Final Group Project

Sales Prediction using R

Riya Srivastava

Akash Sharma

Northeastern University

ALY 6015: Intermediate Analytics

Introduction

Evolution is the change of nature. Evolution in the field of computer science has taken new forms and divided into new branches such as Data Science etc. Data in today's world is being generated at a very high rate every second. Therefore, it has become necessary for business to manage and study that data in order to enhance consumer experience and stay ahead in the competition.

In order to predict and work with data, companies have designated data teams which collect and work with data. In this project, we have collected sales data for a company and will make predictions for the future using various techniques. Predictions are based on the historical trends and purchasing behaviors of the consumers. It is this data which helps us in creating models, applying techniques to train and model our data.

The process on a high level is easy to list out as it involves studying the historical values, clean and wrangle the data and create a model to train predict a few values.

This report presents a step by step process of working with one such kind of data and is explained using relevant plots.

Analysis

This report gives details on the project of Sales Prediction. We have taken the Sales data of a company and analyzed it to predict the Weekly Sales of products. Initially, we had three datasets namely Stores, Sales and Features. The Stores dataset consists of the details of Store, Type and Size. The Sales dataset contains the type of store, department, date, weekly sales and whether that day was a holiday or not. The features dataset consists of store, temperature, unemployment details, fuel price, CPI and markdown details. The files for datasets have been attached as external files.

Step 1: Installing required libraries

The first step was to install and invoke the necessary libraries that were required for this project. It was important to identify the libraries needed for the implementation of code such as : ncvreg, bigmemory, data.table, glmnet, dplyr, stats, data explorer. Apart from this, there were a few libraries that we installed while writing the code that were required to execute some pieces of code.

Step 2: Reading and Merging the data

The next step is to read the data files of our data using read.csv() and choose the files. The features file was read first followed by stores and sales. The next step was to merge the data. We merged the Sales and Features datasets and then merged the resultant with Stores naming it Final_df. The key column in merging the data sets was ‘store’ column because it is the primary key and is the common column in all three datasets.

Step 3: Reforming String Values

We reformed the string values of Is_Holiday column into 0 and 1 so that all columns in final_df have numeric values.

Step 4: EDA and Correlation Matrix

A vital step in working with data is the process of performing visual exploratory analysis to know more about the attributes of the data. To know more about our data, we created a correlation matrix to figure out the correlations between the variables of our data. We used plot_correlation() present in DataExplorer library. This function ignores two columns of Dates and plots all the other variables.

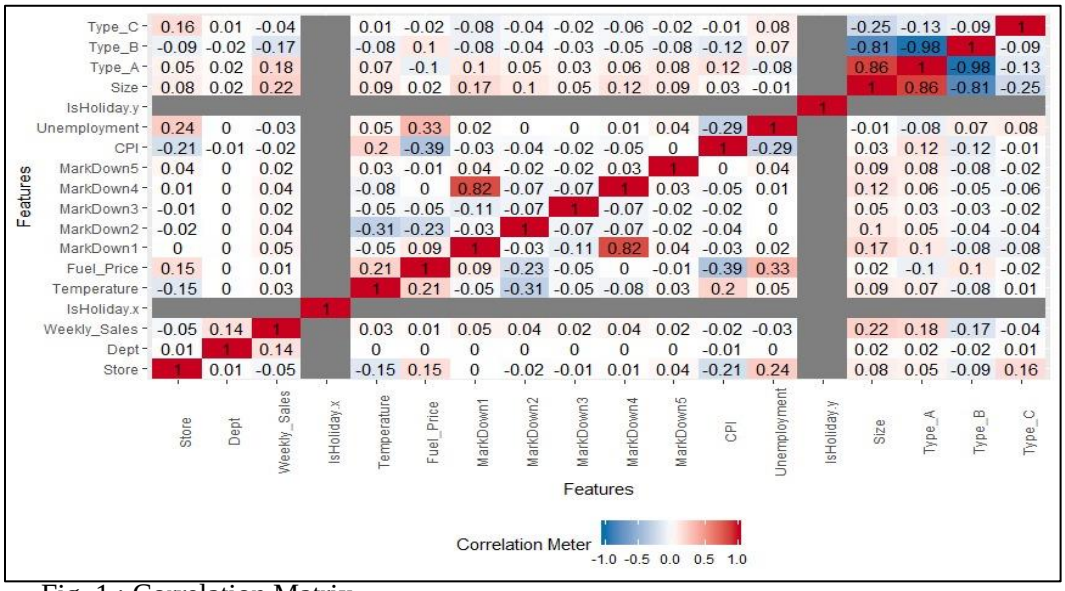

Fig. 1 : Correlation Matrix

In Fig. 1, we can see that the highly correlated variables are highlighted in red and the negatively correlated variables are highlighted in blue. There is no or negligible correlation between the variables in white boxes. We can eliminate the variables with high correlation as they will not contribute much in the analysis and might cause overfitting such as Markdown 4 and Markdown 1.

Step 5: Creation of Models for Predicting Sales

The next step is to create models for our data and predict our target variable that is Weekly Sales.

We decided to create LASSO model which is a part of regression as it gives the prediction accuracy for the data. This is a technique that helps in minimizing overfitting for data. It works by creating co-efficients for the variables and then penalizing them[1].

We begin by creating a matrix of our final dataset final_df using the as.matrix() function and then apply glmnet() and use Weekly Sales in the final dataset as y component. For performing LASSO regression, we set the value of alpha which is the elastic net mixing parameter to 1.

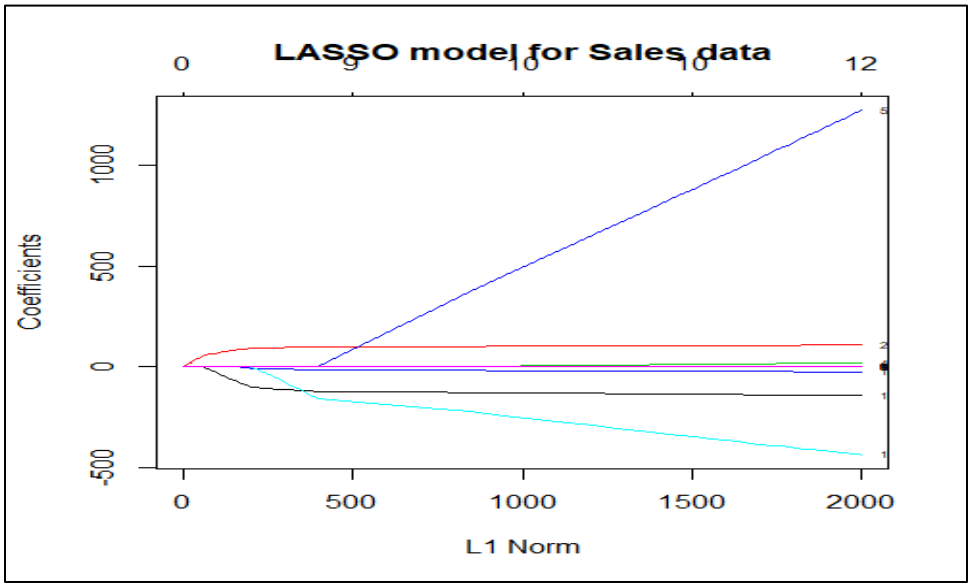

Fig.2: LASSO plot

From Fig. 2, we can see that each curve in the plot belongs to a variable in the order of the columns in our code. All the curves are shrinking towards a central point. Variable 5 that is fuel price is the last to enter the model with maximum deviation from the central point. As our tuning parameter λ varies, each value of co-efficients of variables is plotted against L1 norm which penalizes the model. As the value of L1 norm increases, the variables form their own path. The output of LASSO has been attached as external file. The values of lambda stop generating once the percentage of deviance starts to generate similar values with respect to the degrees of freedom.

Step 6: Time Series

We will use time series to predict the future values of our Weekly Sales based on the previous values of Weekly Sales from 2010 to 2012. To create time series for our data, we need to ensure that our time series is stationary.

```
         Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
2010  8229.77  9262.33 11284.27  9612.87  9587.35  7784.82  8311.17  7699.00  7634.97  8111.01  9296.13 11746.83
2011 12338.08 11420.63 10199.01 12748.13 11994.48 11947.03 12178.80 11448.84 12109.21 10747.37 10565.39 11730.26
2012 11109.25
```

We can see from the above output for time series that the data has been divided into short-term cycles of 12 for each month and are present from January, 2010 to January, 2012.

Then, we need to figure out whether our data is seasonal or not. If the data is non-seasonal, we apply smoothing techniques such as SMA(Simple Moving Average). If the data is seasonal, we find it's components and then decompose the time series.

In this case, we have seasonal data and we have divided it in cycles of 12. We have first found the seasonal component of our decomposed time series [2]. After which we have seasonally adjusted the time series by subtracting the seasonal component from the original times series. We now have our seasonally adjusted component.

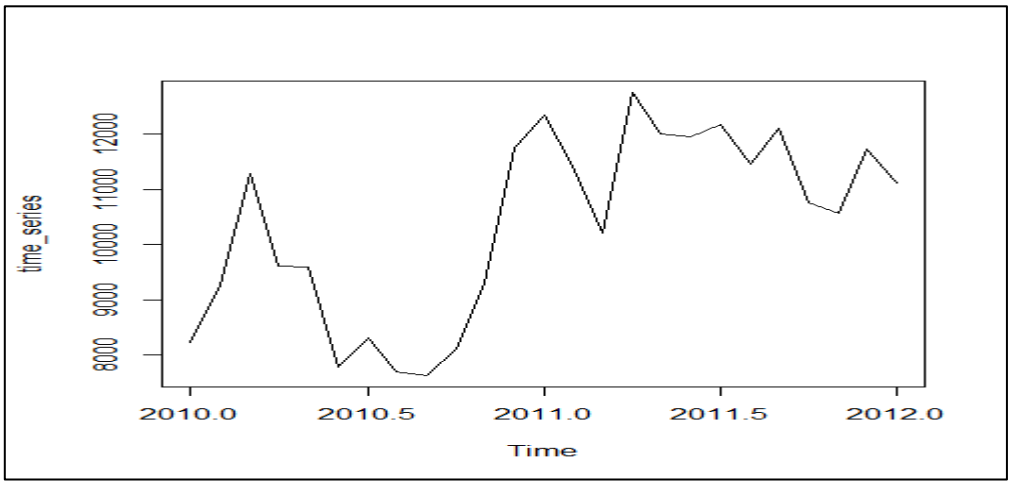

Fig.3: Original time series

It is clear from the above figure that the time series is not stationary and needs to go through the process of smoothing.

Figure 4 displays the decomposed components of the time series namely irregular, trend, seasonal and random.

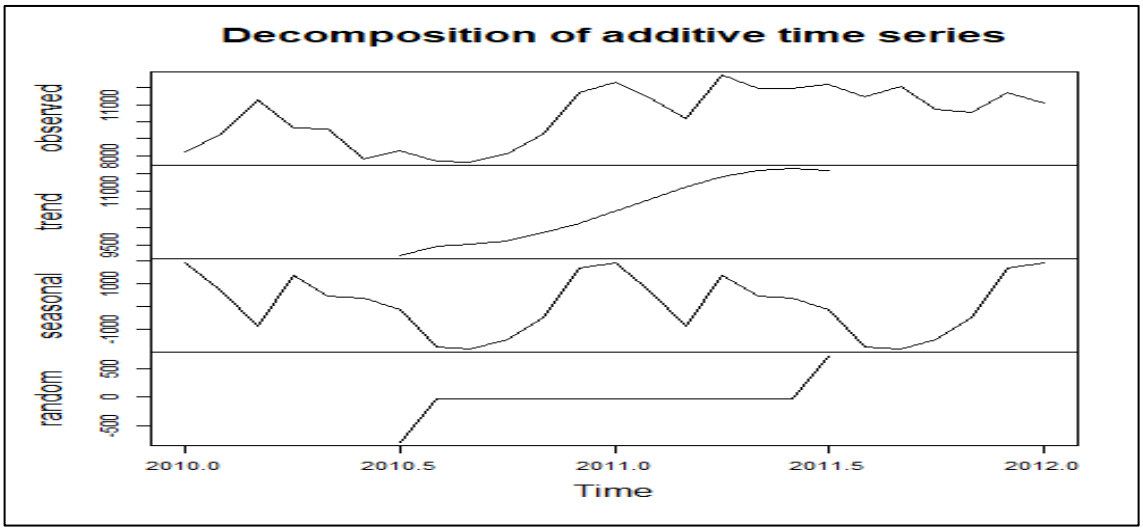

Fig.4: Decomposed components of original time series

Thereafter, we have performed the ADF test which tests the presence of unit root[3]. The ADF test helps in finding the lag order and test statistic. Following, we will create an ARIMA model and it's order. To create an ARIMA model, we need three components namely p, d and q. The value of p is obtained from the autocorrelation function acf(). The next autocorrelaton function is derived from the pacf() which gives us the value of q. The value of d depends on the value of

differencing order. In this case, the value of differencing is 3 as we have obtained a relatively stationary plot at order 3.

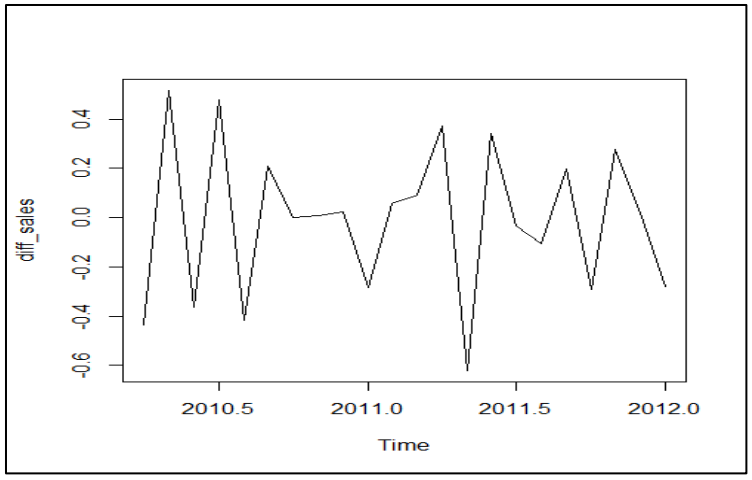

Fig. 5: Differencing plot

After applying acf() to the differenced series with lag.max of 20, we obtain the following result:

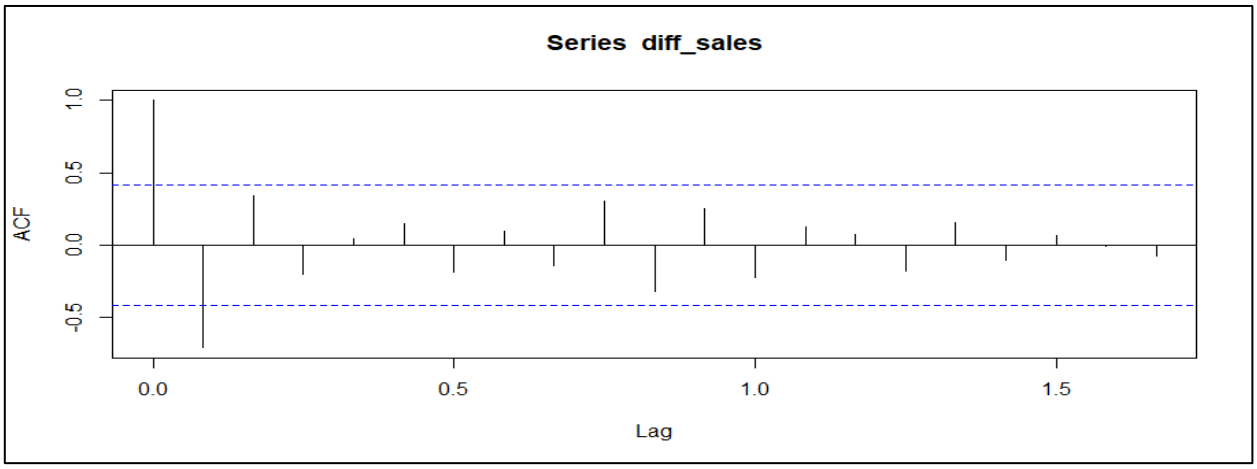

Fig.6 : Acf correlogram for lag values

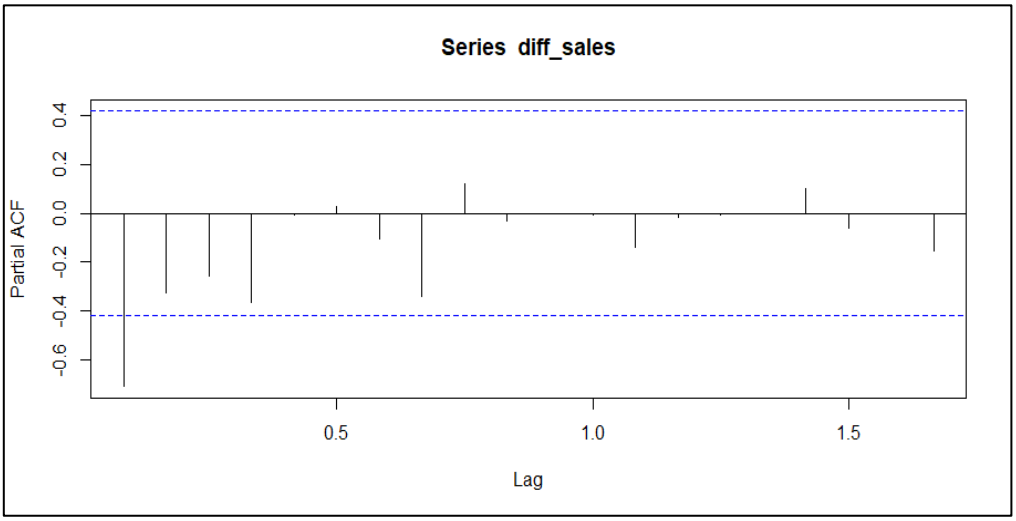

Fig.7 : Pacf correlogram for lag values

As we can see from Fig. 7, the plot for partial correlogram has mostly negative lag values. And the function pacf() has been used with lag order of 20.

The principle of parsimony has used to obtain the best model [4]. After trying a few models, we concluded that the best model for arima is the(3,0,0). The next step is to forecast values using an ARIMA model. To do the same, we used the forecast library in R to plot our results for the model.

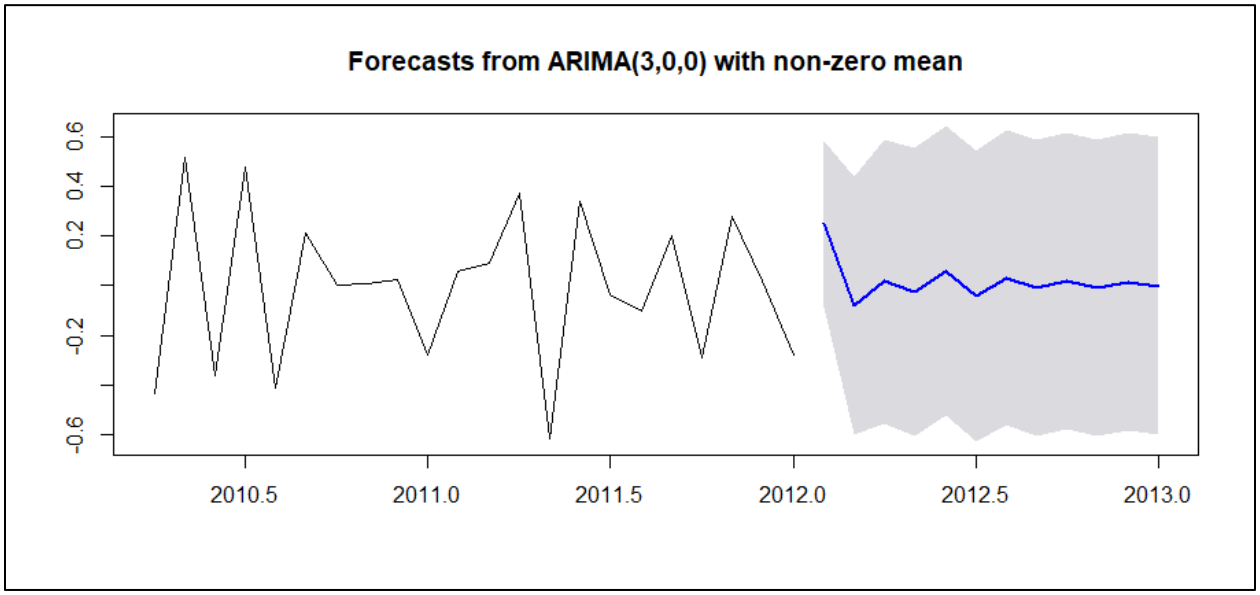

Fig.8 ARIMA model forecasting values for the year 2013

From Fig. 8 we can see that the model has predicted values for the year 2013. The confidence interval used is 95 %. The predicted values for Weekly Sales are at a high in the beginning of 2012 and then follow a downward trend and later are somewhat constant over time for the rest of the year.

Conclusion

We have applied two techniques of Intermediate Analytics in this project namely LASSO model and Time Series Analysis to predict the time series for our model. The methods of statistics have been used in R to work with the data and conclude meaningful derivations from it. Time Series Analysis is the most important when it comes to predicting a variable over time. The topic is vast and only a part of it has been used in this project. The learnings after performing this project have been immense.

References

[1] LASSO model: https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net

[2] Time Series Analysis : Time Series Analysis by James D. Hamilton

[3] ADF test : https://www.statisticshowto.datasciencecentral.com/adf-augmented-dickey-fuller-test/

[4] ARIMA: https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html#decomposing-time-series