Albin Lindfors

Demand Forecasting in Retail: A Comparison of Time Series Analysis and Machine Learning Models

Master' Thesis in Information Systems

Supervisor: Xiaolu Wang

Faculty of Social Sciences, Business and Economics

Åbo Akademi University

Åbo 2021

Abstract

Subject: Information Systems

Writer: Albin Lindfors

Title: The Forecasting Performance of Time Series Analysis and Machine Learning Models in the Retail Industry

Supervisor: Xiaolu Wang

Abstract:

Having an accurate forecast of the upcoming demand is of utmost importance to a retail company, as it helps the retailer plan the day-to-day activities and optimize the supply chain. At the same time, retailers also gather a substantial amount of data about everything from weather conditions to promotional campaigns, having the potential to improve the forecasts when used right. The aim of this thesis is to compare time series analysis, which only utilize the past sales data, to machine learning models, which can also utilize other data, when forecasting retail sales figures. The comparison is conducted by a thorough literature review and an empirical study where forecasting is performed on a set of Walmart sales data. The forecasting methods used in the empirical study are ARIMA, Holt-Winter's exponential smoothing, linear regression, decision tree and artificial neural networks, and the results of the empirical study suggest that ARIMA and Holt-Winter's exponential smoothing are the best performing models on this particular dataset.

Keywords: Forecasting, Machine learning, ARIMA, Exponential smoothing, Moving average, Linear regression, Decision tree, Artificial neural networks, Retail, Supply chain

Date: 21.6.2021 | **Number of pages:** 114

Contents

1. Introduction/background 1
1.1. Introduction 1
1.2. Objective 3
1.3. Method 4
1.4. Structure of thesis 4
2. Literature review 5
2.1. Time series analysis 5
2.1.1. Time series patterns 6
2.1.2. Forecast accuracy 8
2.2. Exponential smoothing 9
2.2.1. Single exponential smoothing 9
2.2.2. Double exponential smoothing 12
2.2.3. Triple exponential smoothing 14
2.2.4. Exponential smoothing as a forecasting method 16
2.3. Moving average 19
2.3.1. Simple moving average 19
2.3.2. Weighted moving average 21
2.3.3. Moving average as a forecasting method 23
2.4. ARIMA 25
2.4.1. Stationarity and differencing 26
2.4.1.1. Is differencing needed? 29
2.4.2. Autoregressive models 30
2.4.3. Moving average models 30
2.4.4. The ARIMA model 31
2.4.4.1. Seasonal ARIMA model 32
2.4.5. ARIMA as a forecasting method 33

2.5. Time series regression 35
2.5.1. Simple regression analysis 36
2.5.2. Multiple Regression Analysis 39
2.5.3. Linear regression as a forecasting method 41
2.6. Decision trees 44
2.6.1. The structure of a decision tree 44
2.6.2. Decision tree as a forecasting method 48
2.7. Artificial neural networks 51
2.7.1. Structure of an artificial neural network 51
2.7.2. How an artificial neural network learns 56
2.7.3. Artificial neural networks as a forecasting method 57
2.7.3.1. Early forecasting use of artificial neural networks 57
2.7.3.2. Deep belief networks 60
2.7.3.3. Artificial neural networks as a retail sales forecasting method 62
3. Empirical study 65
3.1. Method 65
3.2. Data overview 66
3.2.1. Data manipulation 68
3.2.2. Data exploration 72
3.3. Forecasting 79
3.4. Results 82
4. Discussion 88
4.1. Empirical study discussion 88
4.2. Research questions 91
4.3. Future research 93
5. Svensk sammanfattning 94
5.1. Introduktion 94

5.2. Litteraturöversikt 95
5.3. Empirisk studie 97
5.4. Resultat 98
References.....100

1. Introduction/background

1.1. Introduction

For a retail company, there are two main factors thar affect its profitability: a positive customer base of a sufficient size and a healthy cost structure. Maintaining the customer base requires knowing the customers, whereas maintaining the cost structure is more complicated. The latter involves everything from planning the upcoming shifts so that neither over- nor understaffing occurs, to maintaining a proper inventory level to ensure that the stock is not depleted at the same time as trying to avoid overstocking, and hence prevent unnecessary costs. Most of the factors affecting the cost structure of a retail company have one common underlier – the demand.

Knowing, or at least having an idea of, the upcoming demand for the products being sold has many benefits. For example, it has a significant positive impact on planning the upcoming shifts, mitigating the situations where over- or understaffing occurs (Defraeye and Van Nieuwenhuyse, 2016), it facilitates the supply chain maintenance, avoiding as much under- or overstocking as possible (Carbonneau, Laframboise and Vahidov, 2008), and it also assists in making the sales plan (Ma, Fildes and Huang, 2016). For example, when running a grocery store, knowing that the weeks before Christmas tend to be one of the busiest times of the year and the sales of ham is substantial already helps. In addition to this, knowing approximately how much ham is expected to be sold during Christmas mitigates the risk of under- or overstocking, which in turn reduces the number of unhappy customers or unnecessary costs in terms of personnel or inventory costs. Knowing that Christmas is one of the busiest periods of the year is already the result of a demand forecast itself, but a more accurate demand forecast is often desired. The fact that companies around the world spend billions of dollars yearly on consulting fees, software and personnel in order to attain accurate demand forecasts (Aiyer and Ledesma, 2004) is a clear sign that reliable demand forecasts are highly desired.

One of the main incentives behind the high demand for the most accurate demand forecasting techniques is the effect they have on inventory planning. According to Tuovila (2020), the cost of carrying inventory is 20% to 30% of the value of the total inventory. Even though the buy-in price of a product is most often cheaper when bought in bulk (Mueller, 2019), the actual price of the product for the retailer could eventually become higher than the non-bulk price, if there is not enough demand for the product and the inventory carrying cost is considered. Kot, Grondys and Szopa (2011) found that high error in demand forecasting combined with poor communication in the supply chain led to increased costs because of increased inventory levels, and Watson (1987) showed that demand forecast variations increased the annual inventory carrying costs. Even though the major costs related to inventory errors come from overstocking, it is also important to note the disadvantages of understocking. As Beutel & Minner (2012) showed, having a shortage in stock would ultimately lead to customer dissatisfaction, which in turn could lead to loss of customers, reputation damage, loss of profits, etc.

A focal point of the research on demand forecasting in the past decades has been the seasonal variation that different industries face. Many time series analysis models have been developed, including time series regressions, time series decomposition, exponential smoothing and the autoregressive and integrated moving average (Chu & Zhang, 2003). Although these models work – at least to some degree – they have their flaws, mainly because they are all linear models. Hence, they are very user-reliant and depend on the user to specify the model form in order to work properly. In many cases, the user does not have the necessary knowledge of the relationships in the data, meaning that the specified model form will be inaccurate, or at least not as accurate as desired. (Chu & Zhang, 2003). To tackle the problem of the high user dependency of the time series analysis models, many machine learning models have been developed over time, most notably the decision tree and the artificial neural networks (ANN). An ANN is a model designed to follow the decision-making process of a human brain and it can be used for many different types of complex problem solving, out of which forecasting is only one application (Nielsen, 2015). While the structure and design of an ANN can be highly complex, the major advantage of an ANN as a forecasting method is that it does not need any human input other than a training dataset in order to be able to generate forecasts. The ANN is

automatically able to determine the relationships in the data and, therefore, eliminates the problem of the time series analysis (Nielsen, 2015).

Even though some research has been conducted regarding the differences between time series analysis and machine learning models in the retail industry, much of the research on the topic has focused on other industries. In addition to this, the fact that forecasting methods are highly case sensitive warrants further research on the topic of demand forecasting in the retail industry. Thus, this thesis will aim to clarify the differences between the most common time series analysis and machine learning models through a thorough literature review. It will also apply selected time series analysis models as well as an ANN and a decision tree model to a set of retail sales data in order to determine if there is a substantial difference in the performance of the models.

1.2. Objective

Given the lack of previous research regarding the differences between time series analysis and machine learning models for forecasting in the retail industry, the objective of this study is to provide a thorough comparison between them.

In order to address this research objective, the thesis attempts to answer the following two research questions:

1. *Do forecasting methods utilizing machine learning techniques perform better than the time series analysis models?*

2. *Which forecasting method gives the best performance in sales forecasting?*

1.3. Method

To answer the first research question, a literature review treating the most popular forecasting methods and an empirical study of sales forecasting in the retail industry will be conducted. The literature review will be conducted first as it will be used to select the appropriate forecasting models to be used in the empirical study. The empirical study will then be conducted by applying the chosen forecasting models to a dataset of the sales data of Walmart, which has been acquired from the machine learning community Kaggle. The forecasting models will be developed using various software and predesigned models for Python, as it makes it possible to include more models in the comparison due to less programming skills being required from the author.

The second research question will be answered by examining the literature review and the results of the empirical study. The forecast accuracy metrics of the empirical study will be compared to each other, whereafter an answer will be suggested based on the findings of the literature review and the empirical study.

1.4. Structure of thesis

The upcoming work of the thesis will be structured as follows. First there will be a literature review focusing on time series analysis models and some alternative, more progressive machine learning models. The results of the literature review will be analyzed, and the time series analysis models will be compared to the alternative models. After this, a few models will be chosen and applied to a dataset in Python to compare their real-world performance in order to find an answer to the research questions. The methods will be explained and the results will be discussed, and finally further research opportunities will be defined.

2. Literature review

This chapter will provide a thorough literature review of the most common time series analysis models, as well as some more advanced machine learning models. The time series analysis models to be discussed are the moving average, the exponential smoothing and the ARIMA model, and the machine learning models covered will be the linear regression, the artificial neural network and the decision tree. While linear regression is a form of time series analysis, it utilizes machine learning and will therefore be considered a machine learning model in this thesis. We will first go through the previously mentioned time series analysis models, after which we will examine the machine learning models. All sections will follow the same structure – the models will first be presented and explained, whereafter the previous research on the models will be presented.

2.1. Time series analysis

In order to produce accurate forecasts, it is crucial to have a complete understanding of the time series data that the forecast is based on. Different forecasting methods are best suited for different types of data and, therefore, it is of utmost importance to know the characteristics of the time series in order to be able to choose the most appropriate forecasting method. The most important characteristics to know about a time series when choosing a forecasting method are the patterns, such as seasonality, and the trend of the time series. In this section, the different patterns of a time series will be discussed, and some forecasting methods suited for each pattern will be mentioned. Anderson, Sweeney and Williams (2011) describe time series analysis as follows:

"The objective of time series analysis is to discover a pattern in the historical data or time series and then extrapolate the pattern into the future; the forecast is based solely on past values of the variable and/or on past forecast errors."

2.1.1. Time series patterns

A time series can have many kinds of patterns and in order to attain a basic understanding of the patterns, Anderson, Sweeney and Williams (2011) suggest that a simple plot of the time series should be constructed. The graphical presentation of the time series can often be enough for the user to identify that the data exhibits a clear pattern. However, in order to fully understand the characteristics of the pattern, more advanced methods than a time series plot might needed. The pattern of a time series is an important part of understanding the past behavior of the time series, and if this behavior can be expected to continue in the future, the pattern can be used to help selecting a proper forecasting method. (Anderson, Sweeney and Williams, 2011)

Some of the most common patterns of a time series are horizontal patterns, trend patterns, seasonal patterns, and cyclical patterns (Anderson, Sweeney and Williams, 2011). A horizontal pattern is a pattern that moves around the mean of the time series and it is always present in a stationary time series (discussed further in Section 2.4.1). Even though a horizontal pattern moves around the mean of the time series, the level of the pattern can change in some cases, for example if new distribution contracts that increase the sales of a product are signed. In these cases, we can observe a quick rise of the level of the trend whereafter the trend continues as horizontal. (Anderson, Sweeney and Williams, 2011)

One of the most typical patterns of a time series is the trend pattern, or just simply the trend of a time series. A trend is when a time series constantly moves towards lower or higher values during a long period of time. The trend pattern does not exclude the existence of other patterns, such as seasonality, in the time series, but trend and horizontal

patterns cannot exist at the same time, as they are mutually exclusive. The reason we have trends are usually long-term factors such as population growth or decreases, or change of consumer preferences. (Anderson, Sweeney and Williams, 2011)

The seasonal pattern and the cyclical pattern have similar characteristics but are still notably different from each other. A seasonal pattern is recognized by observing the same pattern in the time series being repeated every season. For example, the sales of ice skates would probably be expected to increase every winter before decreasing to a lower level as we approach summer, showing the characteristics of a seasonal pattern. Even though the seasonal pattern might be thought to occur yearly, it can also be seen in shorter periods, or seasons, such as months or even days when the time series represents data with shorter seasons. A cyclical pattern on the other hand is a pattern in which the observations alternate between points below and points above the trend line for longer than a year. Cyclical patterns are often observed in economic time series as a result of multiyear business cycles caused by periods of moderate inflation followed by periods of rapid inflation. Anderson, Sweeney and Williams (2011) emphasize that due to the high difficulty of forecasting business cycles, the cyclical effects are often combined with the long-term trend. (Anderson, Sweeney and Williams, 2011)

When the pattern of the time series has been identified it is of utmost importance to select the appropriate forecasting method or otherwise the accuracy of the forecast will most likely suffer. When there is a horizontal pattern and the time series is stationary, an average of past observations may be used. If there is a horizontal pattern but the time series is not stationary because of changes in the level of the pattern, the naïve method, which uses the last observation as the forecast for the next observation, may be used. The simple moving average, weighted moving average and exponential smoothing are other methods suited for a time series with a horizontal pattern. If we have a trend pattern, on the other hand, linear regression or Holt's linear smoothing will most likely be the most suitable forecasting models. If we have a season

al pattern, then a multiple regression model with dummy variables for each season could be used, and if we have both seasonality and trend, a trend variable might be added to the regression model. (Anderson, Sweeney and Williams, 2011)

2.1.2. Forecast accuracy

As there are usually many different models suitable for capturing the pattern in the time series, the accuracy of the forecasts can be measured in order to decide which method to use. Forecast accuracy is measured by running the forecasting method on data that already exists, and then measure how close to the actual observations the forecasts are. When this is done, the method with the lowest forecast error can then be chosen to forecast the upcoming observations. (Anderson, Sweeney and Williams, 2011)

Anderson, Sweeney and Williams (2011) define a forecasting error by the following formula:

$$\text{Forecast Error} = \text{Actual Value} - \text{Forecast} \tag{2.1}$$

There are many ways of measuring the forecast error of a time series forecast but some of the most common methods are the mean absolute error (MAE), the mean squared error (MSE) and the mean absolute percentage error (MAPE):

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{2.2}$$

$$MSE = \frac{\sum_{i=1}^{n} (y_i - x_i)^2}{n} \tag{2.3}$$

$$MAPE = \frac{\sum_{i=1}^{n} |(y_i - x_i)/y_i|}{n} \quad , \tag{2.4}$$

where y_i is the predicted value at time i, x_i is the predicted value at time i, and n is the number of observations.

The MAE is the average of the absolute value of all the forecast errors whereas the MSE is the average of the squared forecast errors, both eliminating the problem of positive and negative forecast errors offsetting each other. While both methods give an easily

interpretable forecast error, the scale of the data highly affects the result. To solve this problem, we have the MAPE which is similar to the MAE but, instead, measured in percentage. To calculate the MAPE we must first measure the absolute percentage error for each forecast, whereafter we calculate the average of all absolute percentage errors. According to Lewis (1997), if we get a MAPE below 10% it can be considered a good forecasting error and a sign that the forecasting method we use is fairly accurate. (Anderson, Sweeney and Williams, 2011)

2.2. Exponential smoothing

Exponential smoothing was introduced in the late 1950s and early 1960s by Holt (1957, reprinted in 2004), Brown (1959) and Winters (1960) and has since been used as a base for many successful forecasting methods. The exponential smoothing method generates forecasts that are weighted averages which use exponential weighing, giving the most recent observations more weight. (Hyndman & Athanasopoulos, 2021)

In this section we will first present the three most important variations of the exponential smoothing method, and then look at the research treating exponential smoothing as a forecasting method in the retail industry.

2.2.1. Single exponential smoothing

The single exponential smoothing, or simple exponential smoothing as Hyndman and Athanasopoulos (2021) called it, is the most basic form of exponential smoothing. The method is used when there is no clear seasonal pattern or trend in the dataset, and when the naïve and average methods are unable to produce forecasts with a sufficient

accuracy. The naïve method assumes that the future forecast equals the last observed point in the series, whereas the average method assumes that the future forecast equals a simple average of the observed data. The single exponential smoothing on the other hand, can be seen as a middleman between these two models. It calculates the forecast using a weighted average, where the weights are exponentially distributed, giving a higher weight to the observation the newer the observation is. To compare it, the naïve method could be seen as a weighted average giving the latest observation a weight of 1 and all the other observations a weight of 0. (Hyndman & Athanasopoulos, 2021)

The Naïve method, Simple Average method and Single Exponential Smoothing method are given as follows:

Naïve

$$\hat{y}_{T+h|T} = y_T, \text{ for h = 1, 2, …} \qquad (2.5)$$

Simple average

$$\hat{y}_{T+h|T} = \frac{1}{T}\sum_{t=1}^{T} y_t, \text{ for h = 1, 2, …} \qquad (2.6)$$

Single exponential smoothing

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 Y_{T-2} + \cdots \qquad (2.7)$$

where $0 \leq \alpha \leq 1$ is the smoothing parameter.

As the smoothing parameter is a constant value between one and zero the model above clearly shows how the smoothing parameter controls the rate at which the weight

exponentially decreases by time. Table 2.1 below shows the weight of the six previous observations given four different smoothing parameters.

Table 2.1

	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
y_T	0.2000	0.4000	0.6000	0.8000
y_{T-1}	0.1600	0.2400	0.2400	0.1600
y_{T-2}	0.1280	0.1440	0.0960	0.0320
y_{T-3}	0.1024	0.0864	0.0384	0.0064
y_{T-4}	0.0819	0.0518	0.0154	0.0013
y_{T-5}	0.0655	0.0311	0.0061	0.0003

Hyndman & Athanasopoulos (2021), *Forecasting: Principles and Practice, Chapter 8.*

As can be seen here, the higher the smoothing parameter is, the more weight is given to recent observations and the rate at which the weight decrease is higher. If we go to the extremes where α = 1 and α = 0, we get forecasts equaling the ones of the naïve and average method. Also, the more weight is given to recent observations, the faster the model is in reacting to changes.

When using a single exponential smoothing method there are two problems that need to be solved before the model can be used. A starting point – meaning how many past observations are included – and an α value for the calculation must be chosen. If the dataset is relatively small it might be acceptable to include all past observations in the calculations, but as the dataset grows larger it fast becomes unnecessary to include all observations. Due to the exponential smoothing, the weight of the past observations reaches close to zero quickly, and therefore, it is an unnecessary strain to include too many observations in the calculation as the older observations would carry a weight close to zero. When it comes to selecting a proper value for α, it can in some cases be chosen subjectively by the forecaster, but it is usually feasible to obtain the unknown parameters from the observed dataset. This is done by finding the values that minimize the sum of the

squared residuals, or sum of the square errors (SSE) as it is also called. The formula for SSE is the following and we want to find the values that minimize the outcome

$$SSE = \sum_{t=1}^{T}(y_t - \hat{y}_{t|t-1})^2 = \sum_{t=1}^{T} e_t^2 \ . \qquad (2.8)$$

To solve this minimization problem, an optimization tool such as R must be used, as the problem includes a non-linear minimization problem that cannot be solved by using any formula. (Hyndman & Athanasopoulos, 2021)

2.2.2. Double exponential smoothing

The double exponential smoothing method was developed by Holt (2004) to extend simple exponential smoothing to be able to forecast data with a trend. Thus, the model is also known as Holt's linear trend model. The main difference between the single and double exponential smoothing models, apart from the ability to predict based on data with a trend, is that the double exponential smoothing uses two smoothing factors whereas the single exponential smoothing model only uses one. (Hyndman & Athanasopoulos, 2021). The formulas for double exponential smoothing are given as follows:

Forecast equation: $\hat{y}_{t+h|t} = l_t + hb_t \qquad (2.9)$

Level equation: $l_t = \propto y_t + (1-\propto)(l_{t-1} + b_{t-1}) \qquad (2.10)$

Trend equation: $b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \qquad (2.11)$

where $h = 1, 2, \ldots$, and $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$ are smoothing factors. The forecast equation consists of two separate equations for finding the level and the trend of the equation, respectively, both of which are weighted averages based on past values of the equations. As we see in Formula 2.9 and 2.10, the level equation, l_t, is a weighted average of y_t and a one step ahead forecast for time t. The trend equation, b_t, is also a weighted average based on the difference between the level at time t and t-1, and a one step ahead forecast of the past trend, b_{t-1}. The smoothing factors used in the equations for l_t and b_t are estimated in the same way by minimizing the SSE, as with the single exponential smoothing. (Hyndman & Athanasopoulos, 2021).

Hyndman and Athanasopoulos (2021) describes the core idea of double exponential smoothing as follows:

> *"The forecast function is no longer flat but trending. The h-step-ahead forecast is equal to the last estimated level plus h times the last estimated trend value. Hence the forecasts are a linear function of h."*

The problem with the double exponential smoothing model is that it tends to over-forecast in the long run as the model displays a constant trend (Hyndman & Athanasopoulos, 2021). Due to this, Gardner and McKenzie (1985) developed an improved double exponential smoothing model that includes a damping-parameter which flattens the trend into a flat line in the long run. According to Hyndman and Athanasopoulos (2021), methods which include a damped trend tend to perform much better than methods with a constant trend. The improved double exponential smoothing model from Hyndman and Athanasopoulos (2021) is as follows:

Forecast equation: $$\hat{y}_{t+h|t} = l_t + (\varphi + \varphi^2 + \cdots + \varphi^h)b_t \quad (2.12)$$

Level equation: $$l_t = \propto y_t + (1-\propto)(l_{t-1} + \varphi b_{t-1}) \quad (2.13)$$

Trend equation: $$b_t = \beta(l_t - l_{t-1}) + (1-\beta)\varphi b_{t-1} \quad (2.14)$$

Where $0< \varphi <1$ is the damping parameter, and if the damping parameter would equal 1, the method would be identical to Holt's linear method. The lower the damping parameter is, the faster the model is to dampen the curve and vice versa. In practice, however, the damping parameter is often restricted to a value between 0.8 and 0.98, as a value lower than 0.8 results in a strong damping effect that dampens the curve too fast, and a value over 0.98 results in a damping effect that is nearly impossible to distinguish. (Hyndman & Athanasopoulos, 2021)

2.2.3. Triple exponential smoothing

Holt (2004) and Winters (1960) further developed the double exponential smoothing method so that it could take seasonality into account. The model they developed is the triple exponential smoothing model, or Holt-Winters' seasonal method, and in addition to the two smoothing equations for level and trend in the double exponential smoothing method, it also includes a third equation for seasonal components. Thus, it also includes a third smoothing factor, and the name of the model is explained. (Hyndman & Athanasopoulos, 2021)

There are two different versions of the triple exponential smoothing method, which are used depending on the type of seasonality in the data. If the seasonal variations are constant throughout the series, the additive method is used, and if the seasonal variations are changing proportionally to the level of the series, the multiplicative method is used. The additive and multiplicative method are defined as follows:

Holt-Winters' additive method

Forecast equation: $$\hat{y}_{t+h|t} = l_t + hb_t + s_{t+h-m(k+1)} \quad (2.15)$$

Level equation: $$l_t = \propto (y_t - s_{t-m}) + (1-\propto)(l_{t-1} + b_{t-1}) \quad (2.16)$$

Trend equation: $$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \quad (2.17)$$

Seasonality equation: $$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m} \quad (2.18)$$

Holt-Winters' multiplicative method

Forecast equation: $$\hat{y}_{t+h|t} = (l_t + hb_t)s_{t+h-m(k+1)} \quad (2.19)$$

Level equation: $$l_t = \propto \frac{y_t}{s_{t-m}} + (1-\propto)(l_{t-1} + b_{t-1}) \quad (2.20)$$

Trend equation: $$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \quad (2.21)$$

Seasonality equation: $$s_t = \gamma \frac{y_t}{(l_{t-1}+b_{t-1})} + (1-\gamma)s_{t-m}\ , \quad (2.22)$$

where m is the frequency of the seasonality, i.e., m=12 if we have monthly data, and k is the integer part of (h-1)/m, which is used to ensure that the estimates of the seasonal indexes used comes from the final year of the sample. In the additive method the seasonal component, s_t, will add up to approximately zero within each year, whereas it will sum up to approximately m in the multiplicative method. (Hyndman & Athanasopoulos, 2021)

A damped version of the triple exponential smoothing method can also be applied to both the additive and multiplicative version, and according to Hyndman & Athanasopoulos (2021) it often provides an accurate forecast for the multiplicative version of triple exponential smoothing. The dampening follows the same principles as explained in Section 2.2.2 and, therefore, the formula for triple exponential smoothing with multiplicative seasonality and a damped trend is as follows:

Forecast equation: $$\hat{y}_{t+h|t} = [l_t + (\varphi + \varphi^2 + \cdots + \varphi^h)b_t]s_{t+h-m(k+1)} \quad (2.23)$$

Level equation: $$l_t = \alpha\left({}^{y_t}/_{s_{t-m}}\right) + (1-\alpha)(l_{t-1} + \varphi b_{t-1}) \quad (2.24)$$

Trend equation: $$b_t = \beta(l_t - l_{t-1}) + (1-\beta)\varphi b_{t-1} \quad (2.25)$$

Seasonality equation: $$s_t = \gamma \frac{y_t}{(l_{t-1}+\varphi b_{t-1})} + (1-\gamma)s_{t-m} \quad (2.26)$$

2.2.4. Exponential smoothing as a forecasting method

This section will present research that has been conducted using exponential smoothing as a forecasting method in the retail industry. It is evident that the single exponential smoothing is a rarely used model when it comes to the retail industry and therefore this section will mainly consist of research around the double and triple exponential smoothing models.

While exponential smoothing has been a popular forecasting method in recent years, it has not always had the same popularity. Gardner (2006) describes the reasons behind the lack of popularity in the early days of exponential smoothing as follows:

> *"When Gardner (1985) appeared, many believed that exponential smoothing should be disregarded because it was either a special case of ARIMA modeling or an ad hoc procedure with no statistical rationale. Since 1985, the special case argument has been turned on its head, and today we know that exponential smoothing methods are optimal for a very general class of state-space models that is in fact broader than the ARIMA class."*

Even though we have acknowledged the optimality of exponential smoothing models as stated by Gardner (2006) above, Gardner (2006) also points out that little

progress has been made on the identification and selection of exponential smoothing models. Gardner and McKenzie (1988), Shah (1997) and Meade (2000) have all addressed the subject of method selection, but the results are still to some degree inconclusive. While the aforementioned research may have developed models for selecting the proper exponential smoothing model, Gardner (2006) still believe it is too early and therefore difficult to tell if they have enough qualifications to select the right model. Despite this, Gardner (2006) points out that the damped models almost always beat the base models in forecasting accuracy.

In a study by Jeong (2017) the additive version of the triple exponential smoothing was found to be the best performing exponential smoothing model when predicting the sales of discount stores and department stores. The exponential smoothing methods compared by Jeong (2017) were the double exponential smoothing, the damped double exponential smoothing and the additive and multiplicative triple exponential smoothing method. When forecasting the sales of discount stores, Jeong (2017) chose the additive triple exponential smoothing model, the double exponential smoothing model as well as the damped double exponential smoothing model for comparison based on the characteristics of the sales data. He found that the additive triple exponential smoothing model outperformed the two other models on all measurements. For example, the R-squared was notably higher, and the MAPE and RMSE were notably lower than in the other two models. The additive triple exponential smoothing model also outperformed an ARIMA-model on many measurements, such as the previous three measurements. When forecasting the sales of department stores, Jeong (2017) only compared the additive and multiplicative triple exponential smoothing because of the highly seasonal sales data of the department stores. Once again, the additive model was superior only being outperformed by the multiplicative model on MAE which is used for estimating the worst-case scenario. This time, however, the models performed much closer to each other, meaning that the forecasts would probably be quite similar no matter which model was to be chosen. One criticism on Jeong's work would have to be the fact that he never followed up on his predictions to see how accurate they really were. Even though the results on the test data might be promising it is hard to say how good the actual predictions are without following up on how close they are to the actualized sales. This was also shown by the

findings of the research by Chu and Zhang (2003), in which their regression model had good performance on the test data but significantly worse accuracy when applied to the forecast data.

Another study on exponential smoothing as a forecasting method in the retail industry was made by Alon, Qi and Sadowski (2001), who compared different forecasting models when forecasting US aggregate retail sales. They forecasted the aggregate sales of two periods and performed both a one-step and a multiple-step forecast for the four methods they compared. Even though the triple exponential smoothing was one of the two worst performing models in both periods when comparing the average MAPE for the one-step and multiple-step forecast, it was the only model to perform notably better in the multiple-step forecasts in both periods. In period two it was actually the best performing model in the multiple-step forecasting but the rank was brought down by the bad performance of the one-step forecast. However, even though the results show that the triple exponential smoothing performed better in a multiple-step forecast in both periods, the results are also varying a lot between the two periods. In period one, the MAPE for the one-step forecast was 3.11% and for the multiple-step forecast 2.27%, whereas the same numbers for period two were 2.22% and 1.16%. This was not exclusively the case with triple exponential smoothing but with all the forecasting models compared, indicating that the models are highly case sensitive.

A study by Makatjane and Moroke (2016) compares multiplicative Holt-Winters' triple exponential smoothing to the seasonal ARIMA method when forecasting short term car sales. They, as the previously mentioned studies, use the MAPE, R-squared and the MAE to measure the forecast errors and to choose the optimal model. However, unlike Jeong (2017) and Alon, Qi and Sadowski (2001), Makatjane and Moroke (2016) realized that the error measurements were not enough to draw conclusions about the best model and therefore they also calculated a power test which showed that the triple exponential smoothing model had about 0.3% more predictive power. Despite this, Makatjane and Moroke (2016) did not follow up on the actual sales of the twelve months they predicted with the triple exponential smoothing either, and therefore we once again cannot know the true accuracy of the predictions.

2.3. Moving average

Another traditional way of time-series forecasting is done by calculating the moving average. The moving average, also known as the rolling average, takes a predetermined set of past observations and calculates the average of these observations. As the time moves forward, the oldest observations are dropped and replaced by newer observations, maintaining an always up-to date recent average of the measured development. By doing this, the moving average smooths out the short-term fluctuation in a time series and focuses on finding the trend of the time series (Anderson, Sweeney and Williams, 2011). The moving average is mainly used for predicting the development of financial instruments (Gunasekarage & Power (2011), Metghalchi, Marcucci and Chang (2012) and Fifield, Power and Knipe (2008), amongst others), but it can also be used for predicting retail sales (Winters, 1960) although more developed versions, such as SARIMA, has had greater success (Arunraj, Ahrens and Fernandes, 2016).

Many different versions of the moving average have been developed over the years and in this section, we will present the two most common versions, the simple moving average and the weighted moving average. The more developed models using moving averages, such as ARIMA and SARIMA, deserve their own section and will therefore be discussed later in this thesis.

2.3.1. Simple moving average

The most basic form of the moving average, the simple moving average, is calculated by adding together all observations (e.g., the daily sales of a product or closing price of a stock) for a predetermined number of days and then dividing the sum by the number of observations.

$$SMA = \frac{P_1+P_2+P_3+\cdots+P_n}{n} \quad (2.27)$$

where P_1, …, P_n = the most recent past observation and n = number of observations included in the moving average. When determining the number of observations included in the calculation of the moving average there are no rules on how many observations the user should consider. However, as James (1968) points out,

> *"It is a fact, well known to statisticians, that the variance of an average of sample observations decreases as the number of observations increase."*

In other words, as the number of observations we include in our moving average increase, the variance of the moving average decreases. This means that a moving average consisting of a large number of observations is slower to react to new observations than a moving average consisting of a smaller number of observations (James, 1960). For example, a 30-day moving average reacts faster when the price of a stock rises quickly and adjusts to the short-term trend fast, whereas a 200-day moving average might barely react to the quick price rise of the stock, needing many new observations of a higher price before adjusting the more long-term 200-day trend notably.

When using the simple moving average model for forecasting, the assumption is that the recent average performance is a good predictor of the future performance (Eppen et al., 1993). Also, the key to attaining the most accurate prediction is to select the proper number of past observations to be used in the simple moving average model (Gentry, Wiliamowski and Weatherford, 1995). This is not exclusive to the simple moving average model as the number of historical observations chosen to be used for forecasting could be crucial to other moving average models which are to be discussed later, but it plays a bigger role here as there are no other factors affecting the simple moving average model. As was discussed earlier, the number of observations affects how fast the model reacts to changes, and therefore the user must consider how sensitive they need their model to be.

Usually, the further ahead we want to predict, the less sensitive the model needs to be. For example, Gentry, Wiliamowski and Weatherford (1995) compared the performance of the simple moving average model to neural networks and some other models on a one week ahead and three weeks ahead forecast, and used observations ranging all the way from two up to eight weeks' worth of data in their moving average models, showing that different number of observations are suitable for different applications. Anderson, Sweeney and Williams (2011) also recommend that the best number of observations included in the moving average model should be determined by trial and error, finding the number of observations that minimize the forecasting error.

2.3.2. Weighted moving average

The weighted moving average follows the same principles as the simple moving average with the addition that it gives more weight to the most recent observations, whereas all observations were equally weighted in the simple moving average model. The weights follow an arithmetic sequence and the sum of the weights given to all observations should always be 100%, or 1. (Anderson, Sweeney and Williams, 2011)

The formula for the weighted moving average is as follows:

$$WMA = \frac{(P_1 * n + P_2 * (n-1) + P_3 * (n-2) + \cdots + P_n}{((n * (n+1))/2)} \quad (2.28)$$

where P_1, …, P_n = the most recent past observations and n = number of observations included in the moving average.

Let us look at the following example to make it clearer.

Table 2.2 Fictional sales

Date	Sales	Weighting
February 4	1847.50€	4/10
February 3	2011.79€	3/10
February 2	1999.99€	2/10
February 1	1810.49€	1/10

Given the observations in Table 2.2 we want to calculate the weighted moving average of the recent sales. To do this we simply multiply the given sales each day by their weights and then total all the values to get the weighted moving average.

WMA = (1847.50 * 4/10) + (2011.79 * 3/10) + (1999.99 *2/10) + (1810.49 * 1/10)

WMA = 1923.584€

The benefit of using the weighted moving average instead of the simple moving average is that the user gets a more sensitive and faster reacting model. As the recent observations are assigned more weight, the weighted moving average is naturally faster to react to changes. On the other hand, this also gives the model a higher volatility compared to simple moving average as more sensitivity means more movement in the line.

2.3.3. Moving average as a forecasting method

When it comes to using moving averages as forecasting methods, the basic models are rarely used, especially when it comes to retail demand forecasting. Most researchers focus on models based on the autoregressive and integrated moving average (ARIMA) which will be covered in the next section. However, there is some research focusing on the basic moving average models as forecasting methods. Although most of it does not focus on the retail industry, it could to some degree be argued that the research could be applied to the retail industry as well.

One of the few studies where the moving average is used for predicting retail demand is done by Chen et al. (2010). They compare a simple 7-day moving average to a 7-day, 14-day, 21-day and a 28-day back-propagation neural network (BPNN) and measure the accuracy when predicting the sales of 10 different food products of a Taiwanese convenience store chain. They found that while the 28-day BPNN was the most accurate BPNN, the moving average still outperformed the 28-day BPNN on almost all measurements. The combined mean-squared error was by far the lowest for the moving average (1.0036 compared to 1.6286 for the 28day-BPNN), the number of times with no prediction error was notably higher for the moving average (51.79% vs. 37.14%) and the number of times with a prediction error of one or less was also higher for the moving average (82.50% vs. 78.57%). The moving average also has a smaller percentage of disqualified performances – performances with a prediction error greater than 2 – compared to the BPNN (17.50% vs. 21.43%). The only areas where the BPNN outperformed the moving average were the mean-squared error of three individual products as well as never underestimating the upcoming sales. In the eyes of the customer, it is obviously positive for a model to never underestimate the sales as there will never be a situation where the product is out of stock. However, the retailer must consider if the cost of losing a sale is bigger than the cost of having too much inventory, or in other words, if it is more profitable to have a positive or negative forecasting error.

Samvedi and Jain (2013) compared the performance of the simple moving average, the weighted moving average, and a few other forecasting models in different supply chain

scenarios. They used what they call a two-stage scenario, which only includes the customer and retailer, to compare the performance of the models in simulated scenarios with different demands. They found that both the simple moving average and the weighted moving average performed best when the demand was steady and there were no disruptions in the demand. The only difference between the models in this scenario is that the weighted moving was a little bit faster to react to changes in demand. When disruptions in the form of sudden peaks in demand were introduced, the moving average models were immediately beaten by other forecasting models since moving average models tend to stay close to the mean value. The higher the degree of smoothing of the interruption (the more steps it takes to reach the peak of disruption) was, the better the moving average models performed, but they were still outperformed by faster reacting models. However, it is worth noting that Samvedi and Jain (2013) did not specify how many past observations they used for calculating the moving averages. Therefore, the performance of the moving average models could possibly be explained by Samvedi and Jain (2013) using too many observations for their calculations, leading to unresponsive moving average models.

Anusha, Alok and Ashiff (2014) conducted a study where they applied the simple moving average as well as three exponential smoothing models to forecast the demand for two products sold by an Indian pharmacy chain. One of the products is used to treat allergies and therefore has a seasonal variation in its demand, whereas the other product is used to treat high blood pressure and, thus, has a relatively steady demand throughout the year. Anusha, Alok and Ashiff (2014) found that the simple six-month moving average was the most accurate model when forecasting the demand for the product with no seasonality, but when it comes to the product with seasonal demand, an exponential smoothing model performed better. This result confirms the findings of Samvedi and Jain (2013) as they also concluded that moving averages performed best when the demand is steady, but when disruptions are introduced, the moving averages are outperformed by other models. In the case of Anusha, Alok and Ashiff (2014) the seasonality can be seen as a form of disruption, as seasonal demand means that there is a high variation in the demand.

Finally, Kalaoglu et al. (2015) compared the simple moving average, the weighted moving average and a linear regression model when forecasting the demand for a Turkish clothing retailer. They found that the moving averages outperformed the regression model in the predictions, but the performance of the simple and weighted moving average models did not show any major differences when compared to each other. Also, in this study, the moving averages performed better on products with less seasonality, confirming the findings of Samvedi and Jain (2013) and Anusha, Alok and Ashiff (2014).

To conclude the findings about the moving average as a retail forecasting method, they work quite well and accurately when the demand for the product being forecasted is steady and has few disruptions. This is because the moving average tends to stay close to the mean of a time series and is relatively slow to react to changes. When the demand has more disruptions, for example seasonality, the moving averages are often outperformed by advanced models as Anusha, Alok and Ashiff (2014) showed. However, it is important to remember that the essence of moving averages is more about finding trends (Fong & Yong, 2005) than predicting exact values. This is also why the moving averages are more commonly used for analyzing stock markets where the trends play a bigger role, than for predicting product demand and sales where the actual sales number plays a bigger role.

2.4. ARIMA

The autoregressive integrated moving average (ARIMA) model, along with exponential smoothing, are the two most widely used models for time series forecasting. While exponential smoothing is based on finding and describing the trend and seasonality in the data, the ARIMA focuses on finding the autocorrelation in the data. The ARIMA model is based on the ARMA model with the difference that ARIMA can use non-stationary data while ARMA requires stationary data in order to work properly. To be able

to use non-stationary data the ARIMA model applies differencing to the time series data, making non-stationary data stationary. (Hyndman and Athanasopoulos, 2021).

In the upcoming section we will describe the three components of an ARIMA model – stationarity and differencing, autoregressive models and moving average models – as well as how ARIMA model are constructed, whereafter we take look into the research made on the use of ARIMA for retail demand forecasting.

2.4.1. Stationarity and differencing

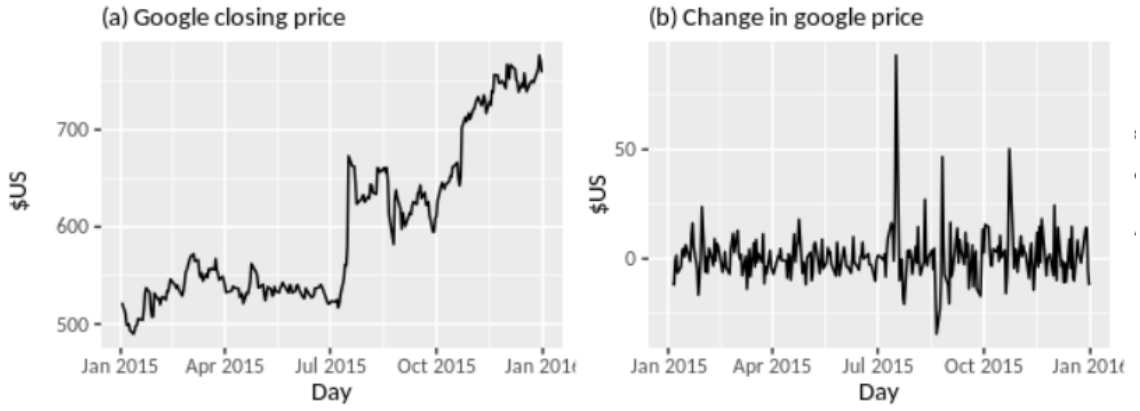
> "*A stationary time series is one whose statistical properties do not depend on the time at which the series is observed."* Hyndman & Athanasopoulos (2021)

Because of this, time series with seasonality are not stationary as the seasonality affects the value of the time series at different times. However, as Anderson, Sweeney and Williams (2011) point out, a stationary time series will always have a horizontal trend, but a horizontal trend does not automatically mean that the time series is stationary. In general, time series without predictable patterns in the long term are stationary (Hyndman and Athanasopoulos, 2021). Hyndman and Athanasopoulos (2021) emphasize that it in some cases can be confusing to determine if a time series is stationary or not, as a time series with cyclic behavior does not have to be non-stationary. Even though a time series with cyclic behavior might resemble a time series with seasonality this does not necessarily have to be the case as we can have cycles without a fixed length, meaning that we cannot be sure of the peaks of the cycle before we observe the data. Therefore, we can have cyclical time series that might look non-stationary at the first glance but ultimately are still stationary.

To tackle the problem of stationarity, differencing has been integrated (hence the I in ARIMA) into the ARIMA model. Differencing is a method where a seasonal time

series or a time series with a trend is transformed into a stationary time series by computing the differences between consecutive observations (Hyndman and Athanasopoulos, 2021). As can be seen in Figure 2.2 by Hyndman and Athanasopoulos (2021), differencing has removed the trend of the time series when we observe the change between the closing prices each day instead of the closing prices each day.

Figure 2.2



Closing price vs. Change in closing price of Google stock - Hyndman & Athanasopoulos (2021), *Forecasting: Principles and Practice, Chapter 9.1*

There are two main ways to conduct the differencing – the random walk model and the seasonal differencing model (Hyndman and Athanasopoulos, 2021). The random walk model follows Formula 2.29:

$$y'_t = y_t - y_{t-1} \qquad (2.29)$$

where y'_t is the difference for each observation t in the original time series and a differenced series of the original series with t-1 values is created. The series only contains t-1 observations as we cannot calculate a difference for the first observation in the original

series. Sometimes, the data might still be non-stationary after the differencing and in those cases, it might be necessary to difference the data once more. This is called a second-order differencing (Hyndman and Athanasopoulos, 2021) and it follows the following formula:

$$\begin{aligned} y''_t &= y'_t - y'_{t-1} \\ y''_t &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ y''_t &= y_t - 2y_{t-1} + y_{t-2} \end{aligned} \tag{2.30}$$

where y''_t is the difference between each observation in the already differenced series, and this time the series will have t-2 values. According to Hyndman and Athanasopoulos (2021), second-order differencing is almost always enough to attain a stationary time series and in practice, further differencing is rarely used.

Seasonal differencing follows the same principles as the random walk model, but it calculates the difference between an observation and the previous observation from the same season (Hyndman and Athanasopoulos, 2021). In other words, the difference attained from seasonal differencing is the difference between one season and the next. The Formula 2.31,

$$y'_t = y_t - y_{t-m}\,, \tag{2.31}$$

looks similar to the one of the random walk model, but m = number of seasons is introduced. In some cases, such as with the random walk model, one degree of seasonal differencing might not be enough. In those cases, we can difference once more using the random walk model resulting in the following formula:

$$
\begin{aligned}
y''_t &= y'_t - y'_{t-1} \\
y''_t &= (y_t - y_{t-m}) - (y_{t-1} - y_{t-m-1}) \\
y''_t &= y_t - y_{t-1} - y_{t-m} + y_{t-m-1}
\end{aligned}
\tag{2.32}
$$

2.4.1.1. Is differencing needed?

It might not always be clear if the time series is stationary or if another level of differencing is needed. However, to get a more objective answer to the question, a unit root test, which is a statistical hypothesis test of stationarity (Hyndman and Athanasopoulos, 2021), can be performed. Many different unit root tests have been developed such as the ones by Levin, Lin and Chu (2002), Harris and Tzavalis (1999), and Hadri and Larsson (2005) but the most cited one is the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test (Kwiatkowski et al., 1992). In the KPSS test, the null hypothesis is that the time series is stationary, and we try to prove the null hypothesis false. To do this the time series is expressed as a sum of a deterministic trend, a stationary error and a random walk, and a Lagrange multiplier test is used to test the hypothesis that the random walk has zero variance (Kwiatkowski et al. 1992). If the KPSS test returns low p-values, the conclusion can be drawn that further differencing is needed.

The unit root tests can be quite complicated but luckily some unit root tests, such as the KPSS test, can be found built in or in downloadable packages for statistical programs such as R or Stata. Therefore, it is usually enough to know how to interpret the results of the test, eliminating the need for deep understanding of the processes behind the test.

2.4.2. Autoregressive models

In an autoregressive model, we use a linear combination of past values of the variable to predict the variable of interest. (Hyndman and Athanasopoulos, 2021). An autoregressive model of order p, or an AR(p) model, is written as:

$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + u \qquad (2.33)$$

where u = error term, and the lagged values of y_t are the predictors, or the explanatory variables. Changes in the parameters ϕ_1 to ϕ_p results in different time series patterns whereas variance in the error term only results in changes of the scale of the pattern. Also, autoregressive models are restricted to stationary data with some constraints required for the parameters. The restrictions for an AR(1) model are $-1 < \phi_1 < 1$, and the restrictions for an AR(2) model are $-1 < \phi_2 < 1$, $\phi_1 + \phi_2 < 1$, $\phi_2 - \phi_1 < 1$. When an autoregressive models has $p \leq 3$, the restrictions are more complicated but most statistical programs have built-in or downloadable packages that take care of them. (Hyndman and Athanasopoulos, 2021)

2.4.3. Moving average models

The moving average model within an ARIMA model uses the past forecasting errors to create the following moving average, or MA(q), model of order q:

$$y_t = c + u + \theta_1 u_{t-1} + \theta_2 u_{t-2} + \cdots + \theta_q u_{t-q} \qquad (2.34)$$

where u is the error term. This model reminds us of a regression model but as we do not observe the values of u, it is not a regression in its usual sense (Hyndman and Athanasopoulos, 2021). As can be seen, each value of y_t can be viewed as weighted moving average of the past forecast errors with weights of θ_1 to θ_q. As with the AR(p) models, changes in the parameters θ_1 to θ_q results in different time series pattern whereas changes in the error term only results in changes of the scale of the pattern. Also, we once again have fairly simple constraints for the parameters of MA(1) and MA(2) and more complicated constraints as q is 3 or higher. The restrictions for a MA(1) model are $-1 < \theta_1 < 1$, and the restrictions for an MA(2) model are $-1 < \theta_2 < 1$, $\theta_2 + \theta_1 > -1$, and $\theta_1 - \theta_2 < 1$. (Hyndman and Athanasopoulos, 2021)

2.4.4. The ARIMA model

When combining the differencing, autoregression and the moving average model we get a non-seasonal ARIMA model that can be written as

$$y'_t = c + \varphi_1 y'_{t-1} + \cdots + \varphi_p y'_{t-p} + \theta_1 u_{t-1} + \cdots + \theta_q u_{t-q} + u \quad (2.35)$$

where y'_t is the differenced series and the predictors on the right side include both lagged errors and lagged values of y_t. This model is also called an ARIMA(p, d, q) model where p is the order of the autoregression, d is the degree of differencing of the original time series data, and q is the order of the moving average. The same constraints that are used on autoregressive models and moving average models also applies to the ARIMA(p, d, q) model. (Hyndman and Athanasopoulos, 2021)

Even though we have automated functions that can choose the values of p, d and q for us, it is important to understand the behavior of the ARIMA model. Table 2.3 shows

the effects of the relationship between c and d on the long-term forecasts. The value of d also affects the prediction intervals so that a higher value of d results in a more rapidly increasing prediction interval size. Also, if d is 0 the long-term standard deviation will go towards the standard deviation of the historical data. (Hyndman and Athanasopoulos, 2021).

Table 2.3

	d=0	d=1	d=2
c=0	To zero	To non zero constant	Follows straight line
c≠0	To mean of data	Follows straight line	Follows quadratic trend

Direction of long-term forecast using ARIMA(p, d, q) model. (Hyndman and Athanasopoulos, 2021)

2.4.4.1. Seasonal ARIMA model

While the basic ARIMA(p, q, d) model is a non-seasonal model, ARIMA models can also be used on seasonal data. In order to do so, the ARIMA(p, q, d) model is modified to include additional seasonal terms to become the ARIMA(p, d, q)(P, D, Q)m model (Hyndman and Athanasopoulos, 2021), or SARIMA as it is also known as. In the SARIMA model the m stands for the seasonal period of the time series (e.g., 12 for monthly data and 52 for weekly data) and the uppercase P, D and Q are the seasonal part of the model. Hyndman and Athanasopoulos (2021) explains the difference between the seasonal and non-seasonal parts of the model as follows:

> *"The seasonal part of the model consists of terms that are similar to the non-seasonal components of the model, but involve backshifts of the seasonal period."*

As an example Hyndman and Athanasopoulos (2021) give an ARIMA(1,1,1)(1,1,1)$_4$ model for quarterly data as m=4, that can be written as follows.

$$(1-\varphi_1 B)(1-\boldsymbol{\varphi}_1 B^4)(1-B)(1-B^4)y_t = (1+\theta_1 B)(1+\boldsymbol{\theta}_1 B^4)u \qquad (2.36)$$

Where the first two parentheses represent AR(p) and AR(P), the third and fourth parentheses represent the differencing (d) and (D), the final two parentheses represent the MA(q) and MA(Q) and u = error term, (Hyndman & Athanasopoulos, 2021).

2.4.5. ARIMA as a forecasting method

The statement presented in the beginning of this Section 2.4 by Hyndman and Athanasopoulos (2021), that ARIMA is one of the most popular time series forecasting methods alongside exponential smoothing, is confirmed by the amount research that can be found on the topic. In this section, the most relevant research that treats ARIMA models as a forecasting method in the retail industry will be presented.

Da Veiga et al. (2016) compared the performance of the triple exponential smoothing and ARIMA models alongside a neural network and a fuzzy system when forecasting the demand of three perishable food products. To estimate the parameters of the triple exponential smoothing and ARIMA models, da Veiga et al. (2016) used the Minitab statistical package and to compare the accuracy of the forecasts they used both the MAPE and U-Theil measures, which both take a lower value the better the forecast is. When comparing the forecast accuracy of the ARIMA models and the triple exponential smoothing models, the ARIMA model outperformed the triple exponential smoothing on the forecast of one product and the triple exponential smoothing outperformed the ARIMA models on the forecast of the two other products. However, all three forecasts with both

models had a MAPE ranging from 4.91 to 8.57, suggesting that they all performed reasonably well as a MAPE below 10 usually is a sign of a forecast that has performed well (Lewis, 1997). This, once again, shows that finding the best performing model is highly case specific, as has been shown previous in this thesis. Also, it is worth noting that while the time series analysis models performed well, they were outperformed by one or both machine learning models on the forecasts of all three products.

The study by Alon, Qi and Sadowski (2001) brought up in Section 2.2.4. also has ARIMA as one of the compared models. The results of this study confirm the findings of da Veiga et al. (2016) as the time series analysis models performed well but were still beaten by a neural network. The MAPE of the time series analysis models ranged from 1.18 to 3.11 which is notably lower than the ones by da Veiga et al. (2016). It is hard to pinpoint the exact reason for the better forecasting results as it could be anything from better optimized models to more suitable data. However, one notable result of the study by Alon, Qi and Sadowski (2001) is the fact that their ARIMA model was able to beat the neural network in forecast accuracy on one occasion. This shows that time series analysis models are capable of outperforming more advanced machine learning models when the conditions are optimal. As with de Veiga et al. (2016), Alon, Qi and Sadowski (2001) also found the performance of the ARIMA models and triple exponential smoothing to be close to each other, ARIMA having MAPEs between 1.18-2.20 and triple exponential smoothing having MAPEs between 1.16-3.11.

When it comes to data with seasonality, the seasonal ARIMA model is the most advanced time series analysis model that has been widely successfully tested (Chu and Zhang, 2003). Chu and Zhang (2003) conducted a comparative study of time series analysis and machine learning models for aggregate retail sales forecasting. They compared a seasonal ARIMA model to a linear regression model as well as to neural networks. They used a forecasting software called Forecast Pro to automatically generate the best ARIMA model based on the data, and the seasonal ARIMA$(0,1,1)(0,1,1)_{12}$ model was generated. Chu and Zhang (2003) found, as the previously mentioned research, that while their ARIMA model was the best performing time series analysis model, it was outperformed by neural networks. The performance of the ARIMA model was by no

means bad as the MAPE was only 2.30 which is significantly lower than the MAPE of 5.36 of the regression model but the neural networks had MAPEs as low as 1.69. However, Chu and Zhang (2003) emphasize that while the MAPE of the models were low and the forecasts were fairly accurate, both the ARIMA model and the neural networks generated forecasts that were consistently lower than the actual sales. In other words, this is a clear under-forecasting situation where the retailer would constantly lose sales if they blindly trusted the forecast and did not have any safety stock.

There is also some criticism against and disadvantages of using an ARIMA model. Stevenson (2007) raised concerns that if the need for differencing is varying between different periods in the time series and the size of the time series is limited, the suitability of ARIMA models can be questioned. The data used by Stevenson (2007) contained 60 observations, which is more than the suggested minimum of 50 observations (McGough and Tsolacos, 1994, as well as Tse, 1997) when using ARIMA, but the observations only changed between four consecutive periods. This suggest that the need for differencing only exists for the observations around these four periods, and therefore, more suitable forecasting models might exist for this dataset. Another disadvantage of the ARIMA model is that it does not allow to estimate the direct relationship between two series (Jenkins, 1979), but Chamlin (1988), who studies the relationship between crime and arrests, points out that this is not a problem if you are interested in the lagged relationship between two variables. Other issues brought up by Chamlin (1988) is that ARIMA tests have been criticized for being too conservative when studying the causation between two series, as well as multivariate ARIMA models requiring long time series in order to produce reliable estimates of the parameters (McCleary et al., 1980).

2.5. Time series regression

Time series regression, or regression analysis using time series data, is a statistical method used for predicting the future outcome based on historical data. Depending on

how advanced the regression model is it can be divided into simple linear regression analysis or multiple regression analysis. (Woolridge, 2012). In order to gain an understanding of how time series regressions are used for demand forecasting we will first look at the theory behind simple and multiple regression analysis, and thereafter we will examine how they can be applied to demand forecasting.

2.5.1. Simple regression analysis

A simple regression model is used to study the relationship between two variables; how is *y* affected by a change in *x*. In retail forecasting this could be translated into, for example, how the sale of a product is affected by a price reduction. The model can often be quite limiting since it only studies two variables, but nevertheless it has its uses, and it is crucial to understand how a simple regression model works in order to understand multiple regression. Woolridge (2012) describes a simple linear regression model as follows:

$$y = \beta_0 + \beta_1 x + u. \qquad (2.37)$$

In this model *y* and *x* are the two variables we want to find the relationship between. Y is called the *dependent variable* or *explained variable*, x is the *independent variable* or *explanatory variable*, u is the *error term* or *disturbance* of the relationship representing other factors affecting the relationship, β_0 is the *intercept* and β_1 is the *slope parameter* (Woolridge 2012). In other words, y is the unknown variable we want to predict, x is the variable which affects the predicted variable, β_1 explains to which degree x affects y, β_0 tells what the expected y is when x = 0, and u represents everything else not specified in the model that is affecting the relationship between y and x. However, in order to get reliable estimators of β_0 and β_1 of a data set we assume that E(u) = 0, because

otherwise we will not be able to estimate the ceteris paribus (all other things equal) effect, or β_1. Woolridge (2012) rationalized the assumption as follows:

> *"Before we state the key assumption about how x and u are related, we can always make one assumption about u. As long as the intercept* β_0 *is included in the equation, nothing is lost by assuming that the average value of u in the population is zero."*

In other words, we can assume that u = 0 because it is assumed that all other factors effecting y are equal in all cases and are therefore already included in the intercept, β_0.

In order to create the regression model, the intercept β_0 and the slope parameter β_1 must be attained. This can be done using the *least squared method*, which uses sample data to provide the values for the slope and intercept at which the sum of the squares of the deviations between the observed values of the independent variable and estimated values of the dependent variable are minimized (Anderson, Sweeney and Williams, 2011). What happens in practice is that we use Equation 2.39 and 2.40 presented by Anderson, Sweeney and Williams (2011) in order to find the values that minimize the least squares criterion (Equation 2.38).

$$min \sum (y_j - \hat{y}_i)^2 \qquad (2.38)$$

where

y_j= the observed value of the dependent value for observation i

$\hat{y}_i$= the estimated value of the dependent value for observation i

$$\beta_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} \quad (2.39)$$

$$\beta_0 = \bar{y} - b_1\bar{x} \quad (2.40)$$

where

x_i = value of the independent variable for the *i*th observation

y_j = value of the dependent variable for the *i*th observation

$\bar{x}$ = mean value for the independent variable

$\bar{y}$ = mean value for the dependent variable

For small datasets, Equation 2.39 and 2.40 can be used to calculate the intercept and slope parameter by hand, but the datasets used for forecasting are usually of such a large size that this is not a viable option. Also, as a minimization problem is included by Equation 2.38, the processing power of a computer is needed to solve the values of the slope parameter and the intercept. Luckily, there are various forecasting and statistical programs, such as Stata, which can be used to automatically attain the intercept and slope parameter from the chosen set of sample data. In addition to these, there are also predesigned functions such as the LinearRegression function by Scikit-learn that can be used in a python environment in order to develop a linear regression model.

Finally, to make it clear, let us look at the following example of a regression model describing the expected exam score in relation to how many lectures a student attended.

$$Examscore = 3.14 + 0.15lectures \quad (2.41)$$

This regression model is interpreted so as that the exam score is expected to be 3.14 if the student has skipped all lectures, but for every additional lecture attended the score is

expected to be 0.15 higher. If for example a student attends 5 lectures, the exam score is expected to be 3.14 + 0.15 * 5 = 3.89.

Even though the single regression model is easy to understand and gives a clear relationship between x and y, it has a major drawback which often makes it an unrealistic model for empirical use. It is difficult to draw the ceteris paribus conclusions on how x affects y because in practice all other things are seldom equal (Woolridge, 2012). Looking back at the previous example you could seldom say that the exam score only depends on how many lectures are attended. There are most likely other factors such as previous knowledge and hours spent studying for the exam that have a noticeable effect on the exam score and therefore it is impossible to say that all other factors would be equal.

2.5.2. Multiple Regression Analysis

Multiple regression analysis tackles the problem of drawing a definite ceteris paribus effect using single regression analysis. As the name implies, multiple regression analysis takes multiple different factors that influence the dependent variable into account, making the statement of all other things being equal more accurate than in the case of single regression analysis. If we add more factors that can be used to explain the dependent variable y to our model, it naturally leads to a bigger portion of the variance in y to be explained. This is a clear advantage compared to a single regression analysis, and therefore the ceteris paribus effect can also be said to be more accurate, and more accurate prediction models are created as a result. According to Woolridge (2012), the fact that multiple regression analysis allows us to control many different factors that simultaneously affects the dependent variable, resulting in a more accurate ceteris paribus analysis, is crucial when we are using a model to test nonexperimental data. For example, Woolridge (2012) mentions testing economic theories and evaluating policy effects, but a

multiple regression analysis could equally well be applied to predict the upcoming sales. (Woolridge, 2012)

If we look at the example in the previous section, we can turn it into a model for multiple regression analysis simply by adding another factor into the equation.

$$y = \beta_0 + \beta_1 x + \beta_2 z + u \qquad (2.42)$$

We have now added the independent variable z with the slope parameter of β_2 to make the single regression model into a multiple regression model. Once again, we can also assume that E(u) = 0 as all other factors affecting y are included in the intercept to make the model ceteris paribus. In this example we only added one independent variable to the equation but in theory we could add as many as necessary if we know the slope parameter for each added variable. Let us look at the example treating exam scores from the previous section. The regression model presented in Equation 2.41 indicates that every lecture attended resulted in a 0.15 higher exam score. Let us assume that after making a survey amongst the students in the class, a relationship between the exam score and hours studied for the test, as well as taking an algebra course in the previous year has also been found. We now update the model, and the result is the following:

$$Examscore = 2.04 + 0.15lectures + 0.05hours + 0.3algebra \qquad (2.43)$$

The new models add the independent variable hours with a slope parameter of 0.05 and the independent dummy variable algebra with a slope parameter of 0.3. The intercept has now also decreased from 3.14 to 2.04. What this means is that the expected exam score without attending lectures, studying and taking the algebra course is now 2.04. The big decrease in the expected exam score is explained by the fact that both hours studied and

the previous knowledge from the algebra course were wrongfully included in the intercept in the simple regression model as we did not know that they had a direct relationship with the exam score. Using the example above the calculation for the expected exam score could look like this:

$$\text{Examscore} = 2.04 + 0.15*5 + 0.05*16 + 0.3*1$$

$$\text{Examscore} = 3.89$$

As we see here, in addition to the 5 lectures attended it also required 16 hours studied for the exam as well as attending the algebra course to get the expected exam score of 3.89 that the single regression model gave us. The multiple regression analysis showed us that the single regression analysis made earlier was in fact quite inaccurate and all other things were not equal although we assumed they were.

2.5.3. Linear regression as a forecasting method

A lot of research has been made regarding linear regression as a forecasting method. Burger et al. (2001) compared different methods for forecasting the tourist demand in Durban, South Africa, and found that multiple regression was a fairly accurate model in their case, but it was also quite a limited model. Multiple regression is limited by the relatively small number of coefficients in the model and in the case of Burger et al. (2001) they could only predict one month in advance with multiple regression, whereas other models were able to predict multiple months into the future. Bougadis, Adamowski and Diduch (2005) studied the forecasting of short-term water demand in Ottawa, Canada, with the help of both simple and multiple linear regression models, amongst other models. They got quite accurate results using multiple regression although not as accurate as the ones by Burger et al. (2001). Also, Bougadis, Adamowski and Diduch (2005) had different

versions of the multiple regression model working best with the training data set and the test data set, showing that the multiple regression models are highly case specific.

When forecasting using time series regression it is also common to use a modified version of the multiple regression model. For example, Chan (1993) uses a sine wave time series regression approach to forecasting the number of tourists arriving at a destination during a month. Chan (1993) developed the following model:

$$y = a_1 + a_2 t + a_3 \sin(a_4 + a_5 t) + u \tag{2.44}$$

where

y = seasonally adjusted number of tourist arrivals at time t

t = time in month with respect to a fixed reference point

a_1 = intercept of the linear model

a_2 = slope of the linear model

a_3 = amplitude of the sine function

a_4 = phase angle of the sine function

a_5 = frequency of the sine function

u = error term

The addition of the sine function to the multiple regression model makes it a non-linear model. Chan (1993) used historical data from 221 previous months to predict the number of tourists for the next 19 months. Three of the forecasted months had a forecast error of over 5% when comparing the forecast to the actual number of tourists, but for the other months the error was around 3% or lower and the lowest forecast error achieved was only 0.27%. Even a forecast error of 5% can be considered as a low forecast error, as per Lewis

(1997), and therefore, the results by Chan (1993) show that time series regression models are capable of producing accurate forecasts.

When it comes to research focusing on the retail industry the two most cited research papers are the ones by Chu and Zhang (2003) and Alon, Qi and Sadowski (2001). Chu and Zhang (2003) conducted a study where they compared linear regression to a seasonal ARIMA model and a few neural network models when forecasting the monthly aggregated US retail sales. While they found that all forecasting models performed well, achieving MAPEs below 7%, the linear regression model was one of the worst performing models. While the linear regression model achieved a MAPE of 6.67% on an out-of-sample forecast, the ARIMA model and some of the neural networks managed to achieve forecast errors as low as 1.69-2.30%, majorly outperforming the linear regression model. Chu and Zhang (2003) also noted that the forecast error of their linear regression model was almost five times larger on the out-of-sample forecast compared to the forecast error of the validation sample. This shows that a good performance on the sample data does not always result in a model that forecasts well, and in the case of linear regression it could be a result of, for example, over-fitting of the model.

In the study by Alon, Qi and Sadowski (2001), the US aggregate retail sales was once again forecasted. This time a regression model was compared to an ARIMA model, a triple exponential smoothing model and an ANN. The findings of Alon, Qi and Sadowski (2001) are similar to the findings of Chu and Zhang (2003) as they also found that the regression model the worst performing models even though it had an average MAPE of 2.75%. It was also found that the regression model performed worse on a multiple step ahead forecast, hinting that a regression model might work better on short-term forecasts. This is somewhat contradictory to the performance of the other forecasting models used, as on average the forecast performance of the multiple-step forecasts were lower than the one step forecasts. On the other hand, as Alon, Qi and Sadowski (2001) point out, it would be more logical that the performance of a one-step forecast is more accurate than a multiple-step forecast as the data used is more up to date in the case of a one-step forecast. Therefore, it can be concluded that the performance of the regression model is as expected.

2.6. Decision trees

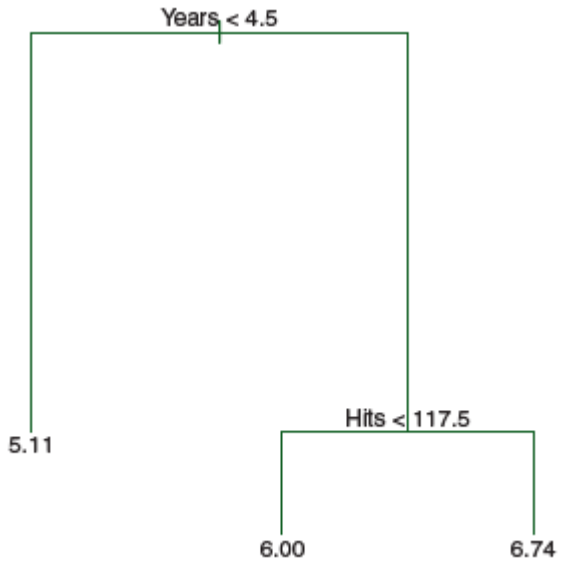
A Decision tree is a method of sequential decision-making which can easily visualized in a straightforward way (Anderson, Sweeney and Williams, 2011). As the name suggests, decision trees are mainly used for decision-making but they can also be applied to other purposes, such as forecasting in the case of this thesis. In the upcoming section the structure of a decision tree for machine learning will be briefly explained, whereafter the research on the topic of decision trees as a forecasting method will be presented.

2.6.1. The structure of a decision tree

When a decision tree model is used for forecasting, it involves a tree-based method for regression, classification, or a combination of both. Due to the structure of a decision tree being characterized by a set of splitting rules for segmenting, the visualization of the model has the likeness of a tree, explaining the name of the model. (James et al. 2013)

To understand the basic structure and functionality, let us look at the regression tree presented by James et al. (2013):

Figure 2.3



Regression tree – James et al. (2013), *An introduction to statistical learning*, Chapter 8.1

This simplified regression tree is used to predict the salaries of baseball players. It has two splitting rules assigning the observations to different branches based on years played and hits made. First, starting from the top of the tree, the regression tree splits the observations so that all players that have played less than 4.5 years are assigned to the left branch and are predicted to have a salary equaling the mean salary of all players who have played less than 4.5 years. In this case the salaries are log-transformed, measured in thousands of dollars, and rounded to the nearest hundredth in the visualization, meaning that the salary for this group of players would be $e^{5.107}$ thousands of dollars, or $165,174. The group of players that have played longer than 4.5 are further split into two groups based on how many hits they have made last season. The players who made less than 117.5 hits are predicted to earn $e^{5.999}$ * $1,000 = $402,834, and the players managing more than 117.5 hits are predicted to earn $e^{6.740}$ * $1,000 = $845,346. These three groups of players can also be written as:

$$R_1 = [X|Years < 4.5] \quad (2.45)$$

$$R_2 = [X|Years \geq 4.5, Hits < 117.5] \quad (2.46)$$

$$R_3 = [X|Years \geq 4.5, Hits \geq 117.5], \quad (2.47)$$

where R_1, R_2, R_3 are known as *terminal nodes* or *leaves*. In addition to these, the points at which the splitting rules are applied are called *internal nodes*, and the segments that connect the internal nodes are called *branches*. (James et al., 2013)

When a regression tree is built, the goal is to find the groups R_1, …, R_J that minimize the root sum squared (RSS), given by the equation:

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2, \quad (2.48)$$

where $\hat{y}_{Rj}$ is the mean response for the training observations in the j^{th} group (James et al., 2013). Since it is impractical to consider every possible partition when creating the groups, a top-down, greedy approach known as recursive binary splitting must be taken when building a regression tree. The process begins at the top layer of the tree where the best split is made before moving on to the next layer of internal nodes where new splits are made minimizing the RSS, again moving to the next layer creating new splits minimizing the RSS, and so on. The process is called greedy since at each step of creating splits, the future splits are not considered. Instead, only the best split at the current level is determined, not looking forward and selecting the split that would possibly result in a better tree in the future. When all the groups R_1, …, R_J have been determined and created, the response for each given observation is calculated using the mean of the training observations in each group. (James et al., 2013)

According to James et al. (2013) creating a regression tree this way often leads to a model that performs well on the training set but likely overfits the data, resulting in poor performance on the test set. As a way of dealing with this problem, James et al. (2013)

presents the process of *cost complexity pruning* or *weakest link pruning* as it is also known as. Pruning is a process of first creating large tree and then prune it back to obtain a smaller subtree. The goal of pruning is to select the subtree with the lowest test error rate which could be done using cross-validation. However, estimating the cross-validation error for every subtree is a demanding task taking a lot of time. The cost complexity pruning solves this by considering a sequence of trees with a tuning parameter α, instead of considering every possible subtree. There is a corresponding subtree for every value on α where $T \subset T_o$ and

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \propto |T| \qquad (2.49)$$

is minimized. |T| is the number of terminal nodes in the tree, R_m is group corresponding with the m^{th} terminal node, and $\hat{y}_{Rm}$ is the predicted response for R_m. (James et al., 2013)

A classification tree is similar to the regression tree described so far, with the difference that a classification tree is used to predict qualitative responses instead of the quantitative responses of a regression tree. Whereas a regression tree returns the mean response for the group, a classification tree returns the most commonly occurring response in the group. Also, whereas the regression tree used RSS as a criterion for making the splits, a classification tree can use the classification error rate which measures the fraction of the training observation that do not belong to the most common class of the group. However, according to James et al. (2013), the classification error rate is not always accurate enough. In those situations, the Gini index which measures the variance across all groups can be used. A low Gini index is a sign that a large portion of the observations in a node are from a single class, hence explaining why the Gini index is sometimes also called the index of node purity. Another alternative to use is entropy, which is similar to the Gini index. Like the Gini index, entropy takes a small value if a node is pure. When building a classification tree both the Gini index and entropy can be used to determine the quality of a split. However, when pruning the tree, the classification error rate is preferred as it usually results in better prediction accuracy. (James et al., 2013)

While decision trees like the ones described above can be designed from scratch, it is usually not a simple task. However, there are some predesigned functions and programs available, such as the DecisionTreeRegressor for python by Scikit-learn, which applies machine learning techniques to develop the most suitable regression tree for forecasting based on the historical data provided by the user. Forecasting results attained by decision trees will be presented in the next section.

2.6.2. Decision tree as a forecasting method

While decision trees are a popular research topic, a vast majority of the research available focuses on the use of decision trees for classification problems, and when it comes to the research treating decision trees as a forecasting method, it mainly focuses on other fields than the retail industry. For example, Liu et al. (2017) used a machine learning algorithm based on decision tree to forecast the price of copper, Kumar (2013) used decision tree for weather forecasting, Lai et al. (2009) forecasted the upcoming stock prices with a fuzzy decision tree and Liao and Sun (2010) used a decision tree method to forecast the water quality of Chao Lake. While these research papers did not focus on the retail industry, they still had promising results showing the forecasting capability of a decision tree. For example, Liu et al. (2017) achieved MAPEs lower than 4%, indicating that the model produces accurate forecasts, and Liao and Sun (2010) showed that the decision tree used produced more accurate forecasts than neural networks.

When it comes to the research on decision tree as a retail sales forecasting method, the work of Thomassey and Fiordaliso (2006) is one of the most quoted articles. They developed a hybrid forecasting model based on both clustering and decision tree, to be used in the highly versatile textile market. As the textile industry is highly competitive and has specific constraints such as the short lifetime of the sold items and the enormous number of new items released to the market, the time series analysis models are unsuitable for the industry and a new forecasting model is needed (Thomassey and Fiordaliso, 2006).

The model developed by Thomassey and Fiordaliso (2006) tries to solve these problems by using clustering to group products with similar sales profiles and then finding the links between the clusters using the decision tree. When applied to the data of a French retailer, Thomassey and Fiordaliso (2006) got highly accurate results with their model. The average RMSE of the model when tested on 285 items was only 12.7E-3 which is about 10% lower than the best benchmark model used by the researchers. While the model used in this paper is specifically developed for the textile industry, Thomassey and Fiordaliso (2006) claim that the model is also applicable to other industries characterized by many historical items but with no historical data to base the predictions on. In addition to this, descriptive criteria do also need to be available for their model to work. Situations with these characteristics usually arise when the sales of new products are forecasted.

In another research paper, Cheriyan et al. (2018) studied the use of different machine learning techniques for sales trend prediction. They applied a decision tree along with two other machine learning models to a dataset of e-fashion sales data in order to compare the predicting performance of the models. The results of the study by Cheriyan et al. (2018) are somewhat contradictory to the results of Thomassey and Fiordaliso (2006). While Thomassey and Fiordaliso (2006) managed to develop a highly accurate model, Cheriyan et al. (2018) had a much more inaccurate decision tree model. Cheriyan et al. (2018) calculated the RMSE, the MSE and the absolute error, and presented the average of these as the error rate of the model. The error rate for the decision tree model used by Cheriyan et al. (2018) was 29% which cannot be considered an acceptable forecasting error of an accurate model. The study also showed that other machine learning techniques managed to achieve error rates as low as 2%, proving that either the data used was unsuitable for a decision tree model or that the decision tree model was poorly developed.

In the final relevant research paper to be presented on decision tree models used as a retail sales forecast method, Wen et al. (2013) study de use of a support vector machine (SVM) model to forecast the grape sales of a fruit supermarket and compare the results to an artificial neural network (ANN) and a decision tree model. The sales of grapes is characterized by its high seasonality due to grapes becoming ripe in the summer months

and the short shelf life due to the perishability of grapes. When forecasting, Wen et al. (2013) use the weather data and type of date (weekday or weekend) in addition to the sales quantity and grape price in order to develop more accurate forecasting models to be used on the three different grape species forecasted. They found that the average relative errors were somewhat different for each of the grape species, differing about 10 percentage units between the most accurate and least accurate forecast for each forecasting method. While the average relative errors varied somewhat between each grape species, the three forecasting methods still performed consistently in relation to each other for every grape species forecasted. The forecast generated by the ANN was always the worst performing model out of the three, trailing by an average relative error of 0.03 to the second-best performing model which on average was the decision tree model. Even though the decision tree model on average was outperformed by the SVM model, it was always close in forecast accuracy and even managed to beat the SVM model on one occasion. Another important finding by Wen et al. (2013) worth noting is the fact that the decision three was able to execute the forecasts substantially faster than the other two models. The decision tree took about 5 to 6 seconds to execute, whereas the fastest ANN took about 50 seconds to execute and the fastest SVM model took about 48 seconds to execute. This indicates that decision trees require substantially less computing power in order to be executed which in certain situations can be a deal breaker.

To summarize the findings on decision tree models as a retail sales forecasting method it can be said, as with most of the previously mentioned methods, that the performance of the decision tree is highly case specific. Because all three research papers examined in this section used different measures of forecasting accuracy, it is difficult to compare the results to each other but the extremely low RMSE attained by Thomassey and Fiordaliso (2006) is probably still the lowest forecast error of the three articles. This shows that hybrid models also have the potential to be even more accurate than models only utilizing one forecasting technique. However, due to the lack of research on the topic, it is impossible to draw definite conclusions based on these three articles, especially since different measures of accuracy were used.

2.7. Artificial neural networks

Artificial neural networks (ANN) are machine learning techniques that are designed to follow the learning pattern of a brain. A human nervous system contains neurons which are connected to each other with axons and dendrites, and the region between them is called synapses. When external stimuli are introduced, the strength of the synapsis changes, and this change is how learning takes place in living organisms. ANNs are designed to follow this same mechanism for learning. (Aggarwal, 2018)

Since ANNs are such a wide topic that they could be covered in a master's thesis of their own, this section will only focus on explaining the basic principles of how an ANN works and providing a review of how ANNs are used to forecast retail sales.

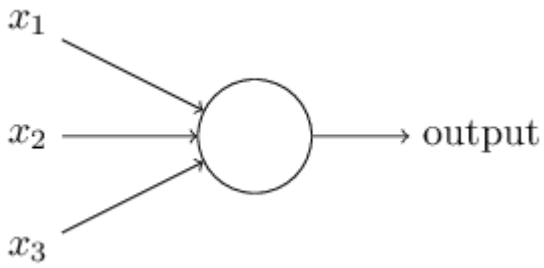
2.7.1. Structure of an artificial neural network

Just as a neural network of an animal, the artificial neural network consists of many neurons connected to each other. The neurons of an ANN are connected by weights which have the same functionality as the synaptic connections in a real neural network, and by changing the weights of an ANN the learning takes place. ANNs also require external stimuli in the form of a training dataset in order to learn. (Aggarwal, 2018)

An ANN can have different types of neurons. In order to understand the basic structure and functionality of an ANN we will begin by explaining perceptrons, which are a basic form of neurons developed in the 1950s and 1960s by Rosenblatt (Nielsen, 2015). Even though the perceptrons are somewhat outdated and not that commonly used today, they are essential in order to understand how the more advanced sigmoid neurons work, and therefore, the perceptrons needs to be covered. The perceptron is a neuron that works with binary inputs and outputs. Depending on if the weighted sum of the input is above or

below a specific threshold, the neuron returns an output of either one or zero. Nielsen (2015) depicts this by the following drawing:

Figure 2.5



The neuron. (Nielsen, 2015)

$$output = \begin{cases} 0, if \ \sum_j w_j x_j \leq treshold \\ 1, if \ \sum_j w_j x_j > treshold \end{cases} \quad (2.50)$$

Where x = input and w = weight of the input. Another way of writing the output of a perceptron is the following (Nielsen, 2015):

$$output = \begin{cases} 0, if \ w * x + b \leq 0 \\ 1, if \ w * x + b > 0 \end{cases} \quad (2.51)$$

Where x = input, w = weight of the input and b = bias. The bias is a measurement of how easy it is to get the perceptron to return a 1, or in other words, how easy it is to get the perceptron to fire. The bigger the bias is, the easier it is to get the perceptron to fire.

An ANN is created by combining many neurons in multiple layers. Each layer answers questions of increasing level of complexity, with the first layer answering the

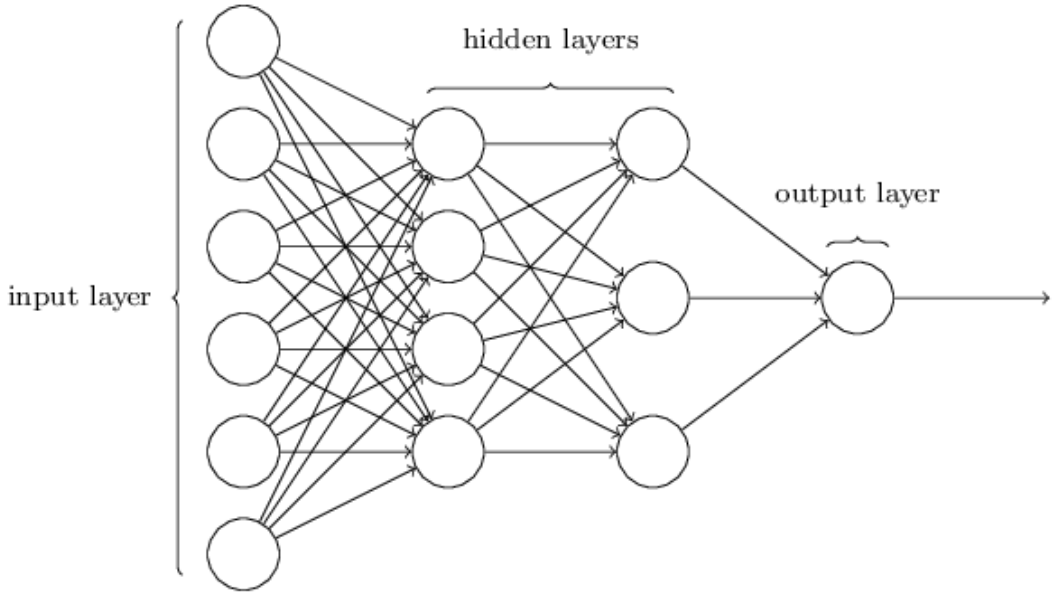
simplest question and the output layer answering the most complicated and complex question. By building an ANN so that it answers a complex question by breaking it down into multiple, less complex questions mimics the behavior of animal brains. For example, if we had an ANN which was to identify if a picture showed a car, it could first ask questions like “Is this a tire?” and “Is this a lug nut?”, which would lead to “Is this a wheel?”, which in turn eventually would lead to the final question “Is this a car?”. (Nielsen, 2015)

The ANN presented above is a deep neural network, described as follows by Nielsen (2015):

> *“It does this through a series of many layers, with early layers answering very simple and specific questions about the input image, and later layers building up a hierarchy of ever more complex and abstract concepts. Networks with this kind of many-layer structure - two or more hidden layers - are called deep neural networks.”*

The first layer of an ANN is called the input layer as it is the layer that processes the input data, and the final layer of an ANN is called the output layer as it is the layer giving the final output of the ANN. All the layers in between the input and output layer are called hidden layers as they are neither input nor output layers. Nielsen (2015) depicts the architecture of an ANN as follows in Figure 2.6.

Figure 2.6



The structure of an artificial neural network. (Nielsen, 2015)

It is important to note that even though the neurons have many output arrows it only represents that the output is used as an input in many hidden neurons, not that there are many different outputs for each neuron. Also, after the input layer, all the decisions made by the neurons are made based on the outputs of the previous layers and the original input data is not used any more (Nielsen, 2015).

The problem with perceptrons is that the output is always a zero or a one. In order to create a learning algorithm inside a neural network we want to see a small change in the output when we make a small change in a weight of a neuron. This is not possible with perceptrons as a small change in the weight of an input might lead to the output flipping from a one to a zero, or vice versa. This in turn leads to all upcoming neurons in the ANN needing complicated restructuring and recalculation, taking a lot of time and processing power each time a weight is slightly adjusted. To solve this problem the sigmoid neurons are introduced. Sigmoid neurons are neurons which take a value between zero and one as the input value instead of the binary input values of a perceptron and returns an output value also between zero and one instead of zero or one. (Nielsen, 2015)

The output of a sigmoid neuron is

$$\sigma(w * x + b), \tag{2.52}$$

where w = weight, x = input and b = bias, and it can be defined with the following formulas:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{2.53}$$

Or

$$\sigma(z) = \frac{1}{1+\exp\left(-\sum_j w_j x_j - b\right)} \tag{2.54}$$

If z = w*x+b is a large positive number, then σ(z) ≈ 1, and if it is a large negative number then σ(z) ≈ 0. Therefore,

> *"It's only when w*x+b is of modest size that there's much deviation from the perceptron model." (Nielsen, 2015)*

Even though it has the capability of taking more versatile inputs and giving more versatile outputs, the sigmoid neuron has one problem. When we have a question which can only be answered yes or no, the sigmoid neuron seldom gives a clear answer. Therefore, in these cases it has become common practice that outputs of 0.5 and higher are interpreted as a yes, and outputs below 0.5 are interpreted as a no. (Nielsen, 2015)

An ANN in the form described in this section, where the output of a neuron in the previous layer is used as an input in the next layer, is called a feedforward neural network.

There are also neural networks which utilize looping, called recurrent neural networks, but as of now they are not as effective as feedforward neural networks and therefore they are not that widely used. To briefly explain a recurrent neural network, it is an ANN where the neurons only fire for a limited time before being quietened, stimulating other neurons which fire for a limited time later on in the ANN. Constructing an ANN this way results in loops not causing any problems as the output of a neuron only affects its input at a later point in time. (Nielsen, 2015)

2.7.2. How an artificial neural network learns

As mentioned by Aggarwal (2018) and Nielsen (2015), an ANN learns by modifying the weights and biases in order to reach the desired output when being exposed to external stimuli in form of a training dataset. For example, if we want to train an ANN to predict future sales, we need a training dataset that contains at least the historical sales numbers. If the dataset also contained information such as weather, campaigns, customer flow and so on, the ANN has the potential to produce more accurate forecasts. The ANN would then try to find the proper weights and biases for the inputs in order to reach the realized sales number for that period. In addition to questions related to weather, campaigns, customer flow and so on, the ANN could for example also identify patterns in the training dataset and use that to an advantage when predicting future sales numbers. When the learning algorithm has finished finding and assigning the proper weights and biases, the ANN needs to be tested on a test dataset in order to confirm that the ANN works as intended in other scenarios. The test dataset contains the same type of information as the training dataset, but it could for example be from another time period. If the accuracy of the forecasts on the test dataset is adequate, it can be concluded that the ANN works as intended for it purpose and it can be applied to predict future sales numbers. (Nielsen, 2015)

To conclude, the learning algorithms that do the heavy work are very complex and can be really hard, if not impossible to design "by hand". Luckily, many techniques for developing the learning algorithm have been created in recent years. Nielsen (2015) summarizes the most recent and most important developments as follows:

> *"Since 2006, a set of techniques has been developed that enable learning in deep neural nets. These deep learning techniques are based on stochastic gradient descent and backpropagation, but also introduce new ideas. These techniques have enabled much deeper (and larger) networks to be trained - people now routinely train networks with 5 to 10 hidden layers. And, it turns out that these perform far better on many problems than shallow neural networks, i.e., networks with just a single hidden layer." (Nielsen, 2015)*

2.7.3. Artificial neural networks as a forecasting method

The following section will treat artificial neural networks as a forecasting method. First, the early forecasting use of neural networks will be presented, then, the findings after the deep learning techniques were discovered will be presented, and finally, the research on neural networks as a retail sales forecasting method will be introduced.

2.7.3.1. Early forecasting use of artificial neural networks

In their article "Forecasting with artificial neural networks: The state of the art", Zhang, Patuwo and Hu (1998) provided a comprehensive overview of the early forecasting use and research on ANNs as well as the ANN modeling issues at the time.

First, the advantages of ANNs as a forecasting method were presented. As opposed to the time series analysis models, an ANN is a self-adaptive method that is able to learn from examples and can find even the most subtle relationships in the data, which would otherwise go unrecognized. Because of this, ANNs are suited for problems where adequate data is available, but the answers require knowledge, which is hard to specify (Zhang, Patuwo and Hu, 1998). Zhang, Patuwo and Hu (1998) mentioned that it is often easier to access extensive data than having the knowledge about the underlying laws that a system uses to generate the data. Thus, ANNs become extremely useful as they have the ability to learn the relationships in a dataset. ANNs can also generalize, often leading to accurate forecasts even though the sample data might be noisy (Zhang, Patuwo and Hu, 1998). Finally, ANNs are non-linear as opposed to the time series analysis methods, which use linear statistics. While the time series analysis models have the benefits of being easy to explain and interpret as well as having the ability to be analyzed in greater detail, they might be inappropriate and provide inaccurate forecasts if the underlying system generating the data has non-linear mechanics (Zhang, Patuwo and Hu, 1998).

The earliest use of an ANN for forecasting purposes dates to 1964, when Hu (1964) used the Windrow's adaptive network for weather forecasting. At the time, a training algorithm for multi-layer networks did not exist and it was not until 1986 when the backpropagation algorithm was introduced by Rumelhart et al. (1986) and Werbos (1988) that ANNs began outperforming the time series analysis models in forecasting (Zhang, Patuwo and Hu, 1998). According to Zhang, Patuwo and Hu (1998), the first successful application of an ANN as a forecasting method was made by Lapedes and Farber (1988) when they designed a feedforward neural network, which could accurately mimic the system using a logistic map and Glass-Mackey equation used to generate the time series used for forecasting. Much progress was made in the years after the findings of Rumelhart et al. in 1986, and in 1993, a forecasting competition was organized by Gershenfeld and Weigend (1993), in which the winner of every category used an ANN model, displaying the dominance of ANNs. Despite this, most of the research from this time focused on other fields and industries than retail. Lapedes and Farber (1988), who conducted the first successful forecast using an ANN studied a chaotic time series which mostly occurs in the field of physics and engineering, which led to many authors following

in the same footsteps in the following years (e.g. Jones et al., 1990, Lowe and Webb, 1991, and Poli and Jones, 1994). Other fields where ANNs were popularly used are forecasting stock prices (e.g. Kimoto et al., 1990, and Grudnitski and Osburn, 1993), foreign exchange rates (e.g. Wu, 1995, and Hann and Steurer, 1996) and electric load consumption (e.g. Park and Sandberg, 1991, Bacha and Meyer, 1992, and Ho, Hsu and Yang, 1992). Zhang, Patuwo and Hu (1998) mentioned a few other forecasting problems that were solved by ANNs but did not receive as much attention as the before mentioned areas. Among these problems are, for example, forecasting airborne pollen (Arizmendi et al., 1993), international airline passenger traffic (Nam and Schaefer, 1995) and tool-wear (Ezugwu, Arthur and Hines, 1995).

Much comparative research between ANNs and the conventional forecasting methods was also made at the time of the research of Zhang, Patuwo and Hu (1998), although nearly none focused on the retail industry. For example, Tang, De Almeida and Fishwick (1991) conducted a study on three business time series where they studied the performance of a simple ANN model compared to an ARIMA model. They found that the ANN outperformed the ARIMA model, when the time series had more irregularity or short memory, meaning that present values do not depend that strongly on past values, but when the time series had long memory, meaning that present values depend strongly on past values, the performance of ANNs and ARIMA models are equal. These results were also confirmed by Kang (1991) who achieved similar results in his research. Hill, O'Connor and Remus (1996) found that ANN models were significantly better than traditional time series analysis methods when forecasting monthly or quarterly data, but when forecasting annual data the time series analsis and ANN methods performed almost equally well. Hill, O'Connor and Remus (1996) also found that ANNs are effective when the time series is discontinuous, confirming the findings of Tang, De Almeida and Fishwick (1991). Nelson et al. (1994) as well as Sharda and Patil (1992) discussed the ability of an ANN to learn seasonal patterns. While the results of Nelson et al. (1994) indicated that ANNs are unable to learn seasonal patterns, Sharda and Patil (1992) argued that the seasonality of a time series does not affect the performance of an ANN. Therefore, it can be implied that an ANN can incorporate seasonality in its predictions even though it is not able to recognize seasonal patterns.

The main challenges of using ANNs were, at the time, the design and building of the models. The number of layers, the number of nodes in each layer, the number of connections, the training algorithm, the performance measures and the training and test datasets, amongst others, must be considered when designing an ANN model, and it was a challenge to find the right combination of all components (Zhang, Patuwo and Hu, 1998). As described by Nielsen (2015), these are the same factors that must be considered when building an ANN model today, with the difference that new deep learning techniques today allow us to train larger ANNs.

As described in Section 2.1, the time series analysis models perform best under certain conditions. Tang, De Almeida and Fishwick (1991) as well as Tang and Fishwick (1993), on the other hand, tried to find the conditions under which ANNs outperform the time series analysis models. These two studies found three conditions under which an ANN performed at its best. First, the bigger the forecast horizon is, the better ANNs performed compared to time series analysis models; secondly, ANNs performed better on time series with short memory, and finally, more input nodes in the ANN model gives better forecasting accuracy. However, Gorr, Nagin and Szczypula (1994) made the important remark that the full power of ANNs might not yet have been discovered when they discussed the reasons to why their ANN did not provide any significant forecast improvement to the regression models they compared it to. This remark is in line with the claim by Nielsen (2015) in the end of Section 2.6.2 that ANNs did not reach their full potential until 2006 when a set of new deep learning techniques was presented.

2.7.3.2. Deep belief networks

In 2006, deep belief networks (DBN) and the methods for training them were introduced in research papers (Nielsen, 2015). According to Nielsen (2015), DBNs are not as popular anymore as they were when first introduced in 2006, as feedforward neural networks and recurrent neural networks has since become more popular, but despite this

they still have several interesting properties. “A fast learning algorithm for deep belief nets” by Hinton, Osindero and Teh (2006) as well as “Reducing the dimensionality of data with neural networks” by Hinton and Salakhutdinov (2006) were the first papers to present how DBNs could be properly trained. One of the two main reasons DBNs gained so much interest is because a DBN is a generative model. While a feedforward network (explained in Section 2.6.1) determines the activation of a neuron later in a network based on the inputs, a DBN also allows for specifying the value of a neuron and then running the process backwards in order to figure out the input value needed for activation. Nielsen (2015) gives a more concrete example by concluding that for an ANN designed to recognize handwriting, this backwards process potentially allows for the ANN to learn how to write and be able to generate text that would look like handwriting. Nielsen (2015) found the following similitude between the generative model and the human brain, which also highlights the advantage of an generative model:

> *“In this, a generative model is much like a human brain: not only can it read digits, it can also write them.”*

The second reason to why DBNs are so interesting is the fact that they can learn unsupervised. As an example, Nielsen (2015) mentions that when an ANN is trained with image data, it can learn features that are useful to recognize other images even though the training data would be unlabeled. Despite these two major advantages of DBNs, the feedforward networks and recurrent neural networks have still become more popular today due to their superior performance. Although DBNs are not as popular as they used to be, the work of Hinton, Osindero and Teh (2006) and Hinton and Salakhutdinov (2006) was not made in vain, as their work provided the base for future methods on successfully training multiple level ANNs with up to ten layers.

Even though the findings introduced in 2006 allowed for larger ANNs to be trained, it did not bring a change to the fact that most of the research around ANNs as a forecasting method circled around other topics than the retail industry. Much of the

research on the classic topics such as stock market forecasting (e.g. Vaisla and Bhatt, 2010, and Wang et al., 2011), electricity demand forecasting (e.g. Kandananond, 2011) and weather forecasting (eg. Abhishek et al., 2012) still existed, and especially topics related to water seemed to be gaining in popularity. For example, Darji, Dabhi and Prajapati (2015) as well as Hung et al. (2009) studied the use of ANNs for rainfall forecasting, Mishra and Desai (2006) used ANNs to forecast upcoming droughts, Abrahart et al. (2012) studied ANNs as a method for river forecasting, which is a combination of rainfall and streamflow modelling, and Ghiassi, Zimbra and Saidane (2008) studied how the urban water demand can be forecasted with ANNs.

2.7.3.3. Artificial neural networks as a retail sales forecasting method

Because of the lack of extensive, well-established research on retail sales forecasting using ANNs, both the research presented before and after the findings in 2006 will be presented in the same section. The most cited work on retail sales forecasting using ANNs is the article "Forecasting aggregate retail sales: a comparison of artificial neural networks and tradition methods" by Alon, Qi and Sadowski (2001), which has already been presented in section 2.2.4. and 2.4.5. of this thesis. To add to the previously presented results, it can be mentioned that the ANN they used had the best performance overall when measuring the average MAPE of the two forecasting periods. The ANN had an average MAPE of 1.50%, whereas the ARIMA model had an average MAPE of 1.67%, the exponential smoothing had an average MAPE of 2.19%, and the regression model had an average MAPE of 2.75%. Despite the margins between the top performing models being small, Alon, Qi and Sadowski (2001) found that there were notable differences between the models. They concluded that while the time series analysis models performed well during stable conditions, the ANN provided a significant improvement when the economic conditions experienced turbulent conditions. It is hard to say if the findings of Hinton,

Osindero and Teh and Hinton and Salakhutdinov in 2006 would have resulted in Alon, Qi and Sadowski (2001) finding ANNs outperforming time series analysis methods also during stable economic conditions or not. As the margins between the methods were small, there is certainly a possibility that it could have happened as even a small improvement brings an edge over the other models. However, as the models are often highly case sensitive it is impossible to say if a deeper neural network would have brought any performance improvements.

Another research paper on ANNs as a retail sales forecasting method, published before the findings in 2006, is the paper by Chu and Zhang (2003) which has already been briefly presented in Section 2.4.5. The results of their study found that an ANN outperformed both a seasonal ARIMA model and a regression model when forecasting monthly sales. However, Chu and Zhang (2003) found that their ANN performed even better when the time series data was deseasonalized, even though Sharda and Patil (1992) argued that the seasonality of a time series did not affect the performance of the ANN. This contradictory finding by Sharda and Patil (1992) suggests that while Chu and Zhang (2003) proved that their ANN is more effective than traditional time series analysis methods, it can still be optimized as deseasonalizing the data improved the performance of their ANN.

While the two aforementioned articles were published before the findings in 2006 and are two of the most cited papers on ANNs as a retail sales forecasting method, it is only after 2010 that retail sales started to become a slightly more popular ANN forecasting topic. In 2013, Li et al. (2013) compared an ANN to an ARIMA model when predicting the weekly retail price of eggs. Their ANN clearly outperformed the ARIMA model for all five weeks predicted, having a forecast error between 0.28% and 0.65% compared to the forecast errors between 2.67% to 3.07% for the ARIMA model. Li et al. (2013) concluded that the results indicate that the ANN is a good tool for short-term forecasting due to the extremely high precision of the forecasts. In a case study by Yu et al. (2017), a recurrent neural network is used on 45 weeks point of sale data on 66 different products in order to forecast the upcoming weekly sales. The results of Yu et al. (2017) are somewhat contradictory to the earlier findings regarding the forecasting performance of

ANNs as they only got an acceptable forecasting error on one fourth of all the products. However, this poor performance may to some degree be explained by the fact that Yu et al. (2017) only have 45 weeks of historical data available which they argue is not enough to build a robust neural network. Also, they do not take seasonality and promotional campaigns into consideration, which are factors that highly affect the sales in the retail industry and, therefore, it is nearly impossible to predict the peaks in sales caused by promotions and seasonality. Despite this, Yu et al. (2017) argued that the ANN they used showed potential for short term retail sales forecasting, especially considering that it had decent result on one fourth of the products despite not accounting for promotions and seasonality. Finally, Chawla et al. (2018) used ANNs to forecast the upcoming sales of the American retail corporation Walmart. They developed one ANN in MATLAB and one using the *neuralnet* package in the R programming environment. The ANN developed with R showed more accurate results than the one created with MATLAB but despite this, both ANNs still occasionally had inconsistent forecasts around some peaks in sales. Chawla et al. (2018) explain these inconsistencies with occasional festivities such as the Super Bowl, which still need manual corrections in the predictions by applying a correction factor. Despite Chawla et al. (2018) not disclosing the actual forecast errors, they claim that they achieved almost perfect accuracy in the end. The only limitations they encountered were caused by the lack of processing power as MATLAB was not able to iterate through the data when the number of layers in the ANN was increased. As a solution to this they suggest that a GPU (graphics processing unit or more commonly known as graphics card) based environment be used in order to be able to handle more layers of perceptrons.

3. Empirical study

This chapter will present the process of the empirical study. The main objective of this study is to answer the research questions by identifying the most suitable forecasting model for a set of retail sales data, as well as determining whether the more advanced machine learning techniques bring a substantial advantage over the time series analysis models.

Section 3.1 describes the methodology of the study, Section 3.2 presents the dataset used to conduct the forecasting, and finally, in Section 3.3, the forecasting is conducted using the selected models and the results of the forecasting are presented, which are later analyzed in Chapter 4.

3.1. Method

As per the definition of Williams (2007), the research approach can be selected based on the type of data needed to respond to the research question. In most cases, the choice of research method lies between a quantitative or a qualitative research approach, but sometimes a mix of both methods might be needed. In the case of this study, a quantitative research method is chosen as numerical data is required to answer the research questions (Williams, 2007). As data is used to objectively measure reality, the research thus stays independent from the researcher (Williams, 2007).

The research conducted in this paper follows the main steps of quantitative research, presented by Williams (2007). First suitable data that can be subjected to statistical treatment is identified and collected, and then mathematical models are applied as the method of data analysis. In the case of this study, a suitable dataset is first acquired, then the data is being analyzed and preprocessed whereafter different mathematical models in

the form of forecasting methods are applied to the dataset. Finally, the results are analyzed using different accuracy measurements and the best performing model is selected.

3.2. Data overview

The dataset used in this study is the popular "Walmart Recruiting – Store Sales Forecasting" dataset published on Kaggle on February 20, 2014, for a forecasting competition by Walmart. Even though the competition ended on May 6, 2014, the dataset is still publicly available. As a result, it has been used in many research papers to date.

The competition dataset consists of four different files containing historical sales data for 45 Walmart stores in different regions, with each store containing several different departments. The store.csv file has three columns containing anonymized information about the type and size of every store. The train.csv file has five columns containing the store number, department number, the date, the weekly sales, and information about whether there is a special holiday that week or not. The test.csv file contains the same information as the train.csv file with the exceptions that the data is from another period and the weekly sales numbers have been withheld. This file was originally the file on which the forecasting was to be conducted on in the competition and it was used to rank the competition entries. However, the test.csv file has not been used in the research of this paper as it does not include the sales data, making it unable to validate the forecasts generated in this research. The final file of the dataset is the features.csv file which has twelve columns containing the store number, the date, the temperature, the fuel price, promotional markdowns, the consumer price index, the unemployment rate, and information on whether there is a special holiday that week or not. Table 3.1 to 3.4 lists the attributes and the description of the attributes of each file in the dataset.

In total, the train.csv file contains 421570 rows of historical sales data and the test.csv file contains 115065 rows of future sales to be predicted. While the train.csv file might

contain enough information to predict the future sales using some of the time series analysis models such as the ARIMA model and exponential smoothing, the machine learning methods also need the information from the store.csv and features.csv files. This problem will be tackled in the upcoming Section 3.2.1.

Table 3.1 stores.csv data description

Attribute	Description
Store	ID number of each store
Type	The type of each store
Size	The size of each store

Table 3.2 train.csv data description

Attribute	Description
Store	ID number of each store
Dept	ID number of each department
Date	The date of Friday each week
Weekly_Sales	The weekly sales in USD
IsHoliday	Whether there is a special holiday during the week

Table 3.3 test.csv data description

Attribute	Description
Store	ID number of each store
Dept	ID number of each department
Date	The date of Friday each week
IsHoliday	Whether there is a special holiday during the week

Table 3.4 features.csv data description

Attribute	Description
Store	ID number of each store
Date	The date of Friday each week
Temperature	The average temperature of the week in the region
Fuel_Price	The cost of fuel in the region
MarkDown1	Anonymized data related to promotional markdowns
MarkDown2	Anonymized data related to promotional markdowns
MarkDown3	Anonymized data related to promotional markdowns
MarkDown4	Anonymized data related to promotional markdowns
MarkDown5	Anonymized data related to promotional markdowns
CPI	Consumer price index
Unemployment	Unemployment rate
IsHoliday	Whether there is a special holiday during the week

3.2.1. Data manipulation

In order to use machine learning models to predict retail sales, the data needs to be manipulated to some degree before the machine learning techniques are applied. First, the train.csv, stores.csv and features.csv files must be merged using Python for all useful

features to be found in one single file. The stores.csv and train.csv files are first merged whereafter the features.csv file is merged into the previously merged file. Since the original files contain some duplicate variables between each other, the duplicate variables are dropped in the merging process. In addition to this, the values of the markdowns are corrected so that every missing value is replaced with a zero, as a markdown of zero equals no markdown at all. This must be done since the original features.csv only contained values for the markdowns when a markdown was active; otherwise, the field was left empty. Because of this, there are a lot of missing values in the markdown columns which cause problems for the machine learning techniques. As a result of the merging and manipulation, a file containing all the attributes presented in Table 3.5 is attained.

Table 3.5 Merged data description

Attribute	Description
Store	ID number of each store
Dept	ID number of each department
Weekly_Sales	The weekly sales in USD
IsHoliday	Whether there is a special holiday during the week
Type	The type of each store
Size	The size of each store
Temperature	The average temperature of the week in the region
Fuel_Price	The cost of fuel in the region
MarkDown1	Anonymized data related to promotional markdowns
MarkDown2	Anonymized data related to promotional markdowns
MarkDown3	Anonymized data related to promotional markdowns
MarkDown4	Anonymized data related to promotional markdowns
MarkDown5	Anonymized data related to promotional markdowns
CPI	Consumer price index
Unemployment	Unemployment rate
Date	The date of Friday each week

In addition to the aforementioned data manipulation, further data modifications are necessary for the chosen machine learning models. First, the date could be transformed from the original form to the month number. This is required because the date of the Friday of a specific week can be different each year even though they represent the same week. Some might argue that instead of the number of the month, the week number should be used. However, through trial and error, it was found that the choice of month or week number did not have a significant impact on the forecasting performance. The second problem to be addressed is the "Type" column. There are three different types of stores in the original data, represented by A, B and C. String variables are difficult to be interpreted by machine learning algorithms and therefore a more easily interpretable format is needed. The solution to the problem is transforming the original "Type" column into three new columns containing dummy variables, each representing one of the original store types. By doing this, the string variables are transformed into numerical variables.

Since the original test.csv file provided by Walmart does not contain the weekly sales numbers it is impossible to validate the forecasting results attained by the model fit on the training data. Therefore, the final step is to split the train.csv into a training set and a test set before the data exploration and forecasting begins, as suggested by Domingos (2012). The splitting is done by removing the last 12 weeks of data from our current dataset and placing them in a different file to be used for validation of the attained forecasting results. As a result of this, a new training dataset containing 131 weeks' worth of sales data and a test, or validation, dataset containing 12 weeks' worth of sales data is acquired.

In addition to the data manipulation presented in this section, some minor steps of data manipulation must be taken before each forecasting method is applied. These manipulations will be presented in conjunction with the forecasting in Section 3.3.

3.2.2. Data exploration

Data exploration is an important step of the forecasting process, as was mentioned in Section 2.1, a good understanding of the available data is needed in order to be able to select the most suitable forecasting models.

The first step of the data exploration is to view the size of the new training dataset created in Section 3.2.1. As can be seen in Table 3.6, the training set contains 386,005 rows and 18 columns which should be sufficient to both train the machine learning models and to calculate the parameters of the time series analysis models. There are no definite rules on how much training data is needed to train an effective machine learning model, and a common understanding seems to be that the required size of the dataset depends on the task to be performed (eg. Domingos, 2012). For the task of sales forecasting in this thesis, the training dataset of 386005 rows and 131 weeks' worth of data should be sufficient as, for example, Alon, Qi and Sadowski (2001) managed to generate accurate forecasts using much fewer historical data points.

Table 3.6 Size of training data

Array	Size
Rows	386005
Columns	18

After this, the relationships among variables in the data need to be examined to attain a basic understanding of how each variable affects the weekly sales. The scatter plots in Figures 3.1-3.9 display the relationship between the weekly sales and all other variables except the markdowns. As can be seen in Figure 3.7 and 3.8 the weekly sales vary a lot between the stores and between the departments. In addition to this, the weekly sale seems to be decreasing as the average temperature reaches the high and low extremes,

and bigger stores seem to logically have slightly bigger weekly sales. On the other hand, the fuel price and consumer price index does not initially give the indication of affecting the weekly sales based on the scatterplots. To further confirm the findings of the high variation in sales between stores and between departments, the bar charts of the average weekly sales per store and average weekly sales per department are shown in Figure 3.10 and 3.11.

Figure 3.1 Scatterplot of Weekly_Sales and Fuel_Price

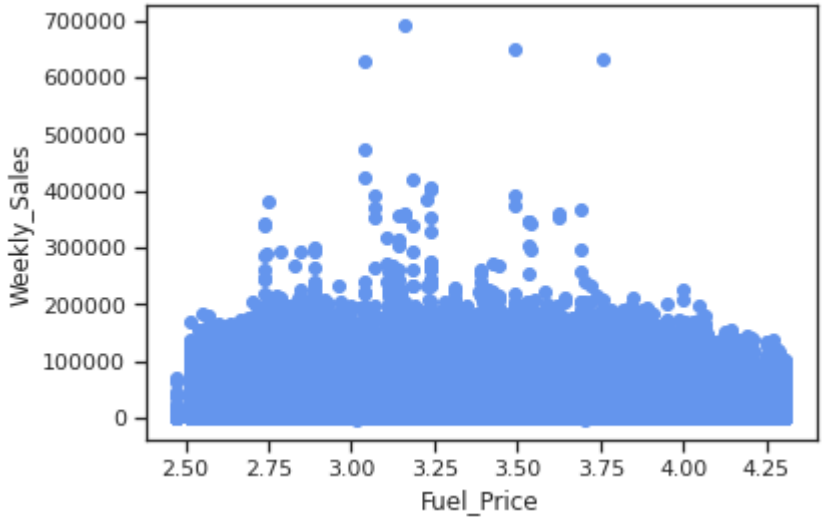


Figure 3.2 Scatterplot of Weekly_Sales and Size

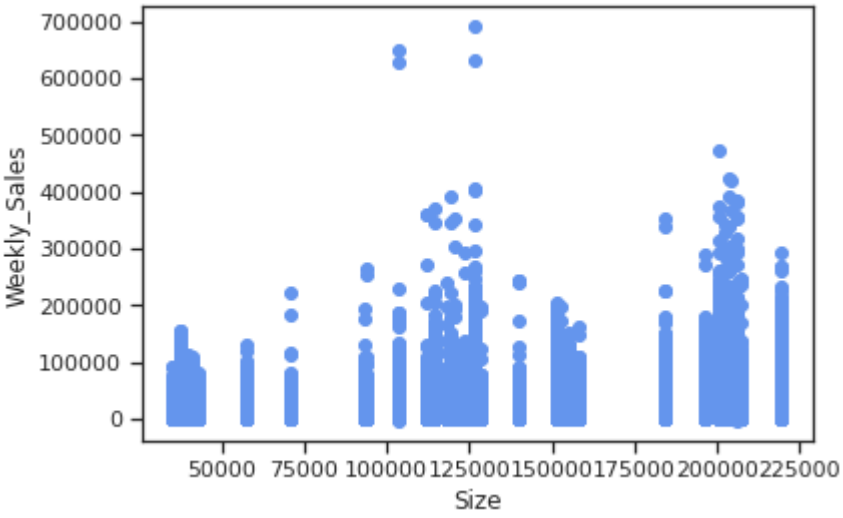


Figure 3.3 Scatterplot of Weekly_Sales and Type

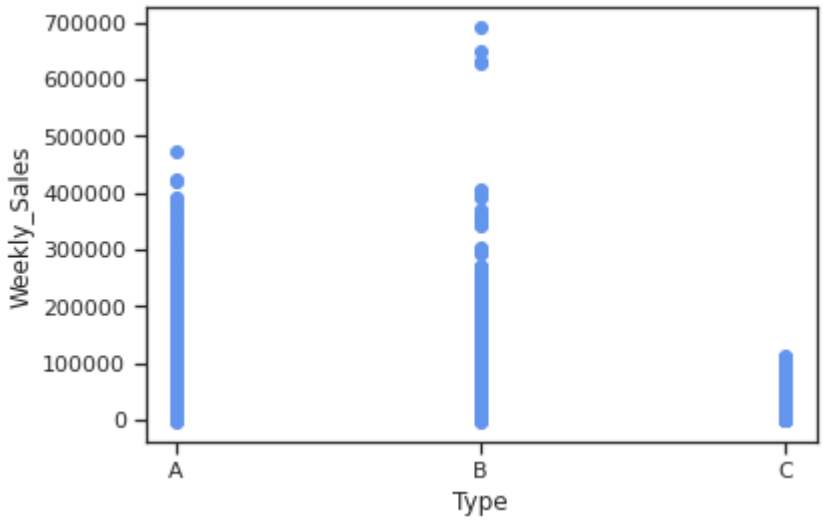


Figure 3.4 Scatterplot of Weekly_Sales and IsHoliday

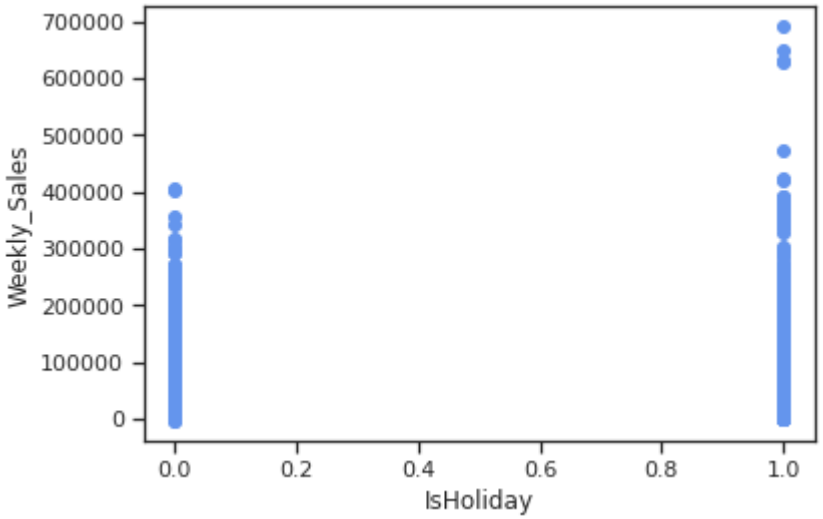


Figure 3.5 Scatterplot of Weekly_Sales and Unemployment

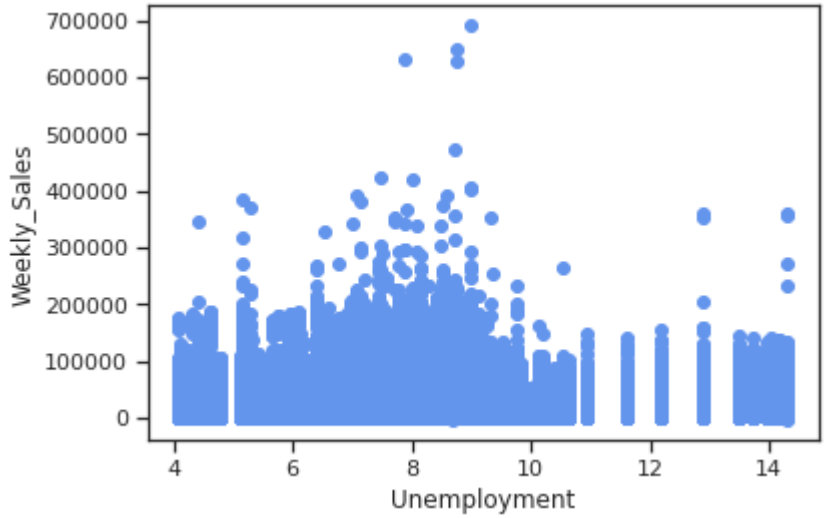


Figure 3.6 Scatterplot of Weekly_Sales and Temperature

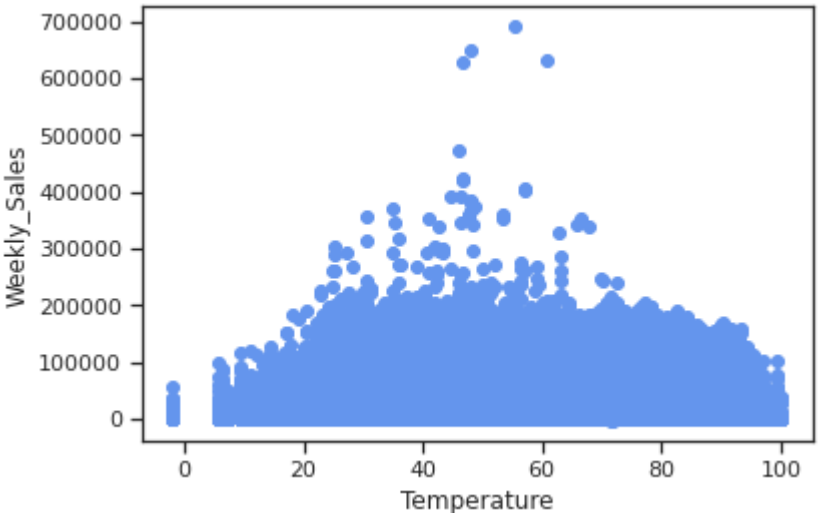


Figure 3.7 Scatterplot of Weekly_Sales and Store

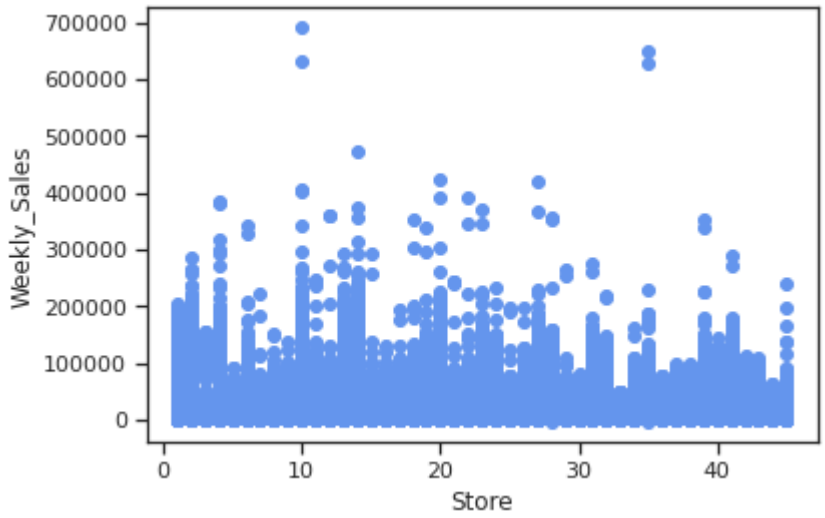


Figure 3.8 Scatterplot of Weekly_Sales and Dept

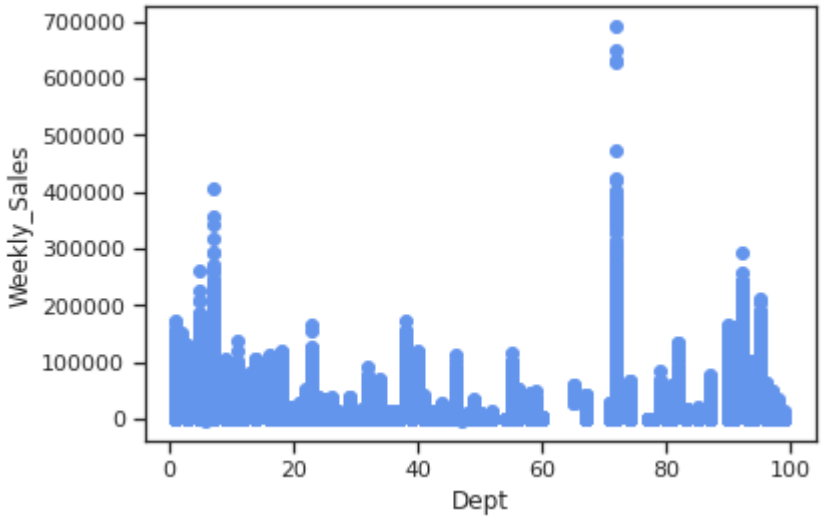


Figure 3.9 Scatterplot of Weekly_Sales and CPI

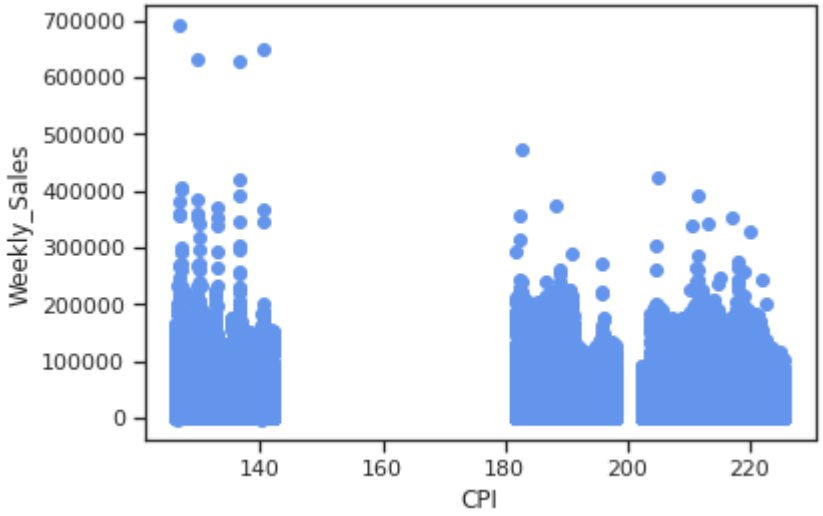


Figure 3.10 Average weekly sales per store

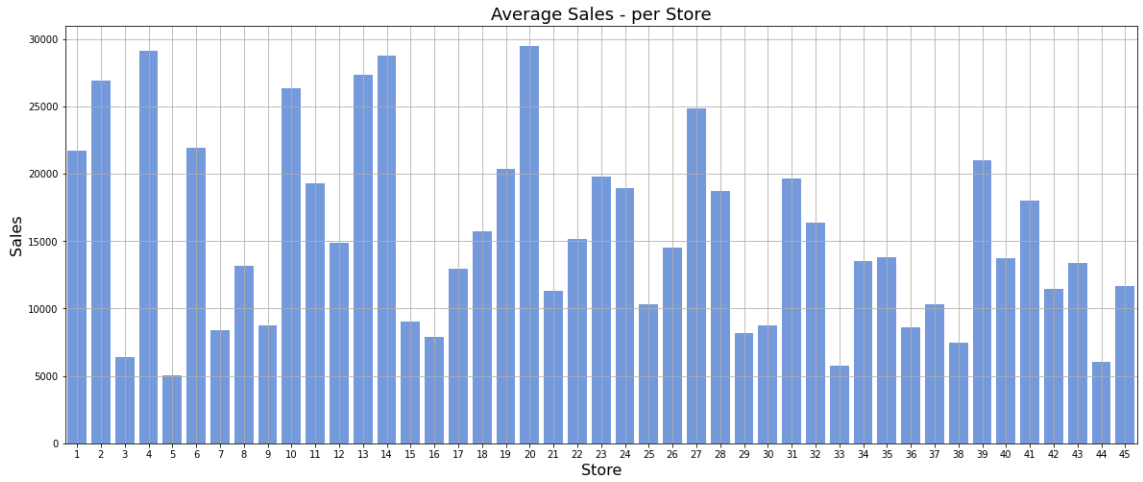
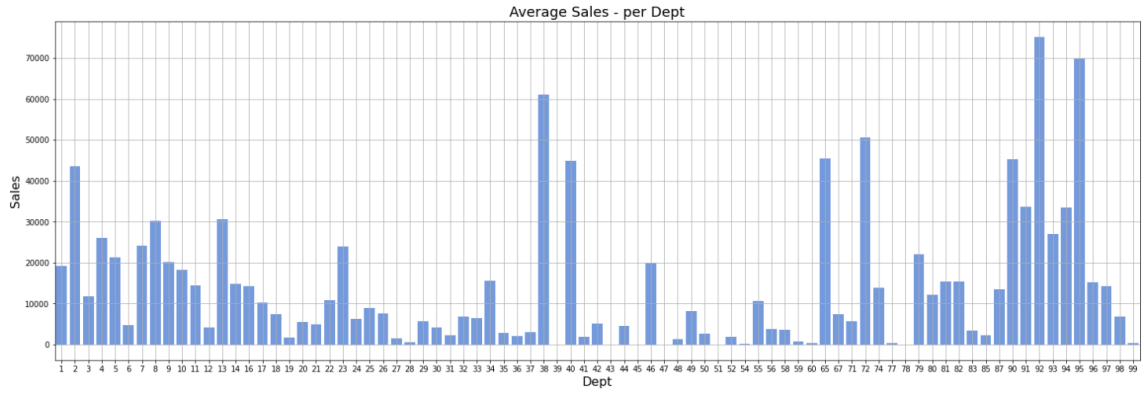
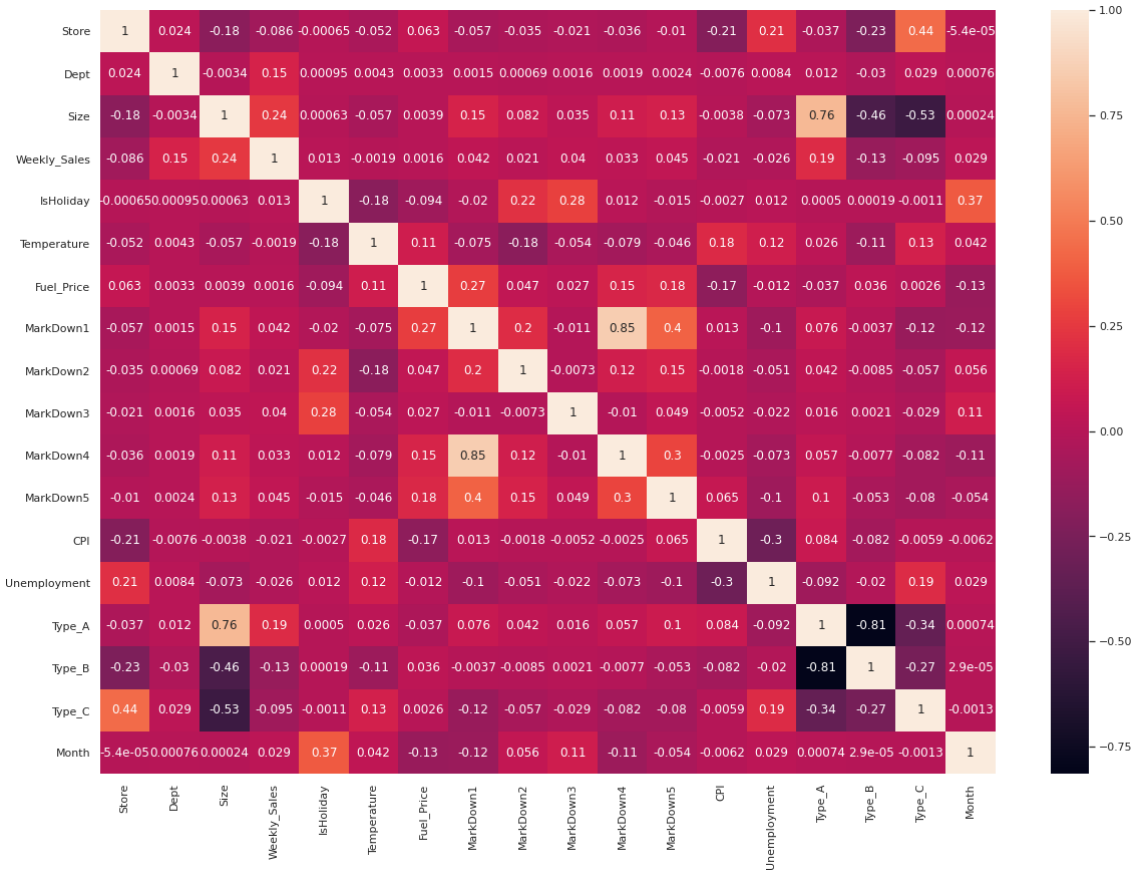


Figure 3.11 Average weekly sales per department



To further examine the relationships in the data, a correlation matrix using the Paerson correlation coefficient is created in Figure 3.12. The Paerson correlation coefficient takes values between 1 and -1 with positive correlation indicating that when one variable increase, the other variable also increases. If the correlation is negative, on the other hand, it indicates that the values move in opposite directions. Finally, correlations close to zero indicates that there is a weak correlation or no correlation at all. (Benetsy et al., 2009). As can be seen from the correlation matrix in Figure 3.12, the size, department and type are most strongly correlated to the weekly sales. In addition to this, other variables which are strongly connected to each other must be found. Since two variables with a strong correlation would bring similar information to the model, one of the strongly correlated variables need to be dropped when the machine learning model is developed and trained. One pair of such strongly connected variables are, for example, MarkDown1 and MarkDown4.

Figure 3.12 Correlation matrix



In addition to the data exploration in this section, some minor data exploration will take place before the final forecasting methods are selected and applied. This data exploration will be presented in Section 3.3 in the process of model selection.

3.3. Forecasting

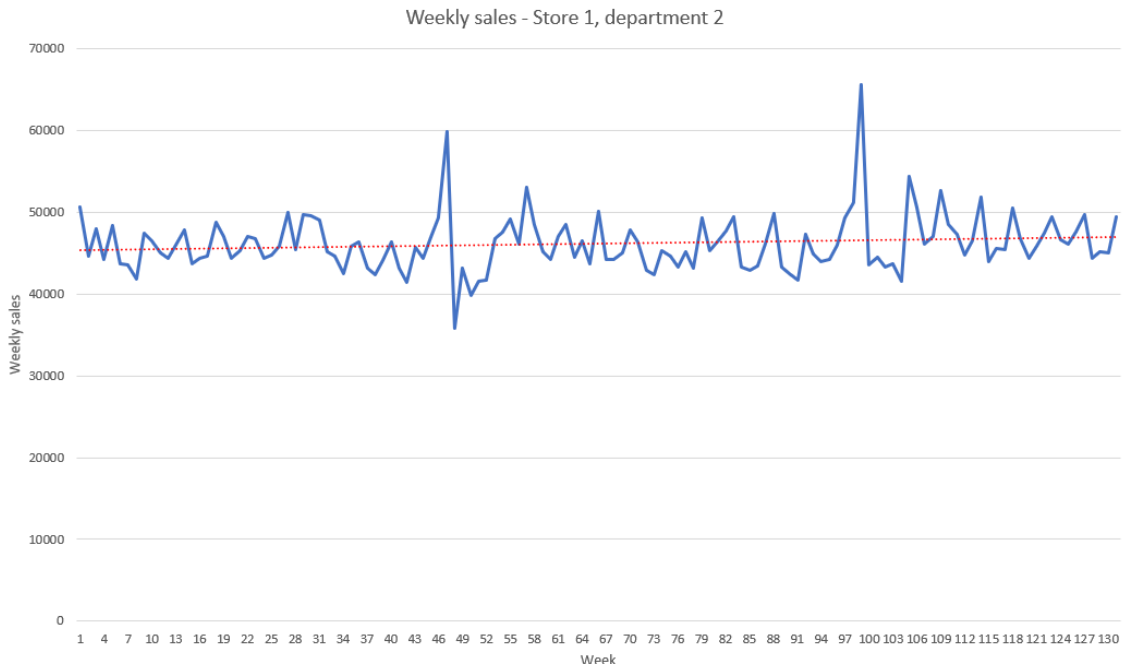
In this section the selection of forecasting methods as well as the forecasting process using all selected methods are presented. First, the time series forecasting methods will be

applied to the data, whereafter the machine learning techniques will be used. Following the forecasting, the results of the forecasting will be presented in Section 3.4.

Since the training dataset contains the sales data of 61 departments in 45 different stores, forecasting the sales of every department of each store would be a heavy task. Therefore, a single department of a single store is selected for the forecasting task of this thesis. Using a random number generator, department 2 of store 1 is selected. The forecasting methods presented in this section and applied to the aforementioned department can also be applied to the other departments to forecast their upcoming sales.

To determine which time series analysis models are most suitable for the dataset, a time series plot is made (Figure 3.13). Since the time series shows clear signs of seasonality with significant sales peeks on the week before Christmas, a seasonal ARIMA model and a triple exponential smoothing, or Holt-Winter's exponential smoothing, are selected from the time series analysis model. In addition to these, a simple moving average will be used as a benchmark model, expected to be outperformed by the other models. As a simple moving average is suitable for a time series without trend but cannot handle seasonality, and the time series plot in Figure 3.13 shows almost no trend but clear seasonality, forecasting models designed to handle seasonality should easily outperform the moving average Therefore, it can be determined that a forecasting model is not reaching sufficient accuracy if it is not able to outperform the simple moving average. In addition to the time series analysis methods mentioned above, a linear regression model, a neural network, and three decision trees will be applied to the dataset.

Figure 3.13 Weekly sales of department 2 of store 1



To determine the optimal ARIMA and triple exponential smoothing models, the software Forecast Pro was used. The best ARIMA model was determined to be ARIMA(1, 0, 1)(0, 1, 0), and the triple exponential smoothing model chosen has a level factor of 0.1614, a trend factor of 0.001611 and a seasonality factor of 0.698. By observing the time series plot in Figure 3.13, these model selections can be deemed reasonable as the time series exhibits barely any trend, excluding the need for differencing in the ARIMA model, and exhibits a clear seasonality, explaining the high seasonal factor in the triple exponential smoothing model. The optimal moving average model was also determined using Forecast Pro. Forecast Pro determined that the most suitable moving average model should include 44 past observations, and, through trial and error, it was confirmed that Forecast Pro had selected the correct number of past observations.

When it comes to the machine learning methods, a different approach is chosen. The software STATA is used to develop the linear regression model while models developed by Scikit-learn for Python are used to build the neural network and decision tree models. Two different linear regression models are developed – one specific model for department

2 of store 1, and one general model that could be applied to every store. This is done to see if a general model can produce forecasts with good accuracy. Furthermore, the decision tree models and the neural network model built by the Scikit-learn models are all general models which can be applied to any department of any store. The adjusted R-squared value, which determines if an additional independent variable brings more explanatory value to the regression model, was used to determine the variables included in the linear regression models. The department-specific model included the variables Fuel_Price, IsHoliday, MarkDown1, MarkDown2, MarkDown4, CPI and Unemployment, while the general model included the variables Fuel_Price, Temperature, IsHoliday, MarkDown1, MarkDown2, and MarkDown4, as well as dummy variables for each store and each department. When it comes to the decision tree models, three different models were chosen. All the models chosen – the decision tree regressor, the extra tree regressor and the random forest regressor – were chosen because of their popularity. Based on the data exploration in Section 3.2.2, the variables chosen for these models, in addition to Weekly_Sales, are Store, Dept, IsHoliday, Temperature, MarkDown1, MarkDown2, MarkDown4, Month and the dummy variables for store type. Therefore, Date, CPI, Fuel_Price, Unemployment, MarkDown3 and MarkDown5 were dropped from the dataset created in Section 3.2.1. Finally, the neural network model selected is the MLPRegressor, or multi-layer perceptron regressor, by Scikit-learn. The same variables used for the decision trees are also used to train the neural network.

3.4. Results

This section will present the results of the forecasting through visualizations and different accuracy measures. The results will also be briefly discussed in this section, but a more thorough analysis will be presented in Chapter 4. The measurement of forecasting error chosen for this study is the MAPE, as it makes it easy to compare the results of

different studies. Other forecasting error metrics, such as the MAE, are dependent on the scale of the data used, and therefore, difficult to be compared between studies.

Figure 3.14 and 3.15 visualize the forecasted sales and the forecast error acquired by all forecasting methods when they are applied to department 2 of store 1 on the new test set developed in Section 3.2.1. As can be seen in the graphs, the neural network underperformed considerably compared to the other models by constantly under-forecasting by at least $12 000. The MAPEs in Table 3.7 also confirm this, as the neural network has an MAPE that is over ten times bigger than the best performing model, and almost three times bigger than the second worst performing model.

By simply comparing the MAPEs in Table 3.7 the ARIMA model has the best performance, followed by the exponential smoothing, the department-specific linear regression model, and the extra tree regressor. However, since the forecasted period is relatively long, stretching over three months, the MAPE does not tell everything. For instance, when looking at the MAPE for each forecasted month, the ARIMA model only had the best performance for the forecast of the final month. The decision tree regressor had the best MAPE for the first month predicted (2.92%) and the extra tree regressor had the best MAPE for the second month predicted (2.78%). In Table 3.9, the MAPEs of the one month ahead, two months ahead and three months ahead forecast are visualized. From this table it can be seen that the performance of the ARIMA model and the extra trees regressor stayed fairly close to each other up until month two, but the forecasting performance of month 3 lead to the ARIMA model achieving the best overall MAPE. The MAPEs of Table 3.9 also show that while the accuracy of most model decreased the further in the future was predicted, the ARIMA model and the random forest regressor had their most accurate results on the three months ahead forecasts. All these findings indicate that most of the forecasting models behave and perform differently, as some models are more accurate on a short-term forecast while others perform better on long-term forecasts.

Figure 3.14 Forecasted weekly sales

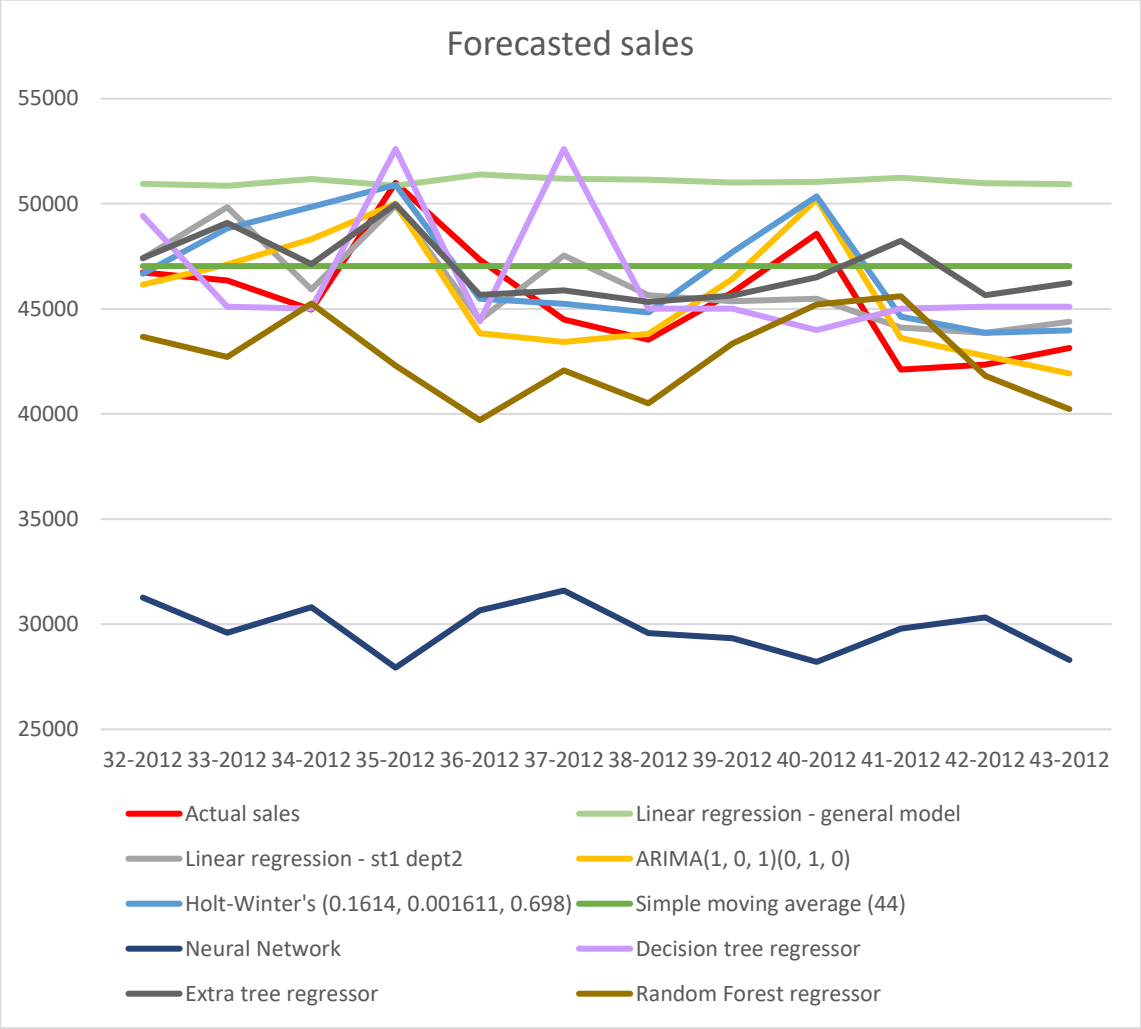


Figure 3.15 Forecast error

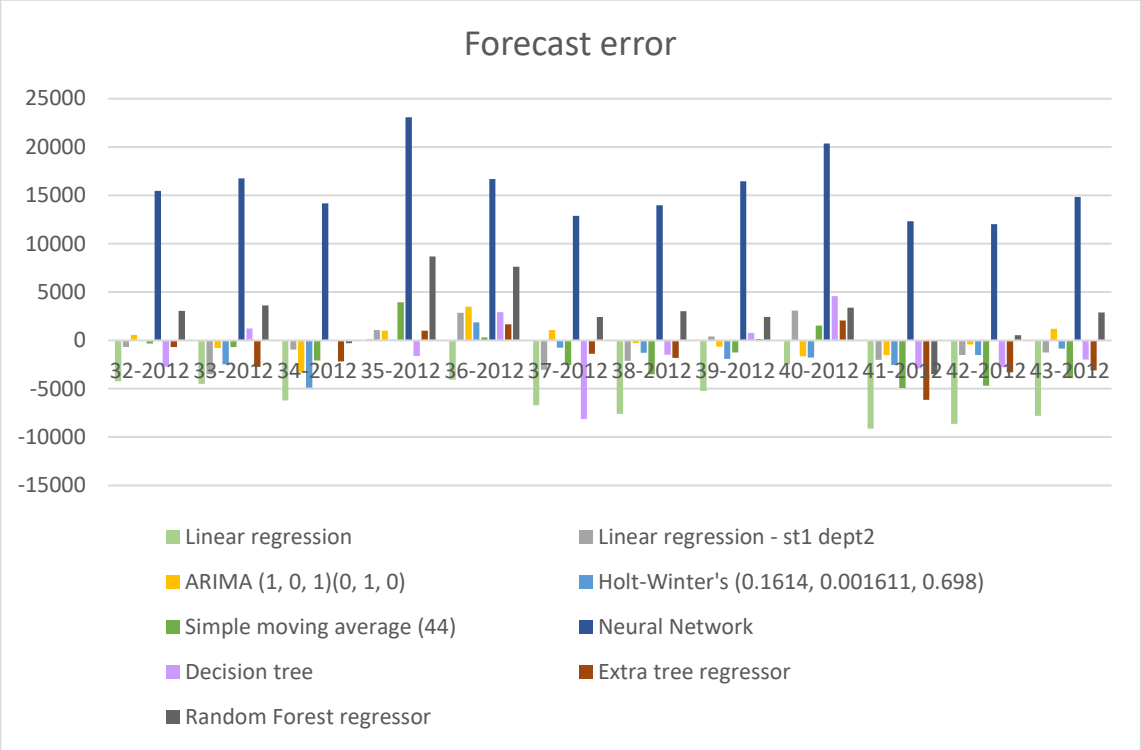


Table 3.7 MAPE

	LRgen	LRst1dept2	ARIMA	ES	SMA	NN	DecisionTreeRegressor	ExtraTreesRegressor	RandomForestRegressor
MAPE	12.54%	4.12%	2.90%	3.71%	5.55%	34.35%	5.72%	4.92%	7.43%

Table 3.8 MAPE per month

	LRgen	LRst1dept2	ARIMA	ES	SMA	NN	DecisionTreeRegressor	ExtraTreesRegressor	RandomForestRegressor
Month1	8.20%	3.31%	3.06%	4.13%	3.62%	36.49%	2.92%	3.54%	8.01%
Month2	13.12%	4.66%	2.95%	3.19%	4.28%	33.06%	7.37%	2.78%	8.46%
Month3	16.30%	4.39%	2.68%	3.79%	8.74%	33.50%	6.85%	8.44%	5.81%

Table 3.9 MAPE of 1 month ahead vs. 2 months ahead vs. 3 months ahead forecast

	LRgen	LRst1dept2	ARIMA	ES	SMA	NN	DecisionTreeRegressor	ExtraTreesRegressor	RandomForestRegressor
1Month	8.20%	3.31%	3.06%	4.13%	3.62%	36.49%	2.92%	3.54%	8.01%
2Month	10.66%	3.99%	3.01%	3.66%	3.95%	34.78%	5.15%	3.16%	8.24%
3Month	12.54%	4.12%	2.90%	3.71%	5.55%	34.35%	5.72%	4.92%	7.43%

In addition to the MAPE there are a few other metrics worth examining when comparing the performance of forecasting models. First, the over- and under-forecasting is examined. As can be seen in Table 3.10 most models tend to over-forecast on more occasions than they under forecast. Also, as shown in Table 3.11, the average over-forecast tends to be bigger than the average under-forecast. In other words, when the model under-forecasts, it tends to under-forecast with a smaller margin than it over-forecasts with. Though, this is not the case with all models as, for example, the random forest regressor is more keen to under-forecast and also under-forecasts by about two times as much as it over-forecast.

Table 3.10 Number of weeks over- and under-forecasted

	LRgen	LRst1dept2	ARIMA	ES	SMA	NN	DecisionTreeRegressor	ExtraTreesRegressor	RandomForestRegressor
Over	11	8	7	9	9	0	8	8	2
Under	1	4	5	3	3	12	4	4	10

Table 3.11 Average over-forecast and average under-forecast

	LRgen	LRst1dept2	ARIMA	ES	SMA	NN	DecisionTreeRegressor	ExtraTreesRegressor	RandomForestRegressor
Over	13.66%	4.25%	2.72%	4.47%	6.11%	-	6.07%	6.11%	4.46%
Under	0.26%	3.87%	3.15%	1.41%	3.86%	34.35%	5.00%	2.53%	8.02%

Another metric worth examining is the number of times a model has an absolute percentage error under a predefined percentage. This is done to determine the frequency of accurate forecasts by a model. Table 3.12 presents the number of weeks the forecasting models reached an absolute percentage error below 0.5%, 2% and 4%. Once again the ARIMA model is one of the best performing models with 8 forecasted weeks with an absolute percentage error below 4%. While the ARIMA model did not manage to produce a single absolute percentage error below 0.5%, it was able to achieve a forecast error below 2% in 6 of 12 forecasted weeks, resulting in the overall MAPE below 3%. The other top performing models on the overall MAPE also performed well on this metrics with at least five forecasted weeks with an absolute percentage error below 5.

Table 3.12 Number of weeks absolute percentage error is below x%

	LRgen	LRst1dept2	ARIMA	ES	SMA	NN	DecisionTreeRegressor	ExtraTreesRegressor	RandomForestRegressor
>0.5%	1	0	0	2	0	0	1	1	0
>2%	1	2	6	4	3	0	2	3	2
>4%	1	6	8	8	5	0	5	5	2

As a conclusion on the forecasting results it can be said that the empirical study confirms, at least to some degree, the findings of the literature review. The different forecasting methods are suitable for different applications, depending on if the long-term or short-term forecast is more important to the retailer. Also, if a forecasting model performed well on one performance metric, it usually also had decent results on the other metrics. Based on the forecasting results on the Walmart data, the conclusion can be drawn that the ARIMA model is the most suitable method for this dataset. This conclusion can be drawn from the consistent results of the ARIMA model throughout the performance metrics. The results of the forecasting will be analyzed more deeply in the upcoming Chapter 4.

4. Discussion

The following chapter focuses on discussing the results of the forecasting done in Chapter 3 on a more in-depth level. In addition to discuss the results, possible reasons to the subpar performance of a few models as well as possible improvements will be discussed in this chapter. Finally, the research questions will be answered.

4.1. Empirical study discussion

When discussing the forecasting results attained in Chapter 3, it is important to note that the results could be completely contradictory to the ones of this study if another department would be chosen for analysis. As was found in the literature review, forecasting methods are highly case specific and therefore a definite truth regarding the best forecasting method cannot be found.

The obvious stand-out in the results is the poor performance of the neural network model. In addition to the MAPE of 34.35%, the neural network was also the only model that constantly under-forecasted and had a MAPE above 10% for each forecasted month. When viewing the forecasted weekly sales in Figure 3.14 it is also apparent that the neural network failed to mimic the pattern of the actual sales, and instead, often moved in the opposite direction of them. The reasons to the poor performance of the neural network is hard, if not impossible to be determined as a neural network produces a model that is nearly impossible to interpret (Nielsen, 2015). However, since the other machine learning techniques used were able to produce far more accurate forecasts using the same data and the same variables, the possibility of an over-specified model can be ruled out. Also, when the neural network was trained only using the historical data of department 2 of store 1, it was able to produce accurate forecasts, leading to the supposition that the neural network had problems separating the stores and departments from each other.

When creating the models for linear regression, two separate models were developed to determine if a general forecasting model outperforms a department specific model, or vice versa. The general model created contained dummy variables for the store number and department number, and the other independent variables are identical to those in the department specific model. As expected, the performance of the general model was poor as it attained a MAPE of 12.54% and over-forecasted on all but one week. The poor performance is most likely caused by the fact that the general model was trained with all available data on all stores, while the specific model was only trained on the data of department 2 of store 1. As a result of this, the general model has to consider characteristics of all departments in all stores, while the specific model only considers the characteristics of the specific department of the specific store, resulting in more accurate results for the specific model.

Quite surprisingly, none of the machine learning models managed to outperform the ARIMA and exponential smoothing models on the whole twelve-week period forecasted. The machine learning models performed well on certain periods, managing to keep up with the time series analysis methods when looking at the performance on one-month periods, but they lacked the consistency to produce accurate forecasts throughout the whole three-month period. For example, the decision tree regressor would most likely have been more competitive had it not been for the major over-forecast in the sixth week forecasted, and the extra trees regressor was a competitive model up until the final three weeks of the forecast. On the other hand, should the pattern of the weekly sales (Figure 3.13) not have been so consistent, the performance of the time series analysis methods could have been much worse as they rely solely on historical sales data. The only major disruptions in the time series are the Christmas weeks of each year, falling on the same week each year. Also, the fact that the forecasted period did not fall during Christmas might possibly be an advantage to the time series analysis models as there is no way knowing how they would have handled the disruption. It is also worth noting that the performance of the simple moving average used as the benchmark model would most likely have been worse had the forecast period been during Christmas.

Comparing the results of the two time series analysis methods to each other, the results were as expected. The MAPE of both the ARIMA model and the exponential smoothing model were within one percent of each other, which confirms the findings of the literature review showing that the models are close to each other regarding forecast accuracy. In the long-term forecast the ARIMA model outperformed the exponential smoothing model, which is also expected as the exponential smoothing model developed by Forecast Pro did not include a damping factor. This most likely results in a slight over-forecast in the long run. Also, the fact that the ARIMA model has zero degrees of differencing while the smoothing weight of the exponential smoothing is close to zero confirms that the models behave as they should, since the time series does not show any substantial trend. One major difference between the two model can be found in the over- and under-forecasting. While both models tend to over-forecast more frequently than they under-forecast, the ARIMA model over-forecasts less than the exponential smoothing model. On the other hand, the exponential smoothing under-forecasts by a smaller margin than the ARIMA model and therefore it is up to the user to decide whether under- or over-forecasting is more desired.

Despite the clear superiority of the time series analysis models, it is worth noting that the MAPE of all models, except for the neural network and general linear regression model, were below 10% which according to Lewis (1997) is a sign of a reasonably accurate forecasting model. Therefore, it can be concluded that none of these models were faulty but the time series analysis models were simply more suitable for the available data. However, the statement of Lewis (1997) is made 24 years ago and therefore somewhat outdated. As technology have advanced and many new models have been developed in the past 24 years, the 10% threshold suggested by Lewis (1997) should be reconsidered. As was found in the literature review and empirical study of this thesis, a forecasting accuracy of 5% or lower is not unusual, and therefore a new threshold could today be set at 5%.

There are a few areas of the forecasting on which a small improvement could possibly have led to a more accurate forecast. First, while the data provided by Walmart include many useful variables, some minor changes could have brought notable performance

boosts to some of the model. While the weekly average temperature is useful information to have, a better measurement could have been the difference to the monthly average temperature. By having the data in the form provided by Walmart, the same temperature is assigned the same weight no matter when it occurs. However, having a, for example, 15 degrees Celsius temperature in February and July most likely has substantially different effects on the weekly sales. Therefore, changing the temperature to the difference to the monthly average gives a more relevant variable to be used in the machine learning models. Another improvement that could result in better forecasting performance would be designing the machine learning models from scratch instead of using predeveloped models. The task would require substantial knowledge and expertise, but a model specifically designed to suit the available data would increase the forecasting accuracy with a high probability. However, it must be considered if a small improvement in forecasting accuracy is worth the extra investment when freely available machine learning models are able to produce forecasts with MAPEs below 5%.

4.2. Research questions

The aim of this thesis has been to understand how the different methods of sales forecasting work as well as comparing the time series analysis methods to machine learning methods. This thesis has, to some degree, been able to answer the following research questions:

1. *Do forecasting methods utilizing machine learning techniques perform better than the time series analysis models?*

2. *Which forecasting method gives the best performance in sales forecasting?*

While there are no definite answers to either research question, both questions can still be answered. The answers to both research questions must be derived from both Chapter 2 and Chapter 3. If the answer to the first question is based solely on the empirical study, the answer is no. However, as many studies presented in the literature review showed, the machine learning techniques have the potential to and can perform better than the tradition forecasting methods. As the results varied, with some studies showcasing superior performance of the machine learning models and others exhibiting superior performance in the time series analysis models, the conclusion can be drawn that determining the best model is highly case specific. Therefore, it is advised that multiple forecasting models are compared before a final model is chosen for the sales forecasting task. It must be noted, however, that while the performance of a forecasting method can easily be measured in the form of a forecasting error, many machine learning models bring the advantage of interpretability. While the time series forecasting models base their upcoming forecasts simply on previous sales figures, most machine learning models gives the user an understanding of which factors affect the sales and to which degree. In some cases with small differences in performance between time series analysis methods and machine learning methods, the advantage might fall to the machine learning methods because of their explanatory power.

The second research question could once again be answered solely based on the empirical study of this thesis, in which case the answer would be the ARIMA model, but the answer would then only reflect one case. As with the first research question, the literature review showed great variety in the answers to this question. Despite this, the best performing model based on the literature review is either an ARIMA, an exponential smoothing, a neural network, or a decision tree model. While the neural network performed poorly in the empirical study of this thesis, it was often one of the top performing models in the studies covered in the literature review. The ARIMA and the exponential smoothing models were also almost always in the top three best performing models in the covered studies, showing their consistent performance. However, since there are countless algorithms and data manipulation options not covered in this thesis, and the performance of the covered forecasting methods has proven to be highly case specific, it

is impossible to name a single forecasting method that for certain is the best performing model for sales forecasting.

4.3. Future research

Some areas of possible future have been discovered during the process of this thesis. The performance of pre-built machine learning models, such as the ones made available by Scikit-learn, could be compared to machine learning models built from scratch. Since the literature review showed that the performance of the forecasting models is highly case specific, there is potential for machine learning models tailored according to the characteristics of the available data to produce more accurate forecasts.

Additionally, the many machine learning models left out of this thesis could also be examined to give a more definite answer to the second research question of this thesis. For example, the gradient boosting, the support vector machines and the k-nearest neighbors algorithms which all have potential to produce accurate sales forecast were not treated in this thesis.

Furthermore, the forecasting performance of hybrid machine learning models would be beneficial to examine. In recent years, hybrid machine learning models have gained popularity (eg. Shon & Moon, 2007, Mohan, Thirumalai & Srivastava, 2019, and Keshtegar et al., 2021) and their suitability for sales forecasting should be studied.

5. Svensk sammanfattning

5.1. Introduktion

För att ett företag inom detaljhandeln ska kunna driva lönsam verksamhet finns det två huvudsakliga faktorer som måste beaktas: en tillräckligt stor och positiv kundkrets samt en hållbar kostnadsstruktur. Medan upprätthållandet av kundkretsen främst handlar om att känna till vem kunden är, innefattar upprätthållandet av den hållbara kostnadsstrukturen betydligt fler ämnesområden. Allt från planering av arbetsscheman för att undvika över- eller underbemanning till att upprätthålla en korrekt lagernivå för att undvika överskottslager samtidigt som produkten inte får ta slut, är involverat i upprätthållandet av en hållbar kostnadsstruktur. Trots att det finns väldigt många faktorer som påverkar kostnadsstrukturen kan de flesta faktorer ändå härledas till samma påverkande faktor: efterfrågan på den sålda produkten. Ifall en detaljhandlare har tillgång till prognoser för den kommande efterfrågan eller försäljningen av en produkt kan hen utnyttja informationen till att främja kostnadsstrukturen. Till exempel kan situationer då över- eller underbemanning förekommer undvikas då bemanningen kan planeras enligt den förväntade försäljningen (Defraeye och Van Nieuwenhuyse, 2016). Utöver detta underlättar vetskap om kommande efterfrågan även bland annat vid hantering av försörjningskedjan i och med att över- eller underskottslager då kan undvikas (Carbonneau, Laframboise och Vahidov, 2008). Detta gäller även vid planering av kommande försäljningskampanjer (Ma, Fildes och Huong, 2016).

Syftet med denna pro gradu-avhandling är att undersöka hur de existerande prognostiseringsmetoderna kan användas för prognostisering av försäljning inom detaljhandeln. Detta sker i form av en jämförande studie där traditionella metoder för tidsserieanalys jämförs med maskininlärningsmetoder. Att jämföra metoder för tidsserieanalys med maskininlärningsmetoder har varit ett populärt ämne de senaste åren, men en marginell del av forskningen har fokuserat på detaljhandeln. I och med att data som skapats inom olika branscher kan vara av mycket olika karaktär, finns det luckor i

den tidigare forskningen som denna pro gradu-avhandling ämnar fylla. Avhandlingen består av en litteraturöversikt i vilken de olika prognostiseringsmetoderna först beskrivs i detalj och tidigare forskning presenteras, samt av en empirisk studie i vilken några av de i litteraturöversikten beskrivna modellerna sedan används för att prognostisera Walmarts försäljning. Forskningsfrågorna som studien försöker besvara är:

1. *Producerar prognostiseringsmetoder som utnyttjar maskininlärning mer exakta prognoser än metoderna för tidsserieanalys?*
2. *Vilken prognostiseringsmetod presterar bäst då försäljning inom detaljhandeln prognostiseras?*

5.2. Litteraturöversikt

Metoderna som presenteras i litteraturöversikten är glidande medelvärde, exponentiell utjämning, ARIMA, regressionsanalys, beslutsträd och artificiella neurala nätverk. Av dessa är det glidande medelvärdet, den exponentiella utjämningen och ARIMA metoder för tidsserieanalys, medan regressionsanalys, beslutsträd och artificiella neurala nätverk är metoder som utnyttjar maskininlärning.

Det glidande medelvärdet omfattar flera olika metoder av vilka det enkla glidande medelvärdet och det viktade glidande behandlas i denna avhandling. Ett glidande medelvärde är ett medelvärde av de senaste observationerna i en tidsserie som uppdateras genom att lägga till den senaste observationen i medelvärdet samtidig som den äldsta utelämnas. Därmed fås ett glidande medelvärde som alltid hålls uppdaterat. Till skillnad från det enkla glidande medelvärdet där alla observationer är lika viktade, ger det viktade glidande medelvärdet större vikt åt de senaste observationerna. Detta kan vara användbart bland annat då man vill ha en modell som snabbare reagerar på förändringar.

Den exponentiella utjämningen påminner till viss del om det glidande medelvärdet men en utjämningsfaktor introduceras i den exponentiella utjämningen, vilket resulterar i märkbart olika resultat. Den exponentiella utjämningen kan delas in i enkel exponentiell utjämning som lämpar sig för tidsseriedata utan trend och säsongsmässighet, dubbel exponentiell utjämning som lämpar sig för tidsseriedata med trend, och trippel exponentiell utjämning som lämpar sig för tidsseriedata med både trend och säsongsmässighet.

Den sista metoden för tidsserieanalys är ARIMA, som står för autoregressive integrated moving average (fritt översatt: autoregressivt integrerat glidande medelvärde). Så som namnet på modellen låter förstå består modellen av flera delar. Det är en kombination av autoregression, differentiering och glidande medelvärde där differentieringen transformerar använda data till stationära data och autoregressionen och det glidande medelvärdet används för att prognostisera. ARIMA och den exponentiella utjämningen är de två mest populära metoderna för tidsserieanalys.

Regressionsanalys är en metod där en regressionsfunktion bestående av beroende och oberoende variabler skapas för att beskriva ett dataset. Den beroende variabeln är det man försöker förutspå, i detta fall försäljningen, och de oberoende variablerna är de faktorer som påverkar försäljningen. I denna avhandling har regressionsanalysen delats upp i enkel linjär regressionsanalys, där det endast finns en oberoende variabel, och multipel regressionsanalys, i vilken det ingår flera oberoende variabler. I och med att en regressionsanalys i teorin kunde utföras utan maskininlärningsmetoder på ett väldigt litet dataset, skulle regressionsanalysen också kunna räknas som en metod för tidsserieanalys. Eftersom det i praktiken ändå oftast är fråga om väldigt stora dataset vid prognostiseringsproblem krävs det oftast maskininlärningsmetoder för att lösa regressionsanalysen.

Den mest populära maskininlärningsmetoden för prognostisering är beslutsträd. Ett beslutsträd löser komplexa problem genom att dela in problemet i flera enklare frågor som sedan besvaras i tur och ordning för att till slut nå en slutsats. Existerande träningsdata används för att träna upp modellen så att rätt slutsats nås då en specifik kombination av frågor eller förhållanden matas in. När inlärningen är färdig kan sedan uppkommande

förhållanden matas in vilket resulterar i att modeller prognostiserar ett framtida utfall. Namnet på modellen härstammar i att en visualisering av ett beslutsträd liknar en trädkrona med sina grenar.

Den sista maskininlärningsmetoden är det artificiella neurala nätverket. Ett artificiellt neuralt nätverk är designat att följa samma process för beslutsfattande som människohjärnan genom att besvara komplicerade frågor genom att, likt beslutsträdet, dela upp dem i flera enklare frågor. Uppbyggnaden av ett artificiellt neuralt nätverk skiljer sig dock märkbart från beslutsträden. Ett artificiellt neuralt nätverk består av flera lager neuroner, i vilka de enklare frågorna besvaras, vilka är ihopkopplade med vikter. Baserat på de inkommande vikterna fattar en neuron ett beslut och skickar sedan en utgående vikt till nästa lager. Nätverket lär sig genom att justera vikterna utgående från träningsdata och kan därmed till slut fatta egna beslut. Prognostisering sker sedan genom att data för den prognostiserade perioden matas in i nätverket, varmed det artificiella neurala nätverket utgående från dessa data kan fatta ett beslut om framtida utfall.

5.3. Empirisk studie

Den empiriska studien kan delas upp i tre delar – dataanalys, datamodifiering och prognostisering. De data som används i den empiriska studien är Walmarts försäljningsdata som publicerats på Kaggle i samband med en maskininlärningstävling år 2012, och modellerna valda för prognostisering är enkelt glidande medelvärde, trippel exponentiell utjämning, ARIMA, regressionsanalys, beslutsträd och neurala nätverk. Programmet Forecast Pro används för att ta fram de optimala ARIMA- samt exponentiella utjämningsmodellerna, medan Python-modeller utvecklade av Scikit-learn används för beslutsträden och det neurala nätverket, och Stata används för att skapa modellen för regressionsanalys.

Dataanalysen visar att datasetet består av data innehållande veckomässig försäljningsinformation om upp till 99 olika avdelningar i 45 olika butiker. Datasetet

innehåller information om butikens typ och storlek, datum för observationen, huruvida det är helgdag eller inte, medeltemperaturen, bränslepriset i området, anonymiserad prisreduceringsinformation, konsumentprisindex samt arbetslöshetsgraden. Om två variabler har stark korrelation utelämnas den ena ur prognostiseringsmodellerna. Andra modifieringar som måste göras i datasetet är att de tomma fälten i prisreduceringsinformationen fylls med nollor så att de kan hanteras av datainlärningsmodellerna, och datumfältet ändras från ett specifikt datum till endast månadens nummer för att maskininlärningsmodellerna ska kunna koppla ihop observationer från olika år. I sitt ursprungliga format representerade datumet alltid fredagen varje vecka men eftersom detta datum kan ändras från år till år behövs ett mera lättolkat format. Variablerna som slutligen valdes för beslutsträden och det artificiella neurala nätverket är försäljningen, butiksnumret, avdelningsnumret, helgdag, temperaturen, prisreduceringsinformation 1, 2 och 4, och butikens typ. Variablerna för regressionsmodellen valdes utgående från det justerade R^2-värdet och var bränslepriset, helgdag, prisreduceringsinformation 1, 2 och 4, konsumentprisindex samt arbetslöshetsgraden.

5.4. Resultat

För att mäta resultatet av prognostiseringen används MAPE-formeln. MAPE är en förkortning av mean absolute percentage error och mäter medelvärdet av det absoluta procentuella mätfelet för en modell. Den bästa modellen var ARIMA med MAPE på 2,90 %, som följdes av exponentiell utjämning på 3,71 %, regressionsanalys på 4,12 %, beslutsträd på 4,92 %, glidande medelvärde på 5,55 % och neurala nätverk på 34,55 %.

För att besvara forskningsfrågorna måste även litteraturöversikten beaktas. Om den första forskningsfrågan skulle besvaras endast utgående från den empiriska studien skulle svaret vara att modeller som utnyttjar maskininlärning inte producerar mera exakta prognoser. Litteraturöversikten visade dock att resultaten av en prognos är väldigt

fallspecifika och en modell som ger goda resultat på ett dataset kan ge dåliga resultat på ett annat dataset. Vissa studier nådde bättre resultat med maskininlärningsmodeller medan andra studier nådde bättre resultat med modeller för tidsserieanalys. Därmed kan ingen definitiv slutsats dras, men det bör påpekas att flera prognostiseringsmetoder bör testas på det egna datasetet före den slutliga prognostiseringen utförs.

Vad gäller den andra forskningsfrågan är svaret ARIMA-modellen om svaret åter endast baseras på den empiriska studien. I och med att litteraturöversikten visade varierande resultat och då det finns otaliga prognostiseringsmetoder som lämnades utanför denna pro gradu-avhandling, är det dock omöjligt att ge ett definitivt svar på frågan. Trots det bör det nämnas att ARIMA, exponentiell utjämning, beslutsträd och artificiella neurala nätverk var de bäst presterande modellerna enligt litteraturöversikten, vilket delvis stöds av den empiriska undersökningen. Då även alla modeller förutom det artificiella neurala nätverket hade ett MAPE-värde under 10 %, vilket enligt Lewis (1997) är ett tecken på en väl presterande modell, kan man inte säga att någon av dessa modeller skulle ha misslyckats i sin prognostisering. När det kommer till den andra forskningsfrågan kan man därmed också dra slutsatsen att ett flertal prognostiseringsmetoder bör testas på det egna datasetet före den slutliga prognostiseringsmetoden väljs.

References

Abhishek, K., Singh, M.P., Ghosh, S. and Anand, A., 2012. Weather forecasting model using artificial neural network. *Procedia Technology*, *4*, pp.311-318.

Abrahart, R.J., Anctil, F., Coulibaly, P., Dawson, C.W., Mount, N.J., See, L.M., Shamseldin, A.Y., Solomatine, D.P., Toth, E. and Wilby, R.L., 2012. Two decades of anarchy? Emerging themes and outstanding challenges for neural network river forecasting. *Progress in Physical Geography*, *36*(4), pp.480-513.

Aggarwal, C.C., 2018. Neural networks and deep learning. *Springer*, *10*, pp.978-3.

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), pp.147-156.

Anderson, D.R., Sweeney, D.J. and Williams, T.A., 2011. *Statistics for business & economics*. Nelson Education.

Anusha, S.L., Alok, S. and Ashiff, S., 2014. Demand forecasting for the Indian pharmaceutical retail: A case study. *Journal of Supply Chain Management Systems*, *3*(2), p.1.

Arizmendi, C.M., Sanchez, J.R., Ramos, N.E. and Ramos, G.I., 1993. Time series predictions with neural nets: application to airborne pollen forecasting. *International journal of biometeorology*, *37*(3), pp.139-144.

Arunraj, N.S., Ahrens, D. and Fernandes, M., 2016. Application of SARIMAX model to forecast daily sales in food retail industry. *International Journal of Operations Research and Information Systems (IJORIS)*, *7*(2), pp.1-21.

Aye, G.C., Balcilar, M., Gupta, R. and Majumdar, A., 2015. Forecasting aggregate retail sales: The case of South Africa. *International Journal of Production Economics*, *160*, pp.66-79.

Bacha, H. and Meyer, W., 1992, June. A neural network architecture for load forecasting. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks* (Vol. 2, pp. 442-447). IEEE.

Benesty, J., Chen, J., Huang, Y. and Cohen, I., 2009. Pearson correlation coefficient. In *Noise reduction in speech processing* (pp. 1-4). Springer, Berlin, Heidelberg.

Beutel, A.L. and Minner, S., 2012. Safety stock planning under causal demand forecasting. *International Journal of Production Economics*, *140*(2), pp.637-645.

Bougadis, J., Adamowski, K. and Diduch, R., 2005. Short-term municipal water demand forecasting. *Hydrological Processes: An International Journal*, *19*(1), pp.137-148.

Brown, R.G., 1959. *Statistical forecasting for inventory control*. McGraw/Hill.

Burger, C.J.S.C., Dohnal, M., Kathrada, M. and Law, R., 2001. A practitioners guide to time-series methods for tourism demand forecasting—a case study of Durban, South Africa. *Tourism management*, *22*(4), pp.403-409.

Carbonneau, R., Laframboise, K. and Vahidov, R., 2008. Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research*, *184*(3), pp.1140-1154.

Chamlin, M.B., 1988. Crime and arrests: An autoregressive integrated moving average (ARIMA) approach. *Journal of Quantitative Criminology*, *4*(3), pp.247-258.

Chan, Y.M., 1993. Forecasting tourism: A sine wave time series regression approach. *Journal of Travel research*, *32*(2), pp.58-60.

Chawla, A., Singh, A., Lamba, A., Gangwani, N. and Soni, U., 2019. Demand forecasting using artificial neural networks—a case study of American retail corporation. In *Applications of artificial intelligence techniques in engineering* (pp. 79-89). Springer, Singapore.

Chen, C.Y., Lee, W.I., Kuo, H.M., Chen, C.W. and Chen, K.H., 2010. The study of a forecasting sales model for fresh food. *Expert Systems with Applications*, *37*(12), pp.7696-7702.

Cheriyan, S., Ibrahim, S., Mohanan, S. and Treesa, S., 2018, August. Intelligent Sales Prediction Using Machine Learning Techniques. In *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)* (pp. 53-58). IEEE.

Chu, C.W. and Zhang, G.P., 2003. A comparative study of linear and nonlinear models for aggregate retail sales forecasting. *International Journal of production economics*, *86*(3), pp.217-231.

da Veiga, C.P., da Veiga, C.R.P., Puchalski, W., dos Santos Coelho, L. and Tortato, U., 2016. Demand forecasting based on natural computing approaches applied to the foodstuff retail segment. *Journal of Retailing and Consumer Services*, *31*, pp.174-181.

Darji, M.P., Dabhi, V.K. and Prajapati, H.B., 2015, March. Rainfall forecasting using neural network: A survey. In *2015 international conference on advances in computer engineering and applications* (pp. 706-713). IEEE.

Defraeye, M. and Van Nieuwenhuyse, I., 2016. Staffing and scheduling under nonstationary demand for service: A literature review. *Omega*, *58*, pp.4-25.

Domingos, P., 2012. A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), pp.78-87.

Eppen, G.D., Gould, F.J., Schmidt, C.P., 1993. Introductory Management Science, *Prentice Hall, New Jersey*, 4th ed., p787

Ezugwu, E.O., Arthur, S.J. and Hines, E.L., 1995. Tool-wear prediction using artificial neural networks. *Journal of materials Processing technology*, *49*(3-4), pp.255-264.

Fifield, S.G.M., Power, D.M. and Knipe, D.G.S., 2008. The performance of moving average rules in emerging stock markets. *Applied Financial Economics*, *18*(19), pp.1515-1532.

Fong, W.M. and Yong, L.H., 2005. Chasing trends: recursive moving average trading rules and internet stocks. *Journal of Empirical Finance*, *12*(1), pp.43-76.

Gardner Jr, E.S. and McKenzie, E., 1988. Model identification in exponential smoothing. *Journal of the Operational Research Society*, *39*(9), pp.863-867.

Gardner Jr, E.S., 1985. Exponential smoothing: The state of the art. *Journal of forecasting*, *4*(1), pp.1-28.

Gardner Jr, E.S., 2006. Exponential smoothing: The state of the art—Part II. *International journal of forecasting*, *22*(4), pp.637-666.

Gardner, E. S., & McKenzie, E. (1985). Forecasting trends in time series. *Management Science*, *31*(10), 1237–1246. https://doi.org/10.1287/mnsc.31.10.1237

Gentry, T.W., Wiliamowski, B.M. and Weatherford, L.R., 1995. A comparison of traditional forecasting techniques and neural networks. *Intelligent engineering systems through artificial neural networks*, *5*, pp.765-770.

Gershenfeld, N.A., Weigend, A.S., 1993. The future of time series: learning and understanding. In: Weigend, A.S., Gershenfeld, N.A. (Eds.), Time Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley, Reading, MA, pp. 1–70.

Ghiassi, M., Zimbra, D.K. and Saidane, H., 2008. Urban water demand forecasting with a dynamic artificial neural network model. *Journal of Water Resources Planning and Management*, *134*(2), pp.138-146.

Gorr, W.L., Nagin, D. and Szczypula, J., 1994. Comparative study of artificial neural network and statistical models for predicting student grade point averages. *International Journal of Forecasting*, *10*(1), pp.17-34.

Grudnitski, G. and Osburn, L., 1993. Forecasting S&P and gold futures prices: An application of neural networks. *Journal of Futures Markets*, *13*(6), pp.631-643.

Gunasekarage, A. and Power, D.M., 2001. The profitability of moving average trading rules in South Asian stock markets. *Emerging Markets Review*, *2*(1), pp.17-33.

Hadri, K. and Larsson, R., 2005. Testing for stationarity in heterogeneous panel data where the time dimension is finite. *The Econometrics Journal*, *8*(1), pp.55-69.

Hann, T.H. and Steurer, E., 1996. Much ado about nothing? Exchange rate forecasting: Neural networks vs. linear models using monthly and weekly data. *Neurocomputing*, *10*(4), pp.323-339.

Harris, R.D. and Tzavalis, E., 1999. Inference for unit roots in dynamic panels where the time dimension is fixed. *Journal of econometrics*, *91*(2), pp.201-226.

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

Hinton, G.E. and Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. *science*, *313*(5786), pp.504-507.

Hinton, G.E., Osindero, S. and Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), pp.1527-1554.

Ho, K.L., Hsu, Y.Y. and Yang, C.C., 1992. Short term load forecasting using a multilayer neural network with an adaptive learning algorithm. *IEEE Transactions on Power Systems*, *7*(1), pp.141-149.

Holt, C.C., 2004. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, *20*(1), pp.5-10.

Hu, M.J.C., 1964. *Application of the adaline system to weather forecasting* (Doctoral dissertation, Department of Electrical Engineering, Stanford University).

Hung, N.Q., Babel, M.S., Weesakul, S. and Tripathi, N.K., 2009. An artificial neural network model for rainfall forecasting in Bangkok, Thailand. *Hydrology and Earth System Sciences*, *13*(8), pp.1413-1425.

Hyndman, R.J., & Athanasopoulos, G. (2021) *Forecasting: principles and practice*, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3. Accessed on 11.2.2021.

James, F.E., 1968. Monthly Moving Averages--An Effective Investment Tool?. *Journal of financial and quantitative analysis*, pp.315-326.

James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer.

Jenkins, G.M., 1979. Practical experiences with modelling and forecasting time series.

Jeong, D.B., 2017. Prediction of sales on some large-scale retailing types in South Korea. *The Journal of Business Economics and Environmental Studies*, *7*(4), pp.35-41.

Jones, R.D., Lee, Y.C., Barnes, C.W., Flake, G.W., Lee, K., Lewis, P.S. and Qian, S., 1990, June. Function approximation and time series prediction with neural networks. In *1990 IJCNN International Joint Conference on Neural Networks* (pp. 649-665). IEEE.

KALAOGLU, Ö.İ., Akyuz, E.S., Ecemiş, S., Eryuruk, S.H., SÜMEN, H. and Kalaoglu, F., 2015. Retail demand forecasting in clothing industry. *Tekstil ve Konfeksiyon*, *25*(2), pp.172-178.

Kandananond, K., 2011. Forecasting electricity demand in Thailand with an artificial neural network approach. *Energies*, *4*(8), pp.1246-1257.

Kang, S., 1991. *An investigation of the Use of Feedforward Neural Network for Forecasting, Kent State University* (Doctoral dissertation, Ph. D. Thesis).

Keshtegar, B., Gholampour, A., Thai, D.K., Taylan, O. and Trung, N.T., 2021. Hybrid regression and machine learning model for predicting ultimate condition of FRP-confined concrete. *Composite Structures*, *262*, p.113644.

Kimoto, T., Asakawa, K., Yoda, M. and Takeoka, M., 1990, June. Stock market prediction system with modular neural networks. In *1990 IJCNN international joint conference on neural networks* (pp. 1-6). IEEE.

Kot, S., Grondys, K. and Szopa, R., 2011. Theory of inventory management based on demand forecasting. *Polish journal of management studies*, *3*, pp.147-155.

Kumar, R., 2013. Decision tree for the weather forecasting. *International Journal of Computer Applications*, *76*(2), pp.31-34.

Kwiatkowski, D., Phillips, P.C., Schmidt, P. and Shin, Y., 1992. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?. *Journal of econometrics*, *54*(1-3), pp.159-178.

Lai, R.K., Fan, C.Y., Huang, W.H. and Chang, P.C., 2009. Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert Systems with Applications*, *36*(2), pp.3761-3773.

Lapedes, A. and Farber, R., 1988. How neural nets work. In *Evolution, learning and cognition* (pp. 331-346).

Levin, A., Lin, C.F. and Chu, C.S.J., 2002. Unit root tests in panel data: asymptotic and finite-sample properties. *Journal of econometrics*, *108*(1), pp.1-24.

Lewis, C.D., 1997. *Demand forecasting and inventory control: A computer aided learning approach*. Routledge.

Li, Z.M., Cui, L.G., Xu, S.W., Weng, L.Y., Dong, X.X., Li, G.Q. and Yu, H.P., 2013. Prediction model of weekly retail price for eggs based on chaotic neural network. *Journal of integrative agriculture*, *12*(12), pp.2292-2299.

Liao, H. and Sun, W., 2010. Forecasting and evaluating water quality of Chao Lake based on an improved decision tree method. *Procedia Environmental Sciences*, *2*, pp.970-979.

Liu, C., Hu, Z., Li, Y. and Liu, S., 2017. Forecasting copper prices by decision tree learning. *Resources Policy*, *52*, pp.427-434.

Lowe, D. and Webb, A.R., 1991, February. Time series prediction by adaptive networks: A dynamical systems perspective. In *IEE Proceedings F (Radar and Signal Processing)* (Vol. 138, No. 1, pp. 17-24). IET Digital Library.

Ma, S., Fildes, R. and Huang, T., 2016. Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra-and inter-category promotional information. *European Journal of Operational Research*, *249*(1), pp.245-257.

Makatjane, K. and Moroke, N., 2016. Comparative study of holt-winters triple exponential smoothing and seasonal Arima: forecasting short term seasonal car sales in South Africa. *Makatjane KD, Moroke ND*.

McGough, T. and Tsolacos, S., 1994. Forecasting office rental values using vector autoregressive models. In *The Proceedings of the Cutting Edge Property Research Conference*, pp.303-20.

Meade, N., 2000. Evidence for the selection of forecasting methods. *Journal of forecasting*, *19*(6), pp.515-535.

Meidinger, E.E., 1980. *Applied time series analysis for the social sciences*. Sage Publications.

Metghalchi, M., Marcucci, J. and Chang, Y.H., 2012. Are moving average trading rules profitable? Evidence from the European stock markets. *Applied Economics*, *44*(12), pp.1539-1559.

Mishra, A.K. and Desai, V.R., 2006. Drought forecasting using feed-forward recursive neural network. *ecological modelling*, *198*(1-2), pp.127-138.

Mohan, S., Thirumalai, C. and Srivastava, G., 2019. Effective heart disease prediction using hybrid machine learning techniques. *IEEE Access*, *7*, pp.81542-81554.

Mueller, M., 2019. Essentials of Inventory Management. *HarperCollins Leadership*, 3rd edition

Nam, K. and Schaefer, T., 1995. Forecasting international airline passenger traffic using neural networks. *The Logistics and Transportation Review*, *31*(3), pp.239-252.

Nelson, M., Hill, T., Remus, B. and O'Connor, M., 1994, January. Can neural networks applied to time series forecasting learn seasonal patterns: an empirical investigation. In *1994 Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences*.

Nielsen, M.A., 2015. *Neural networks and deep learning*. San Francisco, CA: Determination press.

Park, J. and Sandberg, I.W., 1991. Universal approximation using radial-basis-function networks. *Neural computation*, *3*(2), pp.246-257.

Poli, I. and Jones, R.D., 1994. A neural net model for prediction. *Journal of the American Statistical Association*, *89*(425), pp.117-121.

Raymond, Y.C., 1997. An application of the ARIMA model to real-estate prices in Hong Kong. *Journal of Property Finance*.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J., 1986. Learning representations by back-propagating errors. *nature*, *323*(6088), pp.533-536.

Samvedi, A. and Jain, V., 2013. A grey approach for forecasting in a supply chain during intermittentdisruptions. *Engineering Applications of Artificial Intelligence*, *26*(3), pp.1044-1051.

Shah, C., 1997. Model selection in univariate time series forecasting using discriminant analysis. *International Journal of Forecasting*, *13*(4), pp.489-500.

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

Shon, T. and Moon, J., 2007. A hybrid machine learning approach to network anomaly detection. *Information Sciences*, *177*(18), pp.3799-3821.

Stevenson, S., 2007. A comparison of the forecasting ability of ARIMA models. *Journal of Property Investment & Finance*.

Tang, Z. and Fishwick, P.A., 1993. Feedforward neural nets as models for time series forecasting. *ORSA journal on computing*, *5*(4), pp.374-385.

Tang, Z., De Almeida, C. and Fishwick, P.A., 1991. Time series forecasting using neural networks vs. Box-Jenkins methodology. *Simulation*, *57*(5), pp.303-310.

Theil, H., 1971. Applied economic forecasting.

Thomassey, S. and Fiordaliso, A., 2006. A hybrid sales forecasting system based on clustering and decision trees. *Decision Support Systems*, *42*(1), pp.408-421.

Vaisla, K.S. and Bhatt, A.K., 2010. An analysis of the performance of artificial neural network technique for stock market forecasting. *International Journal on Computer Science and Engineering*, *2*(6), pp.2104-2109.

Wang, J.Z., Wang, J.J., Zhang, Z.G. and Guo, S.P., 2011. Forecasting stock indices with back propagation neural network. *Expert Systems with Applications*, *38*(11), pp.14346-14355.

Watson, R.B., 1987. The effects of demand-forecast fluctuations on customer service and inventory cost when demand is lumpy. *Journal of the Operational Research Society*, *38*(1), pp.75-82.

Wen, Q., Mu, W., Sun, L., Hua, S. and Zhou, Z., 2013, September. Daily sales forecasting for grapes by support vector machine. In *International Conference on Computer and Computing Technologies in Agriculture* (pp. 351-360). Springer, Berlin, Heidelberg.

Werbos, P.J., 1988. Generalization of backpropagation with application to a recurrent gas market model. *Neural networks*, *1*(4), pp.339-356.

Williams, C., 2007. Research methods. *Journal of Business & Economics Research (JBER)*, *5*(3).

Winters, P.R., 1960. Forecasting sales by exponentially weighted moving averages. *Management science*, *6*(3), pp.324-342.

Winters, P.R., 1960. Forecasting sales by exponentially weighted moving averages. *Management science*, *6*(3), pp.324-342.

Wu, B., 1995. Model-free forecasting for nonlinear time series (with application to exchange rates). *Computational Statistics & Data Analysis*, *19*(4), pp.433-459.

Yu, Q., Wang, K., Strandhagen, J.O. and Wang, Y., 2017, September. Application of long short-term memory neural network to sales forecasting in retail—a case study. In *International Workshop of Advanced Manufacturing and Automation* (pp. 11-17). Springer, Singapore.

Zhang, G., Patuwo, B.E. and Hu, M.Y., 1998. Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, *14*(1), pp.35-62.