Superstore Sales Prediction using Time Series Data

Patrick Brus

25.09.2021

1 Introduction

Data is said to be the gold of the future. Data can be used for various reasons. A very valuable application of data is the prediction of future sales for stores. If a store can accurately predict its future sales, then appropriate quantities of goods can be ordered and stocked. This can ensure that there are always enough products on stock so that no potential sale is lost. In addition, warehousing costs can be saved, because only as many products need to be stored as will be sold. So if no high sales figures are expected in the near future, then fewer products can be stored and thus storage space can be saved.

In this project, the superstore sales dataset from Kaggle [1] is used. This dataset contains sales data and various features of the sales from a retail store of the last 4 years. One important feature is the date of the sales figure. This can be used to represent the sales data of the retail store as a time series. The goal of this project is to analyze this dataset and to find important influences on the sales feature. In addition, the daily sales is to be analyzed in form of a time series and the time series sales data of the past is to be used to make a prediction of the sales data for one week ahead.

The first section describes the dataset and all available features. In the second section, the dataset is explored and analyzed to gain more insights into the available features and their influence on the sales. This section also describes some pre-processing methods applied on the dataset

The second section describes the time series analysis of the time series data of this dataset. It also contains a test whether the series is stationary or not.

In the methodology section, three different forecasting methods are explained and applied on the dataset. Each forecasting method is then used to make a prediction of one week ahead. The predictions are then compared to the ground truth sales data of this week and the root mean squared error is computed. This is then used to compare all methods and to decide which one is the best performing.

The conclusion section briefly summarizes this project, followed by a future work section, which discusses some possible further steps that could be done to maybe further improve the performance of the final model.

The code and all required files can be found on my Github page [1] in Jupyter notebooks.

2 Dataset

The superstore sales dataset contains 9800 samples and 18 features. Table 1 gives an overview of all features and a short description of each feature.

Feature Name	**Description**
Row ID	ID number of the Row as identifier
Order ID	ID of order as unique identifier for the specific order
Order Date	The date on which the order was placed
Ship Date	The date on which the order was shipped
Ship Mode	A mode describing how the order was shipped (i.e. Standard Class, Second Class, ...)
Customer ID	A unique ID for identifying the customer
Customer Name	Name of the customer who placed the order
Segment	Segment of the product that was ordered (i.e. Consumer, Cooperate, ...)
Country	Country where the order was placed
City	City where the order was placed
State	State where the order was placed
Postal Code	Postal Code of the location where the order was placed
Region	Region where the order was placed
Product ID	A unique ID for identifying the product that was ordered
Category	The category of the ordered product (i.e. Furniture, Office Supplies, ...)
Sub-Category	A more fine-grained category of the ordered product
Product Name	The name of the ordered product
Sales	The sales value of the ordered product

Table 1: Overview of all features and a short description for each feature.

3 Exploratory Data Analysis and Pre-Processing

As stated in the chapter 2, the dataset contains 9800 samples and 18 columns. The dataset is loaded into a pandas data frame to allow an easy analysis. The dataset contains 11 missing values for the postal code feature. Therefore, it is first checked which postal codes are missing. Luckily, all missing postal codes are from the city Burlington, such that the missing postal codes can be replaced with the postal code from Burlington, which is 05401. As a next step, the column data types for the order date and ship date are converted to datetime format. This allows a simple time series analysis to be performed later and to look for patterns in the sales data as a function of time. Afterwards, some statistics of the sales data are computed. The mean sales price for a order is at 230.8, with a standard deviation of 626.7. The minimum sales price in this dataset is at 0.44, while the maximum sales price is at 22638.5. In a next step, all features are checked for unique

[1] link to Github: `https://github.com/patrickbrus/IBM_Machine_Learning_Professional/tree/master/Time_Series`

values. This is done, because if a feature is only containing unique values, than it can be dropped, because it is not containing any useful information. This led to the conclusion, that the row ID can be dropped, because every sample contains its own ID. In addition, the country feature can be dropped, because all orders were placed from the United States of America. Therefore, this feature does not contain any useful information for the further analysis.

The sales data is plotted in a histogram (Figure 1) to get more insights into the distribution. As one can see, the data is heavily left skewed with some outliers on the right. To better visualize the outliers, the histogram is also plotted on a logarithmic y-scale. The left skewed data could lead to problems when a regression should be performed. But in this project, the sales data is only used to make forecasts. Therefore, no pre-processing methods for reducing the skewness are applied.

The next feature that is analyzed is the states features. Figure 2 shows a countplot of all available states and the number of products ordered from each state, while Figure 3 shows the top ten states yielding the highest sales. As you can see, California has the highest sales, followed by New York. This also makes sense, because California is the largest state in the United States of America and also one of the wealthier states.

For each ordered product, there is also the city from where the product was ordered. Figure 4 shows the top ten cities having the highest sales. As one would have guessed, New York City is on the first place. Los Angeles (LA) and San Francisco also have a lot of sales. This also makes sense, because California is the state with the highest number of sales and LA and San Francisco are the two largest cities of California.

As a next feature, the customer name is analyzed. This can be a very important feature, because having past data about customers and their sales can be very powerful in predicting their future sales. Figure 5 shows the most valuable customers according to their total sales. The first place goes to Sean Miller, with total sales of 25000. Sean Miller is followed by Tamara Chand, having almost 20000 in sales.

For a retail store, it is also always useful to locate the most valuable segments and product categories. Thus, targeted investments can be made in strong segments or weak segments can be strengthened through advertising. Therefore, the features "Segment", "Category" and "Sub-Category" are further analyzed. Figure 6 gives an overview of the three different segments and the sales achieved for each event. The most valuable segment is the consumer segment, followed by the corporate segment. The home office segment is the least valuable. It would be interesting to also have the sales numbers of the superstore after the corona pandemic, because the home office segment could be more valuable after the corona lock downs and the trend to work from home. Figure 7 shows the different product categories and the amount of products per category, while Figure 8 shows the sales per product sub-category. The phones sub-category is the most valuable, followed by chairs. In general, office supplies is the category with the largest number of sold products. The office supplies product category contains products of all segments. Figure 9 gives an overview of the different categories, their sub-categories and the share of each category in sales. As a last visualisation for getting better insights into the top selling categories and segments, the top selling product names are visualized (Figure 10). A canon copier is the best selling product. This could be due to the reason that also the price of this canon copier is already very large in comparison to other products and therefore the share in total sales is larger.

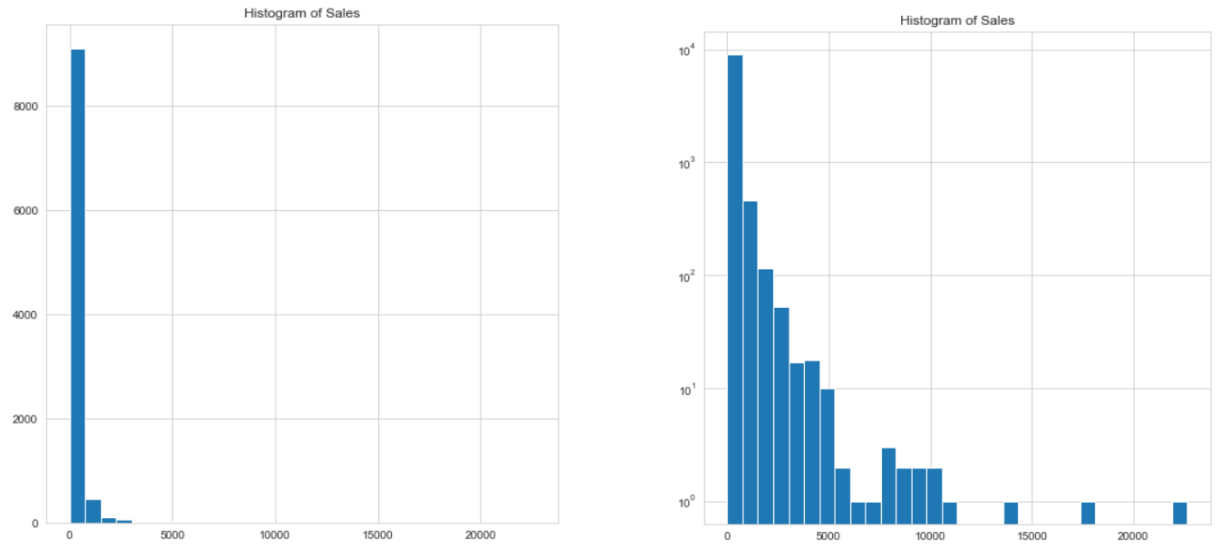

Figure 1: Histogram of the sales data on a normal y-scale (left) and on a logarithmic y-scale (right). The sales data is heavily left skewed, which could lead to problems when a regression task should be performed.

The last analyzed feature is the shipping mode feature. There are four different shipping modes (Figure 11). Most products are shipped with the standard class and only very few are shipped within the same day. This is possibly due to the fact that this is the most expensive shipping class. Figure 12 shows the sales per shipping class. The share in sales of each shipping class seems proportional to the amounts of products sold within each class. Figure 13 shows the correlation heat map of each shipping class to the amount of sales. As one can see, the correlations are very small, so the shipping class is not expected to have a large impact on the sales feature.

4 Time Series Analysis

As time series data, the order date is analyzed. Therefore, the index of the pandas dataframe is set to the order date. Then the frequency of the order date is checked. This is important, because daily order dates and sale values are required to make the one week forecast of sales. There are 1230 unique days in this dataset, but 1457 days in the total date range. This means, that there is not a sales value for each day. Therefore, the dataset is re-sampled, such that daily sales data is available. Linear interpolation is used to replace the not available sales. Figure 14 shows the resulting time series data after re-sampling the missing days. Figure 15 show the daily sales separated into the different years. The daily sales data looks quite stationary. There is also no clear seasonal component recognizable. In this project, one classic forcasting method is applied and compared to deep learning approaches. The classic method applied is the ARIMA model. The ARIMA model can only be applied in case the series is stationary. A stationary series contains a constant mean, constant variance, the autocorrelation structure is constant and there is no periodic component. The ARIMA model itself includes the ability to also difference out a linear or exponential trend. But this is described later. In order to make a first check whether the series is stationary or not, the daily sales data is divided into 10 chunks and the mean and variance are computed for each chunk. This already shows that there are no larger differences in the means and variances, which could

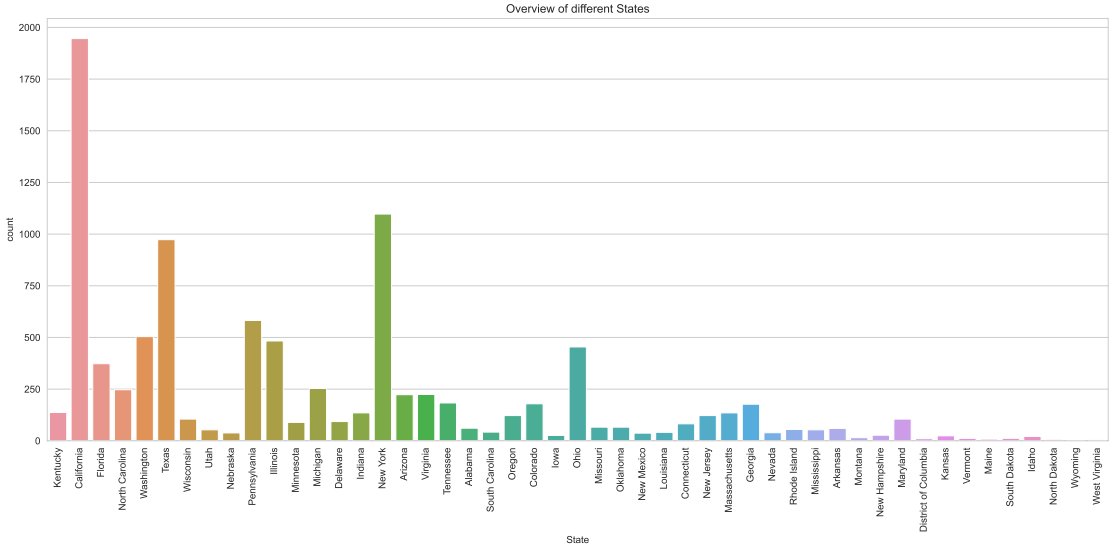

Figure 2: Countplot of all states and the ordered products for each state.

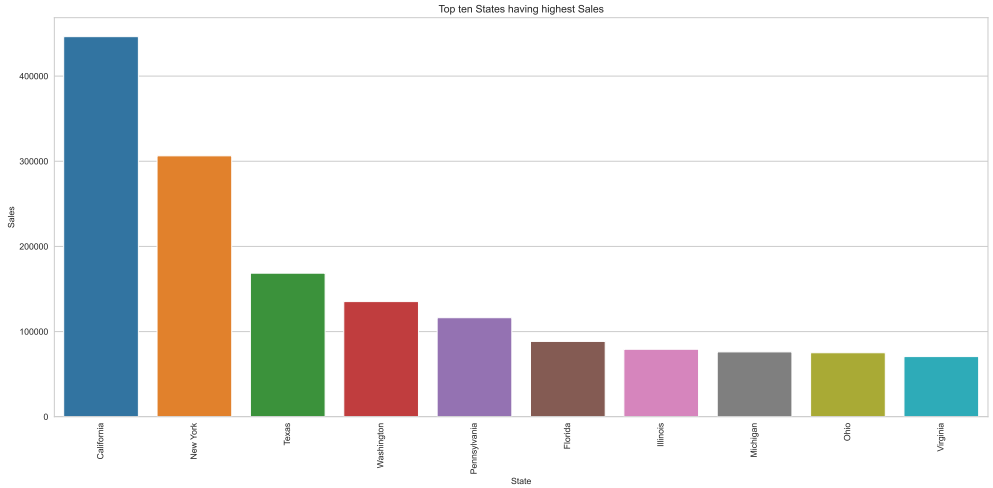

Figure 3: Top ten states having the highest sales.

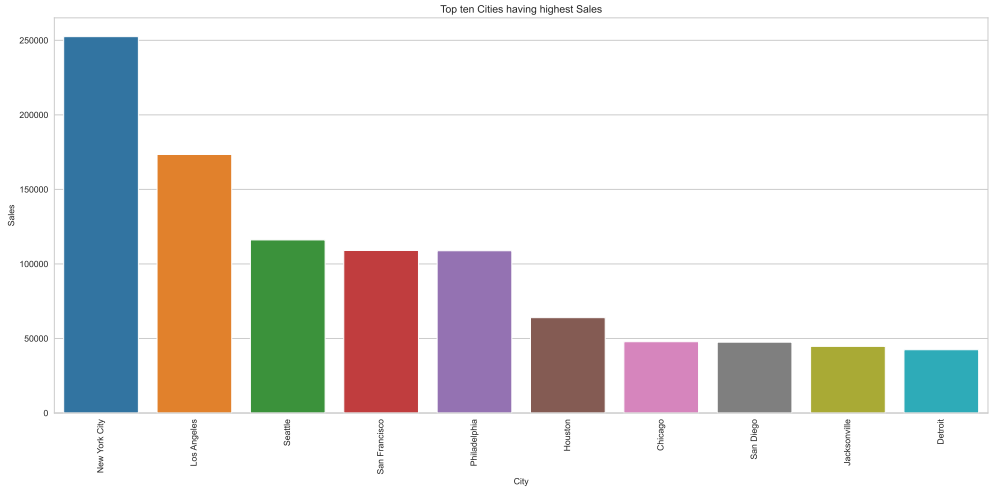

Figure 4: Top ten cities having the highest sales.

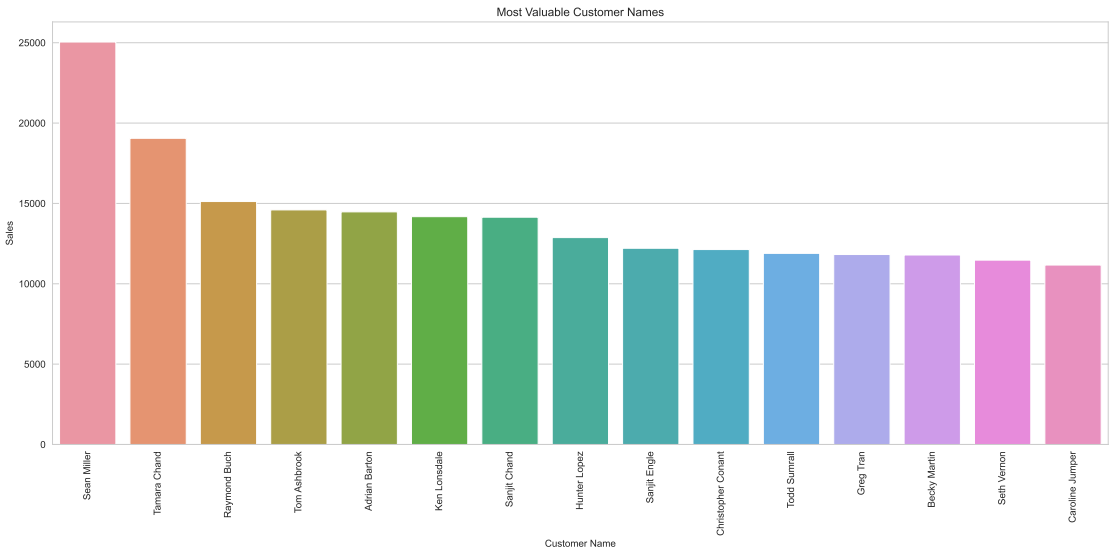

Figure 5: The most valuable customers according to their total sales.

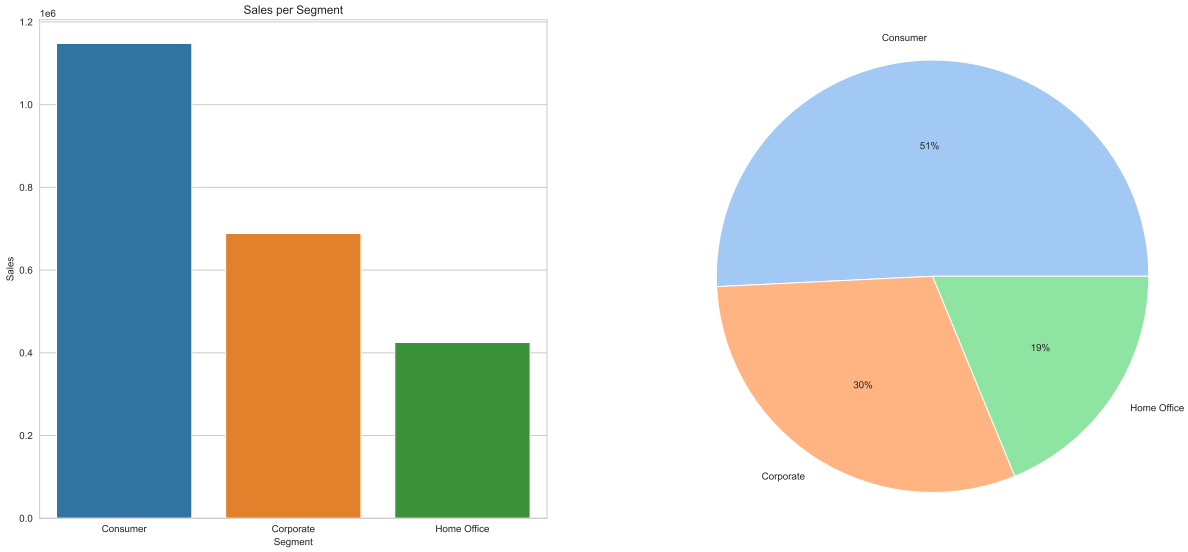

Figure 6: The amount of sales per segment as histogram (left) and as pie-chart (right). The consumer segment is the most valuable segment.

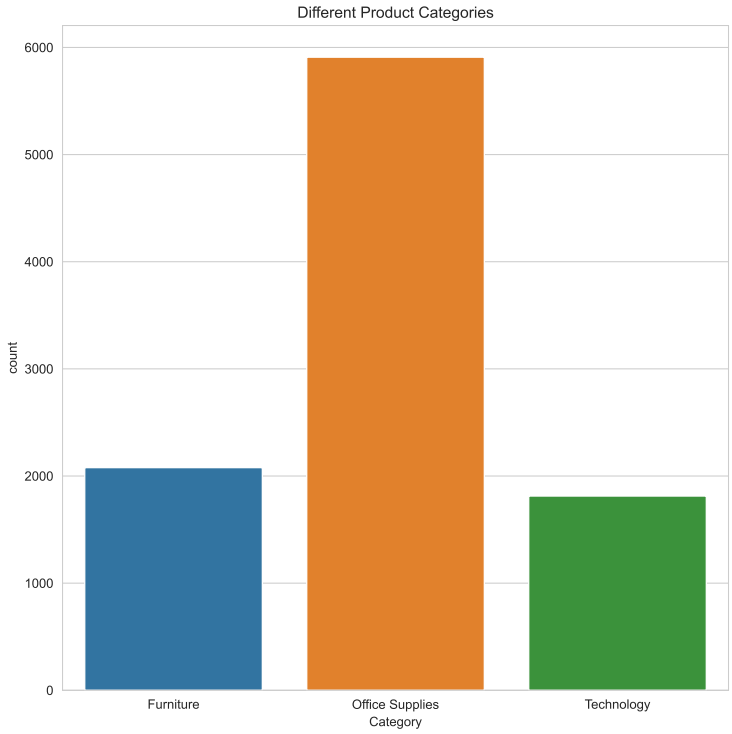

Figure 7: Overview of the different product categories and the sales per category. The office supplies category is the most valuable category.

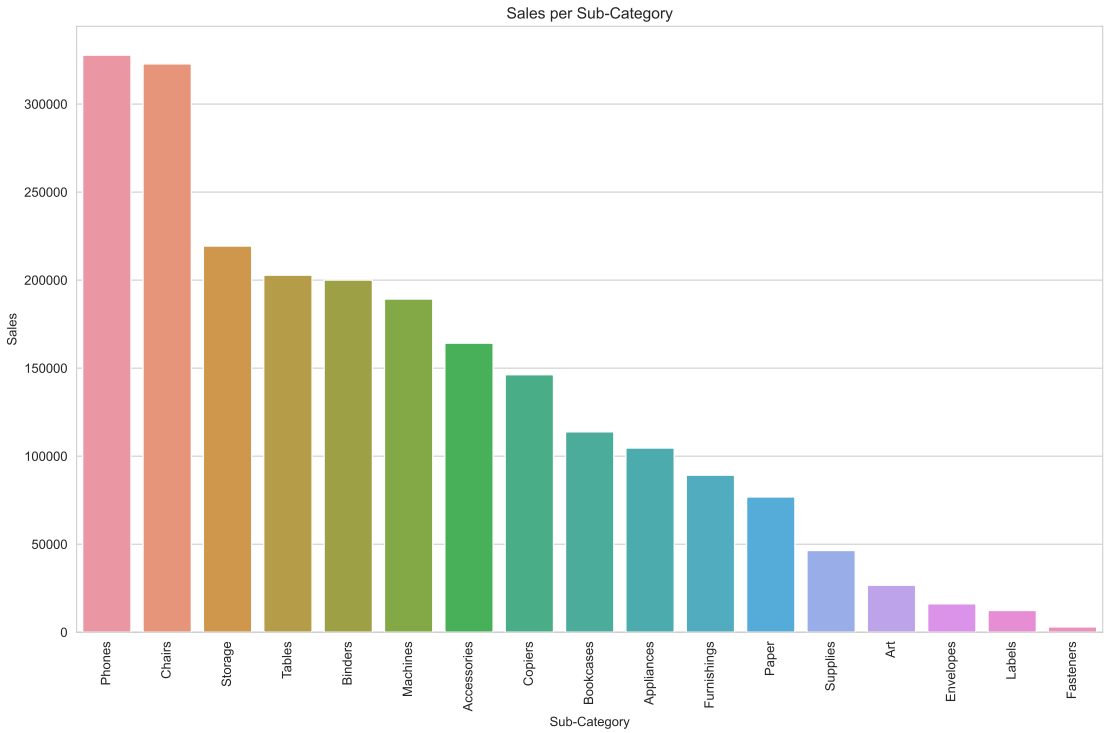

Figure 8: Overview of different product sub-categories and the sales per sub-category. Phone is the most valuable sub-category.

already imply that the data is stationary. The Augmented Dickey-Fuller test is used to make the statistical test whether the data is stationary or not. This test leads to a p-value of zero, meaning that the data is stationary. Figure 16 shows the decomposition of the time series data into trend, seasonality and residuals.

5 Methodology

This section briefly describes the training of three different time series forecasting methods. The first used method is the Auto-Regressive Integrated Moving Average (ARIMA) method. The second one is a simple Recurrent Neural Network (RNN) and the third one is a Long Short Term Memory (LSTM) RNN, which is specifically useful for long sequences. All three methods are trained on a training set and used to predict the sales for one week ahead. The RNN and the LSTM models are also capable of taking more features as input, but in the scope of this project, only the past daily sales data is used to predict the future sales data. Both RNN strategies are using 14 daily sales steps as input and are predicting one step ahead into the future. The test set for all strategies consists of the last seven daily sales samples. These values are not used to fit any method. All three methods are compared using the Root Mean Squared Error (RMSE) of the ground truth daily sales data and the predicted daily sales data.

5.1 ARIMA

The ARIMA model is perfect suited to utilize time series data in order to make a forecast about the future. The ARIMA model is a generalization of the Auto Regressive Moving Average Model

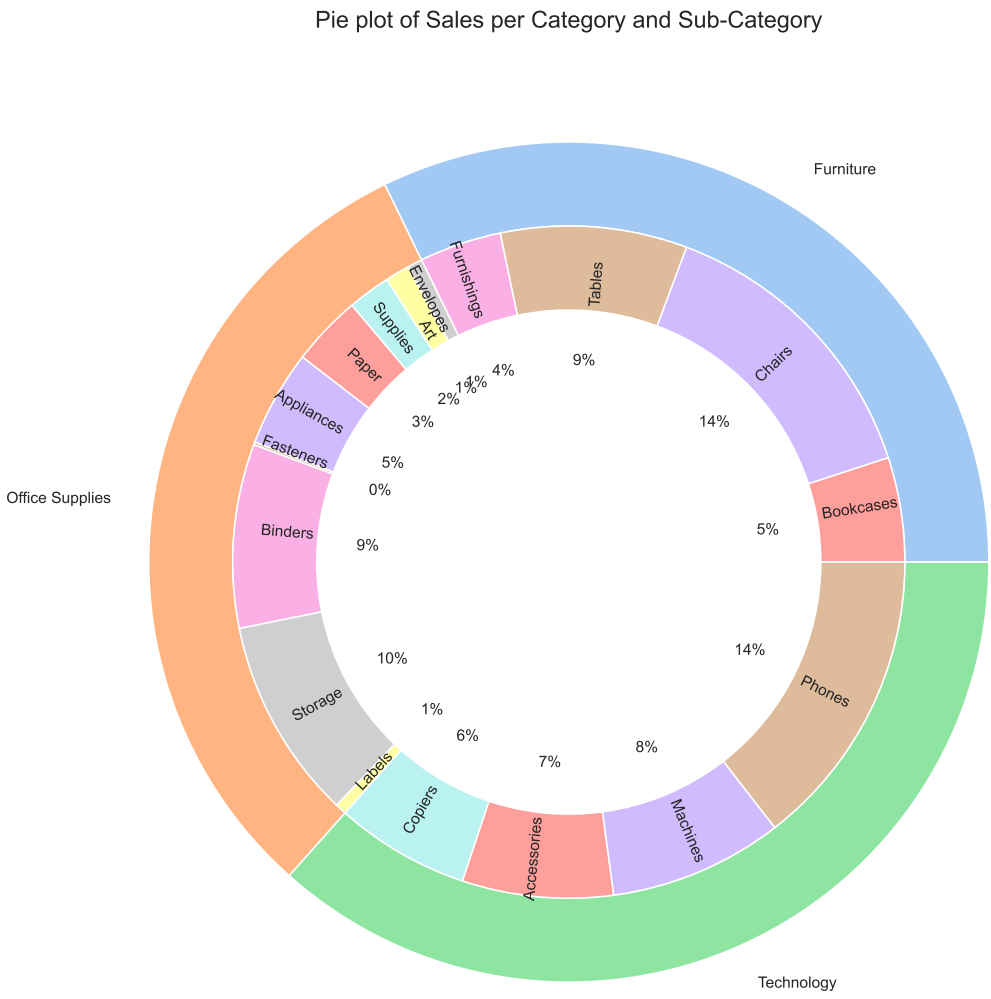

Figure 9: A pie-chart of the different product categories and sub-categories.

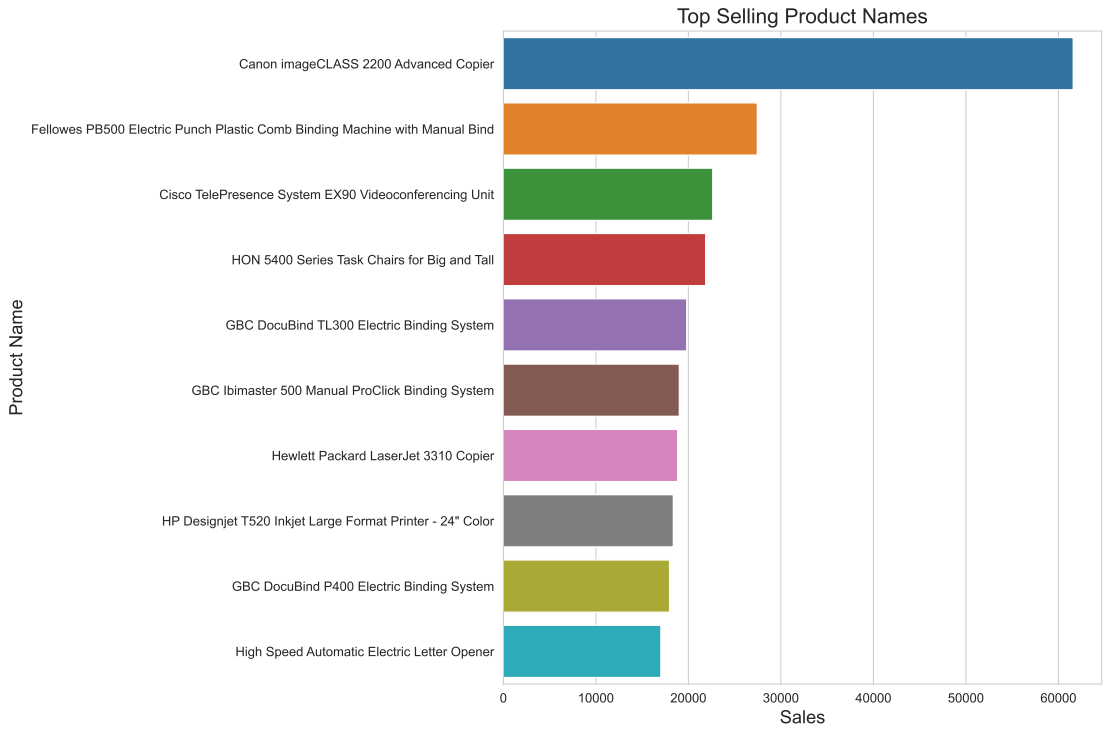

Figure 10: The product names of the top selling products.

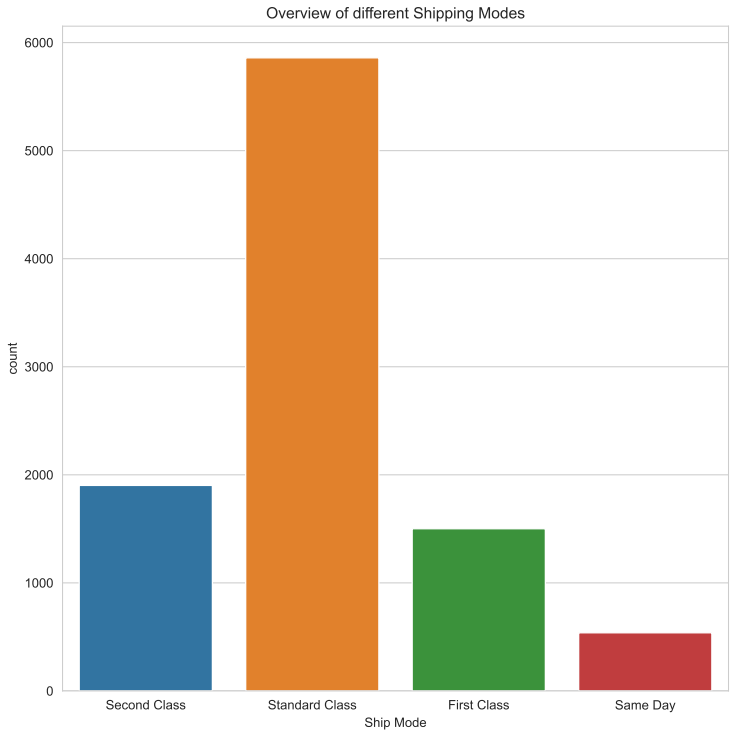

Figure 11: Overview of the different shipping modes and the amount of products for each shipping mode.

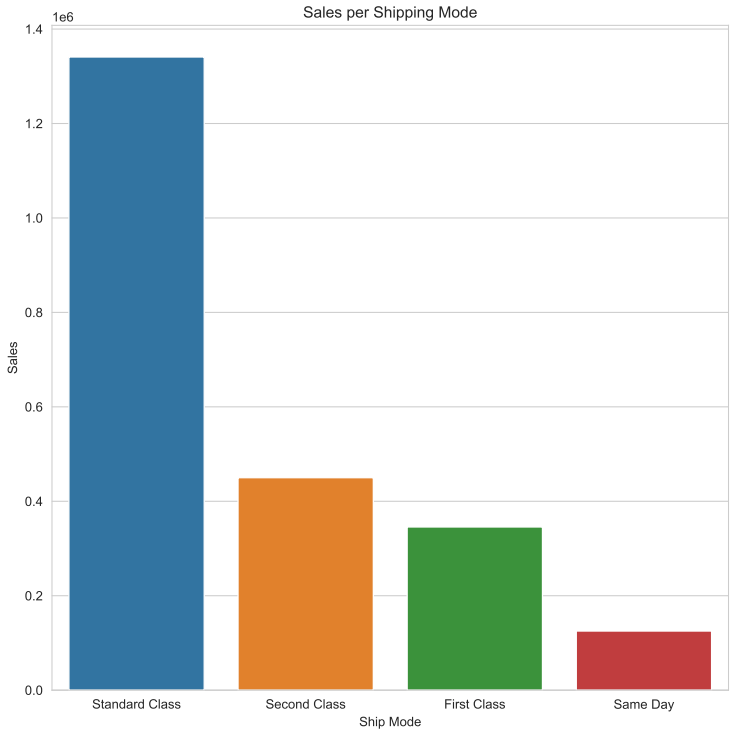

Figure 12: The sales per shipping mode.

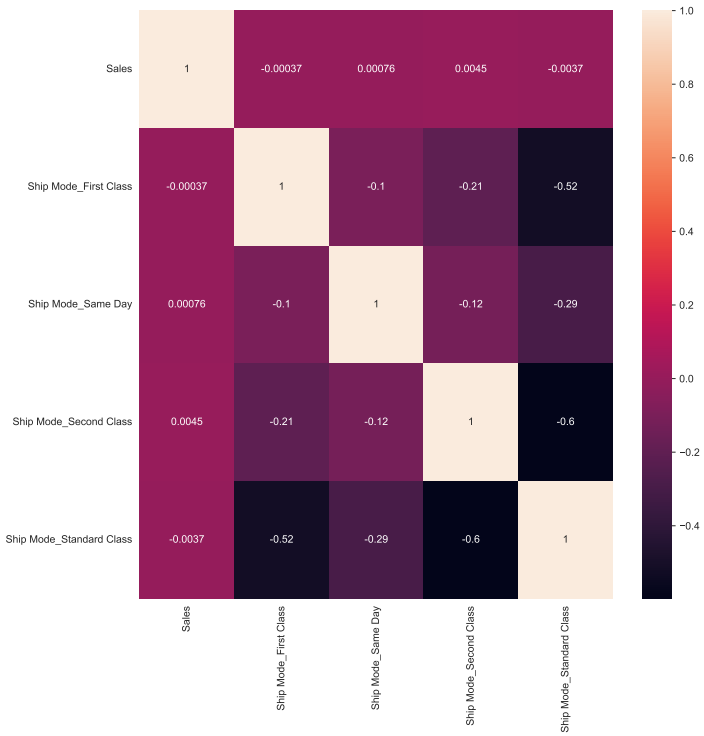

Figure 13: Feature correlation heatmap of the shipping modes and the sales.

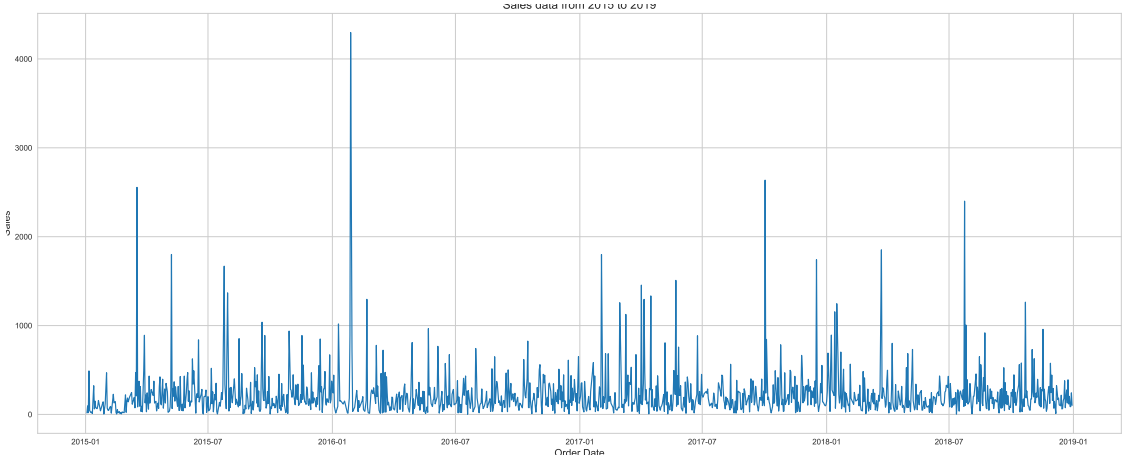

Figure 14: Daily sales data of the full date range after re-sampling the dataset such that daily sales data is available.

Figure 15: Daily sales data per year.

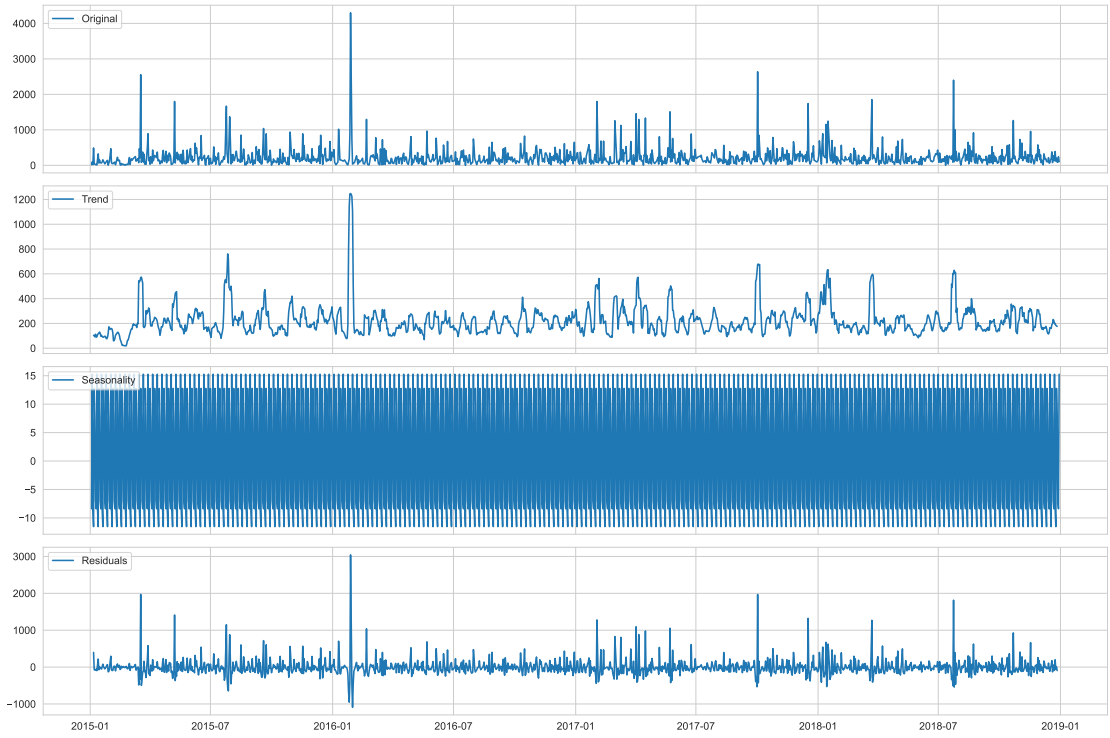

Figure 16: The decomposition of the daily sales into trend, seasonality and residuals.

(ARMA), with the advantage of being able to remove a trend by subtracting the past time step from each current time step. The ARIMA model is perfect suited for making a forecast into the near future, with the drawback of not performing well when doing a forecast long into the future. The ARIMA model as three different hyperparameters. The first one is **p**, which relates to the number of auto regressive terms to use from the past. The second one is **d**, which refers to the number of times the data should be differenced in order to remove a trend. Differencing once means that a linear trend is removed, while differencing two times is able to remove a exponential trend. By setting this parameter to zero, the ARIMA converges to the ARMA model. The last hyperparameter is **q**, which relates to the number of moving average terms to use. The moving average terms are the residuals from the last steps.

A grid search is performed in order to find the best suited hyperparameters. The Akaike information criterion (AIC) is used in order to compare the different ARIMA models to each other and to determine the optimal hyperparameters. The ranges for p and q are set to [1, 5], while the range for d is set to [0, 2]. This leads to the option to fully remove the integrated part or to remove a linear or exponential trend. The best parameters found are (p, d, q) = (3, 1, 1). These parameters are then used to fit the final ARIMA model and to make the forecast.

Figure 17 shows the results of the ARIMA model training, while Figure 18 shows the ARIMA predictions on the full dataset in comparison to the ground truth data. Figure 19 shows the one week forecast of the ARIMA model in comparison to the ground truth data. The RMSE of this forecast is at 115.86. As one can see, the ARIMA forecast is almost constant and is not really following the ground truth sales data.

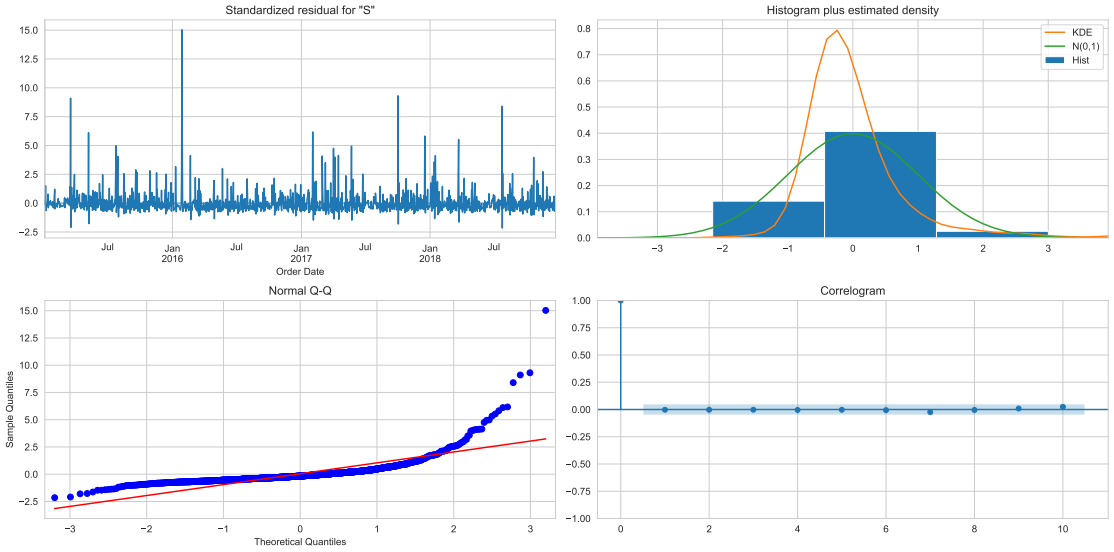

Figure 17: The results of the trained ARIMA model.

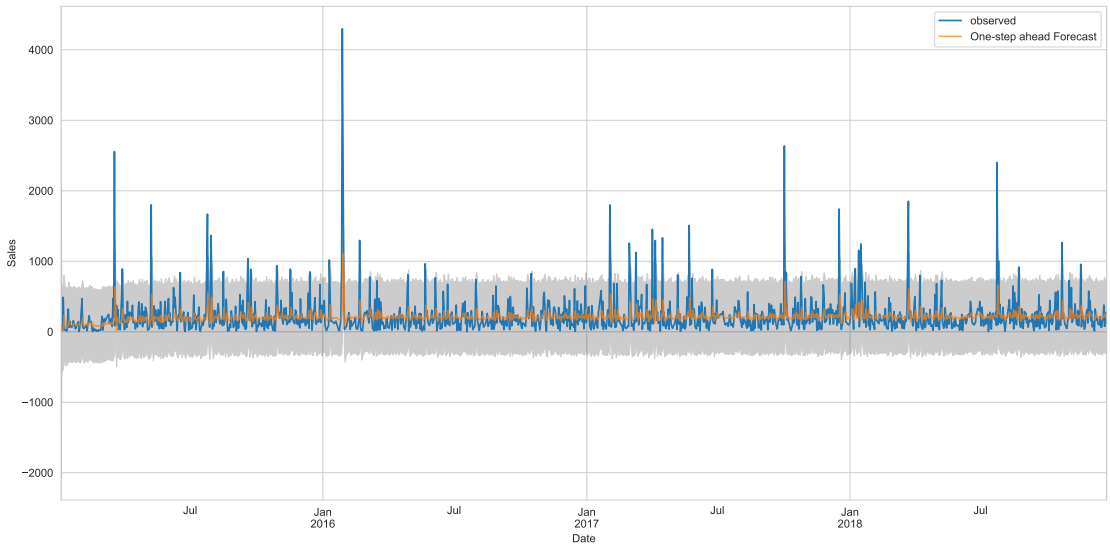

Figure 18: Comparison of ARIMA predictions and the ground truth data.

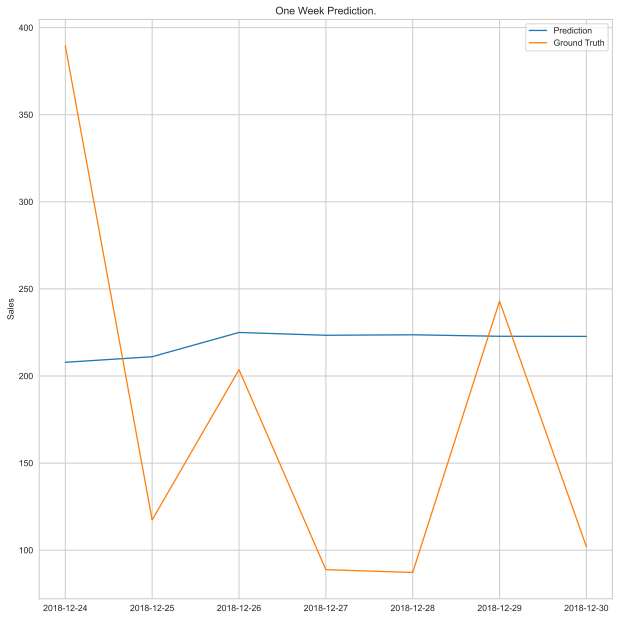

Figure 19: The results of the ARIMA one week forecast and the ground truth data.

5.2 Simple RNN

The Keras Simple RNN API is used for creating the RNN model. This API creates a fully-connected RNN where the hidden state of the previous timestep is to be fed to the next timestep. The number of units is set to 20, the activation function used is the ReLU activation function and the mean squared error is used as loss function. The simple RNN is trained for 200 epochs, with a batch size of 64, a validation size of 20% and ADAM is used as optimizer with a learning rate of 0.001. Figure 20 shows the results of the training process and the prediction in comparison to the ground truth values. As one can see, the training loss is higher than the validation loss. This could mean that the complexity of the model can be increased, by for example adding a higher amount of units or by stacking more RNN layers together. The RMSE of the one week ahead prediction is 120.86, which is worse than the ARIMA RMSE. The simple RNN is not optimized more.

5.3 LSTM

The Keras LSTM API is used for creating the LSTM model. The number of units is set to 70, the activation function used is the ReLU activation function and as loss the mean squared error is used. The LSTM model is trained for 200 epochs, with a batch size of 64, a validation size of 20% and ADAM is used as optimizer with a learning rate of 0.001. Figure 21 shows the results of the training process and the prediction in comparison to the ground truth values. The RMSE is at 107.38, which is lower than the RMSE of the ARIMA model. The LSTM is therefore the best performing of all strategies compared here.

6 Conclusion

The dataset is analyzed and the main sales drivers are found. The time series data is analyzed on stationarity, which is an important criterion for applying the ARIMA method. The methods

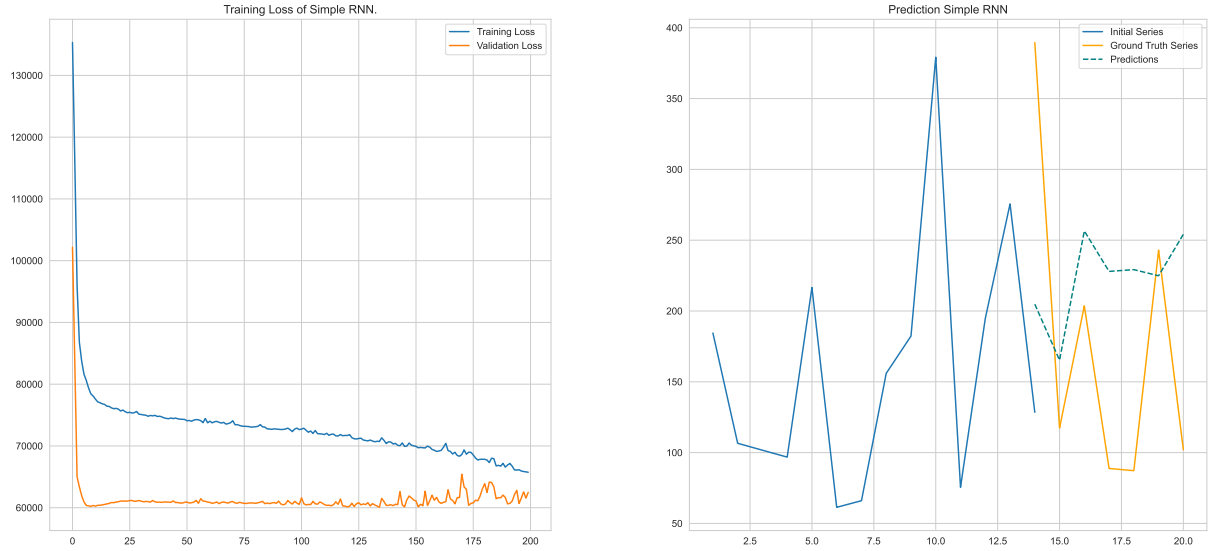

Figure 20: The training loss of the RNN training (left) and the one-week forecast of the trained RNN model in comparison to the ground truth data (right).

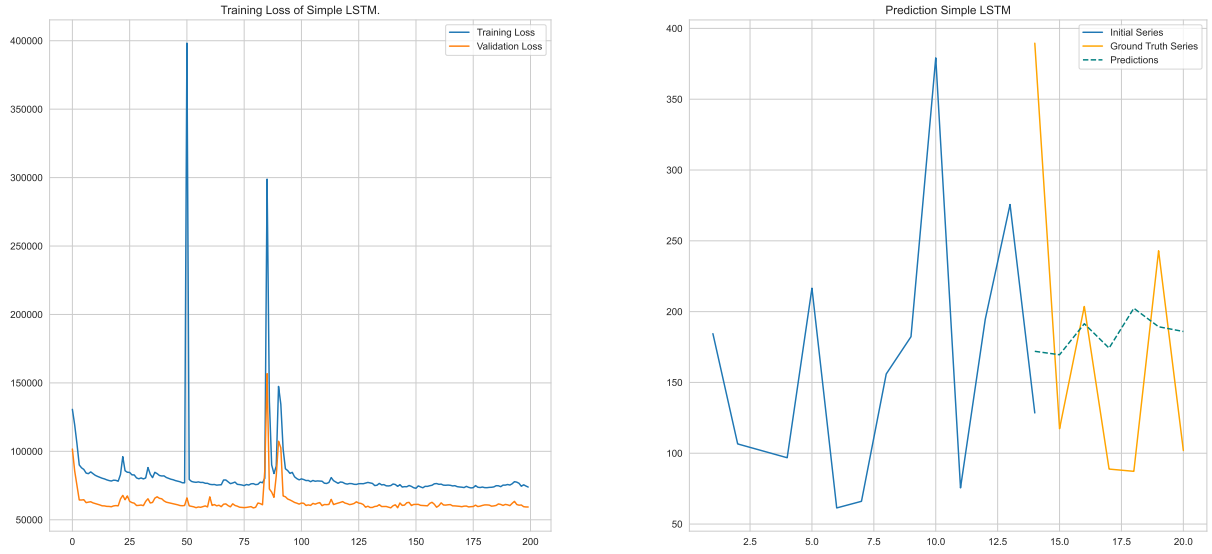

Figure 21: The training loss of the LSTM training (left) and the one-week forecast of the trained LSTM model in comparison to the ground truth data (right).

presented in this project are able to utilize the daily sales data of the past to make future forecasts. The daily sales data is pre-processed such that it can be fed to the recurrent neural networks. The LSTM approach achieves the lowest RMSE in comparison to the other strategies.

7 Future Work

Currently, the LSTM model only takes the daily sales data of the last two weeks as input and makes a one-step ahead prediction. But there are plenty of other features that could be utilized to make a better forecast into the future. As a future project, the other features are to be used as well and the LSTM should be transformed to directly output a one week forecast. In this way, other features of the last two weeks can be used to make the one week ahead forecast.

References

[1] rohit sahoo. *Superstore Sales Dataset*. 2020. URL: `https://www.kaggle.com/rohitsahoo/sales-forecasting` (visited on 09/25/2021).