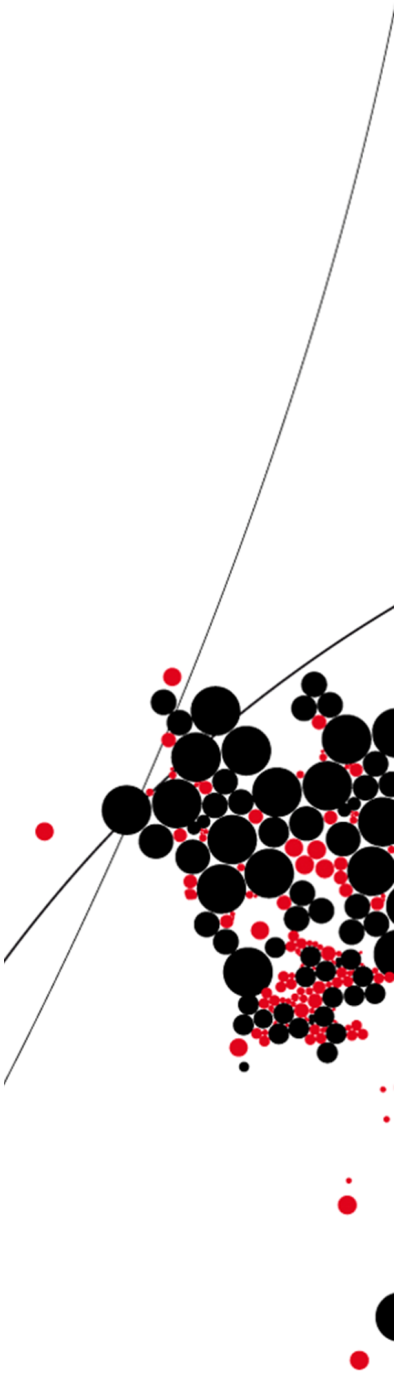
UNIVERSITY OF TWENTE.

University College Twente, ATLAS



Sales Forecasting: A Case Study in the Retail Business

Robin D. de Groot

B.Sc. Thesis

Supervisors:
dr. C.F. Pinho Rebelo de S
dr. E. Mocanu
MSc. R. Weeren
BSc. R. Cornelissen
dr.ir. M. Streng

University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

CONTENTS

Abstract—Sales forecasting has experienced much interest over the past few decades. The potential benefits businesses can gain from this are substantial, and this interest is not expected to decrease any time soon. This thesis looks at a case study in retail, specifically at A.S. Watson, the world's largest health and beauty retailer. This company is interested in applying a sales forecasting algorithm with the implementation being shelf replenishment. For this task, predictions on the level of per product, per store, per day need to be made. Data and a computer to run tests on were provided, which carried their own limitations. Multiple machine learning algorithms were looked at, of which a Neural Network Classifier was tested in depth. A clear step-by-step process was followed for hyperparameter tuning. The results show impressive accuracy values, though other metrics show that accuracy is not always a valid performance metric. In order to arrive at a model that can provide better and more confident results, several limitations of the current thesis and recommendations for future research are provided.

Index Terms—Classification, Neural Networks, Retail, Sales Forecasting

1 Introduction

DATA SCIENCE has seen a major increase in attention and use recently, coinciding with computing power and data availability increases [1]. The computation improvements allow the development of more, and more complex and exotic machine learning algorithms to analyse data and to come up with predictions based on historical data, for example. These new algorithms in turn allow the analysis of new types of data and the application of data science to new fields, such as medical solutions and translation services, and the research output on the topic of data science is vast, as is demonstrated by the large number of research journals focused on the topic [2].

The retail business is another field where there is a growing interest in data science and analytics, motivated by the potentially crucial economic benefits [3]. There is also increased competition from online retailers, pressing on the bottom line of traditional 'brick and mortar' stores due to their superior data collection capabilities, such as being able to see what customers looked at before making a purchase, and using this information in their operations. This is largely impossible for the traditional stores, who usually only see what is bought in the end [4]. Besides, since the cost of labour and other intermediate inputs in the supply chain of products has risen faster than the retail price, there is additional pressure on the bottom line of traditional stores [5].

This means that the will to adopt data driven solutions is definitely there in the retail business field, though many still struggle to properly understand the technology, and therefore struggle to define their exact requirements [6]. McAfee and Brynjolfsson dedicated an article in the Harvard Business Review to the obstacles traditional businesses face, and they describe how (senior) managers are sometimes sceptical about the benefits to be gained from data driven solutions and their complexity [4]. This prohibits a fast transition to (at least partly) base decisions on data, which they claim is always better than not using data.

In order to further the transition to data driven solutions in industry, this thesis will consider a case study at A.S. Watson Group, the largest health and beauty retailer in the world [7]. Their Dutch branch has a subsidiary called Kruidvat, which owns over a 1000 stores in the Netherlands and Belgium. This subsidiary, although it also has an increasing online presence, is still mostly physical store based, and thus is subject to the aforementioned pressure from online vendors. Therefore, they are looking for ways to increase their efficiency, and data driven solutions are a bit part of this. Sales forecasting can play a huge role in this as other research has shown [3], and will thus be the focus of this thesis.

Many different sales forecasting techniques have been devised, of which time series models such as Auto Regressive Moving Average (ARMA) or its derivatives are still widely used, just like the Box-Jenkins method [8]. These often-used algorithms have the downside that they assume that the data is stationary, meaning that (seasonal) trends are removed. This requires data from multiple years, which is not always available. Last year, another study was conducted which looked at the same problem of sales forecasting, though using SARIMAX, a time series model based on ARMA [9]. This method carries the aforementioned downsides and was only performed on a very limited number of products. Another algorithm that is widely used is Gradient Boosted trees, of which the derivative XGBoost has been used in many of the winning solutions on Kaggle [10]. These are just a few examples of the much larger number of algorithm possibilities.

Neural Networks have produced great results in areas like computer vision [11] and natural language processing (NLP) [12], and have also been applied successfully to structured, or tabular, data [13] [14]. They have also been used for sales forecasting [8], with promising results. They have a few distinct advantages, such as the much-decreased need for data pre-processing and feature engineering. Neural Networks can also "be universal function approximators for even non-linear functions" [15]. This means that they are theoretically able to represent the function that fits the data best. Although this is far from a guarantee of a perfect model that actually learns this function, it does show that Neural Networks are flexible and able to extract otherwise hidden relations between the variables in the data.

1.1 Research aim and organisation of thesis

This thesis explores the adequacy of the performance of a Neural Network that, after training on historical data, can

forecast sales for A.S. Watson on a per store, per product, per day basis. Other studies use time series analysis or a form of regression [8], so regression will be tried first and depending on the results, other methods will be tried. Another challenge is the data set since it is at a very low aggregation level, meaning that there are challenges in the data set as will be discussed in the methods section.

The rest of this thesis is organised as follows. In the next section, the background of this thesis will be discussed, both the case study and the more general theoretical background to the methods considered. The third section explains the materials and methods used in this thesis, which is followed by the fourth section which describes the experiments performed and shows the results of those experiments. The final section consists of the discussions of the findings and the conclusion of the thesis.

2 Background

2.1 A.S. Watson Case Study

A.S. Watson Group is the largest health and beauty retailer worldwide. The Group is for 75% owned by CK Hutchinson Holdings and for 25% by Temasek Holdings. CK Hutchinson Holdings is a large corporation with other businesses in, among others, port services and infrastructure.

A.S. Watson counts Watsons, Rossmann, Superdrug among its international brands. In the Benelux it is active through, among other formulas, Kruidvat, Trekpleister, and Ici Paris XL. All their different store brands combined, they own more than 15.000 stores worldwide in 25 different markets in which, combined with their online presence, 5 billion customers shop annually [7]. They have been collecting sales data over the past years, though, so far, the data has not been used to its full extent, and there are still a lot of potential benefits to be gained from extensive analysis of the data.

2.1.1 Possible applications of sales forecasting algorithm

Naturally, sales forecasting can be applied to a plethora of different retail processes and increase the efficiency in those. Examples range from larger scale applications, such as distribution centre optimisation, buying support systems, and promotion planning, which is a significant factor in the sales of the stores. Promotion planning support systems have been proven to be effective in increasing profitability [16], and is likely less intrusive than changing a heavily optimised and planned distribution centre process. Smaller scale implementation examples are store delivery planning, store staff planning, and shelf replenishment.

The application the company wants to look at first, due to its limited intrusion and potential for efficiency improvements, is shelf replenishment optimisation. The current system for shelf replenishment is simple and evidently leaves room for more efficiency. The delivery of new products comes into the store once every seven days (sometimes twice, depending on the store). That same day, all the products that are still in the store stockroom are 'pushed' into the store by the store employees. All products whose store shelves are full and still have extra stock are put in crates categorised by product group. When products are in promotion, there usually is another place in the store where those products are on display, and those shelf places are stocked as well.

Over the remaining days of the week until the next delivery, the restocking happens based on a planning that is not yet supported by sales data. Employees grab a crate from the stockroom, walk to the shelf location of the products in the store and product by product check if more fits on the shelf. Since the employees are not aware beforehand which products can be restocked due to the lack of support by data, many of the products stay in the crates and return to the store stockroom. This causes inefficient use of employee time, which causes inconvenience to the employees and unnecessary costs.

An example of a more ideal situation works as follows. Still, when a delivery comes in, all the products are 'pushed' into the store. The difference now is that, when there is stock of a product that does not fit into the shelf anymore, the employee indicates this in the internal system. The system will then, based on a sales forecast for the coming week, tell the store employee on which day the product shelves need to be restocked again. The employee then puts the products in the crate of the given day. Using this method, the store employees only need to restock the products that are in the crate of that day, saving much time and effort by the employees and reduces the number of times a product is handled.

Altering the shelf replenishment process is relatively limited in intrusion and risk, at least compared to other company processes, such as warehouse policy changes and logistical changes. In the warehouse of A.S. Watson there are very strict regulations on those processes due to the magnitude of a (partial) failure of those processes. Even though the intrusion is limited, the company aims to pilot this project first in one of the stores and only one product group and, if the results are satisfactory, roll it out to more product groups and stores later. The application requires that the model will need to deliver predictions per store, per product, per day, a few days ahead, which is a challenge that will be discussed in the methods section.

In summary, improving the efficiency of the shelf replenishment has the potential to save the company a lot. Since the company has so many stores, any efficiency improvement, if implemented in all or in a large number of stores, is multiplied by the number of stores. This means that a few hours saved per store, per week or month adds up to a lot over all stores on a yearly basis.

2.2 Theoretical Background

2.2.1 Sales Forecasting

The importance of sales forecasting for business success has been recognised in academia for a long time now [17]. Even though computational resources were much more limited back when that paper was written in 1957 compared to today, time series was already a popular, even the most popular, technique for sales forecasting. A contemporary much used variant is Autoregressive Integrated Moving Average (ARIMA), developed by Box et al. [18]. Such time series models look at recent values of the target variable (in this case sales volumes), and using lagged errors of the previous forecasts it provides a forecast for future data.

These models have and important underlying assumption: stationarity of the data. If larger trends are visible in the data, the data has to be differenced in order to remove the trend. Recognising such a trend requires the availability of data from a longer period of time. Still, when such data is available, such models have been proven to be very effective [19].

Other methods have also been applied successfully, such as Support Vector Machines (SVM) [20]. McCarthey et al. found in 2006 that still many firms base their forecasts on straight-line projection [21]. This is the practice of taking the trend line of a previous number of examples and basing prediction based on that, or taking the data from the previous year and assume that is a proper predictor for current data, possibly with slight adjustments. Exponential smoothing, forms of regression, and Naive Bayes models are also often used methods for sales forecasting in industry.

2.2.2 Neural Networks

Even though the focus of research within the Neural Networks domain has primarily been on computer vision and NLP in the beginning, Neural Networks also have great advantages when using it on structured data. Other methods on structured data require a large amount of feature engineering work to be done in order to get good results. The bulk of the data science work when using other methods is done on engineering the features, which requires a great amount of expertise. Neural Networks are able to learn relations between features without requiring that those relationships are explicitly defined beforehand by the data scientist. This significantly decreases the need of feature engineering knowledge. Still, more feature engineering can result in a higher accuracy and decreased training time, also in Neural Networks.

Neural Networks, or versions of them, were applied in many forecasting fields, such as fashion sales prediction [22], building energy demand forecasting [23], and economic time series forecasting [24], and gave favourable results.

Neural Networks can take on different forms and shapes, and they can be used for both regression and classification problems, two abundant paradigms. Regression is used by, among many others, Thiesing et al. in a 1995 study on supermarket sales prediction [25]. The benefit of this model is that the output can take on any continuous value. Classification problems have a few distinct possibilities for the output: a class label of one of the predetermined possible classes. Therefore, these types of machine learning models do not seem immediately logical to use for sales forecasting as sales figures usually exceed reasonable numbers for distinct classification, depending on the scale of the data. They have been used for other prediction tasks, such as bankruptcy prediction and credit scoring [26] and bank failure [27], which shows that Neural Network Classifiers can also be used as predictors.

3 Method and Materials

3.1 Materials

For this thesis, most of the materials were provided by A.S. Watson Group. Their internal databases were used to gather data related to the store sales, stores themselves, and products, among others. The final data set used consisted of data from 2018 and 2019. Besides the internal data gathered from within the company, external data was gathered from the open data platforms of the Dutch statistics bureau (CBS) and the Dutch weather institute (KNMI). The exact variables in the data set can be found in table 11 in the appendix.

- Internal data
- CBS Data [28]:
  - 70634NED: Population
  - 83978NED: Consumer trust
  - 80305NED: Access to services
  - 83648NED: Crime
  - 83651NED: Theft
- KNMI Daily weather [29]

The computations of this thesis were run on a provided laptop since this allows the data to stay within the IT environment of the company. A laptop with the following specs was used (note that everything was run on the CPU):

- CPU: Intel Core i5-8350U, 4 cores/8 threads
- RAM: 16GB DDR4-2400
- ROM: 512 SATA SSD

Although the data gathering process was done using Oracle SQL Developer, all the other code was written in Python. Some of the important packages used were: Pandas and NumPy for data handling, Seaborn for creating plots, imbalanced-learn for imbalanced data tasks, scikit-learn for much of the pre-processing and baseline setting, and Keras with TensorFlow backend for the Neural Networks.

3.2 Methods

3.2.1 Data Set

The company's internal data was stored on Oracle databases that could be accessed through Oracle SQL Developer. The author and his supervisor at the company were not aware of existing documentation of what data could be found where, leading to much time being spent on the data gathering process. This process consisted of finding out in what table certain necessary data was to be found, and afterwards joining that data to the rest of the table. After much work and many hours, the final script consists of, in total, more than 1500 lines of SQL to assemble the data sets used for this thesis.

Part of this process was the adding in of the external data sets mentioned in the materials section. That data was downloaded from the CBS and KNMI data portals, added to the Oracle databases, and joined with the internal data through SQL. The exact variables for each can be found in the appendix.

A.S. Watson is interested in a certain subset of products for this pilot study, namely the so-called 'medium movers'. The products that are sold in large volumes, the so-called 'fast movers', are not interesting for the shelf replenishment planning as they have to be refilled every day already. The products that are not sold often, the so-called 'slow movers', are also not interesting as they don't need to be restocked until the next delivery.

In the end, the product group of Soap Products was chosen due to the assortment being relatively stable and not season-bound, and due to the size of the data set when the medium movers were extracted. Since no strict requirements or characteristics for medium movers were given, the choice was made to take the products within the 0.65 and 0.75 percentile of average sales volume per product. This thesis only had access to detailed sales data from 2018 and 2019, and the external data was naturally also gathered for this time frame.

Not having access to data of the years prior to 2018 limits some possibilities with the data, leading to the inability to take seasonal effects into account. Neural Networks trained on deseasonalised and detrended data have proven to have dramatically reduced forecasting errors compared to raw data [8]. Even though detrending and deseasonalising the data is not possible, there is a lot of data available from 2018 and 2019 and a large number of explanatory variables can be found in the data set.

Due to the pilot study being focused on a store in the region of Utrecht, only data from the province of Utrecht was gathered. This was done to limit the total amount of data the Neural network needs to consider, which greatly helps speed up computation in the limited computation environment of the provided laptop.

3.2.2 Data Exploration

The data set used is split in 2018 data and 2019 data. The 2019 data will be used as the test set, as it is future data which is what the model needs to predict on when implemented. Both sets have the same 41 explanatory variables, which can be found in the appendix, and 1 target variable: sales volume. The data is aggregated on a per store, per product, per day level, meaning that the sales volume consists of the number of products of a certain Stock Keeping Unit (SKU) sold on one day in one of the stores. The 2018 data set consists of 540068 rows/examples, and the 2019 data set has 250302.

As quickly becomes clear when plotting a histogram of the sales volumes, most of the data has sales volume 0, meaning no sales on that day, of that product in that store. The plot can be found in figure 1. There are so few data points for sales volumes higher than four, especially relative to the total number of data points, that a decision was made to leave that data out and only consider sales volumes from zero up to and including four, as the other peaks would be hard for the model to take into account.

The data sets have 12 categorical variables and 29 numerical explanatory variables. To see how the data of the different sales volumes are distributed, Principal Component Analysis (PCA) was used to reduce the dimensionality of the data such that it can be represented in a 2D and 3D plot, which can be seen in figure 2. The reason for inclusion of figure 3 is that figure 2 only seems to only plot the 0 sales volume data points, whereas that is due to the abundance of data with that sales volume value. In figure 3, which depicts a random subset of the data points, it is more visible that other classes are also represented. Still, it is evident from these plots that different sales volume values are overlapping whereas we would like to see a clear distinction between the values. This will definitely be a challenge.

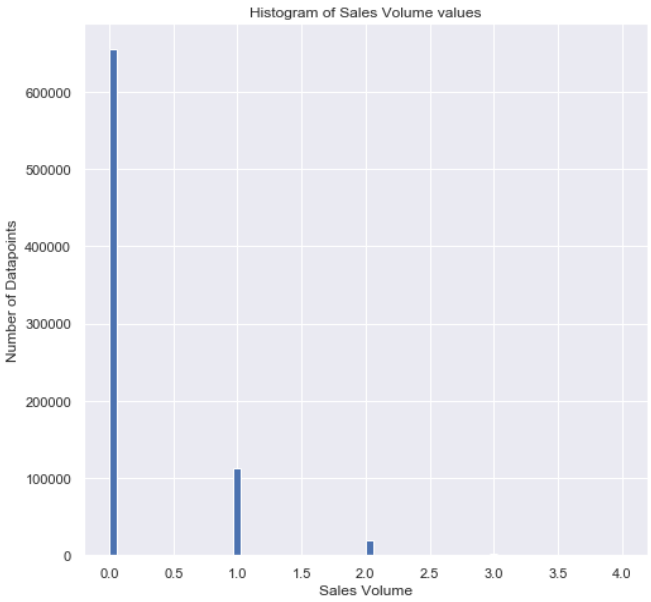


Fig. 1: A histogram of the occurrences of different sales volume values of the complete data set

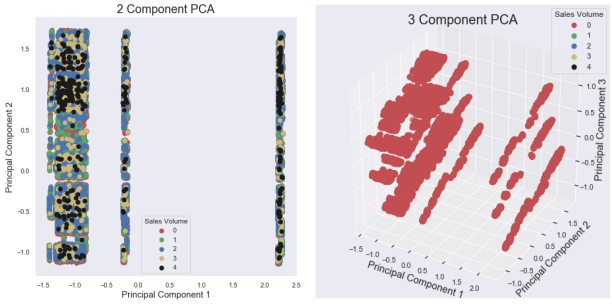


Fig. 2: Two and three dimensional PCA plots of the data

3.2.3 Data Pre-processing

For the data to be ready to be fed into a machine learning algorithm, some more data pre-processing needs to happen first. Like already mentioned, the data points with a sales volume value of over 4 were left out. Only 280 data points in the data set with almost 800,000 data points had to be deleted because of this.

Subsequently, the null values in the data set were filled with zeros. This was possible due to the limited number of null values in the data set and the fact that for most variables, the null value represented an absence of a certain condition (e.g. the null values in the 'extra promotion' represented instances where no such extra promotion applied).

The categorical variables are one-hot encoded (one of the resulting columns was deleted for each variable to prevent the dummy trap [30]), which resulted in a data set with 280 columns. Subsequently, the data set was split such that the 2019 data was used as a test set as it consists of data chronologically after the training data, therefore representing the task the model will have to perform when it is implemented.

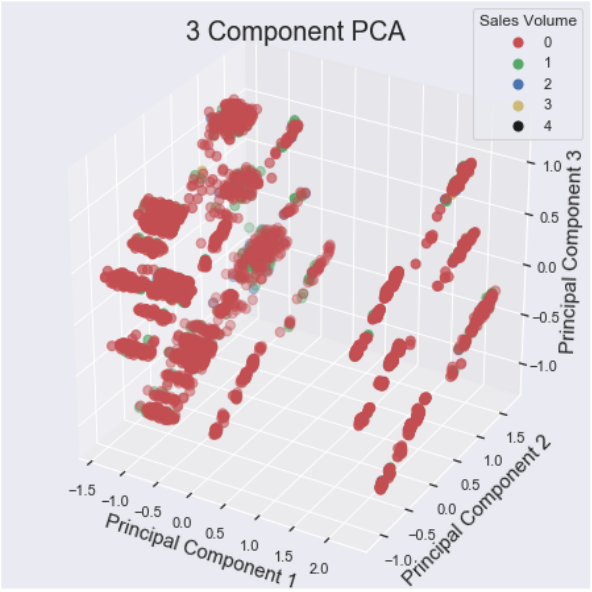


Fig. 3: Three dimensional plot of a random subset of the data

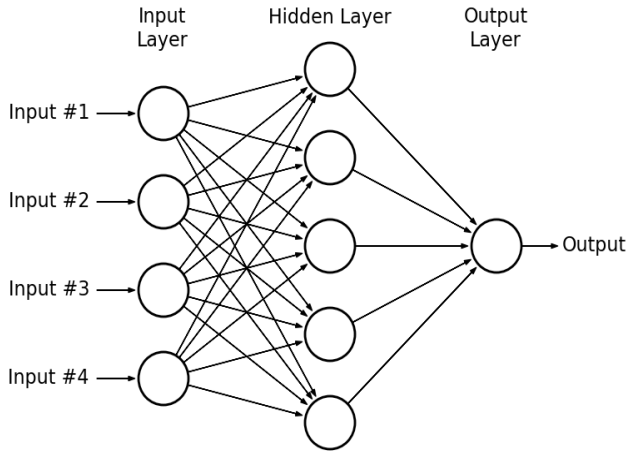


Fig. 4: Simple schematic of a feedforward Neural Network Regressor [32]

The 2018 data was split in the following fashion: The data was sorted on date starting with the 1st of January, after which the first 80% of the data was taken as the training set and the last 20% as the validation set.

Normalization was performed using min-max scaling. The scaler was 'fit' on the training set such that the data in the training set was scaled to a range of [0,1]. Then the same scaler was used to scale the validation and test sets.

Due to the extremely imbalanced distribution of the target variable, sales volume, the data needs to be either undersampled or oversampled in order to level the distribution. Early tests showed that when a Random Forest was used to train and predict on the data, it quickly realised that outputting just zeros would give it a rather high accuracy of over 80%. On paper, this number looks good though it is of course meaningless in terms of the performance of the model. Therefore, Adaptive Synthetic Sampling (ADASYN) oversampling was used. This is a method based on Synthetic Minority Oversampling Technique (SMOTE), which, in summary, looks at the K-nearest neighbours of a minority sample, draws Euclidean lines between them, then selects one line on which the synthetic data point lies exactly in the middle. ADASYN adds to this by having a selection criterion for number of synthetic data points to be created from each real data point [31]. This oversampling results in a data set with many more data points than the original data set. Of course, only the training set was oversampled as validation testing should only be done on real data, which resulted in approximately 350,000 data points per class.

After these steps were completed, the data was ready to be fed into the model.

3.2.4 Regression

Since regression is widespread within forecasting, this is what will be tried at first. Using this test, we aim to obtain useful insights and to understand if regression methods are suitable for this discrete and highly imbalanced sales data set.

Regression is implemented through a feed-forward neural network, a simple schematic of such a model can be found in figure 4. The inputs to the model are the different variables per data point, they are then multiplied by weights saved in the network (symbolised by the arrows), and the result of that multiplication is fed into the hidden neurons in which it is transformed using a non-linear function called the activation function. The results are then multiplied by the next saved weights (the arrows to the right of the hidden layer neurons), non-linearised by the activation function, and 'fed forward' to the next hidden layer, depending on how many hidden layers are used. The output layer functions comparably to a hidden layer, with the major differences that there is only one neuron in that layer for regression and that the activation function in the neuron is linear instead of non-linear.

The result of the output layer is then compared to the actual target value and the error in the prediction is calculated using an error function, in this case mean squared error was used (1):

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

where, x_i represents the input values and y_i represents the output values, while n represents the total number of datapoints.

This resulting error is then propagated back through the network, a process called backpropagation which lies at the basis of neural network learning, with the aim of minimising the overall error over the whole data set. If the error is plotted against all the weights in the network, a landscape with peaks and valleys of the error function becomes apparent, with the valleys representing minima, and in this case optima, in the error function. This means that for every error there is a gradient on that landscape. This gradient is used for updating the weights. The exact implementation of this gradient based learning depends on

the optimiser used, but the process of updating the weights based on the gradient is called gradient descent.

The exact architecture of the regression model used for this thesis has an input layer, two hidden layers, and one output layer. The dimensions are 280, 1000, 500, 1, respectively, based on a paper by Guo et al [13]. The Rectified Linear Unit (ReLU) was used as the activation function for the hidden layers (Equation (2)) and a linear activation function was used for the output layer.

$$f(x) = max(0, x) \tag{2}$$

Adaptive Moment Estimation (Adam) is used as optimiser, which is based on stochastic gradient descent, due to its straightforward implementation and computational efficiency [33].

3.2.5 Classification

A Neural Network Classifier is in many ways very similar to a Neural Network Regressor. However, there are a few distinct differences, the main one being the output. Classification has as many output neurons as possible classes in the data whereas regressors only have one output neuron. Additionally, the error function is different, and Sparse Categorical Cross-Entropy is used in this case since the labels are encoded as integers. The function for Categorical Cross-Entropy and this function can be seen in equation (3), which is extremely similar to Sparse Categorical Cross-Entropy.

$$H(p, q) = -\sum_{x} p(x) log(q(x)) \tag{3}$$

The $p(x)$ factor in the equation is the actual distribution of the correct class per data point and the $q(x)$ represents the probability distribution over the classes given by the model.

These differences between regression and classification are there due to the fact that there is no continuous spectrum of possible outcomes but just a few distinct ones. That means that the mean squared error is not a proper error measure in this case. The other important difference is the activation function of the output layer. In regression, this is a linear function but with classification, the outcomes should be a probability per output neuron, and thus per class, that that specific class is the correct one according to the classifier. So instead, the softmax function will be used. Based on the errors in the classification, the weights are updated in a similar fashion as with regression, and the Adam optimiser is used.

Due to the possible sales volume levels being so limited (only 0, 1, 2, 3, and 4), classification could be a valid method to solve this problem. Therefore, a lot of testing is done to thesis whether classification is indeed a good solution to this specific sales forecasting problem.

The performance of the classifier will be judged baesd on the accuracy of the predictions, meaning what proportion of the data points were predicted to have the correct class. Additionally, the classification outputs a confusion matrix, based on which precision and recall, visualised in 5, can be calculated.

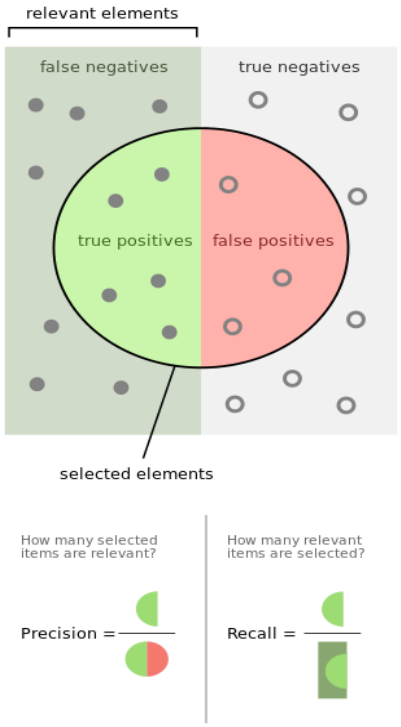


Fig. 5: Visualisation of Precision and Recall [34]

3.2.6 Hyperparameters

Since several hyperparameters will be tuned to arrive at the most optimal version of the model used, some more explanation of the effects of those hyperparameters is justified. The first hyperparameter that will be tested is the architecture, meaning in this case the number of hidden layers and the number of neurons per layer of the Neural Network. A deeper Neural Network allows the network to learn more complex relations since a deep network can be seen as a combination of multiple simpler functions [35], though it can also lead to more overfitting, meaning that the performance on the training set is good but the performance on the validation and test set is much worse.

The learning rate determines the magnitude of the weight update by gradient descent. It is a multiplier on the backpropagated error. A higher learning rate means a larger weight update and a lower rate means a smaller weight update. Generally, models with a higher learning rate learn faster but are more likely to not find the global optimum due to overshoot, and vice versa for models with a smaller learning rate.

Dropout is a technique to reduce the chance of overfitting. The dropout value represents the proportion of units that are dropped from the network during training, with a higher value meaning more units dropped. It has proven to enable a great reduction in overfitting while the performance does not suffer (much) [36].

L2 regularisation, also called weight decay, is a method to control the magnitude of weights by forcing them to stay small [37]. It adds a penalty term to the error function. This penalty term is a function of the sum of the weight values multiplied by some constant λ. This λ can be varied to

determine by how much large weights should be penalised. Minimising large weight values reduces overfitting.

The final two hyperparameters that are tested are batch size and the number of epochs. Batch size determines how many samples are fed through the network until the weights are updated. A lower batch size means that the weights are updated more often though the updates are based on the gradient of the error of just a few examples. The number of epochs represents the number of times the whole training data set goes through the network during the learning process.

3.2.7 Experimentation Methodology

To determine which model and which hyperparameters are best for this solution, many tests are performed. Firstly, a Neural Network Regressor is tested to see what the performance of this model is on the data. Then, based on the results, that model is tested on further or a classifier is chosen to be tested further. A baseline will be set using a different, though also popular, model to judge whether a Neural Network approach will turn out to be more favourable than this baseline.

The hyperparameters of the model that will be used in the end (e.g. learning rate, dropout) will be determined through unit testing. Ideally, this would be done using a technique such as grid search or random search since the hyperparameters likely exert an influence on each other. However, due to computational limits, the testing will be done through unit testing, meaning that the parameters will be tested one by one with the risk of not finding the perfect combination of hyperparameters. The order of testing is as follows: first the architecture of the model is determined, followed by learning rate, dropout, L2 regularisation, batch size, and number of epochs.

The decision for the optimal value for each hyperparameter is based on both the accuracy curve plotted against the number of epochs (both training and validation accuracy) and the confusion matrices on the training, validation, and test sets. The reason for this is that accuracy by itself is not sufficient for good reflection of the performance of the model, as was demonstrated by the test done in the Data Pre-processing section. The imbalance in the data causes the accuracy by itself to not be good enough for decisions. The confusion matrices then function as support in the decision since they can clearly show if the model is just outputting zeros or if the predictions are similar to the actual values.

Each model, with the varied hyperparameter that is tested at that moment, was left running for 30 epochs after which the accuracy plots and the confusion matrices were produced. Longer testing would give a more confident result, though this was not feasible considering the computational limits in this thesis. Repeated tests would also have made the decisions more valid and confident, but unfortunately, again, the available computational power did not allow this. To be more concrete, the laptop took, on average, almost an hour per hyperparameter value to be tested, causing some hyperparameter tests to take more than 10 hours.

Taking this computation and time restraint into account, early stopping was used. This mechanism automatically stops the training of the model and moves on to the next hyperparameter value to be tested if its performance does not improve after a certain number of epochs. This number was set at 15 epochs on the training set for this thesis. The reason to not use a lower value for this is to allow the model the opportunity to get out of a local optimum, if it lands in one. The initial hyperparameter values can be found in table 1.

For the final results, the precision and recall for this multi-class problem were calculated by calculating the mean of the those metrics per class.

TABLE 1: Initial hyperparameter values

Hyperparameter	Value
Input layer dimension	280
Hidden layer dimensions	1000, 500
Output layer dimension	5
Learning Rate	0.001
Dropout	0.3
L2 Regularisation	0
Batch size	1000
Epochs	30

Since the tests generated more than a hundred graphs, the author chose to not include them all in this thesis but to show a few of them so it becomes clear what they look like and how they are used in the decision making process, of which the learning rate section is a good example. All graphs are available from the author on request [1].

4 Experiments and Results

4.1 Experiments

4.1.1 Regression experimentation

The first test was done using a Neural Network Regressor which was implemented using the settings described in the Regression section. The results of which can be found in figure 6. From this visualisation, it becomes clear that the regression model is not learning properly. The error on the training set dives to 1.5 after only a handful of epochs, while the validation error is very unstable and even increases over time, which is contrary to what should happen. The histogram of the predictions on the training set in figure 7 also show that the regressor predicts according to a graph that looks like a Gaussian distribution centred at around the sales volume of 2, which is not how the sales are actually distributed, as all sales volumes are equally represented in the training set after oversampling.

This poor result prompted the search for other methods. As mentioned before, the data shows promising characteristics for a classification problem with only a few distinct possible values for the sales volume.

4.1.2 Baseline setting using Random Forest

In order to find out if a Neural Network Classifier would be a valid option for this problem compared to other methods, a baseline should be established. A Random Forest Classifier was chosen for this task as it is a popular choice and straightforward implementation in the scikit-learn library.

The hyperparameters of the model were taken from the implementation in the programming language R [38], as the

1. Author: Robin de Groot, e-mail address: r.d.degroot@icloud.com

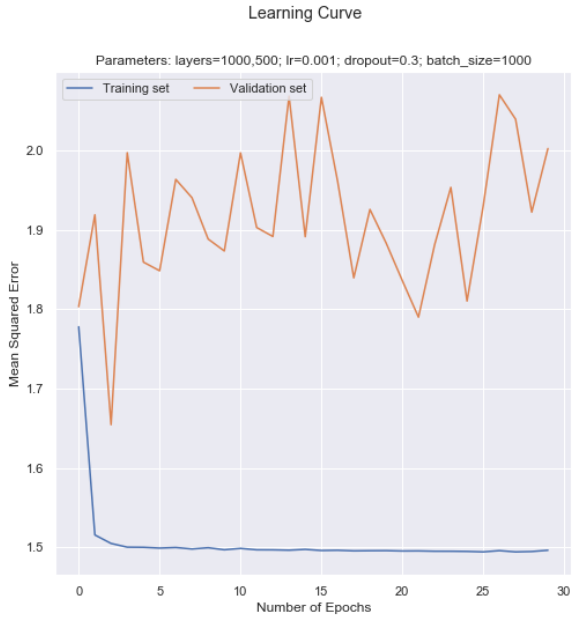


Fig. 6: Accuracy curve of Neural Network Regressor

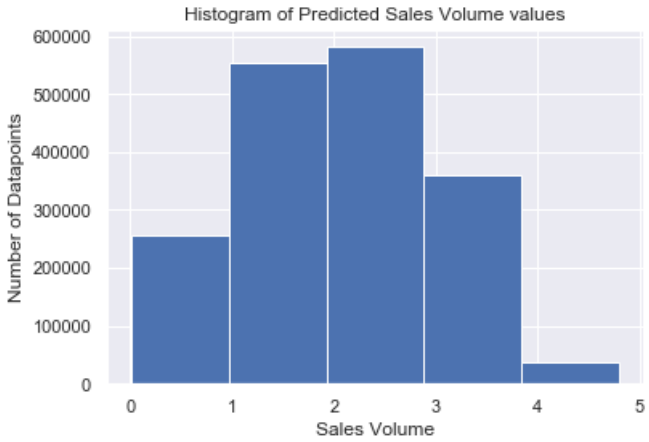


Fig. 7: Predicted Sales Volumes by Neural Network Regressor

default implementation in scikit-learn overfitted extremely. The resulting confusion matrices of the Random Forest can be found in figure 8.

TABLE 2: Random Forest Performance

Data set	Accuracy	Precision	Recall
Training	0.63	0.62	0.63
Validation	0.71	0.25	0.33
Test	0.68	0.25	0.31

Figure 8 and the performance metrics in table 2 show that the model struggles with finding a fitting model on the training set, though it is apparent that it does not overfit, fortunately. The results on the validation and test sets still leave a lot to be desired as a substantial proportion of the data points is not correctly classified, which is emphasised by the poor precision and recall results for the validation

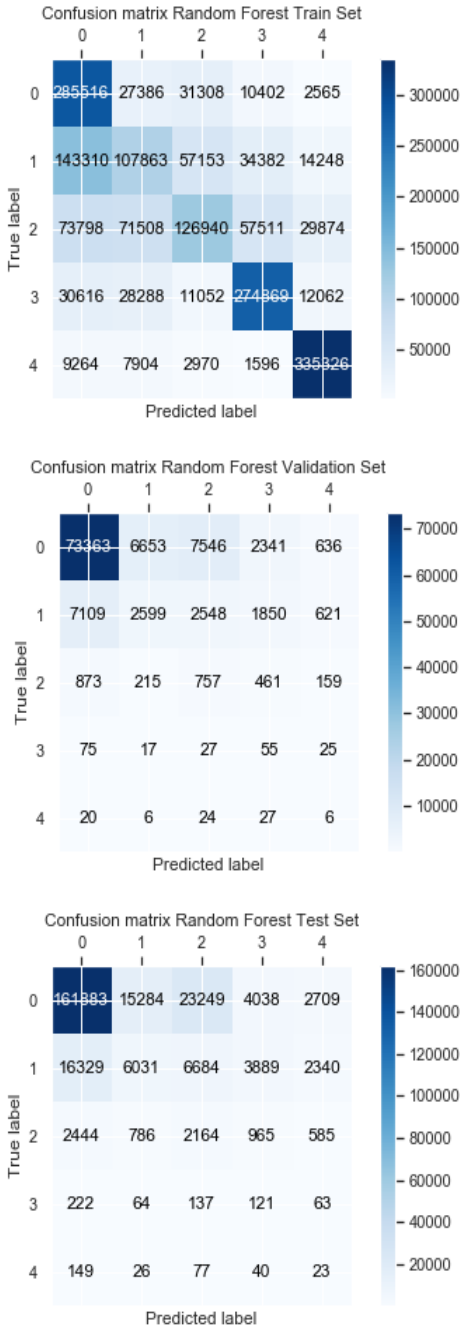


Fig. 8: Confusion matrices from the Random Forest Classifier. Top is on the training set, middle on the validation set, and bottom on the test set

and test sets. The parameters of this Random Forest have not been tuned at all, so there is still room to improve this model. Still it provides a baseline that the Neural Network Classifier will have to beat in order to be considered as a proper solution.

4.1.3 *Architecture*

Now that the baseline has been set, the hyperparameters of the Neural Network Classifier can be tuned. Architecture is chosen as the first parameter to be tuned. The values tested can be found in table 3. The numbers represent the

TABLE 3: Architecture values to test

Test Number	Neurons per hidden layer
1	1000, 500
2	1000, 500, 200, 75
3	1000, 500, 300, 100, 70, 30

TABLE 4: Learning Rate values to test

Test Number	Learning Rate
1	0.0001
2	0.0003
3	0.001
4	0.003
5	0.01
6	0.03
7	0.1
8	0.2
9	0.3
10	0.4
11	0.5
12	0.6
13	0.7
14	0.8
15	0.9

number of neurons per hidden layer and the number of values represents the number of hidden layers. The input and output layers are left as they are and the values for those can be found in table 1.

From the confusion matrices resulting from these rudimentary tests it becomes clear that a deeper Neural Network, meaning a Network with more hidden layers, probably does not improve the results while it does increase the time it takes for the model to train. Therefore, the architecture of the first test is chosen for the neural network, and this architecture is used for the further tests.

4.1.4 Learning Rate

The hyperparameter to be tested next is the learning rate, which determines the magnitude of the weight updates. Even though the values tested might seem extreme, they were chosen this way to make sure a reasonable spectrum of possibilities is considered. The exact values that were tested can be found in table 4.

It is important to note that the Adam optimizer used for this thesis adapts the learning rate based on the moving average of the gradient [33], so the learning rate tested is not kept constant throughout the epochs.

For the learning rate tests, the results can be found in figure 9, which shows the accuracy of the model on the training and validation sets. Keep in mind that accuracy is not necessarily the best performance metric if not combined with another evaluation method. Confusion matrices are used for additional evaluation, as they allow for visual inspection of the performance of the model on the different classes. The confusion matrices of the best result, determined using the accuracy curve and confusion matrices, which turned out to be 0.01, are shown in 10.

This value was chosen due to the reasonable performance in the training and validation set accuracy graphs, combined with a confusion matrix that shows reasonable performance and that this model does not label (almost) everything as zero

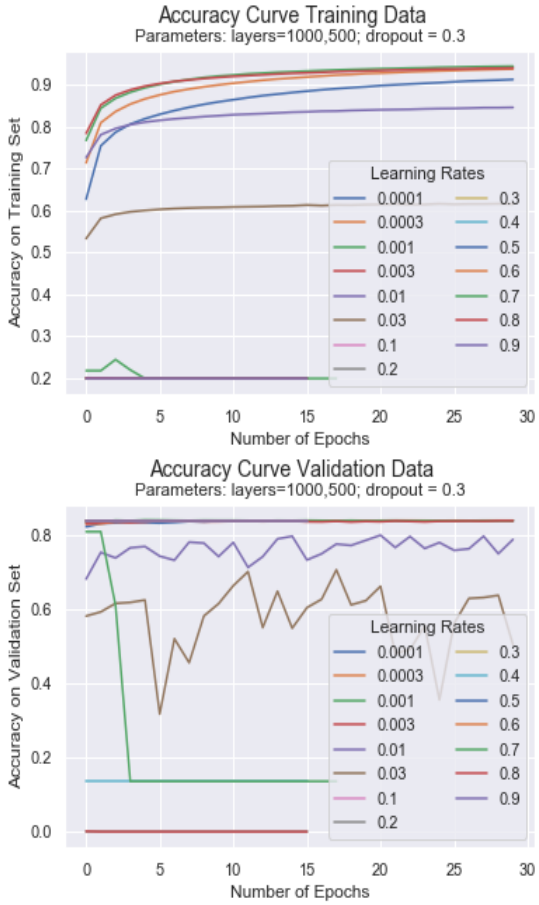


Fig. 9: Accuracy curves of Neural Network Classifier tested with different learning rates

TABLE 5: Dropout values to test

Test Number	Dropout
1	0
2	0.1
3	0.2
4	0.3
5	0.4
6	0.5
7	0.6
8	0.7
9	0.8
10	0.9

4.1.5 Dropout

The resulting best settings for the architecture (1000,500) and learning rate (0.01) are used to test for the dropout value. In short, the dropout value determines what proportion of neurons are 'thrown away', which should lead to reduced overfitting [36].

The values that were tested can be found in table 5. These values were chosen due to the possible values for dropout being in a range of [0,1], with a value of 1 meaning that all neurons are thrown away leaving an empty network. Choosing these values shows the performance of the model over a (rough) spectrum of different dropout values.

A dropout value of 0.3 was chosen as this value gave the best trade-off between the accuracy and the results in the confusion matrix.

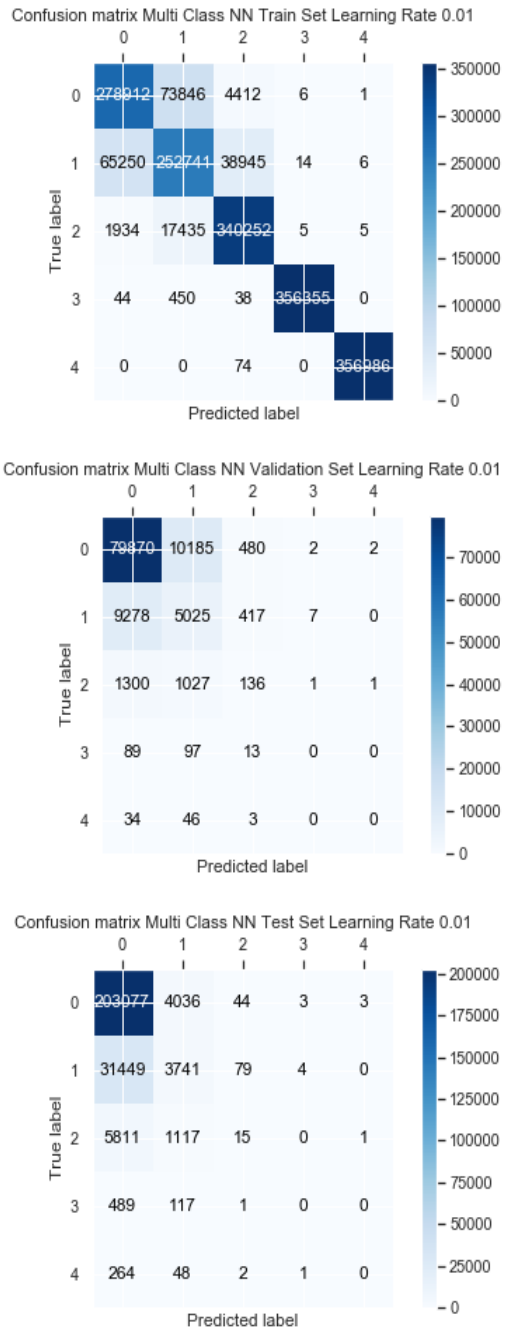


Fig. 10: Confusion matrices from the Neural Network Classifier using 0.01 as the learning rate value. Top is on the training set, middle on the validation set, and bottom on the test set

4.1.6 L2 Regularisation

The next hyperparameter to be tuned is L2 regularisation. This is a method of reducing the overfitting of a Neural Network. The values for λ that are tested can be found in table 6.

It turns out that the best value for λ is 0. This effectively means that no L2 Regularisation is applied.

TABLE 6: L2 Regularisation λ values to test

Test Number	λ value
1	0
2	0.00001
3	0.00003
4	0.0001
5	0.0003
6	0.001
7	0.003
8	0.01
9	0.03
10	0.1
11	0.3

TABLE 7: Batch sizes to test

Test Number	Batch size
1	128
2	1024
3	2048
4	2048
5	4096
6	16384

4.1.7 Batch size

With the aforementioned hyperparameters already tested and decided upon, batch size is the next one to be tested and chosen. This value determines how many data points are fed through the network and on which the gradient is calculated, which in turn determines how much the weights are updated. The values tested can be found in table 7. These values were chosen since they represent both a very small batch size, increasing to a very high batch size. Based on the results it can be determined if a smaller or a larger batch size gives better performance for this task. The reason why they're all a power of two is that the number of processor cores is also a power of 2, leading to all cores being used at their fullest throughout the learning process.

Judging from the accuracy metrics and a simple visualisation of the confusion matrices, a batch size of 1024 works best for this model, and this value is chosen.

4.1.8 Number of epochs

The final hyperparameter to vary is the number of epochs. This hyperparameter determines how many times the whole data set is fed through the network. The values that are tested can be found in table 8. The early stopping mechanism is still used to prevent unnecessary computations.

When testing the values it quickly became apparent that the early stopping mechanism capped the learning at around 200 epochs, though slightly differing per test. Therefore, this number of epochs is chosen for the model, which has the advantage that the model will train quicker since it needs fewer epochs.

TABLE 8: Number of epochs to test

Test Number	Number of epochs
1	200
2	500
3	750
4	1000

4.2 Results

After all the tests were performed, the optimal hyperparameters were chosen and they are displayed in table 9. The resulting accuracy plots can be found in figure 11 and the confusion matrices are displayed in figure 12.

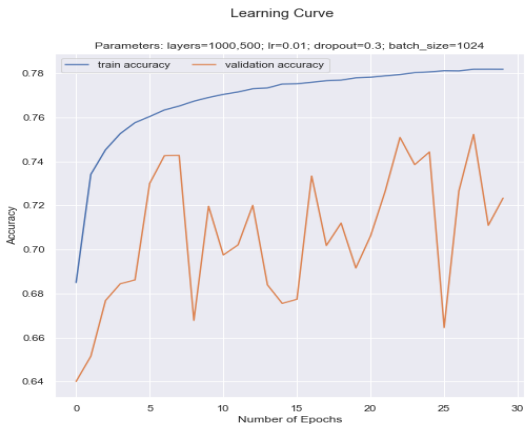


Fig. 11: Accuracy curve of Neural Network Classifier

TABLE 9: Final hyperparameter values

Hyperparameter	Value
Input layer dimension	280
Hidden layer dimensions	1000, 500
Output layer dimension	5
Learning Rate	0.01
Dropout	0.3
L2 Regularisation	0
Batch size	1024
Epochs	200

Initially, the values in table 9 seem (almost) identical to the values in table 1. Even though, indeed, many values are the same or similar, the testing performed has determined these values to be the most optimal.

The outcome that a deeper network does not improve performance is slightly surprising since the opposite would be expected based on the findings of Goodfellow et al. [35], who state that a deeper network usually leads to better performance. This has some possible explanations, of which the following seem most likely to the author. It could be a result of the data not being separable based on the classes, for which the PCA plots in figures 2 and 3 give a good indication, and a deeper network does not solve this problem (much) better than a less deep network. Even though this is a likely explanation, its correctness cannot be guaranteed.

The performance metrics of the final results can be found in table 10.

TABLE 10: Final Neural Network Classifier Performance

Data set	Accuracy	Precision	Recall
Training	0.83	0.86	0.87
Validation	0.76	0.26	0.25
Test	0.82	0.28	0.23

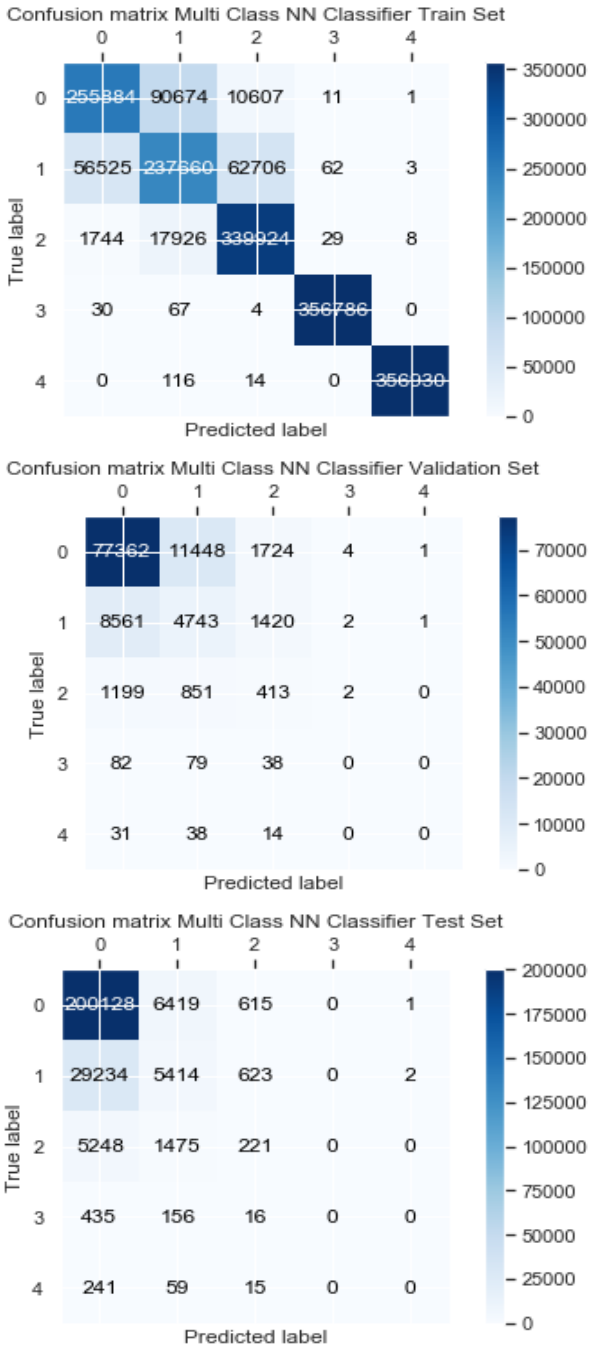


Fig. 12: Confusion matrices from the Neural Network Classifier. Top is on the training set, middle on the validation set, and bottom on the test set

5 Discussion

Even though the accuracy values are impressive, the precision and recall values on the validation and test sets prove that using just the accuracy as a performance metric can be misleading. This is the result of the class imbalance as the accuracy of the model would be very high (approximately 83%) if it would just output zeros. From the confusion matrices in figure 12 it becomes clear that for many of the data points in the test and validation sets the model predicts zero sales volume, further confirming that the accuracy is

not a proper performance metric for this problem.

Compared to the baseline of the Random Forest, the accuracy is 5 to 20% higher, depending on which set is being looked at. The precision of the Neural Network Classifier is slightly higher, thought the recall of the Random Forest is slightly higher. This means that there is a trade-off to be made between accuracy, precision, and recall, and based on the decision which one is deemed more important, the Random Forest or Neural Network Classifier should be chosen. The fact that the hyperparameters of the Random Forest were not tuned at all implies that there are still potential performance improvements to be gained, which could be looked into further.

To get to the implementation of the model for the shelf replenishment process a translation needs to be made. The predictions just give information on how many products of a certain product in a certain store will be sold per day. A second model needs to be implemented to translate this prediction into a message to the store employee to tell them in what crate the leftover products should be put. This second model is also required to implement some business rules in order to account for errors in the predictions. An error causing the store shelf to be empty before the day the model predicts it needs to be refilled can cost the store valuable revenue, and thus safeties should be built in. The requirements for this model should be set by an expert who is also knowledgeable about store processes, and this second model is thus outside the scope of this thesis.

5.1 Limitations and recommendations

5.1.1 Limitations

As has been mentioned a few times before in this thesis, the computing power was a major limitation on this thesis. If the provided system would have had a GPU, Neural Network learning would have been greatly accelerated due to the good implementation of NVIDIA's CUDA technology in Keras. Unfortunately, this was not the case, making a grid search or random search not reasonably possible. Unit testing was used instead, though this method fails to take the influence hyperparameters have on each other into account. This could have had a negative effect the results as there is a chance that a suboptimal combination of hyperparameters was chosen. Another future improvement when more computation power is available is the repetition of the tests. By repeating each test, the results become much more robust and a more informed choice can be made about the best value for each hyperparameter.

The computational power also limited the possibilities of using some possible different methods, such as Recurrent Neural Networks which have shown promise in other forecasting applications [39]. They have also been applied to store sales forecasting in the Rossmann Kaggle competition, in which the authors of [13] participated and received the third-place prize. This is especially impressive due to the very limited feature engineering done compared to other top performers. This model would have been more taxing computationally, making it unrealistic considering the computation environment of this thesis.

Another limitation, which has already been mentioned, is the data. Only data from 2018 was available to train the model on. Data from multiple years would have allowed the seasonal effects and general trend to be visible, which would have probably improved the performance [8].

Another aspect of the data that is the overlapping of the classes which became clear in the PCA plots in figures 2 and 3. This shows that the data set apparently does not contain enough information to properly separate the classes. The final aspect of the data that stood out was the extreme class imbalance with the largest class containing approximately 83% of the data and the smallest class containing much less than 1% of the data. This was tackled using the ADASYN oversampling method, and even though this has given good results in other studies, the synthetic samples it creates are not as meaningful as real data.

5.1.2 Recommendations

Since the results of this thesis are reasonable, though not great, it leaves open the question if this model can significantly increase the shelf replenishment process efficiency, or if different methods should be looked at for further study. One consideration could be to transform the problem in a binary classification problem with the two options of 'sales' or 'no sales' for a certain product, store, and day. This slightly reduces the class imbalance and could allow the model to be able to distinguish between these two better than between the five classes in this thesis. However, it puts much more pressure on the second model to translate the predictions into actionable shelf replenishment planning, if such a binary model is properly implementable at all for this specific problem of shelf replenishment.

Other machine learning algorithms should also be tried on this problem. The aforementioned Recurrent Neural Network could be an option that could potentially perform well. The model choice could also be based on a trigger mechanism that automatically chooses the best model from a few options of models based on the product for which the sales are predicted. Something like this has been implemented by Huang et al. [40] and showed promising results.

A method that could be implemented to improve the current method is collaborative learning, which means that two models are used in tandem. For this problem, an implementation could be to use the classifier as the main model and to use a regression model on the same data to get a better idea of how confident the prediction is. For example, if the classifier outputs 2 sales, and the regression model outputs 0.2, the prediction is probably not very accurate. However, if the classifier outputs 1 and the regressor outputs 0.9, the prediction is probably quite accurate. This level of confidence could be used by the model that translates the sales predictions to an action prompt to the store employee to determine more accurately on what day a product should be restocked.

Considering the data, there are also a few recommendations to be made. The choice for the product group of Soap Products used for this thesis, although based on sound arguments, might not be perfect to test this model on. The product group choice should be based also on shelf capacity, not just sales volume, since they together determine how often a product should be refilled. This data was not easily available for this thesis, and work should be put into acquiring this data for future studies. Another option for change in

the data is to aggregate the data per week, per product, per store. This combats (part of) the problem of the abundance of data points with sales volume zero, which might help in separating the different sales volumes. Doing this probably has an effect on the sales volumes in the data set, and thus there should be critical reflection on the use of a classifier as there might be too many different sales volume values to realistically use a classifier.

Another major data related recommendation concerns the availability of data from more of the previous years. As has been mentioned before, seasonal effects could then be seen and dealt with, potentially leading to better results [8].

In their participation in the Rossmann sales forecasting Kaggle competition, Guo et al. came up with a novel way to treat categorical variables as input to a Neural network [13]. They call it entity embedding, which maps the categorical variables into Euclidean space. This mapping is treated as another layer for the neural network as this embedding is also learnt during the training process. By using a one-hot encoded matrix for this mapping, memory usage is decreased. Additionally, it allows categorical variables that are similar to each other to be mapped closer to each other. These characteristics improve the Neural Network performance.

One final recommendation, though more of a nice-to-have than a must-have, is to use some visualisation of the Neural Network weights. For A.S. Watson, it could be valuable to see which variable influences the sales in what way, giving them the opportunity to use this information in order to increase sales. Olden & Jackson describe a number of methods for "illuminating the black box" of neural networks [41]. One, or a combination, of these methods could be used to provide more insight into the factors that influence sales performance that neural networks usually allow, giving the business pointers as to what they should focus on.

Keeping these possibilities for improvement in mind, there is still a lot that can be improved upon to create a solution that is more favourable than the one presented in this thesis, and the author hopes that these recommendations and considerations will be taken into account when the same, or a similar, problem is encountered in the future.

6 Conclusion

This thesis developed a sales forecasting algorithm using a Neural Network Classifier on the case study at A.S. Watson. It presents the detailed step-by-step process of going from the problem faced by the business to a solution after much testing. Even though there were some substantial limitations, the performance of the model, considering those limitations, is not entirely discouraging. In addition, the use of Neural Network Classifiers for sales forecasting has not been published widely upon. Classification has been used to choose an optimal forecasting algorithm [40], but, as far as the author could tell, not by itself to forecast sales.

In addition, the case study itself has been described in detail and the possible implementation of the results also received much attention, which the business can us in order to plan further study and the implementation of a sales forecasting model.

The limitations and recommendations of this thesis were also elaborated on, giving A.S. Watson and other organisations clear pointers to potential methods to implement a similar and/or improved solution to a similar problem.

Considering the scale of A.S. Watson, the savings that can be made on the shelf replenishment process, if such a sales forecasting model is implemented, are potentially substantial. A few hours saved per store, per month, quickly add up when looked at annually for the whole business. Therefore, there is clear incentive to keep looking at this problem and to find more ways to implement solutions.

Acknowledgements

I would like to thank my supervisors, Claudio Pinho Rebelo de Sa, Elena Mocanu, and Martin Streng (all University of Twente faculty), and Randy Weeren and Robert Cornelissen (both A.S. Watson employees), for providing me with much help in the process of conducting this thesis. They have helped me greatly in a number of different ways. A few examples, though non-exhaustive, are: helping me a lot with the data gathering process, helping me with the structure and planning of the thesis, being critical of the choices I made, and teaching me a lot of valuable data science knowledge in the process. This thesis would have been less in-depth and refined if not for their help. Thank you for your time and effort, it is really appreciated.

A thanks also goes out to A.S. Watson, the company that was so kind as to allow a bachelor student to undertake this thesis, as well as to provide the materials necessary for this thesis to be possible.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[2] "Top journals for machine learning & arti. intelligence," http://www.guide2research.com/journals/machine-learning, accessed: 2019-07-11.

[3] W. Wong and Z. Guo, "A hybrid intelligent model for medium-term sales forecasting in fashion retail supply chains using extreme learning machine and harmony search algorithm," *International Journal of Production Economics*, vol. 128, no. 2, pp. 614–624, 2010.

[4] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton, "Big data: the management revolution," *Harvard business review*, vol. 90, no. 10, pp. 60–68, 2012.

[5] P. DArcy, D. Norman, S. Shan *et al.*, "Costs and margins in the retail supply chain," *RBA Bulletin, June*, pp. 13–22, 2012.

[6] R. Kohavi, L. Mason, R. Parekh, and Z. Zheng, "Lessons and challenges from mining retail e-commerce data," *Machine Learning*, vol. 57, no. 1-2, pp. 83–113, 2004.

[7] "A.s. watson group website," http://www.aswatson.com/, accessed: 2019-07-11.

[8] G. P. Zhang and M. Qi, "Neural network forecasting for seasonal and trend time series," *European journal of operational research*, vol. 160, no. 2, pp. 501–514, 2005.

[9] W. Steenbergen, "Hyperparameter optimization using grid search in sarimax models: Efficiently filling the shelves in kruidvat stores," 2018, bachelor thesis.

[10] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[12] Y. Goldberg, "A primer on neural network models for natural language processing," *Journal of Artificial Intelligence Research*, vol. 57, pp. 345–420, 2016.

[13] C. Guo and F. Berkhahn, "Entity embeddings of categorical variables," *arXiv preprint arXiv:1604.06737*, 2016.

[14] C. Yan, "Convolutional neural network on a structured bank customer data; a case study that proves cnns value even when youre not dealing with computer vision problems." 2018. [Online]. Available: https://towardsdatascience.com/convolutional-neural-network-on-a-structured-bank-customer-data-358e6b8aa759

[15] T. Hill, L. Marquez, M. O'Connor, and W. Remus, "Artificial neural network models for forecasting and decision making," *International journal of forecasting*, vol. 10, no. 1, pp. 5–15, 1994.

[16] J. M. Silva-Risso, R. E. Bucklin, and D. G. Morrison, "A decision support system for planning manufacturers' sales promotion calendars," *Marketing Science*, vol. 18, no. 3, pp. 274–300, 1999.

[17] J. B. Boulden, "Fitting the sales forecast to your firm," *Business horizons*, vol. 1, no. 1, pp. 65–72, 1957.

[18] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[19] N. S. Arunraj, D. Ahrens, and M. Fernandes, "Application of sarimax model to forecast daily sales in food retail industry," *International Journal of Operations Research and Information Systems (IJORIS)*, vol. 7, no. 2, pp. 1–21, 2016.

[20] Q. Wu, H.-S. Yan, and H.-B. Yang, "A forecasting model based support vector machine and particle swarm optimization," in *2008 Workshop on Power Electronics and Intelligent Transportation System*. IEEE, 2008, pp. 218–222.

[21] T. M. McCarthy, D. F. Davis, S. L. Golicic, and J. T. Mentzer, "The evolution of sales forecasting management: a 20-year longitudinal study of forecasting practices," *Journal of Forecasting*, vol. 25, no. 5, pp. 303–324, 2006.

[22] K.-F. Au, T.-M. Choi, and Y. Yu, "Fashion retail forecasting by evolutionary neural networks," *International Journal of Production Economics*, vol. 114, no. 2, pp. 615–630, 2008.

[23] A. H. Neto and F. A. S. Fiorelli, "Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption," *Energy and buildings*, vol. 40, no. 12, pp. 2169–2176, 2008.

[24] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing*, vol. 10, no. 3, pp. 215–236, 1996.

[25] F. M. Thiesing, U. Middelberg, and O. Vornberger, "Short term prediction of sales in supermarkets," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 2. IEEE, 1995, pp. 1028–1031.

[26] C.-F. Tsai and J.-W. Wu, "Using neural network ensembles for bankruptcy prediction and credit scoring," *Expert systems with applications*, vol. 34, no. 4, pp. 2639–2649, 2008.

[27] K. Y. Tam and M. Y. Kiang, "Managerial applications of neural networks: the case of bank failure predictions," *Management science*, vol. 38, no. 7, pp. 926–947, 1992.

[28] S. N. (CBS), "Statline open data," 2019, data retrieved from https://opendata.cbs.nl/statline/#/CBS/nl/navigatieScherm/thema.

[29] T. R. N. M. I. (KNMI), "Daggegevens van het weer in nederland - download," 2019, data retrieved from http://projects.knmi.nl/klimatologie/daggegevens/selectie.cgi.

[30] D. University, "The dummy variable trap," data retrieved from http://facweb.cs.depaul.edu/sjost/csc423/documents/dummy-variable-trap.htm.

[31] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge & Data Engineering*, no. 9, pp. 1263–1284, 2008.

[32] Y.-J. Ma and M.-Y. Zhai, "Day-ahead prediction of microgrid electricity demand using a hybrid artificial intelligence model," *Processes*, vol. 7, no. 6, p. 320, 2019.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] Walber, "Precision and recall," data retrieved from https://upload.wikimedia.org/wikipedia/commons/2/26/Precisionrecall.svg.

[35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[37] T. van Laarhoven, "L2 regularization versus batch and weight normalization," *arXiv preprint arXiv:1706.05350*, 2017.

[38] DataCamp.com, "randomforest," data retrieved from https://www.rdocumentation.org/packages/randomForest/versions/4.6-14/topics/randomForest.

[39] T. G. Barbounis, J. B. Theocharis, M. C. Alexiadis, and P. S. Dokopoulos, "Long-term wind speed and power forecasting using local recurrent neural network models," *IEEE Transactions on Energy Conversion*, vol. 21, no. 1, pp. 273–284, 2006.

[40] W. Huang, Q. Zhang, W. Xu, H. Fu, M. Wang, and X. Liang, "A novel trigger model for sales prediction with data mining techniques," *Data Science Journal*, vol. 14, 2015.

[41] J. D. Olden and D. A. Jackson, "Illuminating the black box: a randomization approach for understanding variable contributions in artificial neural networks," *Ecological modelling*, vol. 154, no. 1-2, pp. 135–150, 2002.

7 APPENDIX

TABLE 11: List of the variables in the data set and their source (target variable in bold)

	Variable	Source
1	MONTH_NR	Internal Time
2	QUARTER_NR	Internal Time
3	YEAR_NR	Internal Time
4	DAY_NR	Internal Time
5	DAY_NAME	Internal Time
6	SKU_NR	Internal Transactions
7	SORTGROUP_NAME	Internal Assortment
8	SUBGROUP_NAME	Internal Assortment
9	STORE_NR	Internal Transactions
10	STORE_MUNICIPALITY	Internal Assortment
11	GENERAL_DISCOUNT_MECHANISM	Internal Promotions
12	DISCOUNT_MECHANISM_EXTRA	Internal Promotions
13	STORE_GROSS_FLOORSURFACEAREA	Internal Assortment
14	STORE_NET_FLOORSURFACEAREA	Internal Assortment
15	**SALES_VOLUME**	**Internal Transactions**
16	DURATION (PROMOTION)	Internal Promotions
17	CONSUMER_PRICE	Internal Promotions
18	DISCOUNT_PERCENTAGE	Internal Promotions
19	TOTAL_INHABITANTS_MUNICIPALITY_OF_STORE	70634NED: Population
20	MALE_INHABITANTS_MUNICIPALITY_OF_STORE	70634NED: Population
21	FEMALE_INHABITANTS_MUNICIPALITY_OF_STORE	70634NED: Population
22	DISTANCE_TO_LARGE_GROCERY_STORE	80305NED: Access to services
23	DISTANCE_TO_TRAIN_STATION	80305NED: Access to services
24	DISTANCE_TO_DEPARTMENT_STORE	80305NED: Access to services
25	CUSTOMER_CONFIDENCE	83978NED: Consumer trust
26	ECONOMIC_CLIMATE	83978NED: Consumer trust
27	WILLINGNESS_TO_BUY	83978NED: Consumer trust
28	CRIME_SUSPECTS	83648NED: Crime
29	REGISTERED_CRIMES	83648NED: Crime
30	SOLVED_CRIMES	83648NED: Crime
31	ALL_THEFT	83651NED: Theft
32	STREET_ROBBERY	83651NED: Theft
33	PICKPOCKETING	83651NED: Theft
34	STORE_THEFT	83651NED: Theft
35	STORE_THEFT_COMPANY	83651NED: Theft
36	AVERAGE_WINDSPEED (DAILY)	KNMI Daily weather
37	AVERAGE_TEMPERATURE (DAILY)	KNMI Daily weather
38	MINIMUM _TEMPERATURE (DAILY)	KNMI Daily weather
39	MAXIMUM_TEMPERATURE (DAILY)	KNMI Daily weather
40	SUNHOURS (DAILY)	KNMI Daily weather
41	HOURS_OF_PRECIPITATION	KNMI Daily weather
42	PRECIPITATION_AMOUNT (DAILY)	KNMI Daily weather