Retail and Marketing Analytics Individual Assignment

Walmart Store Sales Forecasting with Multiple Linear Regression and Neural Networks

CID: 01997714

Table of Contents **Page Number**

1.1 Introduction and Delineation of the Marketing Problem

Marketing analysts across the globe have been analysing an ever-increasing amount of retail data to uncover insights and trends that could improve business operations and enhance growth. Much of this data is used to create forecasts to provide companies with information that may influence decisions surrounding production quantities, supply chain planning, advertising expenditure, promotions, etc. New technology and tools have transformed the retail industry by helping identify opportunities or capturing underlying factors which affect a company's sales.

Sales forecasting and prediction is hugely important in the retail industry to improve performance metrics such as revenue and profitability. The most common approaches used for sales forecasting include traditional statistical methods such as moving averages or linear regression. However, recently, machine learning techniques have become more prevalent in forecasting models. Literature has shown that machine learning models such as neural networks can be beneficial to reveal insights as well as accurately predict sales in the retail industry.

There has been much debate whether black box models are better than interpretable models for sales forecasting. For instance, Kaneko and Yada (2016) find that neural networks predict sales 7% more accurately than using logistic regression. Furthermore, Amiri et al. (2019) find that neural networks obtain a lower minimum squared error (MSE) when forecasting car sales than linear and exponential regression methods. On the other hand, Hallman (2019) states that there seemed to be no significant difference in predictions for powder blend sales between neural networks and linear regression models. Hallman suggests the unusual results could be the result of poor model architecture as the number of neurons in the network was not optimized.

1.2 Report Outline

This report aims to compare the prediction performance of interpretable models which use traditional statistical techniques such as multiple linear regression (MLR), ARIMA and Holt-Winters to black box models which use machine learning methods such as neural networks. Due to the word count and in the interests of brevity, this report compares two models: multiple linear regression and neural networks.

MLR is a statistical technique that uses several independent variables to predict the outcome of a dependent variable. MLR uses ordinary least squares (OLS) to make predictions. On the other hand, neural networks are a form of artificial intelligence that make predictions based off a network of neurons. The input or predictor variables form the input layer, and the predictions form the output layer. In between these layers lies the hidden layers, in which the model calculates weights through an activation function. The main difference between the methods is that neural networks require no statistical assumptions about the data compared to regression.

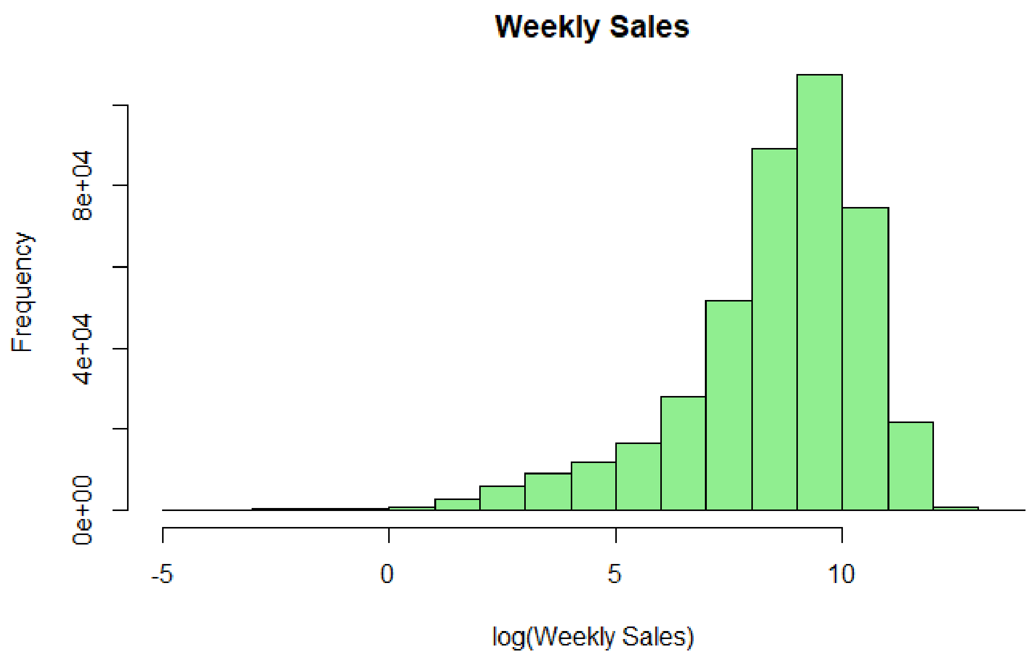
The methods were compared in terms of their prediction accuracy on weekly sales. The performance of the models was determined using two error measures: root mean squared error (RMSE) and mean average error (MAE).

1.3 The Data

The Walmart store sales dataset is used for the assignment which contains weekly sales information spanning from February 2010 to November 2012. There are three separate spreadsheets within the dataset. The features data contains several attributes including numeric data for temperature, fuel price, consumer price index (CPI), promotional markdowns (1-5), unemployment and a holiday dummy. The training spreadsheet contains the store, department, date, and weekly sales information while the stores spreadsheet contains the size and type of each store. The three spreadsheets are merged to analyse the data as one large dataset. There are 45 stores in the dataset, and each store has 99 departments. Therefore, weekly sales forecasts are created for each department of each store at every week of the time frame. Given this added complexity, several assumptions and decisions must be made to produce the best forecasting model.

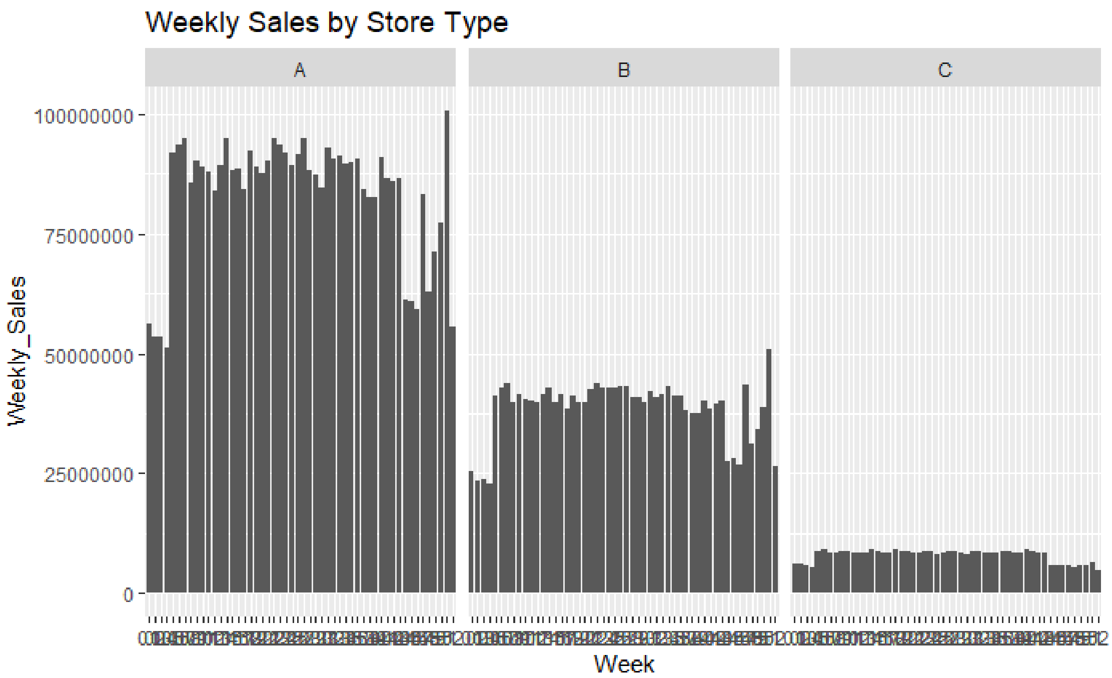
2.1 MLR Model

After merging the datasets, initial exploratory data analysis is taken to clean the data. The following histogram shows the distribution of weekly sales over the time-period:

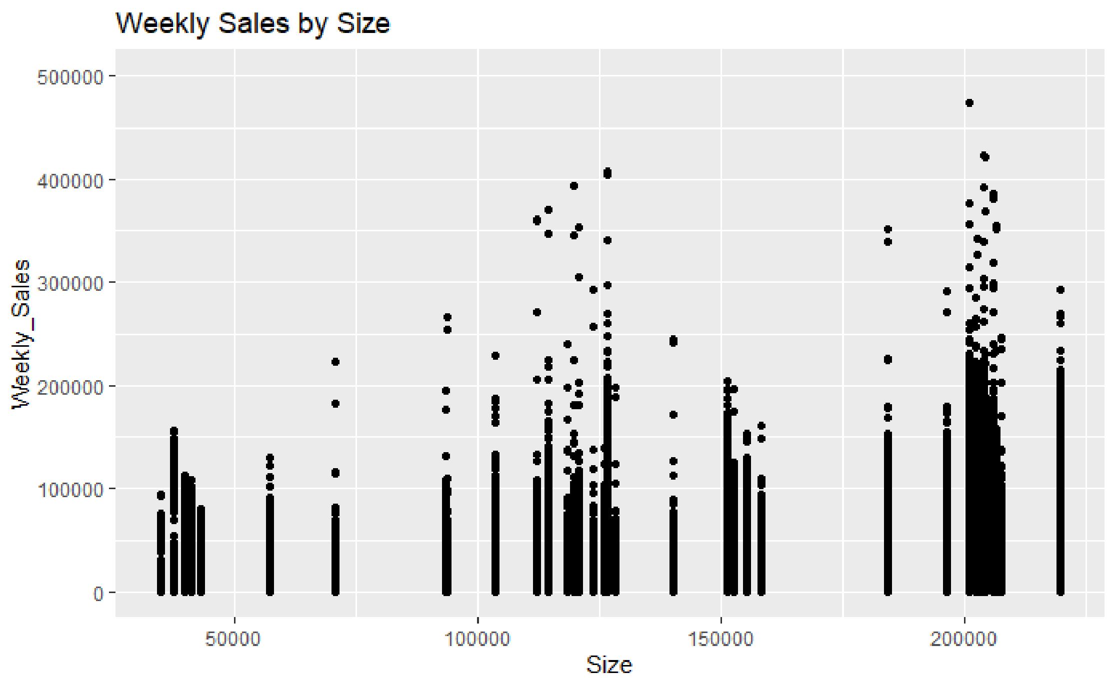


Upon close inspection, the graph above shows that some values for weekly sales fall below zero. This can be confirmed as there are 1285 values below zero. As it is not possible to have negative sales during any given week, these values are removed from the dataset. All promotional markdown columns contain NA values until 2012, hence they are replaced with 0 to avoid computational errors.

Following cleaning, the next step is to identify the significant variables to include in the linear model. Firstly, the 45 stores are grouped into types A, B and C depending on the number of sales they make:



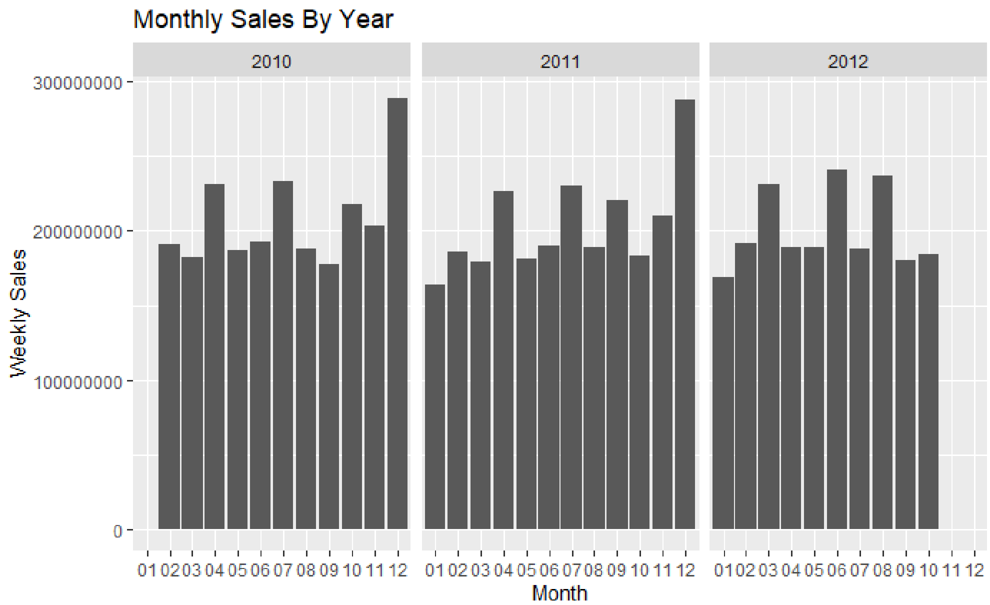
Type A represents larger stores which make more sales whilst type C denotes smaller stores which sell less. Three dummy variables are created, one for each store type, equal to 1 if a store pertains to the store type and 0 if it does not. Type C is used as the base dummy and the effect of it is reflected in the intercept. The graph below indicates that a stores size and weekly sales are positively correlated:

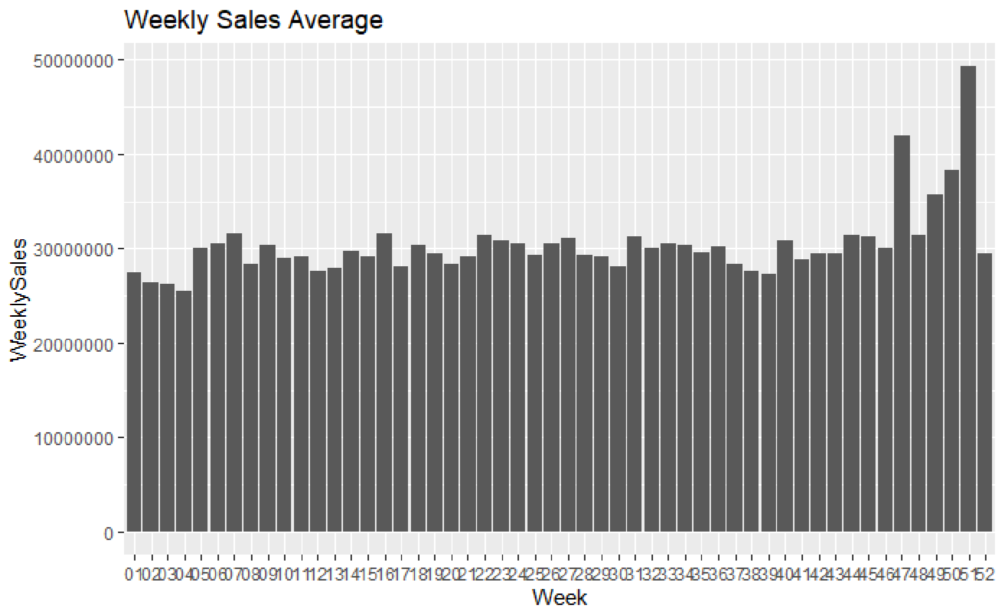


As size (store size) is highly correlated with store type, size is removed from the regression. A lagged dependent variable for weekly sales is created as there is strong reason to believe that the number of sales from the previous week influences the following week's sales. The lagged dependent variable is grouped by store and department. The variables CPI, unemployment, fuel price and the holiday variable all influence weekly sales and are added to the regression. The regression as of now can be written as:

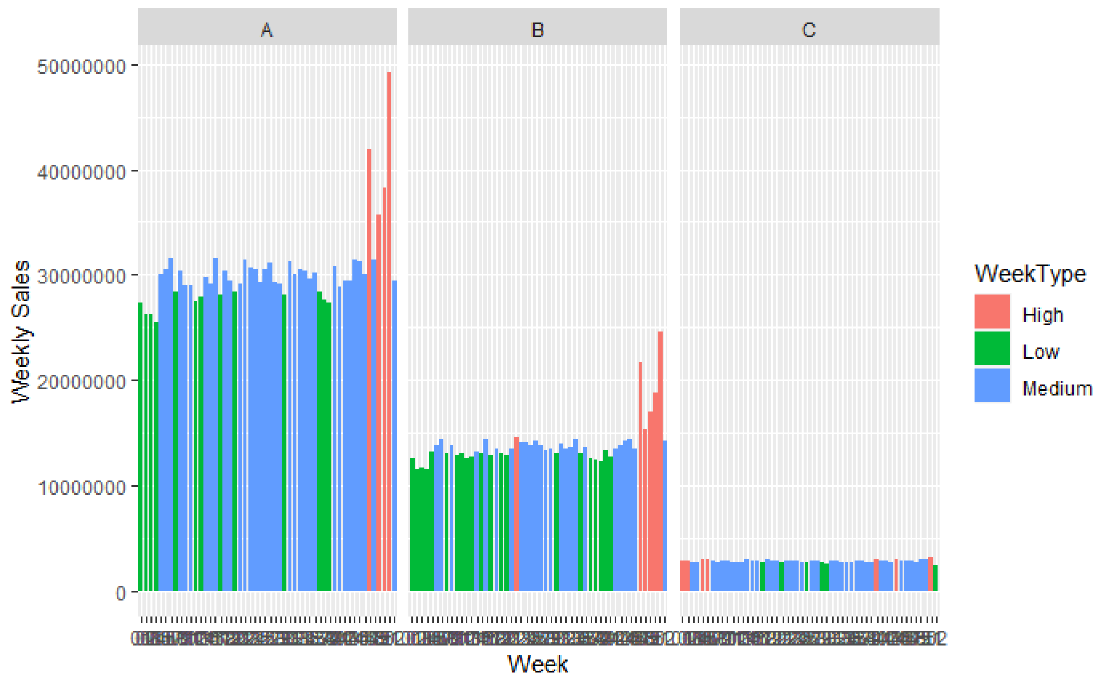
$$\begin{aligned} Weekly\ Sales_t &= \beta + \beta_1 Weekly\ Sales_{t-1} + \beta_2 Type\ A + \beta_3 Type\ B\ +\ \beta_4 CPI \\ &+\ \beta_5 Unemployment\ +\ \beta_6 Fuel\ Price\ +\ \beta_7 IsHoliday\ + \epsilon_t \end{aligned}$$

To detect seasonal trends, monthly and weekly sales by year are visualised:

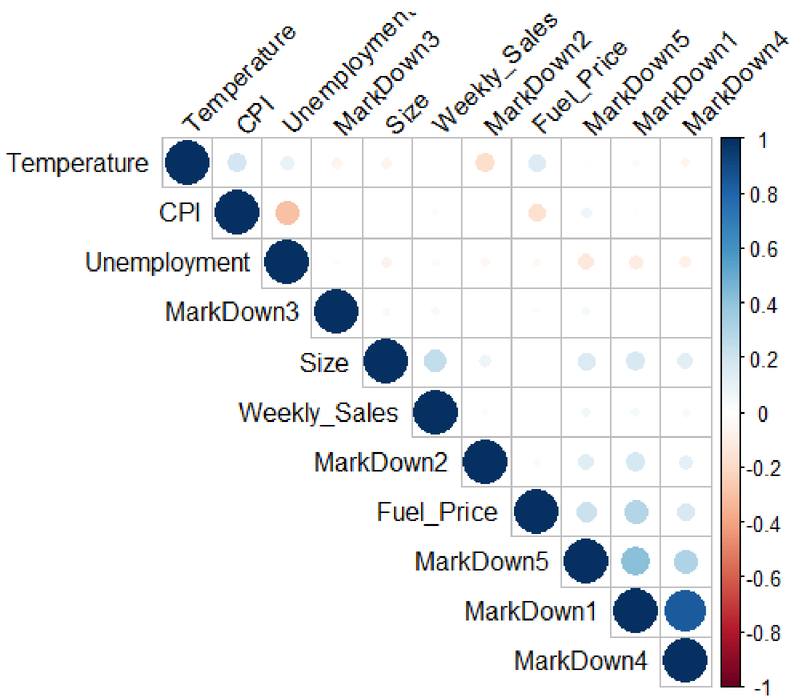




The graphs show large increases in sales towards the end of the year around Christmas, presumably as customers shop for gifts. Furthermore, the data shows that April and July have higher sales in each year and January experiences a decline in sales. As the seasonal effects appear to be irregular, a common approach is to create a dummy variable for each week, 52 in total. However, to avoid overcomplicating the regression with too many variables, the weeks (1-52) are split into three categories depending on their average number of sales: low, medium, and high. The data is split by year to calculate upper and lower thresholds to group weekly sales. If the weekly sales value is 5% greater than the mean, it is categorized as high, or if it is 5% lower than the mean, it is low, otherwise it is medium. Weekly sales grouped by store type and number of sales groups are shown below:



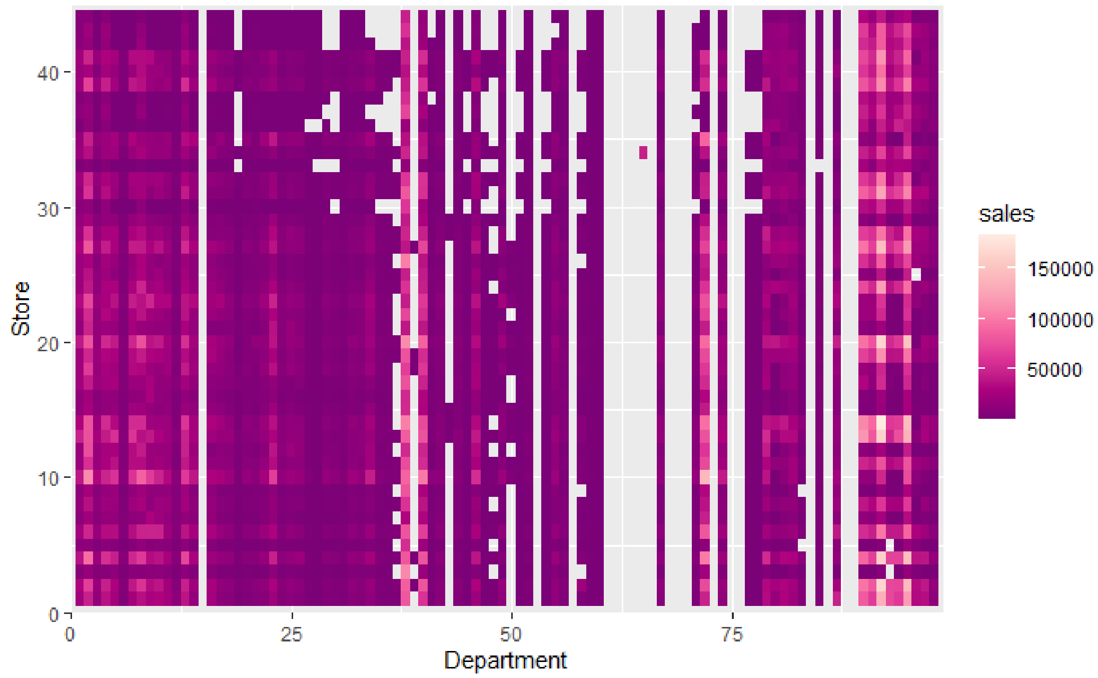
The graph shows that any outliers are categorized as high while weeks which experience lower sales are stored as low. The variable High is stored as the base dummy. Next, the markdown attributes (1-5) represent promotional markdowns which Walmart implement before holidays. By observing a correlogram, the markdowns seem to be multiplicative as they interact with each other and past weekly sales. Therefore, an interaction term is made between the sum of all markdowns and previous weekly sales:



$$
\begin{aligned}
Markdown\_interaction \\
&= sum(Markdown1 + Markdown2 + Markdown3 + Markdown4 \\
&+ Markdown5) * Weekly\_Sales\_Lag
\end{aligned}
$$

Each markdown variable and the interaction term are added to the regression model.

Finally, each store in the data set has around 81 departments (1-99). Instead of creating 81 dummy variables, one for each department, a dummy variable is created only for departments which experience excessively high weekly sales. This naive approach is taken to avoid overcomplicating the regression, which would diminish the power of the other explanatory variables. To observe which departments have high weekly sales, a heat map is created:



The heat map indicates that a few departments exhibit many weekly sales across all stores. Through filtering and observation, departments 38, 92 and 95 experience excessive weekly sales across many stores. Therefore, dummy variables are created for departments 38, 92 and 95. Gaps in the heat map correspond to missing departments from 1 to 99.

To calculate the errors of the models, the data is split into a training and test set to obtain the actual and predicted weekly sales. The Walmart sales files contain both a training and testing set, however, the test set does not contain the actual results for weekly sales needed for the error calculations. Therefore, an 80:20 randomized split is used on the training data to create a train and test set.

An AIC test is performed on the MLR to determine the best model mix. The purpose of using the AIC test was increase model complexity to improve the accuracy of the model whilst preventing overfitting. The model with the lowest AIC score will be used to forecast weekly sales. Interestingly, the interaction term has no effect on weekly sales, unemployment is also removed. The final regression and results are shown below:

$$
\begin{aligned}
Weekly\ Sales_t \\
= \beta + \beta_1 Weekly\ Sales_{t-1} + \beta_2 Type\ A + \beta_3 Type\ B + \beta_4 CPI \\
+ \beta_5 Fuel\ Price + \beta_6 IsHoliday + \beta_7 Low + \beta_8 Medium \\
+ \beta_9 MarkDown1 + \beta_{10} MarkDown2 + \beta_{11} MarkDown3 \\
+ \beta_{12} MarkDown4 + \beta_{13} MarkDown5 \\
+ \beta_{14} Dept38 + \beta_{15} Dept92 + \beta_{16} Dept95 + \epsilon_t
\end{aligned}
$$

MLR Results

The following equation shows the estimated coefficients of the MLR model:

$$
\begin{aligned}
Weekly\ Sales_t \\
= 4{,}462.51 + 0.929 * Weekly\ Sales_{t-1} + 1{,}191.24 * Type\ A + 401.26 \\
* Type\ B - 1.656 * CPI - 212.93 * Fuel\ Price - 1{,}387.7 * IsHoliday \\
- 3{,}016.93 * Low - 2{,}661 * Medium + 0.04 * MarkDown1 - 0.15 \\
* MarkDown2 + 0.08 * MarkDown3 + 0.05 * MarkDown - 0.06 \\
* MarkDown5 \\
+ 3{,}923.7 * Dept38 + 5{,}310.42 * Dept92 + 4{,}827.3 * Dept95 + \epsilon_t
\end{aligned}
$$

The results show that the model has a R^2 of 89.6% [1], meaning that around 90% of the variation in weekly sales can be estimated by the model. Furthermore, the coefficients in the model make sense as store type A which represents larger stores have a greater effect ($+1{,}191.24$) on weekly sales. Low has a negative coefficient of -3,000, indicating that a week which generally experiences lower sales can reduce the total amount by 3,000 units. The departments which experience high sales all have highly positive coefficients. All variables are statistically significant at the 1% level.

After estimating the MLR model using the training set, the weekly sales of each store and its corresponding department at each week can be forecasted. The RMSE can be calculated as:

$$
RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}
$$

Where predicted denotes the predicted weekly sales for each department of each store at a given time. N represents the number of observations in the test set which is 84,097. The MAE is calculated by:

$$MAE = \frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}} |Predicted_i - Actual_i|$$

The results for the errors of the MLR are as follows:

Model	RMSE	MAE
MLR	7654.66	2534.27

Although these results change every time the code is run, they remain approximately the same. Both the RMSE and MAE results are relatively low measured at 7654 and 2534, respectively. General discussions online state that a good model for this data set has a MAE below 3000.

3.1 Neural Networks Model

To apply neural networks to predict weekly sales, the numeric data in the data set must be normalized. As part of data pre-processing, all numeric columns are scaled using a max – min normalize function to scale the data between values of 0 and 1. After the data has been scaled, an 80:20 random train-test split is applied. The same regression formula is applied to the neuralnet library in R to estimate the results.

The number of hidden layers in the neural network is set to 1. Determining the number hidden layers and neurons is an important part of neural network modelling. A larger number of hidden layers creates a high amount of complexity of the neural network and increases the number of adjustable weights and biases of the parameters. In contrast, less neurons in hidden layers can hinder the networks accuracy in describing patterns and relationships between the inputs and outputs. There is only 1 hidden layer in this model as there is limited computational power to estimate a model with a higher number of hidden layers.

3.2 Neural Network Results

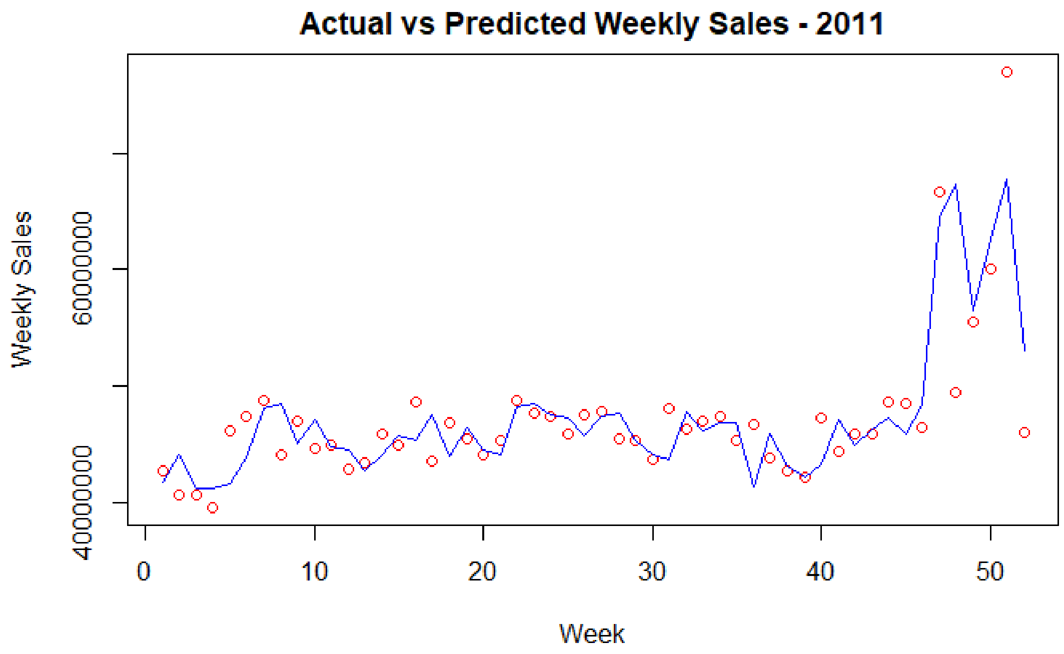
The residual sum of squares error for the neural network is 16.08 which could be considered relatively low given the size of the dataset. The data is unscaled and reverted to its previous form to predict weekly sales for the test set. The RMSE and MAE are calculated for the neural network method:

Model	**RMSE**	**MAE**
Neural Network	26,656.32	13,411.24

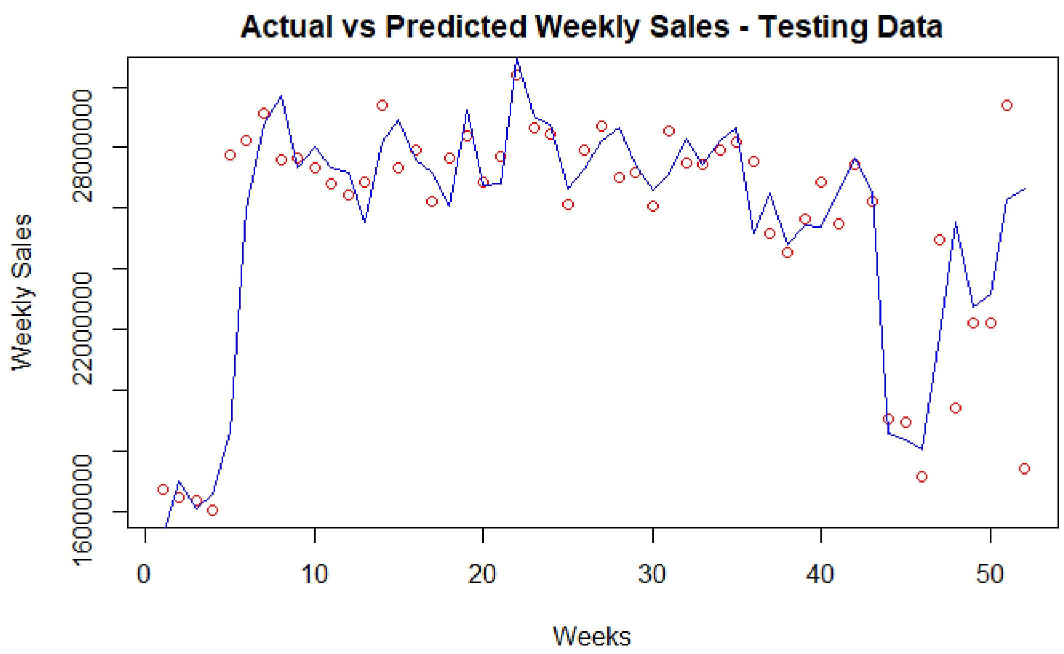
Interestingly, the RMSE and MAE are both lower than the MLR model. The results are further analysed to find out why.

4. Discussion and Limitations

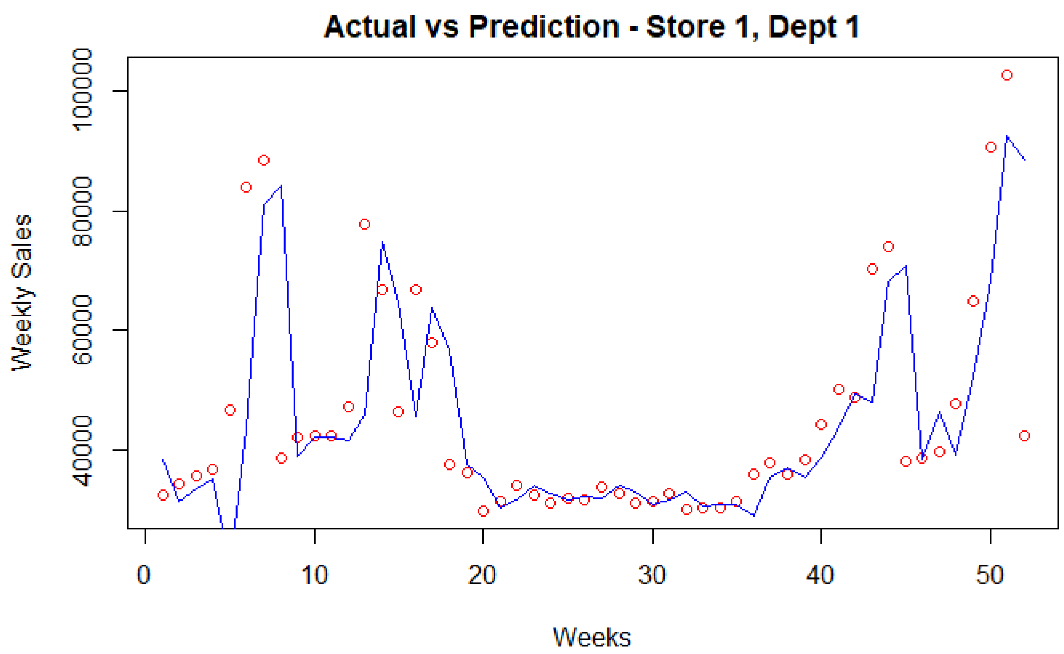
The neural network model performed worse than the MLR based on the RMSE and MAE values. The RMSE of the neural network is particularly high as the RMSE tends to penalize larger errors in the predictions. Many retail stores may want to avoid large RMSE prediction errors in sales as it may cause supply chain problems. To maintain inventory, stores could use sales predictions to automatically order supplies, large predictions errors may lead to stockpiling issues. Several graphs are created to further visualize the predicted weekly sales compared to actual sales, the first shows how the MLR fits the training data:



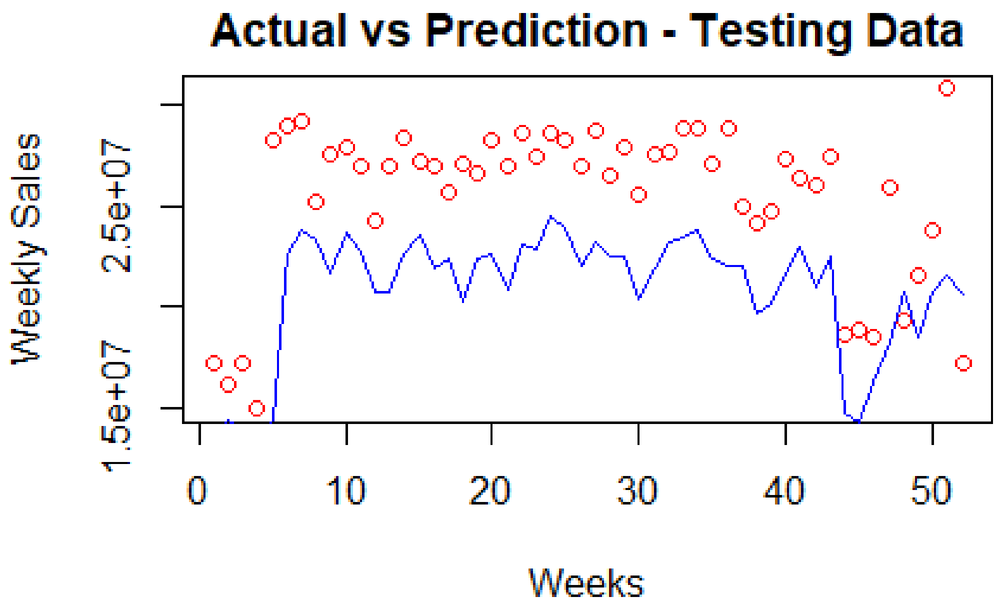
The MLR weekly sales predictions for 2011 are shown by the blue line and the actual weekly sales are represented by the red dots. These are the total weekly sales aggregated over each store and department. As we can see, the predictions fit well and follow the trends in the data. We can also plot the testing dataset:



Again, the actual and predicted sales for the test set move with each other. Therefore, overall, the MLR has a high accuracy. The accuracy of the MLR can be further observed by looking at the weekly sales from a specific department and store from the testing dataset:



The graph shows the actual compared to the predicted weekly sales for department 1 of store 1. Even at a deeper level, the model predicts accurately. The neural network weekly sales predictions can now be visualised:



The graph indicates that although the neural network model follows the general pattern in the testing data, it predicts too low compared to the actual weekly sales. The poor performance of the neural network is unexpected, since several studies praise the forecasting power of neural networks. The neural model shows very high standard deviations in both the RMSE and MAE compared to the MLR. The erratic predictions could be a result of the parameter settings as only one hidden layer is used in the network. The neural network aimed to be optimized by testing the RMSE for 1 to 10 hidden layers, but the computational power was too low given the large dataset. Furthermore, different normalization methods were tested, although the result remained the same. Therefore, the poor results from the neural network were due to both a lack of computational power and lack of sufficient knowledge on how to find the optimal neuralnet settings in R. A more simplified model with better explanatory variables could reduce the run time of the neural network model.

5. Conclusion

In this report, MLR and neural networks were used for predicting weekly sales. The MLR model made very accurate sales forecasting predictions producing both a low MAE and RMSE. The most important takeaway was that the neural network model performed considerably worse in its predictions than the MLR. However, the results could have been hugely improved with an optimized network and better computational power. With proper training, an optimal algorithm over the neural network could have performed better than the MLR model.

6. Reference List

Kaneko, Y. and Yada, K. 2016. A Deep Learning Approach for the Prediction of Retail Store Sales. *ICDMW*. **16**(7-8), [no pagination].

Hallman, J. 2019. A comparative study on Linear Regression and Neural Networks for estimating order quantities of powder blends. Thesis. KTH Royal Institute of technology.

Amiri, N.S., Farahani, D.S. and Momeni, M. 2016. Car Sales Forecasting Using Artificial Neural Networks and Analytical Hierarchy Process. The Fifth International Conference on Data Analytics. ISBN: 978-1-61208-510-4.

7. Appendix

1. MLR Results Table

	Dependent variable:
	Weekly_Sales
Weekly_Sales_Prev	0.920***
	(0.001)
IsHoliday	-1,387.709***
	(53.929)
Temperature	-6.147***
	(0.768)
Fuel_Price	-212.932***
	(30.649)
CPI	-1.656***
	(0.338)
MarkDown1	0.038***
	(0.004)
MarkDown2	-0.148***
	(0.003)
MarkDown3	0.080***
	(0.002)
MarkDown4	0.046***
	(0.006)
MarkDown5	-0.057***
	(0.003)
WeekType_Low	-3,016.953***
	(54.659)
WeekType_Medium	-2,660.947***
	(52.638)
Type_A	1,191.242***
	(45.203)
Type_B	401.264***
	(46.067)
Dept_38	3,923.707***
	(107.012)
Dept_92	5,310.416***
	(110.802)
Dept_95	4,827.301***
	(109.313)
Constant	4,462.513***
	(135.116)
Observations	336,228
R2	0.896
Adjusted R2	0.896
Residual Std. Error	7,306.551 (df = 336210)
F Statistic	170,996.800*** (df = 17; 336210)
Note:	*p<0.1; **p<0.05; ***p<0.01

2. Neural Network Results Table

```
                     Length  Class      Mode
call                       6 -none-     call
response              336228 -none-     numeric
covariate            5715876 -none-     numeric
model.list                 2 -none-     list
err.fct                    1 -none-     function
act.fct                    1 -none-     function
linear.output              1 -none-     logical
data                      33 data.frame list
exclude                    0 -none-     NULL
net.result                 1 -none-     list
weights                    1 -none-     list
generalized.weights        1 -none-     list
startweights               1 -none-     list
result.matrix             23 -none-     numeric
```

```{r}
nn$result.matrix['error',]
```

```
   error
16.07947
```