be expensive and/or risky, because products can not be sold off to any customer. Usually, as a result, companies resort to reactive manufacturing, which means manufacturing only starts as a direct result of a customer order. Reactive manufacturing also poses challenges. If the company and the customer come to an agreement about the delivery date of the order, it is the company's responsibility to adhere to this delivery date. In this case, disturbances in the supply chain are the biggest threat to the company. Accurate demand forecasts can overcome these challenges, because it will allow the company to proactively manufacture products without the risk of high inventory.

This research takes place at KMWE, a high-tech company, specialized in precision engineering and machining of assemblies and components. The goal of this research project is to investigate under what conditions KMWE can achieve a reliable demand forecast. This chapter will contain a introduction to the setup of the research project. Firstly, a general company description is provided, which is followed by a description of the business problem. Next, the research objective and research questions are formulated. The chapter is finished by defining the research scope and an outline of the report.

## 1.1 Company Background

KMWE is a high-tech company based near the airport of Eindhoven in The Netherlands. Its core activities are designing, building and continuously improving high-tech components, modules and systems using precision engineering and machining. KMWE is part of a collaboration among the leading high-tech suppliers of the region, titled Brainport Industries. The head-office is located in a state-of-the-art building, called the Brainport Industries Campus (BIC), which was build as part of the collaboration and houses several of the partnering companies.

The markets on which they operate can roughly be grouped into five categories: Aerostructures, Aero Engines, Semicon, Healthtech and Industrial. Some of their most renowned customers include Airbus, Boeing, Dutch Air-Force, ASML and Rolls-Royce. Figure 1.1 shows a high-level organizational overview. KMWE has divided its internal business into two main divisions: Aerospace (AER) and Mechatronics (ME). These subdivisions operate as two separate companies. AER is mainly focused on the Aero Engines customers, whereas ME focuses on the four other customer segments.

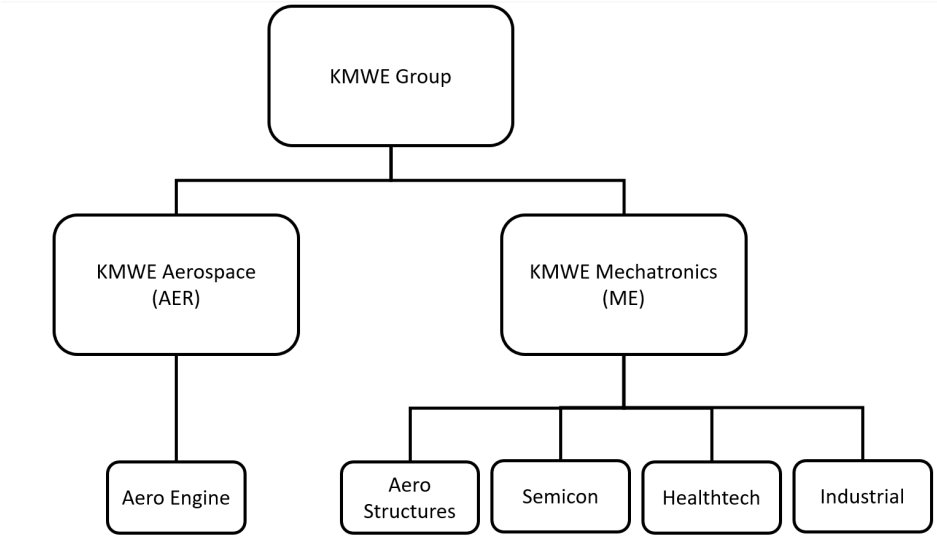



Figure 1.1: High-level organizational structure

KMWE is spread over four locations. The head-office is based in Eindhoven, as well as the Aerospace division. Another location is found near Eindhoven in Oirschot. The international location is located in Malaysia from which KMWE supplies its Asian customers. KMWE offers work to around 600 employees, who together generate an annual revenue of over 100 million euros.

The market on which KMWE operates can be characterized as a high-mix, high-

complexity, low-volume environment. This means that KMWE has a broad product portfolio. The majority of the products sold by KMWE are sold in low quantities. These market characteristics are known to be challenging for companies. Often production is make-to-order, which makes it difficult to manage demand and production capacity (Mahoney, 1997).

## 1.2 Problem Statement

A supplier in a High-Mix/Low-Volume (HMLV) supply-chain is exposed to unique challenges. Often customers are highly dependent on the performance of their upstream supplier, in this case KMWE. Therefore, KMWE carries a great responsibility when it comes to complying to expected due dates, leaving little room for error. On top of that, the components/modules that are produced/assembled are highly complex, custom-built and expensive, making it very economically undesirable to keep inventory. Moreover, customers order to the just-in-time (JIT) principle. This means that the customer aims to have the order delivered just in time for further processing, thereby minimizing the time the item is idle (in inventory). As a result, customers often try to shift the desired delivery date on an their order accordingly. Naturally this leads to an unstable workload distribution because work-orders have to be postponed or moved up, last minute, in the work-order schedule. Consequently, the risk of late delivery of all scheduled orders increases.

Some of KMWE's customers try to support KMWE with these challenges by providing forecast information about future orders. Even-though this forecast information should, in theory, help KMWE in their decision process, in practice they struggle to use the information effectively. Currently, the planning department uses the forecast as a rough indication of the expected order quantities. Due to the uncertainty in this forecast, the planning department only takes action based on real orders. When accepting new orders, KMWE always checks that the estimated lead-time falls within the requested delivery date. In theory, these orders should be delivered in time. However, when disturbances occur or the customers shifts the required delivery date or when production is simply overloaded, on-time delivery becomes more difficult to achieve due to lack of a time buffer.

Ideally, the planners would be able to initiate orders before a customer order has been received. This would give the planners more time respond to changing preferences of the customer. To accomplish this, a reliable forecast of the customer demand is required. Unfortunately, prior internal research (van de Velde, 2021; Vincenten, 2021) revealed that a majority of products are difficult to forecast. Nonetheless, KMWE wishes to investigate how to acquire a reliable forecast. Specifically, KMWE aims to understand under what conditions it can generate a reliable forecast.

## 1.3 Research Objective

The primary goal of this research is to determine under what conditions KMWE can achieve a reliable forecast. Part of the research is to determine which conditions will be considered. Previous internal research has revealed that, in most conditions, forecasting customer demand is generally difficult (Vincenten, 2021; van de Velde, 2021). This research aims to explain under what circumstances KMWE can overcome these difficulties. This will be done by constructing forecasting models under several conditions, and for multiple items, and examining which conditions and product characteristics impact the reliability of the forecasting models.

## 1.4 Research Questions

In order to have a structured approach to the research, research questions are formulated. The main research question will be divided into more concrete sub-questions. The main research question is formulated as follows:

*Under what conditions can KMWE achieve a reliable demand forecast?*

We can answer the main question by addressing multiple sub research questions. The first two questions are related because they both study the impact of customer forecast data. Customer forecast data is a source of information that is a potentially valuable predictor for demand, since it can be seen as a declaration of intent from the customer. Employees are currently barely using this information because they have doubts about the accuracy of the information. Therefore, it can be interesting to discover if using customer forecast data could help achieve a reliable demand forecast. For the first question we will see how adding customer forecast data, as an additional predictor, to models built on historical demand data, will improve the predictive performance of these models (RQ1a). Related to this is the second question (RQ1b), which studies how, using the customer forecast data directly performs, compared to models with customer forecast data. Note that, with using the customer forecast data directly, we mean that the information that is provided by the customer is directly used as a prediction for demand, and that no modelling techniques are applied. This leads to the following two related research questions:

*RQ-1a. Does adding customer forecast data to forecasting models improve the predictive performance of the models?*

*RQ-1b. How well do forecasting models, built on historical demand and customer*

*forecast data, perform compared to direct use of customer forecast data?*

Additionally, we will study how presenting the data to the model impacts its predictive performance. For this question, we will compare the situation, in which the demand data was presented on a monthly aggregation level, to a situation, in which the data was presented on a quarterly aggregation level. Vincenten (2021) showed that aggregating customer forecast data improves it's reliability. Therefore we aim to discover if the same applies for models that incorporate customer forecast data. Moreover, we want to study if these results transfer to a novel simulation metric that provides a more practical insight compared to traditional metrics. This leads to the following research question:

*RQ-2. Does aggregating demand data, from monthly to quarterly, improve the predictive performance of the demand forecasting models?*

Finally, we will investigate how certain product characteristics are related to the reliability of demand forecasting models. Specifically, for a collection of relevant product characteristics, we will see how they relate to the predictive performance scores of demand forecasting models. The research by van de Velde (2021) concluded that it is difficult to achieve a reliable demand forecast in general. Therefore, KMWE wants to discover if it is possible to distinguish predictive performance difference among products with different product characteristics. This leads to the following research question:

*RQ-3. How are certain product characteristics related to the predictive performance of demand forecasting models?*

## 1.5 Scope

The first important consideration, when scoping the project, is on what company process the research is focused. KMWE uses the MRP-II model as navigation for resource planning. In particular, this study will address the challenges in the demand management process. This means that the research will not consider production planning activities. The focus will solely be on the how KMWE can predict and control their demand. The results will have to be interpreted by experts in order to implement improvements in production scheduling processes.

Another important consideration is, which customer to focus on. ASML is one of KMWE's largest customers, representing a large chunk of their revenue. Even-though

the problem as described in section 1.2 is not unique for any specific customer, ASML is a suitable customer to use as research subject. This has three main reasons. Firstly, specific data needed for this research is relatively easily accessible. Secondly, because KMWE has been selling a large variety of items to ASML, which means that there is sufficient historic sales data. And thirdly, because the demand of many other customers, especially in the aerospace industry, has completely stagnated during the worldwide covid crisis.

## 1.6 Report Structure

The remainder of this report will start with a theoretical background of time series, demand forecasting evaluation metrics and variable importance metrics, in chapter 2. Next, we will present the methods that were used to build and evaluate forecasting models, and methods that were used to analyze the importance of conditional variables, in chapter 3. Subsequently, the results of the research are presented, in chapter 4, which is followed by the discussion of the results, in chapter 5. Finally, in chapter 6, we will provide answers to the research questions, provide recommendations to management of KMWE, present the limitations of the research, and suggest directions for future research.

# Chapter 2

# Theoretical Background

In this chapter the theoretical basis upon which the research is built, is provided. The chapter starts with an introduction to demand forecasting, which includes the properties of time series and an overview of some popular forecasting methods. Forecasting is central to our research questions. In order to investigate under what conditions the reliability of a forecast is optimized, we would need a forecasting model to analyze and compare the performance under certain conditions. For comparing the performance, we first need to define performance. Therefore, the subsequent section will cover evaluation metrics. These metrics are capable of quantifying performance of forecasting models. The reliability scores that result from these metrics will be used to make performance comparisons between models, and between the conditions that are inherent to the models. Finally, the last section will discuss the subject of variable importance metrics (VIMs). Conventionally, VIMs are used to quantify the importance of input variables in a model. However, as we will see in that section, particular VIMs (like statistical filters) are able to determine the importance of the variables on which the model was built, independent of the forecasting method. Additionally, we will see that statistical filters can be used to determine the importance of variables that were not directly used as input variables, which also allows them to be used to compare variables over multiple models. As we will see in the following chapter, the setup of this research requires this property in order to quantify the importance of variables.

## 2.1 Demand Forecasting

Demand forecasting revolves around estimating future demand as accurately as possible. According to the book of Montgomery et al. (2015) most demand forecasting techniques involve the use of time series data, which is a chronological sequence of

observations or events. An important feature of time series data is that, successive data point are usually not independent, and therefore the order of observations is critical. Thus, the goal of time series analysis is to find a suitable model that describes the statistical properties of the time series data. This section goes over both the properties of time series and the methods for time series analysis. Recall that the goal of our research is to build forecasting models that can be used to compare the conditions under which they perform optimally. This part of the research will help us to make an informed decision on which methods are useful for building the models.

### 2.1.1 Time Series Properties

Before the forecasting can start, it is useful to take a preliminary look at the data and find out as much as we can about its properties. Variation in time series data is often caused by a combination of multiple sources of variation. Chatfield (2000) use the following four sources of variation for decomposing a time series:

**Seasonality** This variation is caused by what time of year it is. Particular patterns in the data can be found at the same time of year for many years. In that case we can talk about seasonality. A classic example of a time series that is highly driven by seasonality is ice cream sales, which increase during summer and decrease during winter.

**Trend** This type of variation is observed as a steady upward growth or downward decline, for at least a couple of periods. An example of such behaviour is average price of a house in The Netherlands, which has been growing for many years now. A loose definition of trend is therefore "a long-term change in the mean".

**Other cycles** As we know, seasonality is cyclic behaviour on yearly intervals, which means we observe similar behaviour at the same time of year, each year. Other cyclic behaviour is similar but is not bound to yearly intervals. An example of this can be a weekly sales cycle that reoccurs during the weekends.

**Residual variation** As the name suggest, this type of variation is the left over variation that can not be explained by the other three. It is possible that this type is completely random in which case it is difficult to forecast.

A very useful tool for exposing features like seasonality, trend, other cycles, outliers, changes in structure and other abrupt changes, is the time plot. This is a visualization of some variable over time. The time plot can also be decomposed to showcase how each type of variation contributes to the observed time series. In Figure 2.1, we can see how passenger airline data is decomposed into three types

of variation. Sometimes the results of these visualizations can help to determine whether the data should be transformed. Some forecasting methods only work with stationary data. Data is defined as stationary if the distribution of the mean and the variance stays the same over time (Cryer, 1986). More concretely, this means that the data may not exhibit trend or change of variance over time. Visualization can help to discover whether your data is stationary. However, the contribution of such visualizations, to the understanding of the data, is highly dependent on the choice of scale and amount of available data. Non-stationary data can be transformed such that is becomes stationary by means of differencing and/or log transformations.

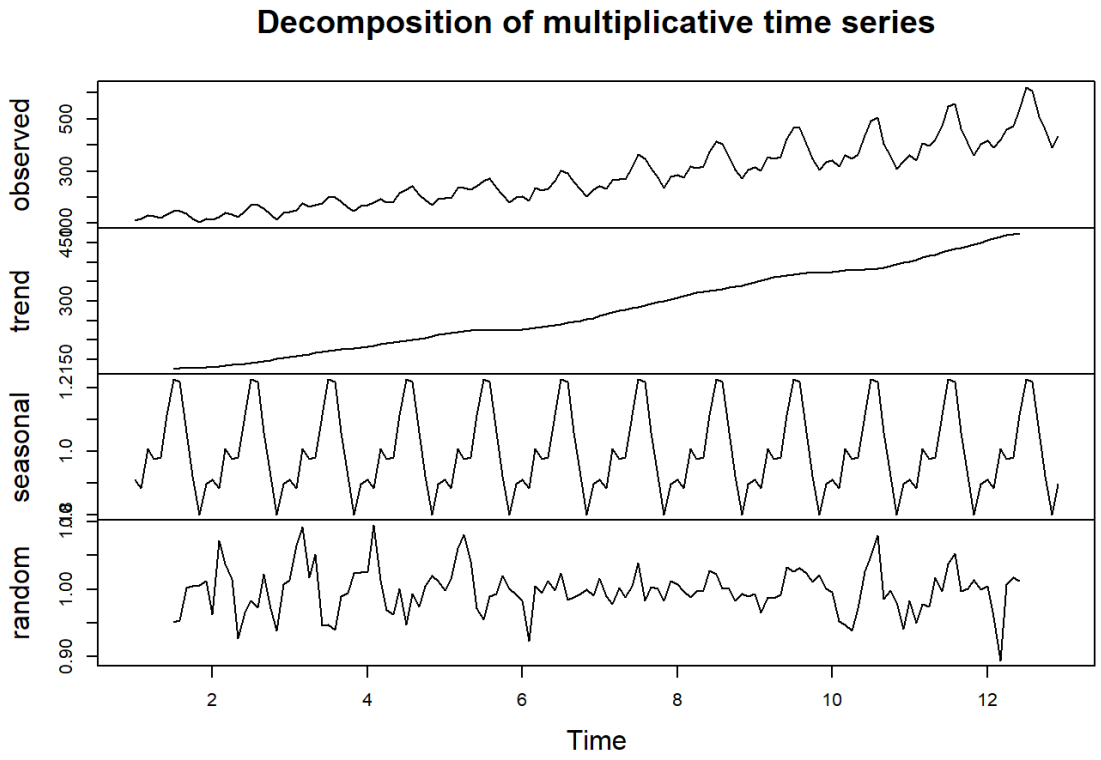



Figure 2.1: A decomposition of publicly available airline data

Another useful tool for discovering properties in your time series was proposed by Williams (1984) and later refined by Eaves (2002) and Syntetos et al. (2005). In particular, they were interested in classifying demand patterns by calculating two values for a time series: squared Coefficient of Variation ($CV^2$) and Average Demand Interval ($ADI$). The fist value will tell you something about the variation in the non-zero demand, and the second gives an indication of the average time between demands.

These values mean nothing until we define some cut-off values to classify the pattern. The study by Syntetos et al. (2005) has thoroughly analyzed this, which resulted in two widely accepted cut-off values: $CV^2 = 0.49$, $ADI = 1.32$. A visualization of how the demand is finally classified is given in Figure 2.2. Much research has been performed for determining the most suited forecasting methods for each of the classes. The difference in characteristics of each class are clear, where the two right-side figures have very irregular demand intervals with many zero demand periods, and where the two top-side figures show high variance in non-zero demand.

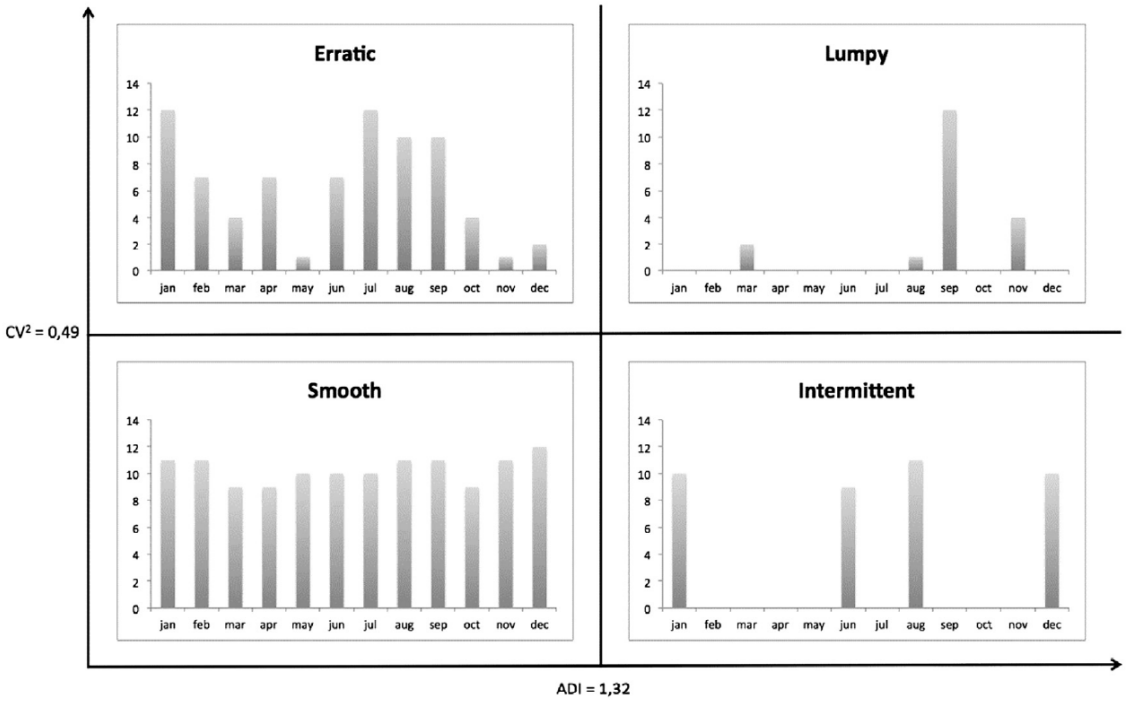



Figure 2.2: Classification of demand patterns, reprint of Costantino et al. (2018)

The exploration of the properties of time series in this way can be very useful for determining which forecasting method to use. Some methods can handle zero demand very well, while other methods can deal with high variability. Some methods can detect seasonality, while others are handle trends properly. Next, we will consider some well-known forecasting methods.

### 2.1.2 Forecasting Methods

As described in beginning of this chapter, forecasting is central to the research questions. A suitable forecasting model will have to be found in order to compare predic-

tion performance under certain conditions. This section will review some of the most popular time series forecasting techniques, which we will help us to choose a suitable model for analysis.

In the past 20 years, time series forecasting literature has shifted its interest from traditional statistical methods towards more complex deep-learning approaches (Cerqueira et al., 2019). Recent research by Makridakis et al. (2018) presents evidence that this shift is not completely justified, as many statistical methods can still outperform machine learning methods. Therefore, we believe it is useful to consider both options carefully in this subsection.

### Statistical Methods

The book of Chatfield (2000) distinguishing two types of statistical forecasting methods: univariate methods exclusively rely on past observations of the variable of interest and multivariate methods can incorporate one or more additional time series, called predictors, to make predictions about the dependent variable. In the latter case, the variation of one series can help to explain the variation in another time series.

Moving average (MA) is one of the most popular and widely used univariate technical analysis methods in the financial field (Zhu and Zhou, 2009). MA comes in many variations but the underlying purpose remains the same, i.e. to track the trend in the time series data. The most straightforward and basic form is the simple moving average (SMA). This form assumes the observations used as input are weighted equally regardless of their location in the time series (Equation 2.1).

$$\hat{y} = \frac{1}{n} \sum_{i=0}^{n-1} y_{t-i} \tag{2.1}$$

An upgraded version of SMA is the exponentially weighted moving average (EWMA) which applies weights to historical observations to make predictions. Specifically, a weighted average of past observations determines the prediction value, where the weights are exponentially decreasing as observations get older (Hyndman and Athanasopoulos, 2018) (see Equation 2.2).

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha) y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + ..., \tag{2.2}$$

SMA and EWMA are often used in practice when dealing with intermittent time series data (Syntetos and Boylan, 2005). But also in the financial field, these methods are found to be very suitable (Hansun and Kristanda, 2017).

MA should not be confused with the Moving Average process (MA(q)) which is, together with the autoregressive process (AR(p)), one of the underlying processes of

the AutoRegressive Moving Average (ARMA) and AutoRegressive Integrated Moving Average (ARIMA) method (Hyndman and Athanasopoulos, 2018). Instead of using past values of the variable, the Moving Average process (MA(q)) uses past errors in a regression-like model, where is a weighted linear sum of q random shocks. This process can be combined with the AR(p) process, where AR(p) is the weighted linear sum of the past p observations plus some noise, to create the ARMA(p,q) model.

An important assumption for ARMA model is that the data is stationary. In subsection 2.1.1 we explained that data can be transformed in order to make it stationary. The ARIMA(p,d,q) model solves this by transforming the time series through differencing individual data points (e.g. subtracting observation 1 from 2, 2, from 3, etc.). Therefore, an additional parameter d is added, which defines the order of differencing. This means that ARMA(p,q) is equivalent to ARIMA(p,0,q). Unfortunately, determining the parametric values p, d and q can be a tedious task, because it requires the use of external methods (e.g. correlograms, unit root tests) to ensure optimality and stationarity. In addition, estimating the parameters is computationally difficult and no efficient algorithm is known (White et al., 2015).

The ARMA process can be extended to handle multivariate data as well (Montgomery et al., 2015). Recall that, multivariate methods can incorporate one or more additional time series, called predictors, to make predictions about multiple output variables. The same limitations hold for the multivariate variants, where estimating the parameters is difficult and can lead to suboptimal results. Moreover, the number of parameters increases exponentially with the dimensionality of the model, increasing the computational difficulty even further (Hipel and McLeod, 1994).

Several studies have investigated the performance univariate and multivariate methods. Meese and Rogoff (1983) found that simpler methods like SMA or random walk can often give as good, or even better results as using more complex and time-costly univariate and multivariate methods.

### Machine Learning Methods

Statistical methods for time series forecasting have been around for many decades already, with some books dating back over 75 years (Maverick, 1945). Even though machine learning is introduced during the 1960s, it only became a serious contender for traditional statistical methods during the last two decades (Ahmed et al., 2010). Machine learning is a form of artificial intelligence that constructs algorithms that can improve automatically through learning structures within the data.

Traditionally, the literature distinguishes three types of learning systems: supervised, unsupervised and reinforcement -learning. For time series forecasting, supervised learning is the most obvious approach, because the goal is to get a point prediction for the dependent variable. With supervised learning, the computer is presented

with input and the corresponding desired output (target), which is called training data. The goal of the model is to learn relationships between in the input and target output. In order to determine how well the model is able to make predictions, the performance of the trained model is measured using unseen data, called test data.

The input variables for time series problems usually are lagged values of the dependent variable(s), and the target output is the next value in time for that same variable (in case of one-step ahead forecasting). In this case, the target variable is continuous, as it is part of a continuous time series. Problems with a continuous target variable are called regression problems. Problems with a categorical target variable are called classification problems.

The most popular supervised machine learning methods for time series forecasting can roughly be grouped into three categories: neural networks, decision tree methods, and ensemble methods. Each method within these categories can be powerful under the right circumstances.

A neural network is a network that is based on workings of human brains. Multiple layers of nodes (neurons) are linked via connections that have an associated weight and bias. The nodes in the input layer are connected through nodes in one or more hidden layers, to nodes in an output layer. The weights and biases of the connections are updated, by minimizing a loss function, as the networks is fed with data. Three types of neural networks can be distinguished: artificial neural networks (ANNs), recurrent neural networks (RNNs) and convolutional neural networks (CNNs). The most important difference between these types is how the algorithm extracts information from the input. This makes each type suitable for a subset of particular problems. ANNs are mostly used for tabular data. RNNs are able to exhibit temporal dynamic behavior because of their ability to store temporal information, making them very suitable for analyzing time series. CNNs are specialized in analyzing image data for image recognition.

Many studies have investigated the predictive power of neural networks for time series problems compared to traditional statistical methods. Sharda and Patil (1992) compared neural networks to ARIMA for multiple time series from a forecasting competition data set. Hill et al. (1996) studied the same competition data using neural networks and traditional methods. Alon et al. (2001) also compared various traditional methods with neural networks for retail sales data. The results from these studies are somewhat mixed, but overall the neural networks outperformed the traditional methods.

The power of neural network comes from its ability to detect non-linear relationships in the data, whereas traditional methods often assume linear dependency between the dependent and independent variables (Khashei and Bijari, 2011). In practice, it is not always easy to determine whether your problem is linear or non-linear. However, the need for sophisticated methods has been questioned, as evidenced

throughout various forecasting competitions (Hyndman, 2020). The research by Lim and Zohren (2021) identifies two keys reason for the under-performance of machine learning methods, like neural networks. Firstly, the flexibility of these methods makes them prone to over-fitting. Hence, simpler methods can do better with datasets with a small number of observations. Secondly, complicated machine learning methods seem to be sensitive to how the data is pre-processed.

Another form of machine learning that can be used for time series forecasting are decision tree methods. A decision tree is a decision support tool that models decisions and consequences in a tree-like manner. Initially, decision trees gathered traction through their usefulness for classification problems, however they can easily be converted for regression problems. By far, the most common strategy for constructing decision trees is by top-down induction (Rokach and Maimon, 2005). At the root node of the tree, all features of the training data are considered and the feature with the lowest loss in accuracy, according to a cost function, is selected for the first split. After this, all features, except the preceding feature, are considered for the next split, using the same cost function as before. This process is continued until splitting the node will no longer improve the cost value, or some predefined minimum value for node samples is reached.

Unlike neural networks, tree-based methods are notoriously more robust against over-fitting, because of a data comprehension technique called pruning. Pruning reduces the size of decision trees by removing non-essential parts of the tree, thereby reducing the complexity. Hence, pruned decision trees are better suited for small datasets (Wu et al., 2008). Moreover, researchers argue that decision trees are popular due to their simplicity and transparency. Also, just like neural networks, decision trees are able to identify non-linear relationships between data (Maimon and Rokach, 2014).

At some point, researchers started experimenting with combining machine learning approaches to create a better models that would make use of the advantages of both techniques. Such combined models are called ensemble models. According to Polikar (2006), ensemble methodology calls upon our second nature to weigh several opinions to make a final decision. Research has shown that combining output of multiple models, can reduce generalization error (Domingos, 1996; Quinlan et al., 1996; Bauer and Kohavi, 1999).

Possibly, the most well-known ensemble method is the random forest ensemble (Breiman, 2001). This method uses many individual decision trees which are created by randomizing the selected feature at each split. The resulting individual trees will likely be less accurate than an single tree that is constructed normally, taking into account accuracy improvement. But, the combination of many suboptimal trees, is often better than a single tree with exact splits. Moreover, random forests are able to handle many input features (Skurichina and Duin, 2002), by ignoring unimportant

features, and are fast to train compared to other machine learning methods, like neural networks (Maimon and Rokach, 2014).

Mussumeci and Coelho (2020) compared the predictive performance of a LSTM neural network and a random forest regression model, for forecasting the spread of seasonal dengue fever. It was found that the difference in forecasting accuracy was small, but in favor of the LSTM network. Yet, they also found that the computational costs of random forest (order of seconds) was significantly lower than than for the neural network (around 10 minutes).

A study by Kumar and Thenmozhi (2006) used several methods, among which were random forest regression and neural networks, to forecast stock index movement on the S&P financial market. Again, the difference in performance between neural networks and random forest was very small, yet in favor of random forest, this time.

In conclusion, neural networks, decision trees and ensemble methods are popular machine learning methods for time series forecasting. They offer several advantages over traditional statistical methods. The most powerful statistical methods like SMA, EWMA and ARIMA are univariate which means that they can only incorporate a single time series. Moreover, estimating the parameters for multi and uni-variate ARMA can be a complicated process, and still lead to suboptimal results. Machine learning algorithms are usually able to deal with non-linear data easily. The LSTM neural network and random forest regression are popular machine learning methods for time series forecasting. Unlike neural networks, tree-based methods are notoriously more robust against over-fitting, leading to better generalization errors. This also means that they are better at dealing with small datasets. Also, the simplicity, transparency and efficiency make random forests much more usable. In terms of performance, the difference between the two techniques is marginal. An overview of the discussed time series forecasting methods can be found in Table 2.1.

| **Method** | **Type** | **Data type** |
|---|---|---|
| SMA | Statistical | Univariate |
| EWMA | Statistical | Univariate |
| MA(q) | Statistical | Univariate |
| AR(p) | Statistical | Univariate |
| ARMA | Statistical | Univariate/Multivariate |
| ARIMA | Statistical | Univariate/Multivariate |
| Neural networks | Machine learning | Univariate/Multivariate |
| Decision trees | Machine learning | Univariate/Multivariate |
| Random forest ensemble | Machine learning | Univariate/Multivariate |

Table 2.1: Overview of time series forecasting methods

## 2.2 Evaluation Metrics

The goal of forecasting methods is to define a model with the most accurate representation of reality. But, how do we quantify accuracy? There are many ways of determining how well your model performs. Probably, the most common is way is to compare model predictions with realized values (Steyerberg et al., 2010). This is possible for both classification and regression problems, although they require other evaluation metrics. For the sake of this research we will focus on metrics for regression problems. Specifically, we will discuss scale independent metrics that allow for comparing models that are built on differently scaled data.

The explained variation ($R^2$), also called the coefficient of determination, is undoubtedly the most common performance measure for continuous outcomes. This metric quantifies how much of the variation in dependent variable is explained by the independent variable(s) (Draper and Smith, 1998). In Equation 2.3 we observe three unknown variables: $y_i$ is the prediction for instance $i$, $f_i$ is the model prediction for instance $i$, and $\bar{y}$ is the mean of the observed data.

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{2.3}$$

In the best case, the predictions exactly match the real values, in which case $R^2 = 1$. In the control case, when the predictions exactly match $\bar{y}$, will have $R^2 = 0$. And models which are worse than predicting the mean, will produce a negative $R^2$.

This metric has some properties which make it such a useful and popular tool. Firstly, it shows the explanatory power of your independent variables. As we have seen in subsection 2.1.1, variation in time series can be decomposed into various sources of variation. Essentially, $R^2$ tells you how much of this variation is captured by the model. Secondly, the metric is scale independent. This means that the scale of the problem is irrelevant, which makes it is possible to compare $R^2$ values over models that are built on other scales. And thirdly, in alignment with the first property, $R^2$ tells something about the predictive quality of the model on out-of-sample data. Unlike other metrics, which measure prediction accuracy of some test set, $R^2$ indicates how the model will perform on arbitrary unseen data (Chicco et al., 2021).

Although this metric is very good, it is not particularly suited for measuring prediction error. $R^2$ does not exactly quantify how far away the predictions are from the real values. For this, metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) have been introduced. However, the 'problem' with these metrics is that they can not be used to compare model which are built on different scales. For example, a model that predicts annual bank robberies will certainly have a lower MAE than a model that predicts annual shoe sales, regardless of how good both models are, simply due to the nature of the problem.

As a results, scale independent error metrics like Mean Absolute Percentage Error (MAPE) and Symmetric Mean Absolute Percentage Error (SMAPE) were created. These metrics normalize the predictions by dividing the errors by the actual observed data. However, this introduces a new problem. The testing data can not contain zero values, since this can lead to division by zero, which produces undefined values (Armstrong and Collopy, 1992).

A relatively recent study by Kim and Kim (2016), combats this problem by introducing a novel metric called Mean Arctangent Absolute Percentage Error (MAAPE) (Equation 2.4), which is developed by looking at MAPE from a different angle (pun intended). The MAAPE uses the inverse tangent function to transform the normal MAPE, which solves two things. Firstly, it maintains a more balanced penalty for small and large errors. And secondly, the bounded range of the arctangent function ensures that the undefined or infinite errors can be avoided as the actual values go to zero.

$$\text{MAAPE} = \frac{1}{N}\sum_{i=1}^{N}\text{AAPE}_i = \frac{1}{N}\sum_{i=1}^{N}\arctan(|\frac{y_i - f_i}{y_i}|) \tag{2.4}$$

The unbounded range of MAPE $[0, \infty]$ has been transformed to the bounded range $[0, \frac{\pi}{2}]$ for MAAPE. However, the authors suggest that there are some limitations to the metric. If very large forecasting errors are considered, and these errors are assumed to be legitimate variations that might have important business implications, MAAPE is not an appropriate metric. This is because very large errors are not penalized proportionally compared to slightly smaller errors. Furthermore, if the actual value is zero ($y_i = 0$), the corresponding AAPE value is always $\frac{\pi}{2}$. As a result, the MAAPE can be inaccurate for time series with many observed zero values.

Evidence shows that using existing accuracy metrics does not always lead to better inventory performance (Gardner, 1990; Syntetos and Boylan, 2005, 2006). Instead, the literature suggest using inventory simulation metrics to determine how well forecasting models perform. Kourentzes (2014) found that seemingly differently performing forecasting methods in terms of the standard accuracy metrics, did not show any substantial difference when inventory simulation was adopted. Similarly, Kourentzes (2013) found that according to conventional metrics, neural networks are inferior when it comes to forecasting intermittent time series, yet the opposite is found when considering inventory metrics. These findings would suggest it can be useful or even necessary to analyze the impact of forecasting models on the (simulated) workings of an inventory system. An important consideration is what inventory metric should be used. Ideally, we would use overage and underage costs of inventory because, in real life, these costs represent how the company is affected. However, as suggested by Kourentzes et al. (2020), these costs are often difficult to obtain and vary over

SKU's. Therefore the authors suggest using other metrics, which blends the overage and underage costs into a single value. The choice of this metric determines whether the simulation method is scale independent, because some metrics adjust for scale while others don't.

In conclusion, there are several scale independent methods for quantifying the accuracy of a forecasting model. Probably the most popular being the $R^2$ metric, which is able to show how much of the variation is explained by the independent variables. For estimating the prediction accuracy, several other techniques were introduced. MAPE, SMAPE and MAAPE are scale independent, but, the first two are not able to deal with zero values. Finally, simulating inventory and recording standard inventory control metrics is a viable way of testing a prediction model. In Table 2.2, an overview of the discussed evaluation metrics can be found.

| **Technique** | **Scale dependency** | **Formula** | **Value range** | **Aim to** |
|---|---|---|---|---|
| $R^2$ | Independent | $1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2}$ | [-$\infty$, 1] | Increase |
| MSE | Dependent | $\frac{1}{n}\sum_{i=1}^{n}(y_i - f_i)^2$ | [0, $\infty$] | Decrease |
| RMSE | Dependent | $\sqrt{\text{MSE}}$ | [0, $\infty$] | Decrease |
| MAE | Dependent | $\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - f_i\rvert$ | [0, $\infty$] | Decrease |
| MAPE | Independent | $\frac{100\%}{n}\sum_{i=1}^{n}\lvert\frac{y_i - f_i}{y_i}\rvert$ | [0, 100] | Decrease |
| SMAPE | Independent | $\frac{100\%}{n}\sum_{i=1}^{n}\frac{\lvert y_i - f_i\rvert}{(\lvert y_i\rvert + \lvert f_i\rvert)/2}$ | [0, 200] | Decrease |
| MAAPE | Independent | $\frac{1}{N}\sum_{i=1}^{N}\arctan(\lvert\frac{y_i - f_i}{y_i}\rvert)$ | [0, $\pi/2$] | Decrease |

Table 2.2: Overview of evaluation metrics

## 2.3 Variable Importance Metrics

The contribution of input variables to performance of a forecasting model can be quantified using variable importance metrics (VIMs). This section shows an overview of techniques that allow the user to extract information about the importance of variables in their prediction models.

(VIMs) (also: feature importance methods) attempt to improve the interpretability of prediction models. It does this by estimating relative importance scores for the input variables, which allows the user to discover on which variables the model relies the most to make its predictions. But, what makes a variable important? Variable importance refers to how much a given model "uses" that variable to make accurate predictions. The more a model relies on a variable to make predictions, the more important it is for the model.

The book by Guyon et al. (2008) distinguishes between three types of VIMs:

filter, wrapper and embedded methods. Filter methods investigate the statistical relationship of individual variables and the target variable. Filters are by definition model-agnostic, which means that they can be used independent of the adopted model. Wrapper methods make use of the performance of the model to estimate the importance of variables. Embedded methods perform variable scoring based on model parameter that result from the training process. Such methods are usually model-specific.

Molnar (2020) propose another characteristic on which VIMs can be grouped: global and local. Global methods describe how variables affect the prediction on average over all samples, while local methods aim at explaining individual predictions. We will exclude methods that only allow for local explanations, because our goal is to understand the general mechanisms of model, instead of how a single prediction made an impact.

In the upcoming subsections we will introduce various VIMs. First Gini impurity is discussed which is an interesting and widely used, model-specific metric for tree-based models. Its interest for our research comes from the fact that it is really effortless to extract variable importance from random forest models because it is implicit to the model structure itself and often a single command in most programming applications is needed. Next, we will discuss the popular model-agnostic methods MDA and PIMP, which can be applied to an arbitrary model. This can be useful in case the selected model does not have its own embedded VIM, or if you simply would like to compare multiple importance scores. Subsequently, a novel embedded approach, called VIANN, which is designed for extracting variable importance from artificial neural networks, is discussed. This is followed by the model-agnostic method RReliefF which is known to be robust against noise and feature interactions. Next, SHAP is introduced. This model-agnostic method is particularly useful if the user is interested in more than just a simple importance score. SHAP generates all sorts of variable impact related information (e.g. variable importance, variable dependence, interactions, clustering and summary plots) which is particularly useful for highly complex models like neural networks. Finally, we will discuss a number of statistical filters that can be used to test dependencies between variables. These methods allow for analysis of variables that were not necessarily part of the input variables of the model, by means of statistically testing relationships between arbitrary input variables and the output variable.

### 2.3.1 Gini Impurity

Some modelling techniques have embedded variable importance scores. For example, the coefficients which are estimated by a linear regression model can be interpreted directly as a crude type of variable importance, under the assumption that

the input is scaled prior to fitting the model. For tree-based models this called the Gini importance (GI) or Mean Decrease in Impurity (MDI). This variable importance metric (VIM) implicitly originates from the Random Forest (RF) classification method (Breiman, 2001). In tree-based models, each node splits the data from its parent node on the variable that gives the greatest improvement in Gini impurity (for classification), or variance reduction (for regression). MDI is defined as the sum of impurity/variance improvement over the nodes using the variable. A major advantage to this VIM is that no additional work is required to calculate the importance scores, as it is implicit to tree-based models, and only a single command is needed in most programming applications. Its obvious shortcoming is its inability to extend to non-tree-based models. Moreover, a study by Strobl et al. (2007) shows that it is possible for MDI to have a bias in favor of variables taking more categories if the variable is categorical. Or in other words, splits are more often sought on variables with a higher number of unique values. Later, we will see which technique is able to solve this bias.

Menze et al. (2009) compared the performance of three variable selection techniques on the improvement of predictive accuracy of three classification models. The models were applied on spectral data for medical diagnosis. The study considered three selection techniques: univariate selection using the p-values of channelwise Wilcoxon-tests, multivariate selection based on MDI, and multivariate selection on partial least squares (PLS) or principal components (PC) regression coefficients. These procedures were used as pre-processing technique, after which the classification task was taken care of either by a Random Forest, discriminant PC regression (D-PCR), or discriminant PLS (D-PLS). A base measurement was constructed by running the models on all variables (without variable selection).

It was concluded that the MDI metric had an overall superior performance over the other selection techniques, regardless of the adapted classification model. After inspection, the MDI metric displayed some bias, suggesting that correlated variables were assigned similar importance. However, their research remained unaffected by the categorical bias because they exclusively used continuous variables with similar ranges.

A more recent study by Dabou et al. (2021) successfully demonstrates the application of MDI in a time series setting, in which the dynamics of a power system were analyzed. It was found that MDI produced very similar results to other variable ranking techniques. In general, MDI is very useful for large data mining problem. As a study by Qi (2012) points out, MDI is a popular choice for biological data mining tasks (many variables), due to its computational speed. This study illustrates that, despite its categorical and correlation bias, in the right experimental setting, MDI can be a valuable metric.

A new version of MDI, introduced by Nembrini et al. (2018), removes the cate-

gorical bias, with similar computational efficiency. The procedure aims at finding the true decrease in impurity, by filtering the noise, which currently together add up to the impurity importance (importance = true impurity decrease + noise). The noise can be defined as the impurity reduction related solely to the structure of a variable. The authors propose a way to remove the noise by performing a random linear reordering $\pi$ on sample ID's, and allowing the model to select values from both original and reordered samples whenever splitting candidates are chosen at the nodes. This results in the actual impurity reduction (AIR), defined as impurity importance from original values minus impurity importance from reordered values.

It was found that AIR is almost as fast as MDI and much faster than permutation importance (discussed in subsection 2.3.2), and simultaneously unbiased with regard to category size. However, the authors acknowledge that the prediction accuracy of the RF might decrease, because the splitting procedure has been altered. Consequently, it is proposed to always run a separate RF for prediction purposes. It should also be noted that AIR does not solve the correlation bias by any means.

There are some studies that have used the AIR metric to determine the most important variables for their model (Wadoux et al., 2019; Messager et al., 2021; Xia et al., 2021). Unfortunately, these studies lack an evaluation/comparison of AIR, and simply implement the metric. Luckily, a study by Loecher (2020) performed a comparison of various variable importance metrics, including AIR. Specifically, the following metrics were analyzed: MDI, SHAP (discussed in subsection 2.3.5), AIR, and $\mathrm{PG}^{\alpha,\lambda}_{\mathrm{OOB}}$. The study consists of two experiments. In the first experiment, a binary $Y$ variable was predicted from a set of 5 predictor variables, which were all independent of $Y$. A reasonable VIM would assign zero importance to all predictors. In the second experiment, $Y$ was dependent solely on a one variable out of five. Also, the number of categories for each variable varied.

It was found that, just like the creators claimed, AIR is able to successfully alleviate the categorical bias, as well as filter out unrelated variables. But more importantly, AIR managed to identify the only real predictor for the binary response variable. Yet, it is worth mentioning that this study did not investigate the complications that correlated predictors can cause.

### 2.3.2 MDA & PIMP

A widely used technique, is the permutation importance (PI) method, proposed by Breiman (2001) for a Random Forest (RF). This method starts by training a baseline model and recording relevant accuracy metrics. Next, the values of a single variable in the test set are shuffled. However, instead of retraining a model like in the drop column technique, the baseline model is applied on the partially permuted test set. The resulting accuracy decrease is an estimation of the variable importance of the

permuted variable. That is why this technique is also called Mean Decrease Accuracy (MDA). Although MDA was designed for RF model, it can be applied to any sort of prediction model.

An advantage of MDA technique is that it can be applied universally to every ML algorithm. The authors do however point out that MDA is still a relatively time-consuming method. But, since computational power has grown significantly since 2010, this argument is possibly invalid now. An argument that has remained valid is the one made by several other studies: MDA is strongly sensitive to correlation between predictors, also called the correlation bias (Archer and Kimes, 2008; Strobl et al., 2008). The problem this bias presents is well described by Zien et al. (2009): *"A change of Xj may imply a change of some Xk (e.g. due to correlation), which may also impact s (output vector) and thereby augment or diminish the net effect."*

Although, theoretically, MDA is a model-agnostic VIM, in the literature it is mostly used in combination with RF models, even if the prediction model is not tree-based (Chae et al., 2016). These type of studies first apply a RF to select useful variables, after which they will build a separate prediction model. Studies that apply MDA in non-tree-based models are limited but available (Date and Kikuchi, 2018; Petneházi, 2019). Unfortunately many studies lack any form of evaluation of MDA.

This is not true for the paper by Date and Kikuchi (2018). This study trained Partial Least Squared (PLS), Support Vector Machine (SVM), Random Forest (RF), and Neural Network (NN) models to predict the geographical origin of yellowfin goby fish based on 106 muscle metabolites profile variables. Apart from the performance of the models, the authors were interested in determining importance scores for the variables. For this, MDA was applied to both the NN and SVM model. A significance test (i.e., a Welch's t test with Bonferroni correction) was performed for validation purposes. This test indicated that most of the variables that were identified as important variables were either significantly more abundant or scarce in the muscles of the gobies derived from the origin, demonstrating that MDA can successfully distinguish important variables. Moreover, MDA was found to be a versatile approach, since for NN and SVM, it resulted in almost identical results.

A variant of MDA, called PIMP, was proposed by Altmann et al. (2010). It works slightly different from the original MDA. Instead of permuting a variable, the response vector is permuted for estimating the random importance of a variable. This is done under the assumption that the random importance of a variable follows a distribution (Gaussian, lognormal or gamma). Next, the probability of the measured importance on the unpermuted response vector is assessed. This results in a p-value, that can be used for measuring the variable importance. The authors point out that this technique successfully alleviates the categorical bias (discussed in subsection 2.3.1), but not the correlation bias.

Let us consider a study by Degenhardt et al. (2019), that has applied MDA and

PIMP to estimate variable importance scores. MDA was used in by several variable selection techniques, and PIMP was used as a variable selection tool by itself. In total, the study compared six variable selection techniques: Boruta, r2VIM, RFE, Vita, Perm, and PIMP. All techniques were used to generate an optimal subset of variables that were adopted by a RF model for prediction based on high-dimensional ($>$ 10000 variables) omics data (biological data such as: genomics, transcriptomics, proteomics, or metabolomics).

These techniques, with the exception of PIMP, use MDA to determine the variable importance scores. Yet, each technique applies MDA differently. For example, Boruta (Kursa et al., 2010) makes a permuted copy of each predictor (shadow variable) and adds it to the data. Next, a model is trained and all predictors importance scores are computed using MDA. A statistical test is used to compare the original MDA score with the highest MDA score over all shadow variables. Variables with significantly larger or smaller importance values are labeled as important or unimportant, respectively.

Overall, the Boruta variable selection technique was found to be the most powerful approach, followed by Vita. PIMP got outperformed on almost all evaluation metrics (RMSE, stability, sensitivity), except the metric that kept track of false-positives. PIMP was able to consistently remove the variables that had no relation to the target variable. Therefore, the method should not be discarded if one's goal is to remove false-positives.

This study also illustrates the usefulness of the MDA score, as it can integrated in a custom algorithm to evaluate which variables should be used for optimizing the prediction accuracy. Therefore we can assume that MDA is an useful method for estimating variable importance scores.

### 2.3.3 VIANN

The paper by de Sá (2019) proposes a novel technique, called VIANN, for deriving variable importance scores from Artificial Neural Networks (ANN) models. The method is based on the underlying principle that, the more important a variable is, the more the weights will change during the training of the model. It works by discretely monitoring the variance of the weights that are connected to the input layer. After training, these variances are combined with the final weights to get the variable importance scores.

A comparison with other well-established techniques show very similar and highly correlated results. However, because the technique is rather new, and applications are scarce, a proper external validation is still necessary. Moreover, the author mentions that when the validation accuracy of the model is low, the variable scores can be misleading, which is not exclusively true for VIANN. On the other hand, VIANN also

shows great promise. According to the author, the technique can be easily extended to recurrent- and convolutional-NNs. Furthermore, VIANN makes it possible to measure the relevance at every node in the model and not only at the input layer, which might open doors for making neural networks more transparent.

One of the few studies that has implemented and evaluated VIANN is the one by Maepa et al. (2020). A SVM and an ANN were used to determine the probability of finding certain minerals in an area using geological layers that show proximity to mineralization. The trained absolute sizes of the coefficients of the SVM were used to identify the most important variables, and were compared to the VIANN importance scores of the ANN. In total 13 variables were considered. It was found that this resulted in similar variable rankings overall, yet, the best variable from VIANN was not even considered among the top 5 variables of the SVM model. So, in general, with the use of both techniques, the authors were confident about the importance of the variables that were selected by both VIANN and the SVM model.

### 2.3.4 RReliefF

A classical model-agnostic method used for evaluating attribute importance in classification problems is the Relief method. It was originally developed to for classification problem with two classes. But soon, many extensions, for a wider range of applications, appeared. In short, the Relief method works like this:

1. Select a random sample from training data
2. Find nearest samples from both classes ('hit' and 'miss')
3. For each predictor, a measure of difference in the predictor's values is calculated between the random data point and the hits and misses.

Essentially, these steps will determine how much a variable has to divert in order for the class to change. The idea is that a predictor that shows a separation between the classes should have hits nearby and misses far away.

This method has been extended by Robnik-Šikonja and Kononenko (1997) to be used for regression problems. The extension, named RReliefF, has a strong advantage over classical variable evaluation techniques because it can be used for settings where the variables are highly correlated. Another upside to the method is its insensitivity to noisy input data. In regression problems, nearest hits and misses can not be used in the same way. Instead, a metric is introduced that measures probability that the predicted values of two samples are different.

Koprinska et al. (2015) adapted and applied three variable importance metrics: Mutual Information, RReliefF, and Correlation-Based Filters to select a subset of

variables that were used to predict electricity load of energy systems. The goal was to reduce a set of 2016 lag variables (past data from target variable) to a subset of around 50 most important variables. It was found that all three techniques successfully selected a subset that achieved good predictive accuracy, with RReliefF being the best. However, while the other techniques delivered results within minutes, RReliefF required 48 hours to compute.

### 2.3.5 SHapley Additive exPlanations

Lundberg and Lee (2017) present a unified framework for interpreting predictions, SHAP (SHapley Additive exPlanations). SHAP is based on the game theoretically optimal Shapley values. Its goal is to estimate for instance x, the contribution of each variable to the prediction. It can be used for local explanations (explaining individual predictions), but can also be globalized over all observations. The Shapley values that are a result of a globalized analysis can be used to determine variable importance. variables with large absolute Shapley values are important. The main power of this method lies in its ability to generate all sorts of variable impact related information (e.g. variable importance, variable dependence, interactions, clustering and summary plots), making it an extremely useful approach for explaining, otherwise incomprehensible models. The main weakness of this method is computational inefficiency because it is required to calculate Shapley values for all instances. But, how does it work? A prediction can be explained by picturing that each variable value of a single instance is a player in game where the prediction is the payout. Shapely values can assign the payout objectively among the variables. Essentially, the Shapley value for each variable (payout) is basically trying to find the correct weight such that the sum of all Shapley values is the difference between the predictions and average value of the model. In other words, Shapley values correspond to the contribution of each variable towards pushing the prediction away from the expected value.

A study by Man and Chan (2021) used variable importance scores from SHAP, MDA and LIME to select variables for several classification and regression problems. LIME is mainly used for locally explaining black-box problems. In total, five datasets were analyzed by a random forest: two synthetic, one breast cancer, one house pricing and one financial trading dataset. The datasets consisted of both categorical and continuous variables. Apart from using standard accuracy metrics (F1, AUC, MSE, MAE, R2) to compare the methods, a novel instability index was used, which accounts for how randomness affects the methods. Let us consider some of their most interesting findings with regards to SHAP:

- SHAP was consistently stable over all datasets

- SHAP was found to be stable, even if many noisy variables were present. Whereas the other metrics were only stable when subsets of the most important variables were used.
- Prediction performance of the RF using SHAP, LIME and MDA for variable selection was very similar for all datasets.

### 2.3.6 Statistical Filters

Until now, we have mostly discussed wrapper and embedded techniques. As mentioned before, there exists a third technique: filtering. This method is independent of the adopted ML algorithm, because it only considers the statistical properties of the data. These statistical methods are often considered simple, effective and efficient. Yet, they also have some clear disadvantages. Let us discuss two of these statistical techniques, as presented by Guyon et al. (2008). Note that there are many methods for determining statistical relationships. We have chosen to only discuss two because it was found that these statistical methods showed similar characteristics when used for variable importance estimation.

#### Pearson correlation

Correlation coefficients are perhaps the simplest approach. The main goal of this approach is to determine the correlation between a variable and the target variable. The linear correlation coefficient of Pearson (Pearson, 1896) is very popular in statistics and is defined as:

$$\rho(X,Y) = \frac{E(XY) - E(X)E(Y)}{\sqrt{\sigma^2(X)\sigma^2(Y)}}$$

$\rho(X,Y)$ will be close to 1 if $X$ and $Y$ are linearly dependent, and close to zero if they are completely uncorrelated. This implies that Pearson correlation is mainly useful if there exists a linear monotonic relationship between two variables. However, a simple way of lifting this restriction is by non-linearly scaling the variables g (e.g., squaring, taking the square root, the log, the inverse, etc.) (Guyon and Elisseeff, 2003).

Wei et al. (2015) investigated the usefulness of the Pearson correlation coefficient for identifying important variables. Two tests were executed using a numerical model where a response variable Y (continuous) is dependent, both linearly and non-linearly, on six X variables generated by the standard normal distribution. Various correlation-based VIMs were compared using these tests. In the first test the input variables were

independent with each other. In the second test, some input variables were correlated on purpose. In both tests, the Pearson correlation was unsuccessful at identifying the importance of the input correctly, if the input was a non-linear predictor of Y. Furthermore, in test 2, the Pearson correlation was found to be very sensitive to correlated input. The authors point out that Pearson correlation is only useful in very specific cases, namely, when input is independent and the relation between X and Y is linear.

**Pearson's $\chi^2$-test**

Another popular method, used for classification problems is Pearson's $\chi^2$-test (Pearson, 1900). This method measures the strength of association between two variables by comparing the expected number of observations ($m_{ij}$), assuming $X$ and $Y$ are independent, with actual observations ($M_{ij}$) and is calculated as follows:

$$\chi^2 = \sum_{ij} \frac{(M_{ij} - m_{ij})^2}{m_{ij}}$$

The chi-square statistic can be used to find the corresponding p-value in a chi-squared distribution, which in turn is used to reject or confirm the null hypothesis about independence.

These statistical methods assume that relevance of variables grows with the correlation between variables and classes, and decreases with growing inter-correlation (between variables). A common criticism is that this can lead to under-evaluation of potentially valuable variables, as some combinations of lower ranked variables can potentially improve the prediction (Guyon and Elisseeff, 2003).

A comparative study by Pirooznia et al. (2008) investigated the performance of three variable selection methods including SVM (weights), chi-square and CFS. As we have seen in subsection 2.3.3, the weights of a SVM can be used directly as an indication of variable importance. As for CFS: this selection tool is based on Pearson correlations. These variables selection methods were applied together with several classification methods: SVM, RBF Neural Nets, MLP Neural Nets, Bayesian, Decision Tree and random forest. Eight microarray datasets containing gene information ($> 7000$ continuous variables) were used to classify for various cancers and diseases. It was found that, in general, the three methods performed similarly with the exception of one dataset. In the breast cancer set, chi-squared did not improve the classification accuracy for most classifiers, which implies that it did not remove the correct variables. The chi-squared metric seemed to work particularly well with the RF classifier. However, overall the SVM weights were deemed to be the best indicator of variable importance.

### Student's t-test

The t-test can determine if the mean of two groups are statistically different (Kim, 2015). For this reason, the test can be used to compare the continuous prediction accuracy of two groups which are constructed based on their value for an arbitrary binary variable. The test works by calculating confidence interval for the difference between groups and determining the possibility that the population mean is equal to zero ($H_0 : \mu_1 - \mu_2 = 0$), according to a predetermined significance level $\alpha$. In case the null-hypothesis is rejected, it can be assumed that the means of the two groups are significantly different. One form of the t-test is the independent or two-samples t-test. For the independent t-test it is required that the groups are unpaired, which means an instance can not be in both groups. Moreover, the groups should be approximately normally distributed. Another form of the t-test is the paired or dependent t-test, which assumes that the group are dependent. This test can be used to measure the impact on the mean of a variable after some intervention that could potentially impact the variable.

### Mann-Whitney U test

The Mann-Whitney U test is the non-parametric alternative to the independent t-test. This means that the test does not make any assumptions about the distribution of the underlying data. It tests the null hypothesis that the underlying distribution of the first sample is the same as the underlying distribution of the second sample. It can be used to test the difference between in location between distributions. Unlike many other statistical test, a higher U statistic indicates a lower difference between the groups.

### Wilcoxon signed-rank test

The dependent variant of the Mann-Whitney U test, is the non-parametric Wilcoxon signed-rank test. Moreover, it is the non-parametric variant of the two-sample t-test. It tests the null hypothesis that two related pairs come from the same distribution. More specifically, it test if the difference between the two related samples is centered around zero. Again, a higher W statistic indicates a lower difference between the paired samples.

In conclusion, VIMs are techniques that can be used to estimate the importance of input variables. Standard techniques like MDI, MDA, VIANN, RReliefF and SHAP mainly focus on analyzing model input variables by either manipulating input and observing the output, or by studying the internal structure of models to find how it relies on variables. On the contrary, statistical filter are much less restricted by

which variables were used to construct the model, and instead can be used to study the relation between arbitrary variables. This can be useful for instances where the model does not necessarily include all variables of which we want to estimate the importance. For example, this can be used in a situation where we want to forecast temperature for multiple countries and build a model for each country. Let's say we want to estimate the importance of the variable that determines on which hemisphere the country is. Adding this variable to each model will not do anything because it will be the same over the entire training data. Therefore, the standard VIMs can not be applied in this example. Instead, we can use statistical filters to compare the performance of the various models to determine how much impact the variable has on the performance. An overview of the discussed techniques can be found in Table 2.3.

| **Technique** | **Root paper** | **Type** | **Model** | **Key characteristics** |
|---|---|---|---|---|
| Mean Decrease Impurity (MDI) | Breiman (2001) | Embedded | Tree-based | Categorical bias<br>Computationally light |
| Actual Impurity Reduction (AIR) | Nembrini et al. (2018) | Embedded | Tree-based | Can influence prediction quality RF<br>Removes categorical bias |
| Mean Decrease Accuracy (MDA) | Breiman (2001) | Wrapper | Any | Moderately computationally heavy<br>Correlation bias |
| PIMP | Altmann et al. (2010) | Wrapper | Any | Alleviates categorical bias<br>Similar to MDA |
| VIANN | de Sá (2019) | Wrapper | Neural Networks | Relatively new: little external validation<br>Easy to implement |
| RReliefF | Robnik-Šikonja and Kononenko (1997) | Wrapper | Any | Tackles correlated variables<br>Insensitive to noise |
| SHAP | Lundberg and Lee (2017) | Wrapper | Any | Generates numerous variable impact related information<br>Computationally heavy |
| Pearson correlation | Pearson (1896) | Filter | Any | Simple & efficient |
| Pearson's $\chi^2$-test | Pearson (1900) | Filter | Any | Efficient<br>Correlation bias |
| Student's t-test | Student (1908) | Filter | Any | Parametric<br>Measure difference in mean |
| Mann-Whitney U test | Mann and Whitney (1947) | Filter | Any | Non-parametric<br>Measure difference in location between distributions |
| Wilcoxon signed-rank test | Wilcoxon (1945) | Filter | Any | Non-parametric<br>Tests if differences are symmetric about zero |

Table 2.3: Overview of VIM techniques and corresponding root papers

# Chapter 3

# Research Methodology

In this chapter we will be discussing the methodology that was used throughout the research. The chapter will be split in two parts. In the first part, an explanation about the demand forecasting model is provided. This model is basis of this research and will be used to answer the research questions. We will explain what data was used to build the forecasting model, how the model was build, and finally what evaluation metrics were used to determine its performance. In the second part, we will show how the forecasting model has been used to answer the research questions.

## 3.1 Demand Forecasting

Demand forecasting is the process in which historical demand data is used to construct a model that can produce predictions about future demand. As we have seen in chapter 2, there are techniques that can incorporate, not only historic demand, but also other time series data, called predictors. In this section, the process of building a forecasting model is described. Firstly, we will present the sources of data that were used, and how they were reduced. Secondly, we will show how the data was prepared, and how it was implemented in a model. And finally, a description of the model evaluation metrics is provided.

### 3.1.1 Data

Our model, in the most basic form, was build on two sources of data. The goal of this model is predict future sales. Usually, a good way of doing this, is looking back and studying the record of previous sales. Also, we can use additional time series data from another variable that is in some way related to the sales. In our case, the additional data is customer forecast data. Below you will find a description of the

data sources that were used to build the model.

**Historic sales:** A dataset containing the order data, for all products sold to ASML for the period 01-01-2017 until 08-09-2021. The relevant information that was used from this file include the following: order date, ship date, product name and quantity ordered. In total this dataset contains orders for 1567 unique products.

**Customer forecast:** A file that contains information about how much a customer expects to buy in the upcoming months, of a particular product. These forecasts are received several times per year and include forecast about for months between 01-01-2017 and 01-01-2022. In total, the file contains forecast information for 180 distinctive products.

Not all products from these datasets were used. Below, we will describe how and why certain products were removed from the dataset, and how the data from multiple products was combined.

**Revisions & Service Items** A product revision happens whenever a product is slightly modified. In reality, this can already happen when a insignificant component of the product is swapped for another. In the sales data, product revisions are separated by adding a revision number to the original product name. In consultation with KMWE experts, it was decided that revisions could be ignored, which meant that all revisions, of the same product, were aggregated under one product name. The same was done for service products. Service products are sold to the customer with the goal of servicing an existing machine, instead of installing it into a new machine. For administrative purposes, KMWE uses different product names, depending on the final purpose of the product. Hence, two identical products with different purposes have different product names. Since the products are identical, we elected to combine the data from non-service and service products. The combining of the revisions and service products, resulted in a richer dataset for some products. [1379 from 1567 products remaining]

**Completeness** An important condition for products to be used for analysis was the quantity and timing of orders. Some products were introduced after 01-01-2017, or phased out before 08-09-2021. With the knowledge of previous internal research that reliable forecasting is difficult to achieve, we decided to only include products that were sold throughout all periods in the dataset. After visual inspection, this could be achieved by only including products that were sold at least once in Q1 2017 and once in Q2 2021. This ensured that, the

products on which the models were build, all had equal time series lengths. A similar check was done for the customer forecast data. The products that did not have a monthly forecast over the period from 01-01-2017 until 08-09-2021 were removed from the data. [274 from 1567 products remaining]

**Customer Forecast Availability** Another requirement was the availability of customer forecast data. Unfortunately, ASML does not provide a forecast for every product. For the purpose of the analysis is was required that customer forecast data was available. By cross-referencing the sales data with the customer forecast data, we removed all products that were not present in the customer forecast data, from the sales data. [127 from 1567 products remaining]

**Inconsistent Delivery Date** Some orders contained information that was deemed inconsistent. These orders had delivery dates that were before the order date, which is not possible. Therefore these products, with all corresponding orders were removed. [114 from 1567 products remaining]

### 3.1.2 Data Preparation

The preceding reduction steps resulted in a final dataset that contained all relevant data for building the forecasting models. Yet, first the data had to be prepared and formatted properly to be used by a model. In chapter 2, we identified several techniques for time series forecasting. Traditional statistical methods have been around for many decades and are still considered very powerful. However, the most popular of these statistical techniques are univariate, which means that they study the statistical properties of a single time series to produce a model that is capable of make predictions. Considering our research goals, we want to incorporate a second time series as well, which forces us to consider other methods. Some popular univariate methods have been converted to handle multivariate data. Literature shows that these methods are, however, difficult to implement and computationally costly, because the number of parameters increases exponentially with the dimensionality of the model (Hipel and McLeod, 1994).

For these reasons, we chose to use machine learning to build a demand forecasting model Specifically, a supervised learning approach was adopted, which is the most suitable machine learning approach for time series forecasting (Bontempi et al., 2012). With supervised learning, a computer is presented with input $X$ and the corresponding desired output $y$ (target), which is called training data. The goal of the model is to learn relationships between the input and target output. In order to determine how well the model is able to make predictions, the performance of the trained model is measured using unseen data, called test data.

In the most complete situation, the model was trained and tested using the following input variables: moving average, lagged demand and customer forecast. Note that we explicitly mention that this was for the most complete situation. As we will see, in the first part of the analysis the input of the model was manipulated by removing the customer forecast. For the remaining part of the analysis, the most complete situation was used. Next, we will give a more formal definition of the input variables.

- Moving average $MA(n_{ma})$ is a single value calculated using the simple moving average (SMA) method. SMA simply takes the average of the past $n_{ma}$ observations of a time series, which in our case is the demand. The reason for adding the moving average as an input variable to the model is motivated by the findings of a paper by Ahmed et al. (2010). The authors argue that adding the moving average allows the forecasting model to focus on the global properties of the time series by smoothing out the noise. Hence, in the interest of improving the forecasting performance for all models, the moving average was added as input to the model.

- Lagged demand $D_{t-n_d}$ is the demand from the past $n_d$ periods. The model can use these variables to study how the history of the demand can lead to the current value of the demand. Let's consider an simple example where we have a time series which consistently alternates between demand is zero and demand is one: $[1, 0, 1, 0, 1, 0, ...]$. By studying the past values in the time series, the model would, if presented with enough data, be able to detect this pattern and make accurate predictions.

- Customer forecast $F_{t-n_f}$ is the forecast that the customer produced $n_f$ periods ago, for time $t$. In other words, at time $t-2$ the customer made the intention to buy $F_{t-2}$ products at time $t$. This variable tells something about the intent of the customer, and should therefore be a good predictor of the demand.

In Table 3.1 we can see an example of what the the input data for the model looked like. In this example, from left to right, we can see the moving average, calculated over two periods, the demand from one ($D_{t-1}$), two ($D_{t-2}$) and three ($D_{t-3}$) periods back, and the customer forecast from one ($F_{t-1}$) and two ($F_{t-2}$) periods back. The final column is the target variable $y$, which is the demand $D_t$. Remember that, for the first analysis, the input data looked slightly different because there we would not have included the customer forecast ($F_{t-1}$, $F_{t-2}$) in the first situation. Note that, in this example, for $n_{ma}$, $n_d$ and $n_f$, we chose 2, 3 and 2 respectively. However, in the actual modelling we did not fix these values, as we will see in subsection 3.1.4.

| | $X$ | | | | | | $y$ |
|---|---|---|---|---|---|---|---|
| $t$ | $MA(2)$ | $D_{t-1}$ | $D_{t-2}$ | $D_{t-3}$ | $F_{t-1}$ | $F_{t-2}$ | $D_t$ |
| 0 | 4 | 2 | 6 | 5 | 5 | 4 | 5 |
| 1 | 3.5 | 5 | 2 | 6 | 7 | 7 | 8 |
| 2 | 6.5 | 8 | 5 | 2 | 8 | 6 | 9 |
| 3 | 8.5 | 9 | 8 | 5 | 6 | 7 | 6 |

Table 3.1: Example data for one product

Let's assume the data from Table 3.1 represents the complete data for an arbitrary product. The goal now is to make a prediction model that is able to predict future demand. The model needs a training set to detect patterns in the data. And a test-set can be used to measure the accuracy of the model. In time series forecasting it is common practice to make a temporal train-test split. For our example this means that we could use 50% of the data to train (t=0, t=1) and 50% to test (t=2, t=3). However, this presents an inefficiency. If we were to make a prediction for t=3, we have not used all the available data. The model only uses t=0 and t=1 to train the model, while in reality the demand data for t=2 would also be available. Hence, we used an algorithm called the walk-forward validation (Brownlee, 2017), that continuously builds a new model with the most recent data. This means that we will still split the data in a train and test set, but now we constantly add the most recent observation to the train set. This ensures that for a new prediction, the model can base its decisions on all available data.

However, we still need to make a test and train split because, the model needs data to base its first predictions on. In order to avoid that the first few predictions are significantly worse than the final predictions, we have used a train-test split of 50%. After visual inspection of a subset of product time series data, we found that this split would be enough to observe the patterns in the data, and would also leave enough data to produce meaningful test scores.

### 3.1.3 Modelling

In chapter 2, we identified the most popular machine learning methods for time series forecasting. Among those are neural networks, tree-based methods and ensemble methods. Unlike neural networks, tree-based methods are notoriously more robust against over-fitting, leading to better generalization errors. This also means that tree-based methods generally are better at dealing with short time series than neural networks. Moreover, in terms of simplicity, transparency and efficiency, neural networks are considered inferior to most other methods. Remember that our goal is not to construct a forecasting model with the highest possible accuracy. Instead, we want

to use the model to make comparisons over various conditions, which does not require the model to be optimized in terms of accuracy. Because of these considerations, neural networks were discarded as candidate for our model. The advantages of both tree-based and ensemble methods pointed us in the direction of the random forest regression, which is an ensemble of individual decision trees. This methods combines the simplicity and interpretability of decision trees with the potential strength of using multiple outcomes and weighing them to get a final prediction. Research shows that individual trees in a random forest are less accurate than single optimized trees, yet the combination of trees in random forest produce superior results (Breiman, 2001). For these reasons, our model was constructed using random forest regression, which was built using the Scikit-Learn implementation in Python 3.8.

### 3.1.4 Hyperparameters

An important aspect of training a model is to determine the correct values of the hyperparameters. Random forest regression has many implicit model hyper-parameters that dictate how the algorithm behaves. For example, the number of observations drawn randomly for each tree, the number of variables drawn randomly for each split, the splitting rule, the minimum number of samples that a node must contain, and the number of trees. Probst et al. (2019) investigated the impact of parameters on the predictive performance of the model. It was found that, the random forest works reasonable well when using the universal default values, for most cases. Also, the authors found that very little guidance can be found throughout the literature about how the parameters should be tuned. They also conclude that the effect of tuning for random forest is much smaller than for other machine learning methods. Recall that the goal of this research is not to optimize predict accuracy, instead we aim to compare the accuracy over varying conditions. For these reasons, it was decided to use the default parameter values, which can be found in Appendix A.

Aside from the implicit parameters, some additional parameters were introduced by the input variables: $n_{ma}$ and $n_d$. These parameters were tuned using a straightforward approach. Because the range of these parameters was bounded by the length of the time series, and random forest regression is very efficient, it was possible to test many combinations of parameter values. We trained models on a random selection of ten items, using all possible combinations of parameters. No correlation between any of the parameters and any of the accuracy metrics was found, which implies the importance tuning the parameters is low. Hence, we chose to define the following default values: for monthly forecasts: $n_{ma} = 6$, $n_d = 6$, for quarterly forecasts: $n_{ma} = 2$, $n_d = 2$.

Another parameter that was introduced by the variables is $n_f$. This parameter requires no tuning. Random forests are known for their ability to deal with high-

dimensional data (Skurichina and Duin, 2002). This means they can effectively ignore uninformative variables. Therefore, we can just add the full range $(F_{t-1}, .., F_{t-12})$ to the model. This is possible because even for the first row of our input frame we would have the information about what the customer predicted 12 months ago. This was done for the demand lags $D_{t-n_d}$, as well, because the range over which the customer forecast was collected, was larger than the range of the sales data. Therefore, if we were to add variable $D_{t-24}$ to our model, we would loose the data from the preceding 23 periods. This can be illustrated more clearly by looking at the example input data from the previous section (Table 3.1). For the first row, it is only possible to have a value for $D_{t-3}$ if we have historic demand from before $t = 0$. Instead, if $t = 0$ is our first data point, we would not have the information, and the cell would be empty. Empty cells are not tolerated by a random forest regression, therefore the row would have to be removed. In other words, in case of the sales data, the more lag variables you add, the more data you lose.

### 3.1.5 Model Evaluation

In chapter 2, we identified various metrics for evaluating the performance of a forecasting model. Some metrics are sensitive to scale (like MAE, MSE, RMSE), while other are not (like SMAPE, MAPE, $R^2$ and MAAPE). Scale sensitive (also: scale dependent) metrics can not be used to compare models that are built on other data scales, because these metrics would almost always favor the models that are built on the smallest data scale. Because we will be comparing models that are built on different products, we need scale independent metrics. Also, because we are dealing with intermittent time series (many zero values), we can not use metrics like SMAPE or MAPE. These metrics will return undefined error scores when the actual value is zero. Therefore, we have chosen to use the following three scale independent metrics that are able to handle zero values: coefficient of determination ($R^2$), Mean Arctangent Absolute Percentage Error (MAAPE). For explanation of $R^2$ and MAAPE, we refer back to section 2.2. We also implemented a third evaluation method that was not scale independent, but is able to provide a more practical answer to how well a model performs. The inventory simulation metric requires some additional clarification.

We built a simulation model that simulates how an inventory system behaves if it would order new products based on the forecast of a model. The goal of this inventory simulation was to reenact what would happen if KMWE implements the model and copies its predictions. Depending on which form of time aggregation was used, the workings of this system was slightly different. If the forecasting model was build on monthly sales data, the model would produce monthly predictions. In this case, the amount of products that were predicted for a month, would arrive at the beginning of that month, after correcting for inventory or back orders. Orders would arrive on

daily basis throughout the month, which would slowly deplete the inventory, until new products would arrive at the beginning of next month. For quarterly forecasts, it worked slightly different. Instead of letting all products arrive at the start of the quarter, we chose to let 1/3 of the quarterly forecast (adjusted for open inventory or back orders) arrive at the start of each month belonging to that quarter. As discussed in the introduction, KMWE tries to avoid holding inventory. Ordering only once per quarter would inevitably lead to high stock levels, especially at the beginning of the quarter. Therefore, according to KMWE, they are more likely to spread the arrival of products, which is why this assumption was made.

The behaviour of the system was evaluated by monitoring the inventory level at the end of each day. For any point in time, the system can either have positive inventory or negative inventory, also called back-orders. In reality, both generate some sort of cost. Positive inventory has to be stored somewhere, which often means paying for storage space. Back-orders arise when an order comes in, but there is not inventory to fulfill the order. This can lead to unsatisfied customers. Cost of positive inventory is easier to quantify than back-orders (however still not easy), because the costs of storage is often known, whereas the cost of an unsatisfied customer is not. Determining these costs for KMWE would require a whole separate analysis. Therefore, we have elected to weigh positive inventory and back-orders equally. This seems like a fair assumption, as the goal has now become to minimize inventory and back orders volume, instead of inventory and back order costs. This resulted in simulation error SIME as presented in Equation 3.1, where $INV_i$ is positive inventory at time $i$, $BO_i$ is back-orders at time $i$, and $N$ is the amount of days over which was simulated.

$$\text{SIME} = \sum_{i=1}^{N} \frac{INV_i + BO_i}{N} \tag{3.1}$$

## 3.2 Analyses

In this research, we have performed three analyses that were used to answer the research questions. This section explains how the analyses were executed. The first analysis was focused on comparing the performance of demand forecasting models build on historic order data, to models that were build on both historic order data and customer forecast data (RQ1a). Moreover, we analyzed how the performance of the forecasting models was, compared to directly using customer forecast (RQ1b). In the second analysis, we compared the performance of a demand forecasting models that were build on monthly order data, to models that were build on quarterly order data (RQ2). In the third analysis we investigated whether we could distinguish

performance differences among products with other characteristics. Specifically, we studied how the performance compared among various products groups (RQ3).

### 3.2.1 Analysis 1: Customer Forecast Data

The goal of this analysis was to compare the performance of forecasting models, under two situations, thereby answering RQ1a. In the first situation we built models using historic demand and the moving average as input variables. In the second situation, we built models using historic demand, moving average and customer forecast data. More specifically, in the first situation, the model was trained and tested using two types of $X$ variables: lagged demand $D_{t-n_d}$ and moving average $MA(n_{ma})$. In total we built forecasting models for 114 products. The models were evaluated using the $R^2$, MAAPE and SIME. In the second situation, a third variable was added: lagged customer forecast $F_{t-n_{fc}}$. In this situation, models were built on exactly the same products as in the first situation. Also, the models were evaluated using the same evaluation metrics.

Both situations produced scores for each evaluation metric and for each product. This information was used to determine the importance of the customer forecast data. This is where we revisit our literature study about variable importance metrics. As we have seen, VIMs are techniques for quantifying the importance of a variable in a model. The standard VIMs, like MDI, MDA, RReliefF and SHAP, are not suitable for this type of analysis because they will only produce importance estimations for a single model, which in this case means a single product. This would not help us to determine whether adding forecast data in general is better. Instead, statistical filter are capable of globally testing the importance of variable.

When choosing a statistical test, an important consideration is what the statistical properties of your data are. Parametric tests make assumptions about the distribution of the underlying data and should therefore be applied with care. Non-parametric test do not make this assumption and are useful when one or more of the common statistical assumptions are violated. One common assumption is that the data should be distributed normally. In order to determine if we can use a parametric test, we have used the Shapiro-Wilk test, to examine if the data was normally distributed. As it turned out, many of the samples were not. Because the goal of this analysis is to make comparisons, we have chosen to use the same test for all samples, which is why were were restricted to non-parametric tests.

Furthermore, for this analysis, the samples were dependent, because the performance scores for a product ended up in both samples. Therefore, we have used the Wilcoxon signed-rank test to compare the samples. This test computes the difference between pairs of samples and tests the null hypothesis that the median difference is equal to zero. If the significance level is below $\alpha = 0.05$, we reject the null hypoth-

esis, and conclude that the samples come from another distribution. Additionally, we looked at the one-sided scores to determine which sample provided the highest values. This test produces W statistic scores that express the magnitude of the difference between the samples. These statistics can be used to compare the magnitude of the difference, as long as the number of paired observations are equal. Note that, the smaller the W statistic, the bigger the difference. It is unusual in this respect: normally, the bigger the statistic, the bigger the difference.

For RQ1b, a very similar approach was applied. In this case, instead of adding the customer forecast data to a model, the customer forecast predictions for each of the products were directly evaluated. This situation was compared to the setting in which we built the model using moving average, lagged demand and customer forecast. Because the setting is similar to that from RQ1a, we have again used the Wilcoxon signed-rank test to determine whether the evaluation metric means from customer forecast was significantly different from the evaluation metrics means for the model.

### 3.2.2 Analysis 2: Time Aggregation

The approach for this analysis was very similar to the approach of the first analysis. Recall that, the goal of RQ2 was to determine how aggregating the data over time would impact the reliability of a demand forecast. For this we simply studied two situations. In the first situation, forecasting models for 114 products were build using monthly forecast data. In the second situation, the same products were used to build models on quarterly aggregated demand data. Aggregation is the process of combining data. In this case, we combined data from three months into a quarter. Naturally, this shortens the time series data threefold, which means less training and test data. Yet, it could potentially stabilize the time series in such a way that the model produces better predictions anyway. In both situations the models were evaluated using $R^2$, MAAPE and SIME. Again, the situations were compared using the Wilcoxon signed-rank test.

### 3.2.3 Analysis 3: Product Characteristics

This final analysis will be used to answer RQ3. Recall that, the goal of this research question was to find out whether we could distinguish performance differences among products with different characteristics. For this, we gathered a collection of product characteristics on which we have differentiated how well a forecasting model, on item level, would perform in terms of prediction accuracy. This collection was accumulated by conducting open interviews with employees who are knowledgeable about the demand management processes. Specifically, 2 managers and 3 scheduling experts were

interviewed. In these sessions, the employees were asked what product characteristics could influence the ordering behaviour of the customer. Additionally, they were asked why, they thought, a particular characteristic would cause that behaviour at the customer. Finally, they were asked to evaluate the resulting product characteristics on three criteria:

**Criterion 1: Expected impact on demand behaviour** For our research, we are interested in finding which characteristics have the most impact on reliability of a demand forecast. Therefore, it is useful to have the employees make an initial assessment of the impact because that means we could leave out characteristics that are potentially irrelevant.

**Criterion 2: Data availability** We wanted the employees to asses what data we need to include the characteristic in the analysis. Data could be dispersed over the company or even only be available outside of the organization. This assessment helped us to determine if it would be feasible to include the characteristic in the analysis.

**Criterion 3: Data quality** The employees were asked to give their opinion on the quality of the data for a given characteristic. Based on this, a decision could be made about whether to include the data.

The evaluation of the characteristics resulted in a definitive collection of product characteristics that were subjected to the analysis. In the interest of modelling decisions and interpretability of the results, we need to inspect the distribution of the products among the various product characteristics. Therefore, we have produced some descriptive statistics over these product characteristics. For this analysis additional data about the product characteristics was used:

**Item parameters:** This file contains basic parameter information about the items that KMWE sells. The majority of product characteristics could be extracted from this file.

**Delivery performance:** For every order, KWME records the delivery performance. An order will receive a score based on the how early (-)/late (+) the order was shipped (see Table 3.2). This file contains order delivery performance information from January 2015 until August 2021. The data from this file was used to calculate an average delivery performance score per product.

| Range | Score |
|---|---|
| $\leq -10$ working days | 0% |
| $> -10$ and $\leq -5$ working days | 30% |
| $> -5$ and $\leq -3$ working days | 70% |
| $> -3$ and $\leq 1$ working days | 100% |
| $> 1$ and $\leq 3$ working days | 50% |
| $> 3$ working days | 0% |

Table 3.2: Delivery performance scores

**Customer quality complaints:** This file contains an overview of quality complaints made by the customer from January 2010 until August 2021. The information from this file was used to count how many times a complaint was made about a product, giving an indication of how the customer perceived the quality of a product.

The data from these three files was combined into one dataset that contained the information for all product characteristics for all items. This dataset was used for the remainder of this analysis. Again, forecasting models for 114 products were build. This time we only used the $R^2$ and MAAPE metrics for evaluating the performance of the models. The reason behind this is that because we are comparing among products instead of situational conditions, we need scale-independent measures. Unfortunately, SIME is scale-dependent, as it will determine the score based on the quantity of positive inventory or back orders. After the employment of the metrics on the models, we ended up with two scores per product. This was used to determine how certain product characteristics were related to the performance.

For this, a distinction between categorical and continuous characteristics was made. Dichotomous variables are categorical variables with only two possible values. Nominal variables are categorical variables with multiple values. For categorical variables we used one of two tests. Either both samples were normally distributed, which meant we could use an independent t-test, or they were not, which is when we used the Mann-Whitney U test. We tested normality using the Shapiro-Wilk test.

The independent t-test assumes that the two samples are independent. It works by calculating the confidence interval for the difference between groups and determining the possibility that the population mean is equal to zero ($H_0 : \mu_1 - \mu_2 = 0$), according to a predetermined significance level $\alpha = 0.05$. In case the null-hypothesis is rejected, it can be assumed that the means of the two groups are significantly different.

The Mann-Whitney U test is the counterpart of the Wilcoxon signed-rank test, as it assumes that the two samples are independent. It is also non-parametric, and tests the null hypothesis that distributions of two samples are the same. Note that, we can not use the U statistic to make comparisons about the magnitude of difference,

because sample sizes are different between pairs of categories.

For the continuous variables, we used the Pearson correlation. This method will determine how two variable are correlated, and with what significance level. This will help us to determine if a continuous product characteristic is somehow related to the reliability of forecasting models. For some correlations, we used visual presentations that showed how the correlations were manifested. In these visualizations we projected a best fit line, which minimizes the error between the line and the data points. It is important to realize that the Pearson correlation coefficient, does not represent the slope of the line of best fit. Therefore, if you get a Pearson correlation coefficient of +1 this does not mean that for every unit increase in one variable there is a unit increase in another. It simply means that there is no variation between the data points and the line of best fit. The goal of this line is simply to illustrate the relation between two variables.

Finally, in this analysis we made an effort of isolating well performing product models and study their related product characteristics. The goal of this was to uncover any particular pattern in the product characteristics and use this to support or dismiss earlier findings. For this, we selected the 5 best performing products according to $R^2$ and MAAPE, as well as for monthly and quarterly aggregated models. This would produce 20 best performing products, however to avoid over-representation of certain characteristics, it was chosen to remove duplicates and use the table notation to clarify which item occurred more than once.

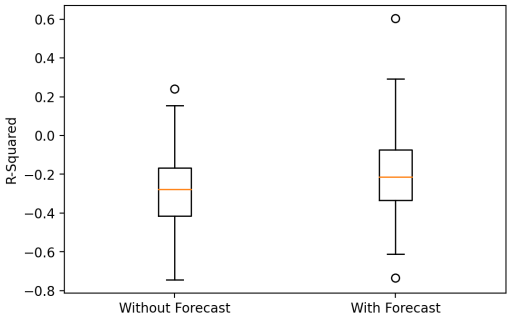
# Chapter 4

# Results

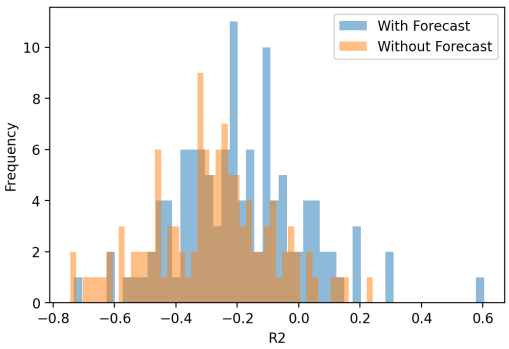
In this chapter, the results from three analyses are presented. In the first analysis we investigated how using customer forecast data impacts the predictive performance of forecasting models. Moreover, we explored the use of customer forecast data directly as a form of forecast method. In the second analysis, we investigated how aggregating data from monthly to quarterly could improve the performance of forecasting. For this, we used to forecasting model with customer forecast to make predictions and generate performance scores per product. Finally, in the third analysis, we studied how certain products characteristics relate to the predictive performance of forecasting models. In this analysis we studied a collection of categorical and continuous characteristics that were gathered with the help of KMWE employees. Subsequently, we looked at some of the best performing products and tried to relate their characteristics to earlier findings.

## 4.1 Analysis 1: Customer Forecast

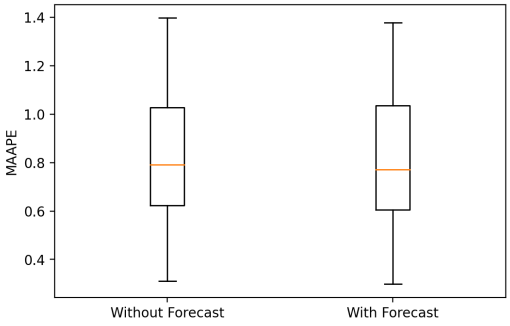
The goal of this analysis was to determine how using customer forecast data affected the predictive performance of demand forecasting models. In Figure 4.1, figures are presented that show the distribution of the performance of models for 114 products, that were built on monthly aggregated data. The figures on the left (a,c,e) are boxplots that show the distribution of model performance for three evaluation metrics. As we can see, for $R^2$, the average performance is slightly better for models that were built with customer forecast data. Yet, for the other two metrics, visually there is no difference. The same can be observed in the figures on the right side. For (b), the seemingly normally distributed performance, is shifted to the right slightly more for the situation with customer forecast data. In the other two right-side figures (d,f), we can not observe a clear difference in distributions.



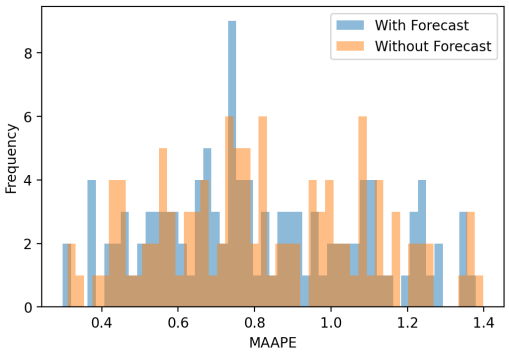

(a) $R^2$



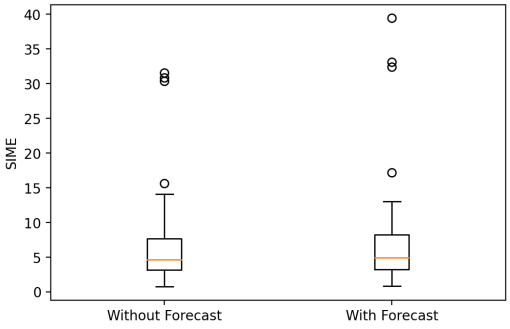

(b) $R^2$



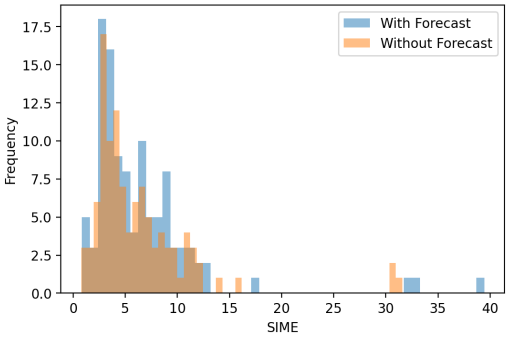

(c) MAAPE




(d) MAAPE




(e) SIME




(f) SIME

Figure 4.1: Distribution plots of model performance: monthly forecast model built with customer forecast vs without customer forecast.

A more exact answer was provided by a rigorous statistical approach, of which the results are presented in Table 4.1. This table shows the values of the test statistics and the significance level. Note that, the grey marked lines indicate the use of the paired t-test, whereas the white lines indicate the use of the Wilcoxon Sign-Ranked Test.

In the table we can observe that $R^2$ for models without customer forecast is significantly lower than for models with customer forecast. Moreover, we can also observe that MAAPE for models without customer forecast data is significantly higher than for models with customer forecast data, which we could not derive visually from the distribution plots.

Table 4.1: Results difference testing: model with customer forecast vs model without customer forecast (n=114) (built on monthly aggregated data).

| **Metric** | **Statistic** | **p-value** | **Outcome** |
|---|---|---|---|
| $R^2$ | 1427.0 | $1.68 \times 10^{-7}$ | Without $<$ With |
| MAAPE | 2054.0 | 0.0005 | Without $>$ With |
| SIME | 2157.0 | 0.0865 | Without $=$ With |

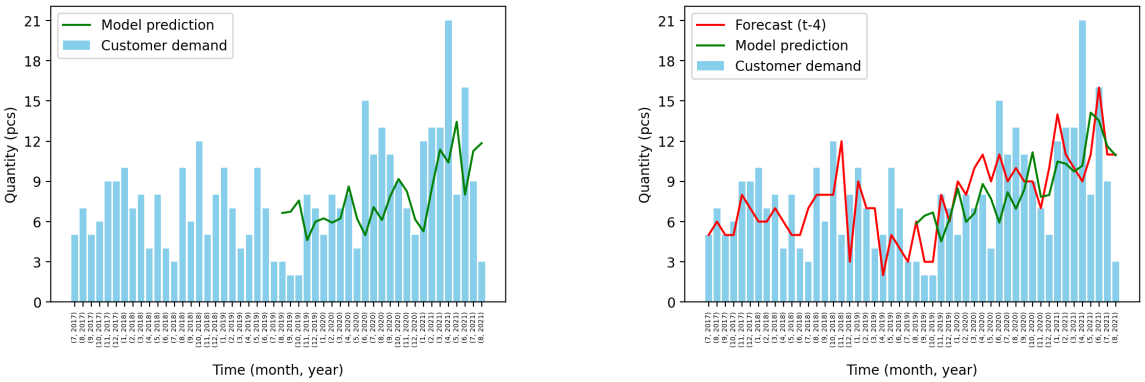
Instead of using the customer forecast as input variables for the random forest models, we also investigated the option of using the customer forecast directly as a prediction. In Table 4.2, we can see the results of comparing models that use customer forecast with directly implemented customer forecast data. For all three metrics, we can observe that the models with customer forecast data are significantly worse than direct use of customer forecast data.

Table 4.2: Paired t-test: model vs customer forecast (n=114).

| **Metric** | **Statistic** | **p-value** | **Outcome** |
|---|---|---|---|
| $R^2$ | 700.0 | $3.15 \times 10^{-13}$ | Model $<$ Customer forecast |
| MAAPE | 1025.0 | $1.91 \times 10^{-10}$ | Model $>$ Customer forecast |
| SIME | 2579.0 | 0.0483 | Model $>$ Customer forecast |

Until now, the results were focused on the global properties of samples of product forecasting models. Although we have to look at the results for all products to make a proper generalization, we can still study the product specific results to help explain some of the findings. We can look at some examples of product-level predictions that were made by the model and the customer forecast. In Figure 4.2, we can see customer demand data, on which we projected the model prediction that was purely built on

historical demand (a), and the model prediction that was built on historic demand and customer forecast (b). Note that the green line only starts halfway because the first half was used for training the model. Visually, it is hard to distinguish a difference between the two model predictions (green). However, we can observe that the customer forecast (red) better fits the real data than the model forecast (green) for this product. Previously, we saw that this is generally the case, as the performance of the direct use of customer forecast was significantly better than the models with customer forecast for all metrics.

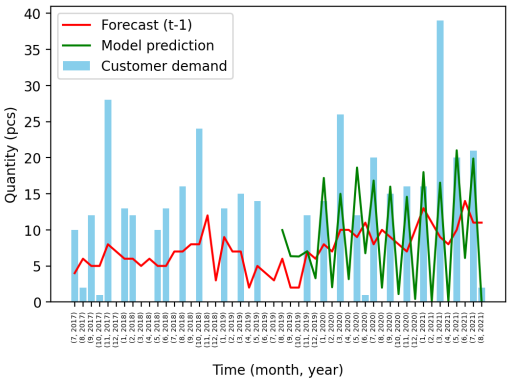



(a) Predictions without customer forecast

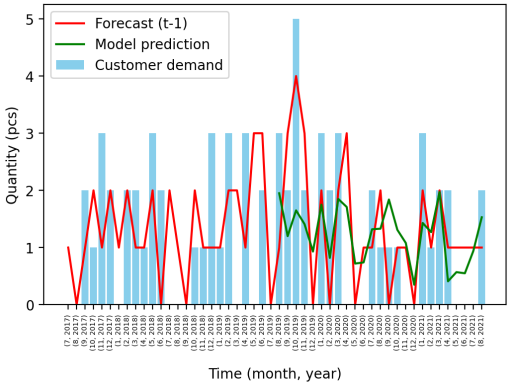
(b) Predictions with customer forecast

Figure 4.2: Monthly predictions made by model and for a single product.

The same can not be observed for all products. In some cases the customer forecast does not fit the real data, as can be seen in Figure 4.3a. In this figure we can clearly see that the model fits the data more precisely than the customer forecast. Instead of following the intermittent demand data, the customer forecast only makes positive predictions. We found this to be common for customer forecasts on intermittent demand data. In general, the customer forecast does not fluctuate very much, even if the underlying demand does. A counter example is given in Figure 4.3b, where we see that the customer forecast does follow a more intermittent pattern. However, this is one of few examples where the customer forecast made several zero predictions. However, for the majority of products, the customer forecast (red line) resembled the one from Figure 4.3a.




(a) Poor fit customer forecast




(b) Good fit customer forecast

Figure 4.3: Examples of predictions with good and bad fit of customer forecast data.

Although the next analysis will be focused on determining the effect of aggregating the times series, we were still interested in studying how well the results from this analysis would transfer to a more densely aggregated situation. As we can see in Table 4.3 and Table 4.4, the results have changed for the new situation, in which the model was built on quarterly aggregated data, instead of monthly aggregated data. In the new situation, we can see that adding customer forecast data to a model is now also significantly better in terms of SIME, whereas in the old situation it was only better for the $R^2$ and MAAPE. Furthermore, if we compare the model with the customer forecast to the direct use of customer forecast data (Table 4.4), in the new situation, we do not see a change in the outcome of the results, compared to the old situation, which means that also in the quarterly aggregated situation, the direct use of customer forecast data is superior to models with customer forecast.

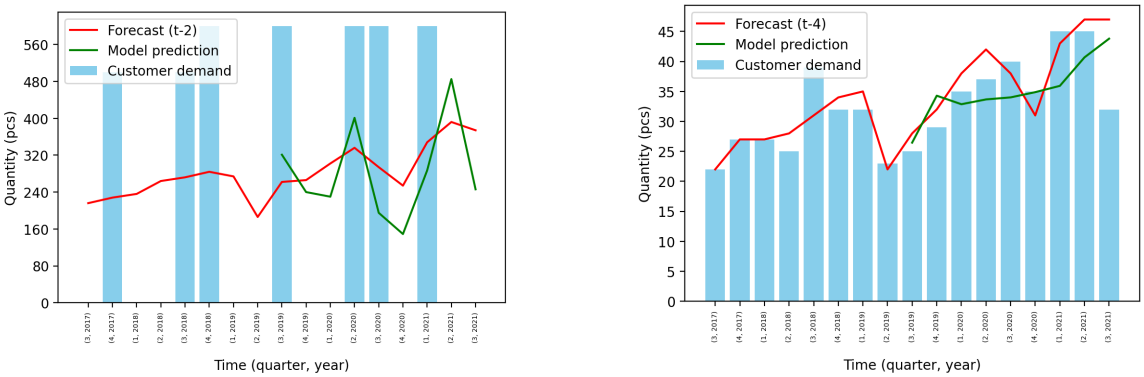
Table 4.3: Paired t-test: model with customer forecast vs model without customer forecast (n=114)(built on quarterly aggregated data).

| **Metric** | **Statistic** | **p-value** | **Outcome** |
|---|---|---|---|
| $R^2$ | 1163.0 | $2.25 \times 10^{-9}$ | Without $<$ With |
| MAAPE | 1784.0 | $2.41 \times 10^{-5}$ | Without $>$ With |
| SIME | 2197.5 | 0.0074 | Without $>$ With |

Table 4.4: Paired t-test: model vs customer forecast (n=114)(built on quarterly aggregated data).

| Metric | Statistic | p-value | Outcome |
|---|---|---|---|
| $R^2$ | 1281.0 | $1.65 \times 10^{-9}$ | Model < Customer forecast |
| MAAPE | 1913.0 | 0.0001 | Model > Customer forecast |
| SIME | 2507.0 | 0.0409 | Model > Customer forecast |

Under the new situation we can also reevaluate some examples of product predictions made on quarterly aggregated data. Just like before, these figures can help us to understand the results. In Figure 4.4 we can still see that the customer forecast struggles with intermittent demand, because it rarely predicts zero demand (a), and so does the model prediction. Yet, for smooth demand series the customer forecast can, sometimes, fit the data accurately (b). These two examples are quite accurate representations how the demand, model prediction and customer forecast look for all 114 products, where Figure 4.4a represents the intermittent data and Figure 4.4b represents the smooth demand.



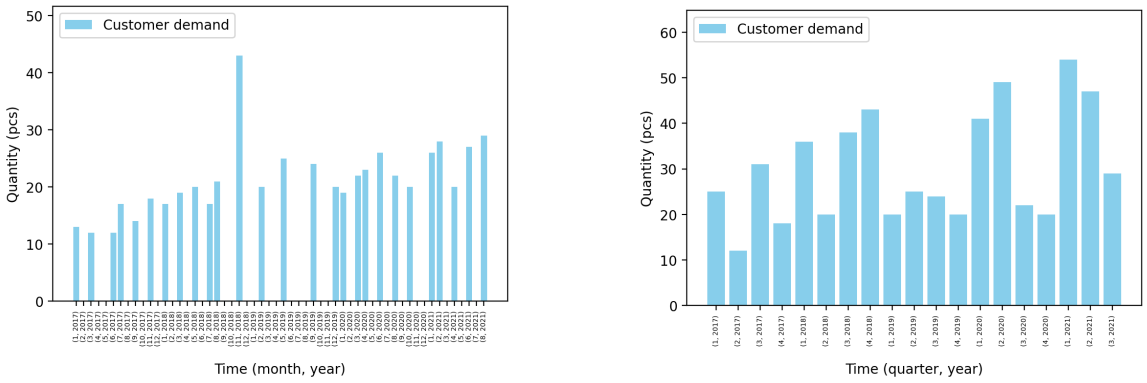

(a) Poor fit

(b) Good fit

Figure 4.4: Customer forecast and model predictions for quarterly aggregated demand data.

## 4.2 Analysis 2: Time Aggregation

The goal of this analysis was to determine how aggregating time series data influences the performance of the forecasting models. Specifically, we have looked at how models, built on monthly data, performed compared to models, built on quarterly data. Note that, for the predictions, we have used the model with customer forecast. In Figure 4.5, the aggregation for a single product is visualized. Straightaway we can observe two things that, theoretically, can influence the performance of a forecasting model. Firstly, aggregating reduces, or in this case removes, zero demand periods. In this case, it is converted from an intermittent, to a smooth time series, which could improve the predictive performance. However, the second observation we can make, is that the time series has becomes significantly shorter (from 56 to 19 data points), which we know is not beneficial for the predictive performance of a model. The question is which of these two properties is more influential.

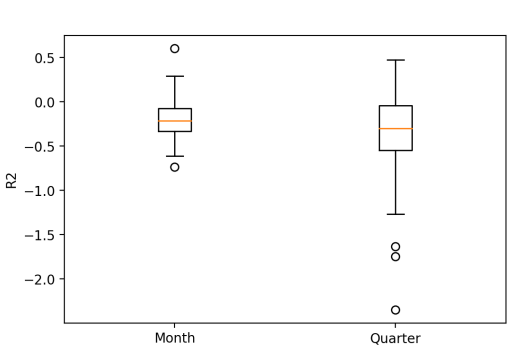



(a) Monthly aggregated data

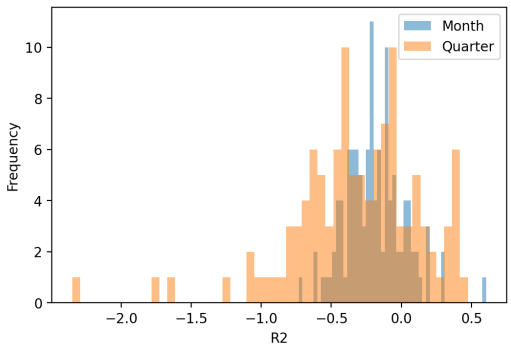
(b) Quarterly aggregated data

Figure 4.5: Time series of a product aggregated monthly and quarterly.

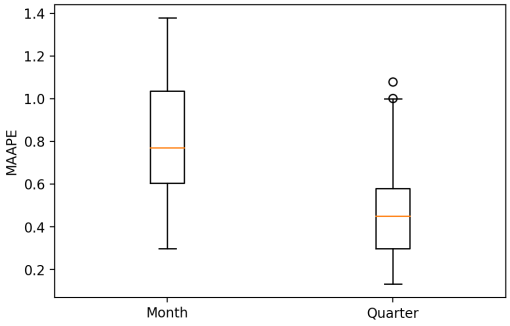
Just like in the previous analysis, we have visualized the distribution of the performance of 114 product level forecast models under two situations. In Figure 4.6, we can see the distribution of performance for three metrics, under the situation with monthly forecasts, and with quarterly forecasts. For the $R^2$, we can clearly see that the variance, with quarterly forecasts, is much higher. Yet, the mean is not visibly different. For MAAPE, we observe that the quarterly forecasts, generally, produce smaller errors. For SIME, visually, it is harder to distinguish a difference. However, we can see that the amount of outliers is smaller in case of monthly forecasts.



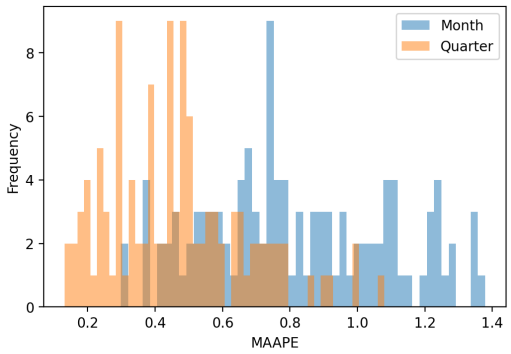

(a) $R^2$



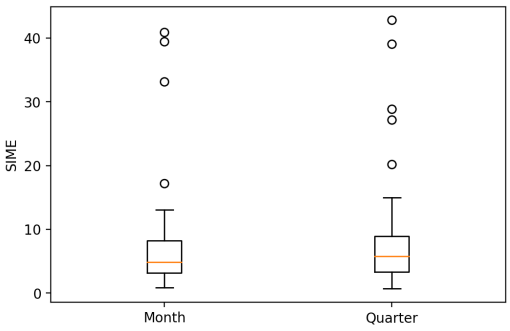

(b) $R^2$



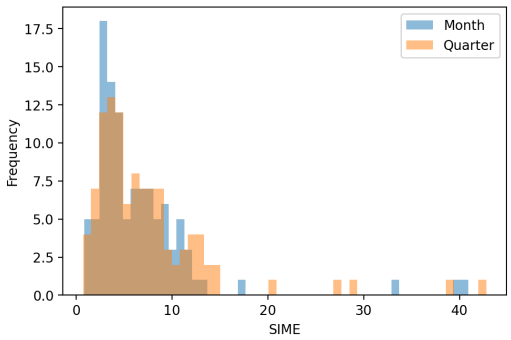

(c) MAAPE




(d) MAAPE




(e) SIME




(f) SIME

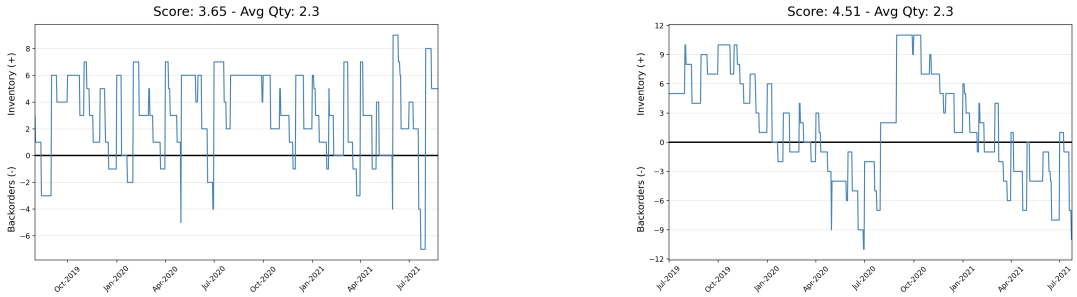
Figure 4.6: Performance distributions for monthly and quarterly forecasts.

Again, a more rigorous method was used to test whether the difference between the two situations was significant. In Table 4.5, the results from statistical difference testing are presented. We can see that, in terms of MAAPE, the quarterly forecast is significantly better than the monthly forecast. The opposite is found for SIME and $R^2$, for which the monthly forecast is significantly better than the quarterly forecast.

Table 4.5: Statistical difference testing: monthly forecast vs. quarterly forecast

| **Metric** | **Statistic** | **p-value** | **Outcome** |
|---|---|---|---|
| $R^2$ | 2290.0 | 0.0052 | Month > Quarter |
| MAAPE | 0.0 | $1.92 \times 10^{-20}$ | Month > Quarter |
| SIME | 1531.0 | $7.89 \times 10^{-7}$ | Month < Quarter |

In the previous analysis we argued that looking at individual product predictions can help to explain the results. For this analysis we can take a closer look at the SIME metric, as it conflicts with the other metrics. In Figure 4.7, we have presented the output of inventory simulations of the same product, for monthly and quarterly forecasts. We can observe that, the inventory appears more stable for monthly forecast, and also stays closer to zero. This is reflected in the SIME score it receives, which is presented above the figure. Again, these figures are representative of the simulation output of most other products, where we also observed a more stable inventory for monthly forecasts.




(a) Monthly forecast

(b) Quarterly forecast

Figure 4.7: Inventory simulation output for same product: monthly forecast vs quarterly forecast.

## 4.3 Analysis 3: Product Characteristics

The first deliverable of this analysis is a collection of product characteristics that could potentially affect the predictive performance of a forecasting model. Recall that this collection was gathered by interviewing employees and asking them to evaluate the characteristic on various criteria, which results in the following final collection:

**Type:** KMWE categorizes its products as either a component or an assembly. Components are usually simpler, smaller and cheaper and often are machined out of one material. Assemblies are more complex products, some of which are assembled in clean rooms.

**Technology:** The products that are sold to ASML all end up being used for one of their machines. The technology within these machines can be grouped into Extreme Ultraviolet (EUV) or Deep Ultraviolet (DUV).

**SMI project:** Some products that are sold to ASML, are part of special project, in which the inventory is managed by KMWE but stored and (partially) paid for by ASML. The goal of this project is to take away the cost and risk at KMWE, while maintaining a good delivery performance.

**Service:** Sometimes products are sold to the customer with the goal of servicing an existing machine, instead of installing it into a new machine. If a product is a service product, it means that is has been used at least once to service a machine. It does not mean the product is exclusively used for servicing.

**Target machine:** The products that are sold to ASML all end up being used for one of their machines. These machines can be grouped into four types: XT, NXT, EXE and NXE. Note that products can be placed into multiple machines, which essentially means products can belong to more than one machine type. XT and NXT machines provide DUV technology, whereas EXE and NXE provide EUV technology.

**Delivery performance:** Every order that is shipped to ASML is marked with how early/late the delivery was. Products received a delivery performance score based on its average performance over all orders.

**Quality complaints:** Customers can file complaints about the quality of the products they have received. This product characteristic is a simple count of how often a

complaint was made about a certain product, which could indicate how the quality of a product is perceived by the customer.

**Average price:** The price of the product. The price for which KMWE sells its products can vary. Therefore, an average over all orders is taken.

**Customer offset:** The customers will use the products that KMWE sells in their own final assemblies. The offset is defined as the time between when the product is required by the customer and when their final assembly should be finished. In other words, if the offset is 1 month, then the customer will require the product 1 month before the final assembly of the machine.

**Setup date:** Whenever a new product is developed, the product is added to the ERP system of KMWE. The setup date is date on which the item was created in the system. This date gives an indication of how old the product is.

In Table 4.6, an overview of the product characteristics is given, with the corresponding format and value range. In total, 5 characteristics are categorical, of which 4 are dichotomous (exactly 2 categories) and 1 is nominal (more than 2 categories), and 5 characteristics are continuous. Note that for setup date, it can be treated as discrete as well as continuous. We will consider it continuous because each date value is continuous down to the shortest measurement of time available, which in our case is days.

Table 4.6: Overview of product characteristics.

| **Product characteristics** | **Format** | **Value range** |
|---|---|---|
| Type | Dichotomous | Component, Assembly |
| Technology | Dichotomous | EUV, DUV |
| SMI project | Dichotomous | True, False |
| Service | Dichotomous | True, False |
| Machine | Nominal | XT, NXT, EXE, NXE |
| Delivery performance | Continuous | 0 - 100 % |
| Quality complaints | Continuous | $\geq 0$ |
| Average price | Continuous | $\geq 0$ |
| Customer off-set | Continuous | $\geq 0$ |
| Setup date | Continuous | 01-01-1998 - 01-01-2017 |

### 4.3.1 Descriptive Statistics

Before we executed the analysis, we first produced some descriptive statistics that will provide insight into how the 114 products are distributed over the characteristics. In Table 4.7, we have presented the distributions over the categorical characteristics. In this table we can see the frequencies of each category, and its percentage share relative to the total set of products. As we can see, for some characteristics, the distribution over the categories is very skewed. However, the statistical tests adjust for sample size differences, and therefore we are still able to analyze these category pairs.

Moreover, we can observe that the frequencies of the machine characteristic do not add up to 114. This is because some products can be present in more than one machine type. The statistical test that we will use to compare the categories, assumes that the samples in each category are independent. Therefore, alternative categories have been created for this product characteristic. This will allow us to use the same statistical test, and still make useful comparisons.

Table 4.7: Frequencies for categorical variables ($N = 114$).

| **Characteristic** | **Categories** | **Frequency** | **Percentage (%)** |
|---|---|---|---|
| SMI project | True | 3 | 2.6 |
| | False | 111 | 97.4 |
| Service part | True | 3 | 2.6 |
| | False | 111 | 97.4 |
| Machine | XT | 42 | 36.8 |
| | NXT | 21 | 18.4 |
| | EXE | 8 | 7.0 |
| | NXE | 64 | 56.1 |
| Technology | DUV | 50 | 43.9 |
| | EUV | 64 | 56.1 |
| Type | Component | 20 | 17.5 |
| | Assembly | 94 | 82.5 |

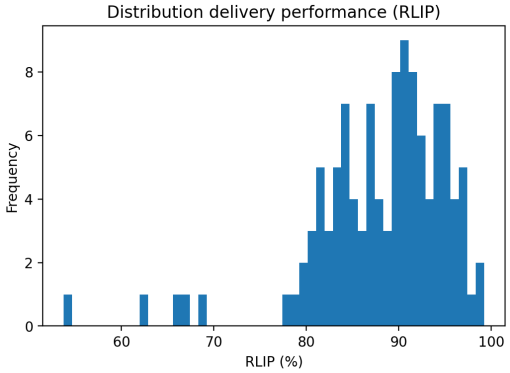
The alternative categories are presented in Table 4.8. The first part of the table consists of categories that describe whether a product is either in the machine (e.g. XT) or not in the machine (e.g. non-XT). Because these categories exclude the possibility of dependency, we can use the statistical test to compare them. The final part of the table defines categories that include products that are exclusive for a single machine. For example, XT only is a category that includes products that are only in

XT and not in any other machine. These categories have been used to make specific comparisons between machine groups. Again, we observe that the products are also not distributed fairly over the categories. However, the statistical test that we will be using can adjust for uneven category sizes. Something worth noting is that products that are in NXT or XT, are never in NXE or EXE, and vice versa. For this reason, we were able to compare, for example XT with EXE, without the risk of dependency.

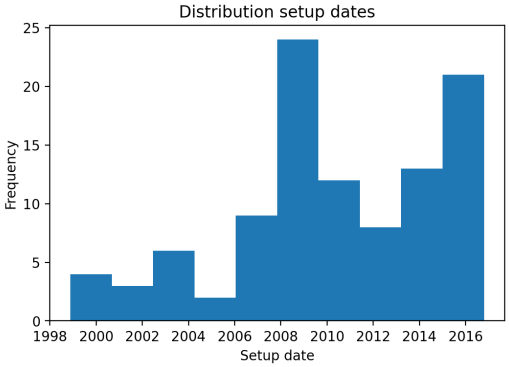
Table 4.8: Distributions alternative categories machines.

| **Categories** | **Frequency** | **Percentage (%)** |
|---|---|---|
| XT | 42 | 36.8 |
| Non-XT | 72 | 63.2 |
| NXT | 21 | 18.4 |
| Non-NXT | 93 | 81.6 |
| NXE | 64 | 56.1 |
| Non-NXE | 50 | 43.9 |
| EXE | 8 | 7.0 |
| Non-EXE | 106 | 93.0 |
| XT only | 29 | 25.4 |
| NXT only | 8 | 7.0 |
| NXE only | 56 | 49.1 |
| EXE only | 0 | 0 |

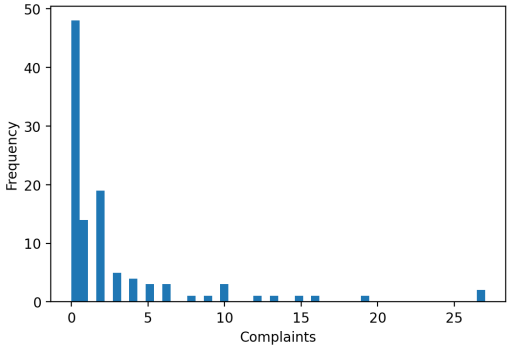
For the continuous characteristics we have produced distribution plots, which can be found in Figure 4.8. For the delivery performance (a) we can see that the data is approximately normally distributed. The setups dates (b) are slightly more represented by younger products. Furthermore, the number of complaints (c) is heavily skewed towards zero and we can see that around 48 products do not have any complaints. Similarly, the average price is skewed towards the left, where the majority of products have a price between 1 and 2500.



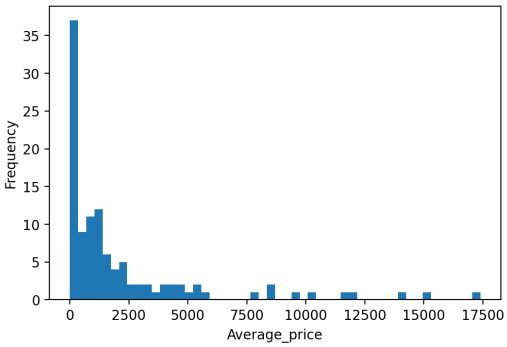

(a) Delivery performance (RLIP)



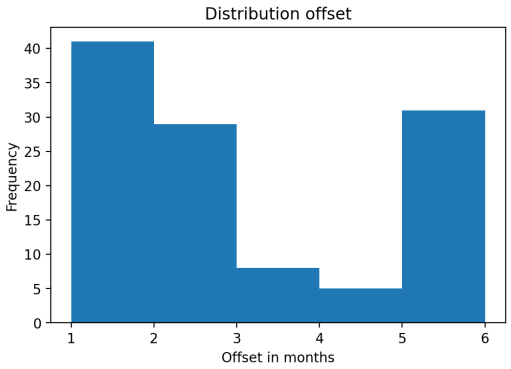

(b) Setup dates




(c) Quality complaints




(d) Average price




(e) Offset

Figure 4.8: Distributions of continuous variables

### 4.3.2 Analysis Results

The results of the statistical difference testing, for the categorical characteristics, can be found in Table 4.9. Pairs of categories have been compared by producing a test statistic and significance level, under various circumstances. Three forms of forecasts have been used to produce the evaluation scores for $R^2$ and MAAPE, for both monthly and quarterly aggregated data. The first form is a random forest model with customer forecast, the second is a random forest model without customer forecast, and the third directly uses the customer forecast as a prediction. Essentially, this table considers the conditions from the first two analyses. The goal of this, is to study the consistency of the results over various conditions. Note that the table contains both U-statistics and t-statistics, depending on how the underlying data was distributed. Also note that, the color of the cell indicates which category was significantly higher according to the test. In other words, the yellow cells indicate that cat 1 was higher than cat 2 under that particular setting and for that particular metric.

One of the most consistent results is the one for components and assemblies, where we observe that MAAPE is significantly higher for assemblies, for all forecasting methods, yet for $R^2$ no significant difference is found.

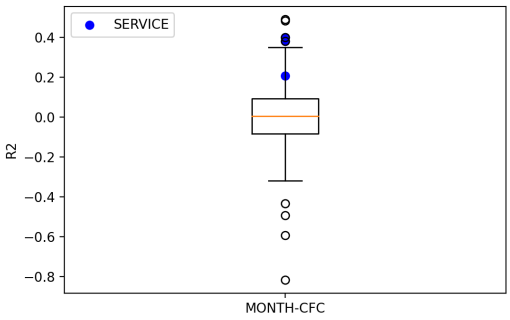
Another consistent result is that of DUV and EUV, where the $R^2$ is significantly higher for DUV products, yet no difference is found for MAAPE. Recall that, DUV products are part of XT or NXT, and EUV products are part of NXE or EXE. In the corresponding machine categories, we can find similar results, where the $R^2$ for DUV machine categories is significantly higher compared to EUV machine categories. This is the case for: NXT vs. non-NXT, NXE vs. non-NXE, NXT vs. NXE, NXT vs. EXE, XT vs. NXE. In these category pairs, we can observe that machine categories belonging to the DUV category, produce better $R^2$ scores than those from the EUV category. Interestingly, the consistent superiority of DUV over EUV, is rarely corroborated by the MAAPE scores.

Moreover, we can observe that the NXT only and XT only categories are significantly different, in favor of NXT only. This does not necessarily mean that NXT is the best performing machine category, since many DUV products are part of both NXT and XT, which makes the NXT only and XT only categories much smaller in size. Additionally, the results are not very consistent over the time periods and forecasting methods.

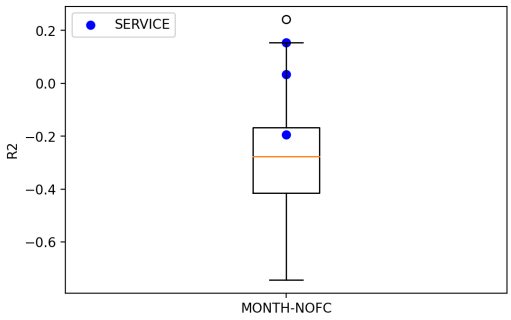
Finally, we see that, when comparing service products, with non-service products, that service products have significantly higher $R^2$ scores. Recall that, the samples size of these categories are highly skewed towards non-service products. Although we know that the independent t-test and Mann-Whitney U test are robust against different sample sizes, we have provided visual support in Figure 4.9, where we can see how the service products perform compared to the non-service products.

Table 4.9: Results statistics from independent t-test & Mann-Whitney U test (* $p < 0.05$, ** $p < 0.01$)

| | | With Customer Forecast | | | | Without Customer Forecast | | | | Only Customer Forecast | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Categories | | $R^2$ | | MAAPE | | $R^2$ | | MAAPE | | $R^2$ | | MAAPE | |
| Cat 1 | Cat 2 | Month | Quarter | Month | Quarter | Month | Quarter | Month | Quarter | Month | Quarter | Month | Quarter |
| NXT | Non-NXT | 1403** | 1442** | -2.1* | 785 | 2.34* | 1382** | -2.47* | 805 | 1254* | 1189 | 763 | 793 |
| XT | Non-XT | 2017** | 1777 | 1757 | 1.98 | 1.41 | 1860* | 1728 | 1.93 | 1858* | 1584 | 1586 | 1678 |
| NXE | Non-NXE | 885** | 1138** | 1526 | 1460 | -2.79** | 1075** | 1546 | -1.47 | 1212* | 1437 | 1677 | 1523 |
| EXE | Non-EXE | -1.19 | 332 | 406 | 420 | -0.46 | 419 | -0.3 | 400 | 324 | 322 | 451 | 416 |
| COM | ASS | 920 | 769 | 1283** | 1264* | 1.21 | 1034 | 2.86** | 1172* | 892 | 924 | 1405** | 1288* |
| DUV | EUV | 2315** | 2062** | 1674 | 1740 | 2.79** | 2125** | 1654 | 1.47 | 1988* | 1763 | 1523 | 1677 |
| NXT only | XT only | 1.05 | 168 | 59* | -1.59 | 2.2* | 151 | 58* | -1.49 | 123 | 134 | 80 | -1.44 |
| NXT | NXE | 1038** | 1024** | 497 | 579 | 2.63* | 998** | -2.1* | -0.36 | 2.07* | 838 | 523 | 574 |
| NXT | EXE | 126* | 2.58* | -0.77 | 73 | 1.25 | 1.28 | -0.97 | -0.22 | 1.25 | 117 | -0.89 | 71 |
| XT | NXE | 1894** | 1667* | 1527 | 1.8 | 2.08* | 1737* | 1501 | 1.8 | 1689* | 1432 | 1373 | 1469 |
| XT | EXE | 226 | 212 | 192 | 0.56 | 0.88 | 203 | 194 | 0.7 | 239 | 207 | 170 | 182 |
| NXE only | EXE only | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| SMI | Non-SMI | -0.18 | 180 | 152 | 149 | 0.43 | 211 | -0.25 | 144 | 232 | 166 | 157 | 140 |
| Service | Non-service | 3.07** | 298* | 96 | 191 | 2.56* | 269 | -1.05 | 215 | 317** | 271 | 95 | 163 |




(a) Only customer forecast - Month




(b) Without customer forecast - Month

Figure 4.9: Boxplot visualizations of distribution of service products compared to non-service products.

The correlation results for the continuous characteristics can be found in Table 4.10. Just like for the categorical characteristics, we studied the consistency of the results for three forms of forecast, with two metrics, for two time aggregations. The Pearson correlation coefficients can be interpreted as follows: a larger correlation (negative or positive) implies that the characteristic has more effect on the performance.

For the setup date, we observe that, only for $R^2$, a significant correlation can found. For the model based forecasts (with customer forecast, without customer forecast), the correlations are negative, which means that as setup date increases, the $R^2$ decreases (Figure 4.10a). Remarkably, the opposite is found for customer forecast

predictions, where, at least for quarterly predictions, the $R^2$ increases with the setup date (Figure 4.10b).

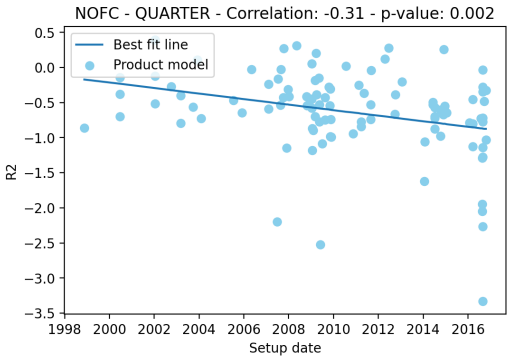
Next, we can observe that for both the offset and delivery performance (RLIP), there is only one instance of circumstances, which shows correlation between the performance and the characteristic. For the offset, only in the case with customer forecast, on monthly data, for $R^2$ a significant negative correlation is found, which implies that, under those circumstances, the $R^2$ decreases as offset increases. For the delivery performance (RLIP), we can observe that, if we use customer forecast directly, on quarterly data, it is negatively correlated to $R^2$. The lack of consistency over the forecasting methods, evaluation metrics and time aggregations, suggests that in general these relations are not very strong.

For the average price, we can see that, over several circumstances, a significant correlation is present, which point towards the same thing: as the price increases, the predictive performance increases (decrease of MAAPE and increase for $R^2$)(Figure 4.10c and Figure 4.10d). This is, however, not corroborated under all circumstances, as for models without customer forecast and directly using customer forecast, the $R^2$ does not appear to have a relation with the price.

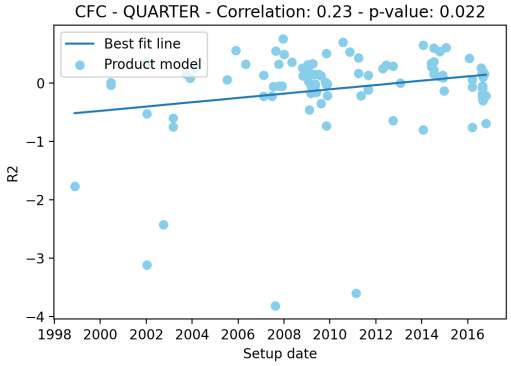
Finally, for quality complaints the results are very consistent. For $R^2$ there is no correlation under any circumstance. Yet, for MAAPE, the results all suggest that there is a negative correlation with the number of complaints. In other words, as the number of complaints increase, the MAAPE decreases (Figure 4.10e and Figure 4.10f).

Table 4.10: Correlation model performance and product characteristics (* $p < 0.05$, ** $p < 0.01$).

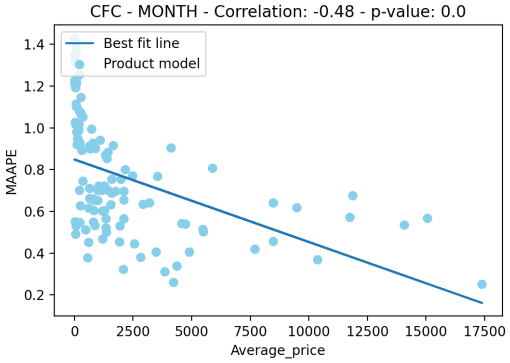
| **Characteristics** | **With Customer Forecast** | | | |
|---|---|---|---|---|
| | $R^2$ | | MAAPE | |
| | Month | Quarter | Month | Quarter |
| Setup date | -0.25* | -0.13 | -0.14 | -0.04 |
| Offset | -0.22** | -0.12 | -0.01 | -0.03 |
| RLIP | -0.01 | 0.02 | -0.01 | -0.03 |
| Average price | 0.34** | 0.29** | -0.34** | -0.18 |
| Quality complaints | -0.03 | 0.15 | -0.4** | -0.37** |
| | **Without Customer Forecast** | | | |
| | $R^2$ | | MAAPE | |
| | Month | Quarter | Month | Quarter |
| Setup date | -0.25* | -0.31** | -0.18 | 0.02 |
| Offset | -0.08 | -0.06 | 0 | -0.06 |
| RLIP | -0.05 | 0.13 | 0.06 | -0.16 |
| Average price | 0.16 | 0.09 | -0.3** | -0.1 |
| Quality complaints | -0.01 | 0.06 | -0.49** | -0.25** |
| | **Only Customer Forecast** | | | |
| | $R^2$ | | MAAPE | |
| | Month | Quarter | Month | Quarter |
| Setup date | 0.07 | 0.23* | -0.17 | -0.16 |
| Offset | -0.11 | -0.05 | 0.03 | -0.01 |
| RLIP | -0.11 | -0.22* | -0.01 | 0.14 |
| Average price | 0.14 | 0.09 | -0.48** | -0.23* |
| Quality complaints | -0.07 | -0.04 | -0.4** | -0.34** |



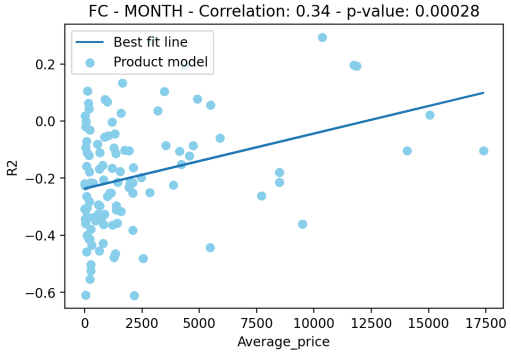

(a) Negative correlation setup date using model without customer forecast.



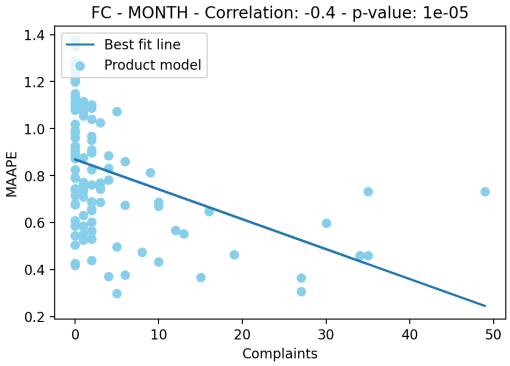

(b) Positive correlation setup date using only customer forecast.



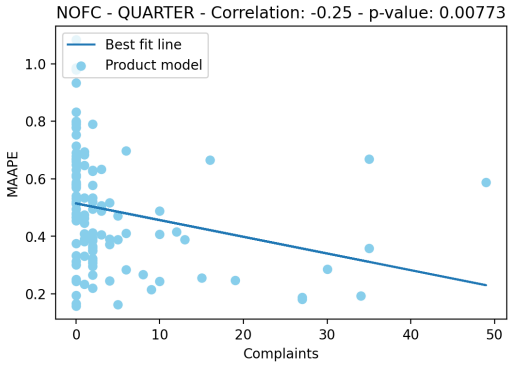

(c) Negative correlation average price using customer forecast only.




(d) Positive correlation average price using model with customer forecast.




(e) Negative correlation number of complaints and MAAPE for model with customer forecast.




(f) Negative correlation number of complaints and MAAPE for model without customer forecast.

Figure 4.10: Correlation graphs between various product characteristics and predictive performance.

We will finish this analysis by presenting the best performing products, and their corresponding product characteristics, according to the models with customer forecast (Table 4.11). Although, we can not use these results to make general claims about the how product characteristics impact the predictive performance of forecasting models, we can relate the results to what we have seen in earlier stages of this analysis and also produce insights into performance in general. For both metrics we found the five best performing products per period, some of which were the best performing products for both periods ("Both" in the table), and one product was best performing according to both $R^2$ and MAAPE (* in table). These measurements were taken in order to prevent over representation of repeating products and its characteristics, which is also why the table does not contain 20 products.

When we look at the type column, we can see that all, except one, products are assemblies. In a way, this contradicts what we have found at the start of the analysis, which showed that for assemblies, on average, the MAAPE was significantly higher. Therefore we must conclude that the majority of assemblies produce worse MAAPE scores, and only a few assemblies actually produce good scores, which ended up in this table.

In the technology column we can see that DUV is more dominant than EUV, for $R^2$, but less so for MAAPE, which is in line with our earlier findings, where we compared the performance difference between DUV and EUV statistically. Recall that the technology also indirectly defines which machines the product can end up in (DUV=NXT,XT and EUV=NXE,EXE), which can be seen in the machine(s) column.

When we look back at Figure 4.8d, we saw that the average price is heavily skewed towards the left, which indicates the majority of products are lower priced (1-2500). Remarkably, many of the best performing products are relatively high priced. This is also in line with previous results that stated that, under many circumstances, the predictive performance increases with the average price.

Table 4.11: Best performing products for models with customer forecast

| Metric | Period | Product | $R^2$ | MAAPE | Type | Tech | Machine(s) | RLIP (%) | Q.C. | Price (€) | Offset | Setup |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $R^2$ | Month | 4022 455 7606 | 0.60 | 0.95 | COM | DUV | NXT, XT | 93.8 | 2 | 311 | 1 | 10/01/2002 |
| | | 4022 639 0868* | 0.29 | 0.37 | ASS | DUV | XT | 83.7 | 4 | 10358 | 1 | 23/07/2010 |
| | | 4022 637 6154 | 0.29 | 0.82 | ASS | DUV | XT | 87.5 | 2 | 2928 | 1 | 19/12/2007 |
| | Both | 4022 635 7279 | 0.20 | 1.22 | ASS | DUV | XT | 80.8 | 0 | 4367 | 1 | 05/10/2007 |
| | | 4022 642 2644 | 0.20 | 0.73 | ASS | DUV | NXT | 84.3 | 35 | 11743 | 2 | 17/04/2012 |
| | Quarter | 4022 656 9951 | 0.47 | 0.22 | ASS | DUV | NXT | 83.1 | 2 | 2110 | 1 | 25/01/2016 |
| | | 4022 669 2863 | 0.41 | 0.57 | ASS | EUV | NXE | 87.2 | 1 | 1929 | 6 | 13/02/2017 |
| | | 4022 642 9925 | 0.41 | 0.59 | ASS | DUV | NXT | 83.1 | 49 | 11861 | 2 | 15/06/2012 |
| MAAPE | Month | 4022 639 3863 | -0.25 | 0.37 | ASS | DUV | XT | 89.9 | 15 | 2089 | 1 | 27/10/2008 |
| | Both | 4022 637 0565 | -0.15 | 0.30 | ASS | DUV | NXT, XT | 95.4 | 5 | 4216 | 1 | 24/08/2007 |
| | | 4022 635 1189 | -0.22 | 0.31 | ASS | DUV | XT | 94.5 | 27 | 3854 | 1 | 20/10/2008 |
| | | 4022 456 2292 | -0.26 | 0.36 | ASS | DUV | NXT, XT | 89.6 | 27 | 7702 | 2 | 06/03/2003 |
| | Quarter | 4022 623 2159 | -0.32 | 0.16 | ASS | EUV | NXE | 91.2 | 0 | 855 | 5 | 28/05/2009 |
| | | 4022 669 0052 | 0.15 | 0.17 | ASS | EUV | NXE, EXE | 84.0 | 1 | 2474 | 5 | 20/09/2016 |

# Chapter 5

# Discussion

In this chapter we will discuss the results from the previous chapter. Firstly, we will review how the interpretation of the evaluation metrics influenced the conclusion that were made. Secondly, we will review the results of the three analyses, by interpreting the outcomes and discussing its implications.

## 5.1 Review of the Evaluation Metrics

When we want to interpret the results, it is important to understand how to interpret the evaluation metrics first. As we have seen, the $R^2$ and MAAPE do not always agree, and sometimes even produce contradicting results. $R^2$ is a metric that represents the proportion of variance, in the dependent variable, that is explained by the independent variable(s). MAAPE is a measure that quantifies the distance between the predictions and the actual values. Essentially, these metrics take a different approach for defining how good a model is. $R^2$ rewards predictions that seem to follow the same pattern in the data, while MAAPE rewards precision of the predictions. One could say that $R^2$ is better for finding models that fit the the data and explain why certain patterns occur in the data, which could lead to good results on new data from the same source. The same is not necessarily true for MAAPE, as good scores on the train and test set do not ensure good results on new data, because precise decisions on train and test data can also happen by chance. Just like MAAPE, SIME measures the precision of the prediction. However, SIME offers a more practical interpretation, because it makes some assumptions that also have to made in reality when implementing a forecasting model. SIME adjusts for current inventory when ordering products for the next period, which is something that would happen in reality as well. Moreover, SIME makes assumptions about the arrival of new products in the inventory, which is also something companies have to think about when implementing forecasting models.

This makes SIME a metric that can actually provide useful insights about what would happen if a forecasting models was to be used in practice. Unfortunately, SIME is scale dependent, which means that it can not be used to compare models built on different data sources, which is why it was also excluded from the third analysis.

## 5.2 Review of the Results

In this section, we will review the results of the three analyses, by interpreting the outcomes and discussing its implications. In the first analysis we studied the impact of customer forecast data on the predictive performance of product level forecasting models. In the second analysis we studied how aggregating data would impact the predictive performance of product level forecasting models. In third analysis, we studied the relation between product characteristics and the predictive performance of product level forecasting models.

### 5.2.1 Analysis 1

In the first analysis, we studied the role of customer forecast data in demand forecasting. In the first part we found that adding customer forecast data to the input of a model increased the $R^2$ and MAAPE of the models significantly, while the SIME did not. If we look at the distribution graph of $R^2$ (Figure 4.1b), we can see that, the distribution for models with customer forecast is narrower than without customer forecast, and also shifted slightly more to the right. So, we know that adding the customer forecast improved the models, in terms of $R^2$. As we know, $R^2$ is a metric that measures how much of variability is explained by the model. In chapter 2, we have seen that the variability in a time series is the result of various forms of variability: trend, seasonality, other cycles and residual variation. Especially the latter is difficult to manage as it often caused by random variation in the data source. It is very likely that the customer forecast data, which is close to the data source, is strongly related to the actual demand, as the customer forecast data can be seen as a declaration of intend by the customer. So, it is not surprising that the customer forecast data generally improves the explainability of the variability. Also, the precision of the forecasts was improved by adding the customer forecast data, as can be seen by the improvement in MAAPE. Although this improvement is significant, it is not large, as can be seen in Figure 4.1d. This is also confirmed by the U statistic of the MAAPE, as the difference with the SIME statistic is very small, and for SIME we did not observe a significant improvement.

For SIME, we did not find the same improvement. We could argue that the customer forecast data is not accurate enough in absolute terms, but instead only

follows a similar pattern as the actual data, which the model then learns. However, in Table 4.2, we can observe that the customer forecast data is significantly more accurate than the models, which implies that the customer forecast data is more valuable than the results from Table 4.1 would suggest. Instead, we can argue that the models are not capable of discovering the value of the customer forecast, which is why the improvement is small. It is surprising that a model with customer forecast performs worse than direct use of customer forecast, since the model has the same information, and additional historical demand data. In theory, the model should learn that the customer forecast is a decent predictor of the demand, and therefore produce predictions that are at least as good as the customer forecast. We argue that in reality the model does not have enough data to detect that the absolute values from the customer forecast are actually better than the predictions made by looking at the historic demand. Surprisingly, the model is able to detect that the pattern from the customer forecast produces better results, which is why the $R^2$ does improve. Yet, the direct use of customer forecast produces an even better $R^2$, so the model is still not able to detect the complete pattern that is presented by the customer forecast. In Table 4.2, we can also observe that SIME is significantly better, when directly using the customer forecast. However, again, when looking at the U statistic and significance level, we can see that the improvement is very small.

We also looked how these results transferred to the quarterly aggregated situation. In this situation, we observed that adding customer forecast to the model improved all three metrics significantly, and also, in terms of MAAPE and $R^2$, the improvement was larger, compared to the monthly aggregation. This could mean that for quarterly aggregation, the model is able to detect that the predictions of the customer forecast are indeed superior to forecast based on historical demand data only, which is surprising, since the model has even less data for quarterly aggregation. So, we argue that the customer forecast has to be much more accurate on quarterly aggregation, to achieve this large improvement. In Figure 4.3a, we illustrated that the customer forecast, sometimes, fits poorly. However, we can also observe that if we were to sum the demand from the first 4 months of 2018 (Q1, 2018), and compare it to the sum of customer demand forecast over the same months, that these values match up much better than when comparing the demand and customer forecast per month. So, we argue that the customer forecast is much more accurate on quarterly aggregation, which is why adding it, shows a larger improvement of the model performance. However, the model was still inferior to the predictions that came directly from the customer forecast, which means that the amount of data is probably still to small to detect the advantage of using customer forecast data fully.

For SIME, the results are surprising. The size of the improvement, compared to MAAPE and $R^2$ was much smaller under all situations. This was true for both the monthly and quarterly aggregated models. If we assume that SIME is an accurate

representation of how KMWE would implement and evaluate the performance of a forecast in practice, then we can not conclude that adding customer forecast, or using it directly, would satisfy KMWE's business needs.

In order to put this in perspective, we have created a baseline prediction method. The goal of this method is to understand how the prediction results relate to a quick and simple prediction. This baseline takes the average over the previous 3 periods, as a prediction for the next period, which is similar to SMA. In case predictions are worse than this baseline, we can argue that the effort of building a more complex forecasting model can not be justified under these circumstances, as it would not beat a simple one. We compared both the models with customer forecast, and direct use of customer forecast data, with this baseline, of which the results can be found in Table 5.1 and Table 5.2.

For both type of forecasts, we found that they were significantly better than the baseline, in terms of $R^2$. For this metric, the difference was very large, as can be seen in Figure 5.1. This is expected, since a simple SMA will not be able to detect patterns in the data, the way a random forest can, which is why the explanation of variability is low for the baseline. Yet, for MAAPE and SIME, we could argue that the difference is still not very convincing. Surprisingly, the MAAPE for models with customer forecasts is even worse than the baseline. So, a higher $R^2$ does not necessarily mean that the model will satisfy the business needs of KMWE.

We argue that explaining variability is not that important when the goal is to minimize inventory and back orders. Instead, the absolute difference between the prediction and actual demand is much more important, since it produces expenses in the form of inventory or back order costs. So, a model that generates predictions which follow a similar pattern as the demand, is not necessarily better than a model that predicts just an average of the past few observations, because it is likely not that far off, in absolute terms.

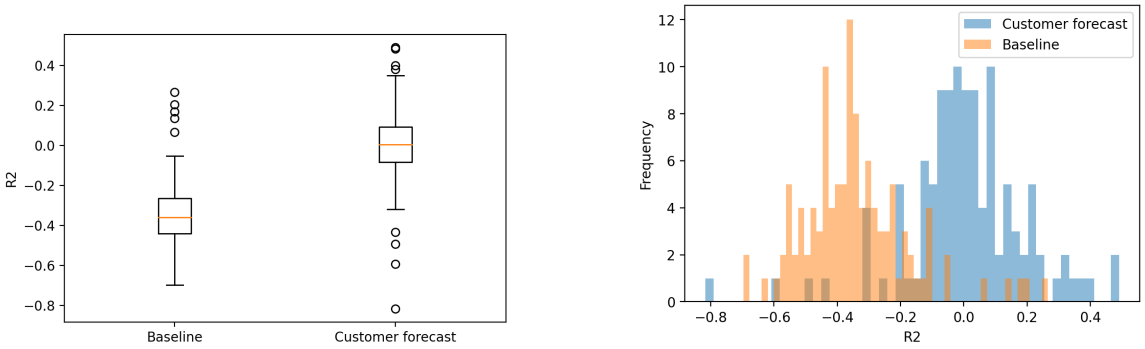
In conclusion, this analysis investigated the impact of customer forecast data to the predictive performance of models, as well as the predictive performance of the customer forecast data itself. It was found that adding customer forecast data to a model mostly improves the model in terms of $R^2$, and not so much for MAAPE and SIME. We also found that, using the customer forecast data directly produces better or equal results as building a forecasting model. However we argued that, the improvement of adding customer forecast, under all circumstances, was not very convincing. Even though the customer forecast was able to explain more of the variability in the demand data, we argued that for inventory management, the precision of the forecast is more important.

Table 5.1: Statistical difference testing: model with customer forecast vs baseline (n=114) (monthly predictions).

| Metric | Statistic | p-value | Outcome |
|---|---|---|---|
| $R^2$ | 1048.0 | $2.91 \times 10^{-10}$ | Model > Baseline |
| MAAPE | 2301.0 | 0.0058 | Model > Baseline |
| SIME | 2151.5 | 0.0022 | Model < Baseline |

Table 5.2: Statistical difference testing: customer forecast vs. baseline (n=114) (monthly predictions).

| Metric | Statistic | p-value | Outcome |
|---|---|---|---|
| $R^2$ | 233.0 | $7.43 \times 10^{-18}$ | Customer forecast > Baseline |
| MAAPE | 2244.0 | 0.0035 | Customer forecast < Baseline |
| SIME | 1675.5 | $5.91 \times 10^{-6}$ | Customer forecast < Baseline |




(a) Boxplot $R^2$

(b) Distribution plot $R^2$

Figure 5.1: Distribution performance: baseline vs. customer forecast

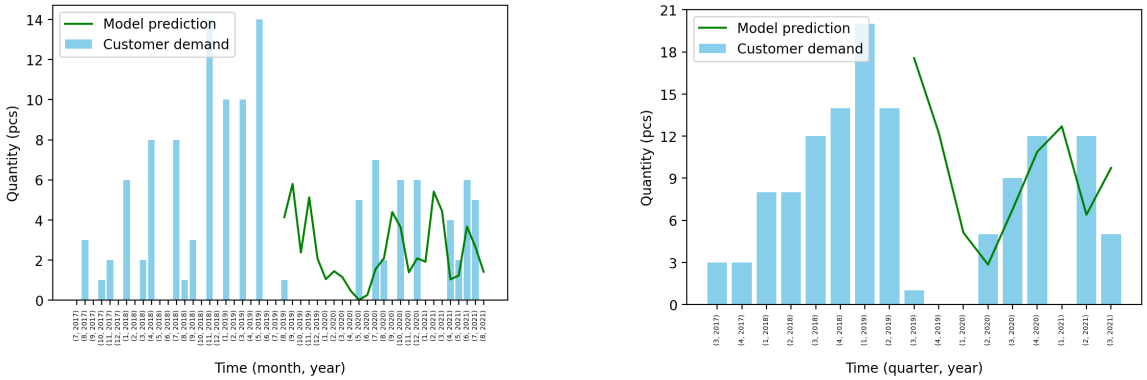
### 5.2.2 Analysis 2

In this analysis, we investigated how aggregating demand data from monthly to quarterly would impact the average performance of product forecasting models. We observed that aggregating data can change two important properties of the time series, namely, the length of the time series, and the number of zero demand observations. The question was how these changes impacted the performance of the forecasting methods.

Firstly, we found that the $R^2$ decreased significantly for quarterly aggregated models, which seemed to be caused by the major increase in the variance in performance for the 114 products (see Figure 4.6b). In other words, when aggregating quarterly, we observed an increase in bad models, as well as an increase in good models, where the increase in bad models was slightly higher. We argue that the increase in performance variance could be caused by a decreased test set length. For quarterly predictions, the same train-test split of 50% was used, however, because the time series are shorter, the test set becomes shorter as well. A shorter test set is more sensitive to extremely bad or good scores, hence the $R^2$ scores for the different products shows more variance.

If we look at Figure 5.2, we can see the predictions for the same product on monthly and quarterly aggregated data. We could argue that, visually, these predictions are not that different in terms of performance. However, as it turns out, for this product the decrease in $R^2$ was the largest of all products, where monthly $R^2 = -0.52$ and quarterly $R^2 = -2.35$. For quarterly forecast, where the test set is shorter, it will be harder to compensate a very bad prediction as it will be for the longer monthly test set.

However, this does not mean that the decrease in $R^2$ from monthly to quarterly is only the consequence of a shortened test set. It can still be true that aggregating quarterly is actually bad for the performance due to the change in properties of the time series data as well. The shortened test set would theoretically lead to an equal increase in both bad and good models. Yet, we found a larger increase in bad models, which is why on average aggregating to quarterly forecasts, decreased the $R^2$. We argue that an explanation for the decrease in $R^2$ could simply be that, the training data has become a lot shorter, which is why the model can not capture the patterns in the data correctly.




(a) Monthly forecast

(b) Quarterly forecast

Figure 5.2: Predictions for product with the highest decrease in $R^2$: from monthly ($R^2 = -0.52$, MAAPE $= 1.15$) to quarterly ($R^2 = -2.35$, MAAPE $= 0.91$).

Secondly, we found that MAAPE decreased significantly for quarterly forecasts. Compared to the change in $R^2$, the change for MAAPE is much larger. Therefore we do not think it could have been caused by the decrease in test set length. Instead, we argue that the decrease in MAAPE was caused by an increase in stability. This is supported by the fact that for monthly aggregated data about 30% of the product was smooth and 64% was intermittent, whereas for quarterly aggregated data 84% was smooth and only 15% was intermittent. This shows a significant shift from mostly intermittent to mostly smooth data. Recall from chapter 2, that in order for this shift to happen, the number of zero demand periods has to decrease. We believe that the models struggle to predict zero demand, which is why it would be beneficial to see a decrease in zero demand.

We can use the Figure 5.2 again, to illustrate this. The MAAPE decreased from 1.15 for the monthly forecast to 0.91 for the quarterly forecast. We can see that, in the monthly forecast, the model only predicts zero demand once, while the training and test set clearly show many instances of zero demand. Because aggregating the data decreases the amount of zero demand periods, it becomes clear why the MAAPE decreases significantly.

Thirdly, we observed that SIME was on average significantly lower for monthly forecasts compared to quarterly forecasts. We can explain this by looking at the assumptions of SIME. For monthly predictions, products enter the inventory at the beginning of the month, and slowly deplete during the month until the start of the next month, when again new products enter inventory. If we would have done the same for quarterly forecasts, the products would have entered at the start of the quarter and slowly deplete until the start of the next quarter. However, we argued that, instead, it would be more realistic to spread the deliveries of inventory evenly

over the quarter, according to KMWE. This means that 1/3 of the product would enter the inventory each month within a quarter. We believe that, the SIME score for quarterly forecasts is, on average, worse because, the timing of when orders request products becomes more crucial.

Even if the accuracy of the prediction for quarter as a whole is better than for a month, the timing of the orders could still mean that we would have too much inventory or back orders for a longer period of time during the quarter. Let's say hypothetically all orders for a product would come in the last month of a quarter, then the precision of the forecast can still be good, but the inventory costs would be immense. This would be avoided with monthly forecasts because it would just predict low quantities for the first two months, and high for the last month. So even if the prediction for the last month is less accurate, compared to the prediction for the quarter, the inventory costs would be much lower. This example nicely illustrates why, in practice, a higher precision for a longer time period is not always better than a lower precision for a shorter period of time.

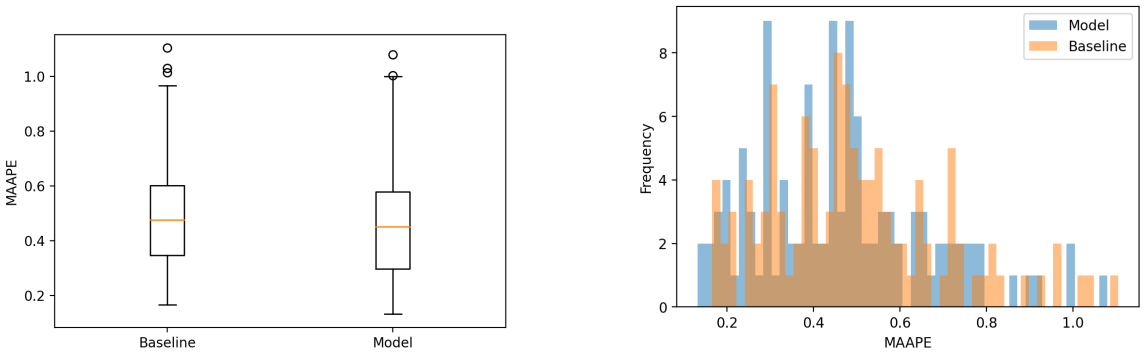
In the previous analysis we found that, when comparing the performance of the models and the direct use of customer forecast to the baseline, on a monthly time aggregation, that customer forecast data was only slightly better in terms of MAAPE and SIME. So, for this analysis we will compare the performance of the forecasting model and the direct use of customer forecast data, on quarterly basis, to the baseline.

In Table 5.3, we can see that in terms of $R^2$ and MAAPE, the models are better than the baseline. Recall that, in the monthly aggregated situation, we saw that the models were only better in terms of $R^2$ and surprisingly worse in terms of MAAPE. So, aggregating has mostly affected the MAAPE, which now favors the models over the baseline. Still, the improvement is small, as can be seen in Figure 5.3. Also, for SIME we can not observe a significant difference between the model and the baseline, which leads us to conclude that building a forecasting model, is difficult to justify under these circumstances, for quarterly forecasts.

In Table 5.4, we can observe that on quarterly aggregated data, the direct use of customer forecast data is better in terms of all metrics. Just like with the monthly aggregation, the direct use of customer forecast shows a larger difference with the baseline, compared to models with customer forecast. Yet, also just like with monthly aggregation, the improvement is relatively small.

Table 5.3: Statistical difference testing: model with customer forecast vs baseline (n=114) (quarterly predictions).

| Metric | Statistic | p-value | Outcome |
|---|---|---|---|
| $R^2$ | 1502.0 | $5.17 \times 10^{-7}$ | Model > Baseline |
| MAAPE | 1409.0 | $1.27 \times 10^{-7}$ | Model < Baseline |
| SIME | 2637.5 | 0.0949 | Model = Baseline |




(a) Boxplot MAAPE

(b) Distribution plot MAAPE

Figure 5.3: Distribution performance: baseline vs. model with customer forecast (quarterly forecasts).

Table 5.4: Statistical difference testing: customer forecast vs baseline (n=114) (quarterly predictions).

| Metric | Statistic | p-value | Outcome |
|---|---|---|---|
| $R^2$ | 804.0 | $2.68 \times 10^{-12}$ | Customer forecast > Baseline |
| MAAPE | 1078.0 | $5.01 \times 10^{-10}$ | Customer forecast < Baseline |
| SIME | 2263.5 | 0.0130 | Customer forecast < Baseline |

In conclusion, we found that aggregating from monthly to quarterly increases the variation of $R^2$ performance scores over the products. We argued that this could have been caused by a shorter test set, which is more sensitive to the precision of the predictions. Yet, this would not explain why the mean of quarterly models is lower. Hence, we can still assume that the aggregating quarterly is bad for $R^2$, although more data is required to gather more evidence. Furthermore, we found that MAAPE

decreased significantly for quarterly forecasts, which we explained by the decrease in zero demand periods. The models struggle to predict zero demand, even if the training data is very intermittent. Also, we found that SIME is significantly lower for monthly forecasts. We argued that his was caused by the timing of orders during the time periods. We concluded that, in terms of SIME, a higher precision for a longer time period is not always better than a lower precision for a shorter period of time. Finally, we compared our findings to the baseline and found that the quarterly aggregated models produce better $R^2$ and MAAPE scores, yet for SIME, the scores did not change significantly. And for direct use of customer forecast data, all scores improved, yet the size of the improvement is small for MAAPE and SIME.

### 5.2.3 Analysis 3

In this analysis, we gathered a collection of categorical and continuous product characteristics that would potentially affect the predictive performance of a demand forecasting model. Per characteristic, we investigated if we could distinguish predictive performance differences over its value range.

For the categorical characteristics, we used a statistical test, that can identify if the performance scores come from the same distribution. The results from these tests can, therefore, not be used to make claims about why the performance scores are different. Instead, we try to make logical explanations in the context of the information that was provided to us.

For the continuous variables, we investigated the correlation between the performance and the characteristic values. As we know, correlation is not causation. Therefore, we can merely logically speculate about why relations in the data are present.

#### Type: Components vs. Assemblies

For the product type, we found that, there was a significant difference between assemblies and components, where the assemblies produced a higher MAAPE. As we know, assemblies are typically sold in smaller quantities. However, MAAPE is scale independent, which means we can not explain the difference in MAAPE by using the argument of differently scaled quantities. Therefore, we have to look for other explanations. When comparing the distributions of demand classification for components (15 intermittent, 3 smooth, 1 lumpy, 1 erratic) and assemblies (58 intermittent, 31 smooth, 5 lumpy), we do not see a very different distribution over the classes, which means we can not attribute the difference to this. Instead, we argue that stability within the intermittent products is better for the components. As we have seen in chapter 2, demand classification is based on predetermined ranges for ADI and $CV^2$.

As a result, two times series, which both are marked as intermittent, can still look differently in terms of stability. Therefore, we believe that the time series for components are still more stable. A reason for this might be that the customer intervenes less with their system generated order suggestions for components as they are often less important for their operations. On the other hand, more important products, like complex assemblies, show less stable demand, because the customer is more considerate about placing orders, as the timing of the arrival of these products is more crucial, which is why the interval and sizing could be less consistent.

The slightly better stability did not improve $R^2$. When studying the best performing products according to $R^2$ and MAAPE, we notice that, the underlying data of good performing $R^2$ models fluctuates much more than the data of good performing models according to MAAPE. In other words, MAAPE favors stable data more than $R^2$, because the error of the prediction is less likely to be high. On the other hand, $R^2$ favors models that manage to make predictions that closely follow a highly fluctuating demand pattern. So, very stable demand patterns are less likely to receive a good $R^2$ score, but more likely to receive a good MAAPE score.

**Technology: DUV vs. EUV**

Next, we found that the forecasts for products in NXT were significantly better than for non-NXT products in terms of MAAPE and $R^2$. Also, the $R^2$ for XT was significantly better under monthly forecast with customer forecast compared to non-XT. As we know, NXT and XT products, make up the DUV category, and NXE and EXE, make up the EUV category. When comparing DUV and EUV, we also find that for $R^2$ DUV shows better performance. Another result showed that the $R^2$ was better for NXT than for NXE, and also better for XT than for NXE. We argue that all these finding are related, and could be explained by the fact that EUV demand (so also NXE and EXE) is more dominated by zero demand observations. EUV products consists for 72% out of intermittent demand patterns under monthly forecasts, whereas for DUV products this is only 54%. This would also explain why the performance differences for quarterly forecasts are generally smaller and less often significant when looking at the category pairs mentioned previously. So, the demand for EUV products more intermittent. Again, we can not know, why this is, for sure, but one possible explanation is that the maturity of a product plays a role. EUV is relatively new technique, and therefore its products are as well. On the other hand, DUV products are more mature, which means that the customer might have a better production flow of those products, which means that the products are order frequently, and therefore show a less intermittent demand pattern, which could be favorable for $R^2$.

### Service vs. Non-service

Additionally, we found that service products showed significantly higher $R^2$ scores than non-service products. We have to be careful when making generalizing statements about this, because the category sizes are heavily skewed towards non-service products. However, as we know, the statistical tests account for this, which means that the 3 service products were very noticeably higher compared to the general distribution. When looking at the demand classification for these 3 products we can see that, for both aggregation levels, the demand is smooth. A reason for this can be that service products are sold more often than other products, because they do not only end up in new machines, but are also used for servicing existing machines. Also, when observing the time series for the service products, we can see that, although they are marked as smooth, they appear to be more erratic than other smooth time series, which would explain why the difference is significant for $R^2$ only. We argue that the more erratic behaviour comes from an equally erratic demand for servicing, which is mostly driven by product breakdowns.

### Setup Date

For the continuous variable setup date, we found that for model based predictions, there was a negative correlation with the $R^2$, which means that newer the products produce worse $R^2$ scores. This is in line with our logic from before about EUV and DUV, where we argued that older products showed less intermittent demand patterns, which is beneficial to the model $R^2$. Surprisingly, we found that, when the customer forecast was used directly, there was an opposite positive correlation between setup date and $R^2$, which means that newer products produce better $R^2$ when directly using customer forecast. This could mean that the customer is able to anticipate its own behaviour better for newer products. In other words, the customer follows its own forecast more closely. Apparently, this does not help the performance of our models, even if they are built on the same customer forecast data. This could imply that the historical data for older products is more stable or understandable for our models, compared to the data for newer products.

### Offset & RLIP

For offset and RLIP we found only one instance, for which the relation with the performance was significant. For offset this was with $R^2$, for models with customer forecast built on monthly data. In this instance, as offset increases, the $R^2$ decreases. For RLIP this was with $R^2$, for direct customer forecasts on quarterly data. In this instance, as RLIP increases, $R^2$ decreases. However, due to the lack of consistency

over the various methods and periods, we were not able to make definitive statements about the actual relation and causation of these results.

### Average Price

For the average price, it was found that for MAAPE, the results are relatively consistent in their conclusion that, as price increases, MAAPE decreases. Also, for the models with customer forecast, average price is positively correlated with $R^2$ for both monthly and quarterly models. In Figure 4.10c and Figure 4.10d, we can see that the data is more skewed towards the lower prices, which affects the validity of these results, as the value range is not evenly represented.

### Number of Complaints

Similarly, for the number of complaints, we found that the relation with MAAPE was very consistent in the conclusion that, if the number of complaints increases, MAAPE decreases. However, in Figure 4.10e and Figure 4.10f, we can see that, again, the data is skewed toward zero, which means there are way more products with few complaints than there are with many complaints. As a result, we can not conclude that there is a linear relationship between number of complaints and predictive performance.

### Best Performing Product Models

Finally, we studied the best performing product models, per evaluation metric and per aggregation period. Surprisingly, we found that the best performing products, according to MAAPE, are all assemblies. And not only for MAAPE, but also for $R^2$, the best performing product are assemblies, except for one. This dominance can be explained by the fact that assemblies are over-represented in the product set. Yet, we would expect some well performing components to end up in this table, since the entire set of components has on average a lower MAAPE than assemblies. This leads us to conclude that the components group is more consistent its performance, whereas the assemblies have some very poor performing products (which increase the average MAAPE), as well as some very good performing products (which end up in this table).

When we looked at the technology type, we found that DUV was more dominant in the best performing models for $R^2$, despite being slightly under-represented in the product set. This is directly in line with earlier findings that suggested that DUV has less intermittent demand patterns because it is an older technique with more mature products. This is confirmed by the setup dates of these products, as they are mostly from before 2013.

Earlier, we questioned the validity of the correlation between the average price and predictive performance, because of the data being skewed towards zero. Remarkably, the best performing products are all relatively expensive, with most being $> 1500$ euros. We can argue that this actually supports the correlation between the average price and predictive performance.

We can not fully explain this correlation. We know that correlation is not causation. So, there can be other characteristics, unknown to us, that are responsible for this relationship. For example, the customer might have specific policies for expensive products that directly impact how the demand pattern is manifested in the data, which in turn impacts how well a model can make predictions. In other words, it is possible that the correlation between average price and predictive performance is caused by unknown, unavailable and external factors.

### Concluding Remarks

In conclusion, we studied the relation between product characteristics and the predictive performance of forecasting models. We found that assemblies produced higher MAAPE scores compared to components, which we explained by how the customer perceives the importance of a product, and how this could affect the stability of the demand pattern. We also argued that MAAPE favors this stability more than $R^2$, which is more likely to favor models that capture a highly fluctuating demand pattern. Next, we found that DUV, and related machine categories NXT and XT, are superior in terms of $R^2$, compared to EUV, and related machine categories NXE and EXE. We showed that EUV demand patterns are more often intermittent, which could by why the $R^2$ was higher for DUV. We argued that DUV is less intermittent because of the maturity of its products. Furthermore, we found that the setup date was negatively correlated to $R^2$, which supported this. Surprisingly, the setup is positively correlated to the $R^2$, for customer forecast data, which could suggest that the customer can better anticipate its behaviour for new products. For the average price and number of complaints we found that the data was skewed, which raised questions about the validity of the results. When studying the best performing models, it was found that DUV is more dominant than EUV, which supports earlier findings. We concluded that, the correlations that were found during this analysis, can also be the results of unknown, unavailable and external factors.

# Chapter 6

# Conclusions

In this chapter we will present the conclusions which we have drawn from the preceding chapters. First, we will revisit the research questions, and formulate the corresponding answers. Secondly, we will present some recommendations for the management of KMWE. And thirdly, we will discuss the limitations of the research and directions for future research.

## 6.1 Research Questions

In this section we will revisit the research questions and formulate the associated findings. Firstly, we investigated the role of customer forecast data in demand forecasting, which was formulated by the following research questions:

*RQ-1a. Does adding customer forecast data to a forecasting model improve the predictive performance of the model?*

*RQ-1b. How well do forecasting models, built on historical demand and customer forecast data, perform compared to direct use of customer forecast data?*

In the fist analysis we have seen that adding customer forecast data to model improved $R^2$ of the models significantly. We argued that the customer forecast data is closely related to the source of the variability in the demand data, as it can be seen as a declaration of intend by the customer. Therefore, we do not find it surprising that the explainability of the variability increased when adding customer forecast data.

Moreover, we did observed a small improvement for MAAPE. First, we argued that, the customer forecast was not very accurate, and instead only followed a similar

pattern as the demand, which would explain why $R^2$ has improved. Yet, we also found that directly using customer forecast data had improved the MAAPE, which invalidated this argument. Interestingly, a model with customer forecast performs worse than the direct use of customer forecast. We argued that the models do not have enough data to learn that, the predictions from the customer forecast are more accurate than predictions based on the historical data. Yet, the amount of data is sufficient for the models to detect that the pattern from the customer forecast, which lead to a more convincing improvement of $R^2$.

Surprisingly, for SIME, the size of the improvement, compared to MAAPE and R2 was much smaller under all situations. If we assume that SIME is an accurate measure for how KMWE would implement a forecasting model in practice, then we can have to conclude that adding customer forecast data, or using it directly would improve the situation of KMWE only slightly. This became even more evident when we compared the direct use of customer forecast, with a simple baseline prediction, and found that the customer forecast was better in terms of $R^2$, but only slightly in terms of SIME, compared to the baseline. We argue that an increase in $R^2$ would not benefit KMWE's situation as explaining variability does not necessarily mean that difference between the prediction and the actual demand is minimized, which is desired by most companies, as this directly affects the inventory and back orders.

In conclusion, adding customer forecast data improves the $R^2$, yet direct use of customer forecast still produces better results, which makes it difficult to justify building forecasting models under these circumstances. Also, compared to a simple baseline, all prediction methods were able to improve SIME scores only slightly, which could mean that, in practice, demand forecasting will not provide the desired results.

Secondly, we studied how aggregating demand data, from monthly to quarterly, would impact the predictive performance of forecasting models with customer forecast data. This was formulated by the following research question:

*RQ-2. Does aggregating demand data, from monthly to quarterly, improve the predictive performance of the demand forecasting models?*

The first observation we made was that two properties of the time series demand data changed: the length of the time series, and the amount of zero demand periods. We argued that a change in performance would possibly be caused by a change in these properties.

When aggregating from monthly to quarterly, we observed a small, but significant decrease in $R^2$ on average. We also observed that the variance over the $R^2$ scores increased , which meant that the number of bad models increased, as well as the number of good models. We argued that this could be explained by a decrease in

test set length, which would be less robust to extremely good or bad predictions. However, we would expect the increase to be equally large for good and bad models, which was not the case. Overall, the number of bad models increased more, which is why overall $R^2$ decreased. So, we argue that the decrease in $R^2$ is caused by a decrease in training data, since the time series length has decreased.

Furthermore, we observed that MAAPE decreased significantly for quarterly forecasts. We argued that this could have been caused by an increase in stability of the underlying data, since we observed that, aggregating to quarterly data, generated an increase in smooth data, as well as a decrease in intermittent data.

Subsequently, we observed that SIME was significantly lower for monthly forecasts, which was explained by looking at the assumptions of SIME. We used an example to illustrate that, because of the timing of deliveries, in practice, a higher precision for a longer time period is not always better than a lower precision for a shorter period of time.

Finally, we found that, also in the quarterly aggregated situation, the SIME of forecasting models is only slightly better than a simple baseline prediction model, which leads us to conclude that it is likely, that in practice, building forecasting models on quarterly data is not desired, for most products.

In conclusion, aggregating from monthly to quarterly, decreases $R^2$ due to a decrease of training data. It also decreases MAAPE, due to an increase in stability of the underlying demand. Furthermore, it increases SIME, which is caused by the nature of the practical assumptions, made by SIME. And overall, in the quarterly aggregated situation, we found a small difference with a simple baseline, which made us question the practical usefulness of demand forecasting.

Thirdly, we investigated which product types would produce the highest predictive performance. This was formulated by the following research question:

*RQ-3a. For which type of products can we produce a demand forecasting model with the highest predictive performance?*

We investigated, for a collection of categorical and continuous product characteristics, how they related to the predictive performance of forecasting models.

We found that components produced lower MAAPE scores than assemblies. We argued that the stability of component orders was better because the customer it less likely to intervene with system generated orders, because the impact on their main operations is smaller. On the other hand, for complex assemblies the customer might be more considerate about placing orders, which could decrease the stability. We also argued that MAAPE favors this stability more than $R^2$, which is more likely to favor models that capture a highly fluctuating demand pattern.

Subsequently, we found that DUV, and related machine categories NXT and XT, are superior in terms of $R^2$, compared to EUV, and machine categories NXE and EXE. We showed that EUV demand patterns are more often intermittent, which could by why the $R^2$ was higher for DUV. We argued that DUV is less intermittent because of the maturity of its products, which could mean that the customer has a better flow in the production of more mature products, which could lead to a decrease in zero demand periods.

Additionally, we found that service products showed significantly higher $R^2$ scores than non-service products. We explained this by the smooth time series of these products, and concluding that they are sold more often because they end up in both new and existing machines. Moreover, we explained that MAAPE did not corroborate the results because the demand was still relatively erratic compared to most other smooth products.

Furthermore, we found that setup date was negatively correlated to $R^2$, which supports the findings for DUV and EUV. Surprisingly, the setup date was found to be positively correlated to the $R^2$, for the direct use customer forecast data, which could suggest that the customer can better anticipate its behaviour for newer products. However, this did not help the performance of our models, even if they were built on the same customer forecast data. So instead, we conclude that the demand for older products is indeed more stable.

For the average price and number of complaints we found that the data was skewed, which raised questions about the validity of the results. However, when we studied the best performing products, it was found that they were all relatively expensive, which supports the results about the average price being negatively correlated with MAAPE. An explanation for this is hard to determine and might be found in unknown factors.

Finally, we found, when looking at the best performing products, that DUV products were most dominant, despite being slightly under-represented in the product set. This was directly in line with earlier finding that suggested that DUV products are better in terms of $R^2$.

In conclusion, we found that components produce better forecasting models than assemblies, in terms of MAAPE, which was explained by a decreased demand stability for more complex and important assemblies. Also, DUV, and related machine categories are better than EUV, and related machines, in terms of $R^2$, which is explained by less intermittent demand data. Additionally, we saw that a few underrepresented service products produce significantly higher $R^2$ scores. Subsequently, we found some weak evidence that the average price is negatively correlated with MAAPE, which was increased in strength by finding among the best performing products. Finally, we found that DUV was dominant among the best performing products, which supported earlier findings.

## 6.2 Recommendations

Based on the findings of this research, some recommendations for KMWE are formulated.

As we have seen, demand forecasting did, in many situations, only slightly beat a simple baseline prediction, when considering the outcome of inventory simulation scores. This raises the question whether demand forecasting will help KMWE reach the desired business goals. We strongly believe that, the sources of variability behind the demand are very unpredictable, and are driven by complex dynamic relations between KMWE's customers and their suppliers. Consequently, we believe even the most intricate forecasting methods will struggle to produce satisfactory results. Moreover, KMWE operates in a highly volatile industry, which is constantly growing, but is also sensitive to changes in its environment. Global pandemics, or shortages in raw materials can have major impacts in the industry, which can lead to unpredictable fluctuations in demand patterns.

Nonetheless, we make some suggestions that could improve the predictive performance of demand forecasting models. Firstly, we propose that KMWE tries to improve existing, or discover additional, predictors for demand. Currently, KMWE receives forecast data from some of its customers. Customer forecasts are declarations of intend, and should therefore be good predictors for demand. Yet, in practice, we have seen that customer forecast information is not always reliable, especially on a monthly time aggregation. We suggest that KMWE presents these findings to their customers, which could convince them to improve the reliability of their own forecasts. Additionally, KMWE can try to discover new predictors. These could be found externally, either at the customer or in publicly available information. Previously, we argued that the variability in demand is, most likely, the result of complex dynamic relations between customers and suppliers. Therefore, additional predictors might be found in the supply chain. An example of a good predictor in the supply chain could be, the demand at other suppliers of the same customer.

Alternatively, KMWE can discover other means for addressing the delivery performance, and the stability in the production process. Increasing inventory levels will allow the process to maintain a more stable flow, as shortages and overages will not directly lead to an increase or decrease on the production demand. Also, increasing inventory should improve the delivery performance, since products can be shipped directly from stock. As for the customer specific products, we suggest that KMWE increases its inventory of semi-finished parts, which are not yet constrained to a single customer. Furthermore, we encourage KMWE to add products to the SMI project, in which KMWE collaborates with its customers to share inventory risk and costs.

Finally, we propose that KMWE attempts to improve efficiency in the production process. Improvement of product lead times would allow KMWE to react more

quickly to changes in demand. Not only would this directly improve its delivery performance, it would also allow KMWE to prepare a more stable production schedule.

## 6.3 Limitations & Future Research

In this section we will discuss some of the limitations of this research, and some suggestions for future research.

Based on the availability of information, a subset of products was selected to be used for the analyses. In order to make general statements that would apply to the complete product portfolio of KMWE, we would need to select the products randomly. Instead, we were bound to the information that was available. Moreover, the products and product characteristics were all focused on the customer ASML. Again, in order to make statements about the complete product portfolio, we would need to select products from all customers, and use only generally applicable product characteristics. Also, because of the restriction that was imposed by the availability of information, the subset of products was small, which makes generalizing statements weaker. This also meant that some product characteristics were generally over-represented in the set, which made it more difficult to make a fair comparison.

Furthermore, the historical demand data that was used to build the models, came from a period with some disturbances. During this periods, KMWE moved the location of its warehouse, which could have affected the ship date of certain products. Also, the global covid pandemic caused disturbances in the demand for some products. During the selection of products, an attempt was made to filter out the products that were affected most by these disturbances, yet, we can not be certain that the remaining products did not show unusual demand patterns.

Another limitations comes from the fact that we used the shipping date of the products to represent the demand of the customer. The shipping date is not always equal to what the customer desires. For example, when products are delivered late, the shipping date does not represent the demand correctly. The reason behind choosing the shipping date was simply because the actual demand of the customer was not available in the systems of KMWE.

When we consider the SIME metric, there have been some key assumptions that affected it's behaviour. Firstly, we made assumptions about delivery intervals for monthly and quarterly aggregation. Although we argue that the assumption are realistic, it could still be the case that KMWE chooses to setup different delivery intervals, which would make the results from SIME less representative of reality. Secondly, the costs of inventory and back orders are difficult to determine, which is why we assumed them to be equal. This leaves a gap for potential improvement, because implementing more realistic cost values, would improve the power of the

SIME metric for KMWE.

Next, we will propose some directions for future research. SIME is inventory simulation metric that can measure the practical impact of a demand forecast on the inventory management of a company. Currently SIME is scale dependent, which means that comparing predictions for products with different scales can not be done. Making SIME scale independent would increase its usefulness, as it would be able to make fair comparisons between products. Additionally, SIME could be turned into an objective function for a machine learning algorithm. The instructions of the model then becomes to minimize SIME, which could lead to forecast that are better suited for the situation of KMWE.

In this research we have studied product-level predictions models. Instead, future work could investigate the aggregation of products with similar resource requirements. So, instead of predicting products individually, it could be interesting to study how predicting a group of products can help KMWE. Preferably, these products would have similar requirements, in terms of machine capacity, supplies, or other resources, as this could increase the practical usefulness of the predictions.

# Bibliography


Ahmed, N. K., Atiya, A. F., Gayar, N. E., and El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric reviews*, 29(5-6):594–621.

Alon, I., Qi, M., and Sadowski, R. J. (2001). Forecasting aggregate retail sales:: a comparison of artificial neural networks and traditional methods. *Journal of retailing and consumer services*, 8(3):147–156.

Altmann, A., Toloşi, L., Sander, O., and Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347.

Archer, K. J. and Kimes, R. V. (2008). Empirical characterization of random forest variable importance measures. *Computational statistics & data analysis*, 52(4):2249–2260.

Armstrong, J. S. and Collopy, F. (1992). Error measures for generalizing about forecasting methods: Empirical comparisons. *International journal of forecasting*, 8(1):69–80.

Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1):105–139.

Bontempi, G., Ben Taieb, S., and Borgne, Y.-A. L. (2012). Machine learning strategies for time series forecasting. In *European business intelligence summer school*, pages 62–77. Springer.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Brownlee, J. (2017). *Introduction to time series forecasting with python: how to prepare data and develop models to predict the future*. Machine Learning Mastery.

Cerqueira, V., Torgo, L., and Soares, C. (2019). Machine learning vs statistical methods for time series forecasting: Size matters. *arXiv preprint arXiv:1909.13316*.

Chae, Y. T., Horesh, R., Hwang, Y., and Lee, Y. M. (2016). Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings. *Energy and Buildings*, 111:184–194.

Chatfield, C. (2000). *Time-series forecasting*. Chapman and Hall/CRC.

Chicco, D., Warrens, M. J., and Jurman, G. (2021). The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *PeerJ Computer Science*, 7:e623.

Costantino, F., Di Gravio, G., Patriarca, R., and Petrella, L. (2018). Spare parts management for irregular demand items. *Omega*, 81:57–66.

Cryer, J. D. (1986). *Time series analysis*, volume 286. Springer.

Dabou, R. T., Kamwa, I., Chung, C., and Mugombozi, C. F. (2021). Time series-analysis based engineering of high-dimensional wide-area stability indices for machine learning. *IEEE Access*, 9:104927–104939.

Date, Y. and Kikuchi, J. (2018). Application of a deep neural network to metabolomics studies and its performance in determining important variables. *Analytical chemistry*, 90(3):1805–1810.

de Sá, C. R. (2019). Variance-based feature importance in neural networks. In *International Conference on Discovery Science*, pages 306–315. Springer.

Degenhardt, F., Seifert, S., and Szymczak, S. (2019). Evaluation of variable selection methods for random forests and omics data sets. *Briefings in bioinformatics*, 20(2):492–503.

Domingos, P. (1996). Using partitioning to speed up specific-to-general rule induction. In *Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models*, pages 29–34. Citeseer.

Draper, N. R. and Smith, H. (1998). *Applied regression analysis*, volume 326. John Wiley & Sons.

Eaves, A. H. C. (2002). *Forecasting for the ordering and stock-holding of consumable spare parts*. PhD thesis, Lancaster University.

Gardner, E. S. (1990). Evaluating forecast performance in an inventory control system. *Management Science*, 36(4):490–499.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L. A. (2008). *Feature extraction: foundations and applications*, volume 207. Springer.

Hansun, S. and Kristanda, M. B. (2017). Performance analysis of conventional moving average methods in forex forecasting. In *2017 International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS)*, pages 11–17. IEEE.

Hill, T., O'Connor, M., and Remus, W. (1996). Neural network models for time series forecasts. *Management science*, 42(7):1082–1092.

Hipel, K. W. and McLeod, A. I. (1994). *Time series modelling of water resources and environmental systems*. Elsevier.

Hyndman, R. J. (2020). A brief history of forecasting competitions. *International Journal of Forecasting*, 36(1):7–14.

Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Khashei, M. and Bijari, M. (2011). A novel hybridization of artificial neural networks and arima models for time series forecasting. *Applied soft computing*, 11(2):2664–2675.

Kim, S. and Kim, H. (2016). A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, 32(3):669–679.

Kim, T. K. (2015). T test as a parametric statistic. *Korean journal of anesthesiology*, 68(6):540.

Koprinska, I., Rana, M., and Agelidis, V. G. (2015). Correlation and instance based feature selection for electricity load forecasting. *Knowledge-Based Systems*, 82:29–40.

Kourentzes, N. (2013). Intermittent demand forecasts with neural networks. *International Journal of Production Economics*, 143(1):198–206.

Kourentzes, N. (2014). On intermittent demand model optimisation and selection. *International Journal of Production Economics*, 156:180–190.

Kourentzes, N., Trapero, J. R., and Barrow, D. K. (2020). Optimising forecasting models for inventory planning. *International Journal of Production Economics*, 225:107597.

Kumar, M. and Thenmozhi, M. (2006). Forecasting stock index movement: A comparison of support vector machines and random forest. In *Indian institute of capital markets 9th capital markets conference paper*.

Kursa, M. B., Jankowski, A., and Rudnicki, W. R. (2010). Boruta–a system for feature selection. *Fundamenta Informaticae*, 101(4):271–285.

Lim, B. and Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209.

Loecher, M. (2020). From unbiased mdi feature importance to explainable ai for trees. *arXiv preprint arXiv:2003.12043*.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777.

Maepa, F., Smith, R. S., and Tessema, A. (2020). Support vector machine and artificial neural network modelling of orogenic gold prospectivity mapping in the swayze greenstone belt, ontario, canada. *Ore Geology Reviews*, page 103968.

Mahoney, R. M. (1997). *High-mix low-volume manufacturing*. Prentice Hall.

Maimon, O. Z. and Rokach, L. (2014). *Data mining with decision trees: theory and applications*, volume 81. World scientific.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3):e0194889.

Man, X. and Chan, E. P. (2021). The best way to select features? comparing mda, lime, and shap. *The Journal of Financial Data Science*, 3(1):127–139.

Mann, H. B. and Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60.

Maverick, L. A. (1945). *Time Series Analysis: Smoothing by Stages*. Paul Anderson Company.

Meese, R. A. and Rogoff, K. (1983). Empirical exchange rate models of the seventies: Do they fit out of sample? *Journal of international economics*, 14(1-2):3–24.

Menze, B. H., Kelm, B. M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W., and Hamprecht, F. A. (2009). A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC bioinformatics*, 10(1):1–16.

Messager, M. L., Lehner, B., Cockburn, C., Lamouroux, N., Pella, H., Snelder, T., Tockner, K., Trautmann, T., Watt, C., and Datry, T. (2021). Global prevalence of non-perennial rivers and streams. *Nature*, 594(7863):391–397.

Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

Montgomery, D. C., Jennings, C. L., and Kulahci, M. (2015). *Introduction to time series analysis and forecasting*. John Wiley & Sons.

Mussumeci, E. and Coelho, F. C. (2020). Large-scale multivariate forecasting models for dengue-lstm versus random forest regression. *Spatial and Spatio-temporal Epidemiology*, 35:100372.

Nembrini, S., König, I. R., and Wright, M. N. (2018). The revival of the gini importance? *Bioinformatics*, 34(21):3711–3718.

Pearson, K. (1896). Vii. mathematical contributions to the theory of evolution.—iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, (187):253–318.

Pearson, K. (1900). X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175.

Petneházi, G. (2019). Recurrent neural networks for time series forecasting. *arXiv preprint arXiv:1901.00069*.

Pirooznia, M., Yang, J. Y., Yang, M. Q., and Deng, Y. (2008). A comparative study of different machine learning methods on microarray gene expression data. *BMC genomics*, 9(1):1–13.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45.

Probst, P., Wright, M. N., and Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, 9(3):e1301.

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

Quinlan, J. R. et al. (1996). Bagging, boosting, and c4. 5. In *Aaai/Iaai, vol. 1*, pages 725–730.

Robnik-Šikonja, M. and Kononenko, I. (1997). An adaptation of relief for attribute estimation in regression. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, volume 5, pages 296–304.

Rokach, L. and Maimon, O. (2005). Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(4):476–487.

Sharda, R. and Patil, R. B. (1992). Connectionist approach to time series prediction: an empirical test. *Journal of Intelligent Manufacturing*, 3(5):317–323.

Skurichina, M. and Duin, R. P. (2002). Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5(2):121–135.

Steyerberg, E. W., Vickers, A. J., Cook, N. R., Gerds, T., Gonen, M., Obuchowski, N., Pencina, M. J., and Kattan, M. W. (2010). Assessing the performance of prediction models: a framework for some traditional and novel measures. *Epidemiology (Cambridge, Mass.)*, 21(1):128.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., and Zeileis, A. (2008). Conditional variable importance for random forests. *BMC bioinformatics*, 9(1):1–11.

Strobl, C., Boulesteix, A.-L., Zeileis, A., and Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics*, 8(1):1–21.

Student (1908). The probable error of a mean. *Biometrika*, pages 1–25.

Syntetos, A. A. and Boylan, J. E. (2005). The accuracy of intermittent demand estimates. *International Journal of forecasting*, 21(2):303–314.

Syntetos, A. A. and Boylan, J. E. (2006). On the stock control performance of intermittent demand estimators. *International Journal of Production Economics*, 103(1):36–47.

Syntetos, A. A., Boylan, J. E., and Croston, J. (2005). On the categorization of demand patterns. *Journal of the operational research society*, 56(5):495–503.

van de Velde, A. (2021). Data-driven Reliability Calculations of Demand Forecast Data. Master's thesis, Technische Universiteit Eindhoven, the Netherlands.

Vincenten, R. (2021). The feasibility of flow production in in a high-mix, high-complexity, and low-volume environment. Master's thesis, Technische Universiteit Eindhoven, the Netherlands.

Wadoux, A. M.-C., Brus, D. J., and Heuvelink, G. B. (2019). Sampling design optimization for soil mapping with random forest. *Geoderma*, 355:113913.

Wei, P., Lu, Z., and Song, J. (2015). Variable importance analysis: a comprehensive review. *Reliability Engineering & System Safety*, 142:399–432.

White, M., Wen, J., Bowling, M., and Schuurmans, D. (2015). Optimal estimation of multivariate arma models. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Williams, T. M. (1984). Stock control with sporadic and slow-moving demand. *Journal of the Operational Research Society*, 35(10):939–948.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., et al. (2008). Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37.

Xia, Z., Stewart, K., and Fan, J. (2021). Incorporating space and time into random forest models for analyzing geospatial patterns of drug-related crime incidents in a major us metropolitan area. *Computers, Environment and Urban Systems*, 87:101599.

Zhu, Y. and Zhou, G. (2009). Technical analysis: An asset allocation perspective on the use of moving averages. *Journal of financial economics*, 92(3):519–544.

Zien, A., Krämer, N., Sonnenburg, S., and Rätsch, G. (2009). The feature importance ranking measure. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 694–709. Springer.

# Appendices

## Appendix A

Table 6.1: Default hyperparameters random forest regression

| **Parameters** | **Default value** |
|---|---|
| n_estimators | 100 |
| criterion | gini |
| max_depth | None |
| min_samples_split | 2 |
| min_samples_leaf | 1 |
| min_weight_fraction_leaf | 0.0 |
| max_features | auto |
| max_leaf_nodes | None |
| min_impurity_decrease | 0.0 |
| bootstrap | True |
| oob_score | False |
| n_jobs | None |
| random_state | None |
| verbose | 0 |
| warm_start | False |
| class_weight | None |
| ccp_alpha | 0.0 |
| max_samples | None |