EIN 6934 - ENGINEERING ANALYTICS 1

Store Sales - Time Series Forecasting

Final Project - Fall 2021

Group 07

POOJITHA BOLLENI

CHANDANI INDRARAJAN

ARAVIND MENON

PAVAN LOHITH RAVI

COURSE INSTRUCTOR

Dr.WALTER SILVA

Contents

Abstract

The sales of different products in the favorita stores in Ecuador plays an important role in the life of retailers. Among which many external contributing factors affect the sales such as holiday events and oil prices etc. This article predicts the sales for the products for a given time period using different Machine Learning Models.

Using the data provided in the " Store sales- Time Series forecasting" of the Kaggle data competition, this article gives overall details of the sales of products and conducts data exploratory analysis on the data to analyze the relation between the variables that will be helpful in predicting the sales of the products. Various techniques such as Linear regressions in R and python, Random forest, Ridge regression, Lasso regression, Neural network and XG boost were built to predict the sales. By comparing the predictions we got from the models , we find out what are the best models to predict the sales.

1 Introduction

The total content of this report is mainly divided into 2 parts namely the prepossessing (which includes Exploratory Data Analysis and Feature Engineering) and the model building. R and Python are the major languages used for this report. We start off by taking the raw data and begin to do the prepossessing steps followed by building the models. The models are trained, fitted and finally the total process end with predictions.

2 Data-set Description

The dataset for this project has been taken from the Kaggle competition Store Sales- Time Series Forecasting. The data has already been divided into train and test sets. Apart from the train and test data sets there are few other sheets which include oil, Holiday events, sample submissions and stores data.

Let us now look at the description of the data set :

Train CSV: The training data includes the following features 1. store nbr indicates store at which the item has been purchased 2. Family indicates the type of product sold 3. Sales indicates the total sales of the product at that particular store 4. On promotion indicates the items that belonged to a particular family that are being promoted.

Test.CSV has the same features that the Test. CSV has.

Stores.CSV City indicates the city in which the store is located State indicates the state in which the store is located Type indicates the type of store Cluster indicates the number of similar stores in that area.

Oil.CSV Date Dcoilwit indicates the price of the oil both during the morning and night

Sample Submission. CSV It includes the id and the predicted sales.

3 Motivation

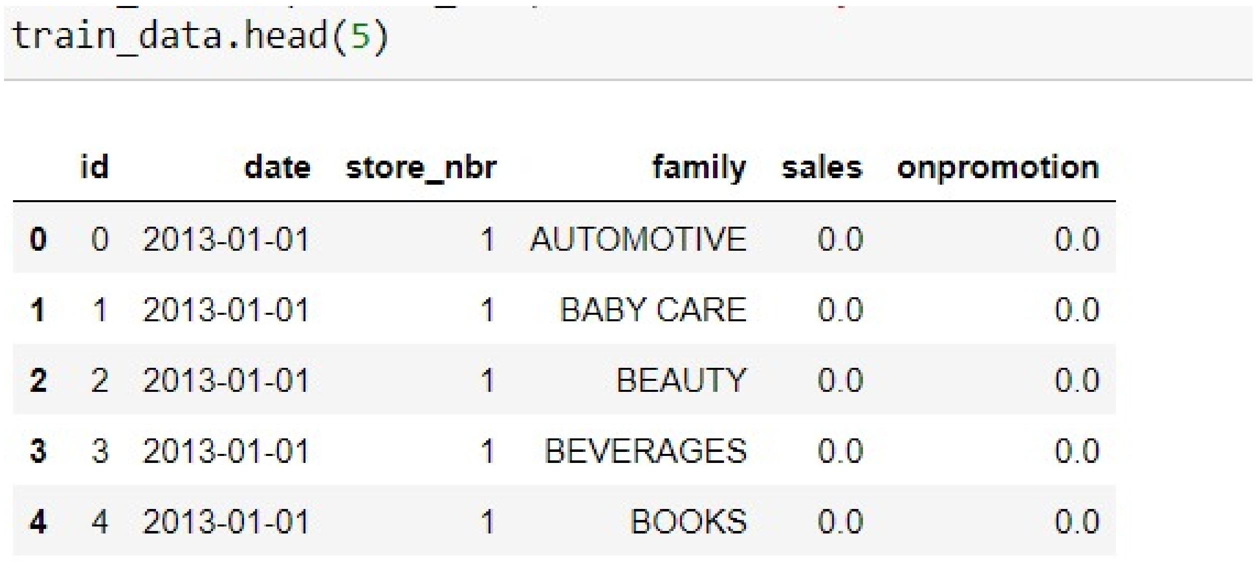
In many stores all-around the world the items are stocked up for long intervals as the retailers do not know how much sales happen on a particular day. There are many disadvantages of this stickup which include wastage of product due to expiry date. If the retailer stocks up little stock than the required amount then there is a chance of customer going back dissatisfied. Predicting the sales properly would help the retailer's stock up correct amount of products which will solve the problem of wastage or the customer going back dissatisfied

4 Prepossessing

4.1 Exploratory data analysis

The sales of different products in the favorita stores in Ecuador plays an important role in the life of retailers. Among which many external contributing factors affect the sales such as holiday events and oil prices etc. This article predicts the sales for the products for a given time period using different Machine Learning Models. Using the data provided in the " Store sales- Time Series forecasting" of the Kaggle data competition, this article gives overall details of the sales of products and conducts data exploratory analysis on the data to analyze the relation between the variables that will be helpful in predicting the sales of the products. Various techniques such as Linear regressions in R and python, Random forest, Ridge regression, Lasso regression , Neural network and XG boost were built to predict the sales . By comparing the predictions we got from the models , we find out what are the best models to predict the sales.

Exploratory data analysis (EDA) is a process where we explore the trends of data by understanding the structure of the data with the help of visual methods such as graphs, tables and chart.Before the Machine Learning models are built the data should be studied thoroughly to understand the trends in the data and clean the data accordingly. Initially we read all the excel sheets required for the project into R or Python. We use the df.head() to see the top few rows just to have an understanding of what is in the table



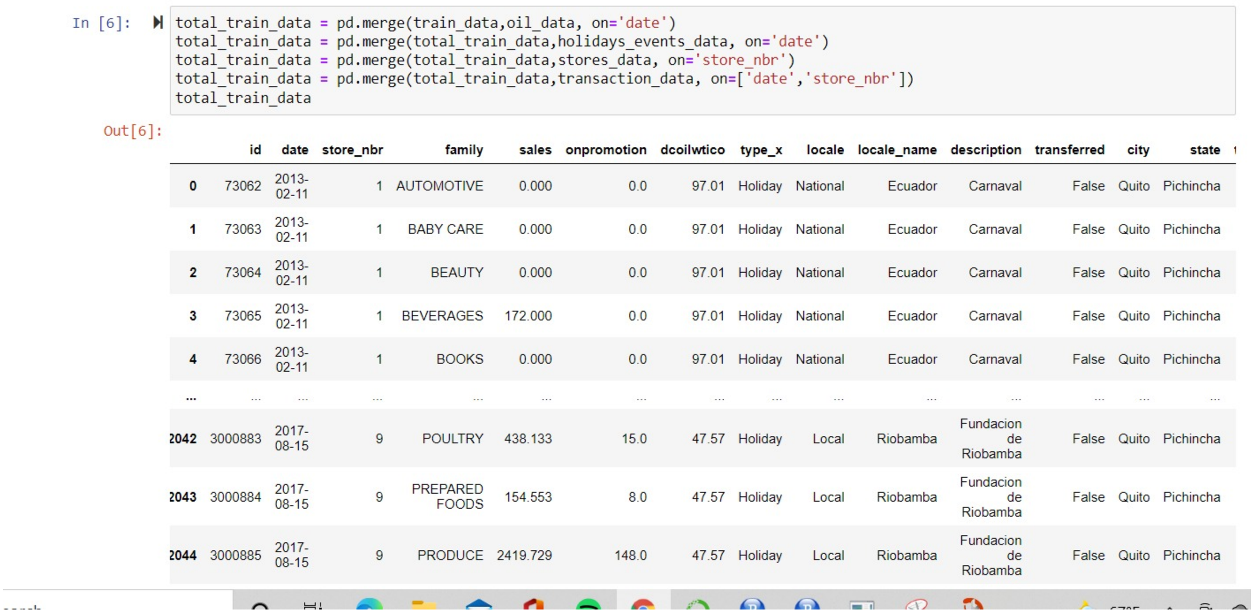
```
train_data.head(5)
```

	id	date	store_nbr	family	sales	onpromotion
0	0	2013-01-01	1	AUTOMOTIVE	0.0	0.0
1	1	2013-01-01	1	BABY CARE	0.0	0.0
2	2	2013-01-01	1	BEAUTY	0.0	0.0
3	3	2013-01-01	1	BEVERAGES	0.0	0.0
4	4	2013-01-01	1	BOOKS	0.0	0.0

Figure 1: Showing top elements of data frame

We usually do this to all the excel sheets which we read into the data frames.

As all the data is in different excel sheets, it is sometimes difficult to find relations between the variables. So we merge all the data frames into one big data frame.



In [6]:
```
total_train_data = pd.merge(train_data,oil_data, on='date')
total_train_data = pd.merge(total_train_data,holidays_events_data, on='date')
total_train_data = pd.merge(total_train_data,stores_data, on='store_nbr')
total_train_data = pd.merge(total_train_data,transaction_data, on=['date','store_nbr'])
total_train_data
```

Out[6]:

	id	date	store_nbr	family	sales	onpromotion	dcoilwtico	type_x	locale	locale_name	description	transferred	city	state
0	73062	2013-02-11	1	AUTOMOTIVE	0.000	0.0	97.01	Holiday	National	Ecuador	Carnaval	False	Quito	Pichincha
1	73063	2013-02-11	1	BABY CARE	0.000	0.0	97.01	Holiday	National	Ecuador	Carnaval	False	Quito	Pichincha
2	73064	2013-02-11	1	BEAUTY	0.000	0.0	97.01	Holiday	National	Ecuador	Carnaval	False	Quito	Pichincha
3	73065	2013-02-11	1	BEVERAGES	172.000	0.0	97.01	Holiday	National	Ecuador	Carnaval	False	Quito	Pichincha
4	73066	2013-02-11	1	BOOKS	0.000	0.0	97.01	Holiday	National	Ecuador	Carnaval	False	Quito	Pichincha
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2042	3000883	2017-08-15	9	POULTRY	438.133	15.0	47.57	Holiday	Local	Riobamba	Fundacion de Riobamba	False	Quito	Pichincha
2043	3000884	2017-08-15	9	PREPARED FOODS	154.553	8.0	47.57	Holiday	Local	Riobamba	Fundacion de Riobamba	False	Quito	Pichincha
2044	3000885	2017-08-15	9	PRODUCE	2419.729	148.0	47.57	Holiday	Local	Riobamba	Fundacion de Riobamba	False	Quito	Pichincha

Figure 2: Merging Data Frames

We build the correlation matrix to see the correlation between the variables.

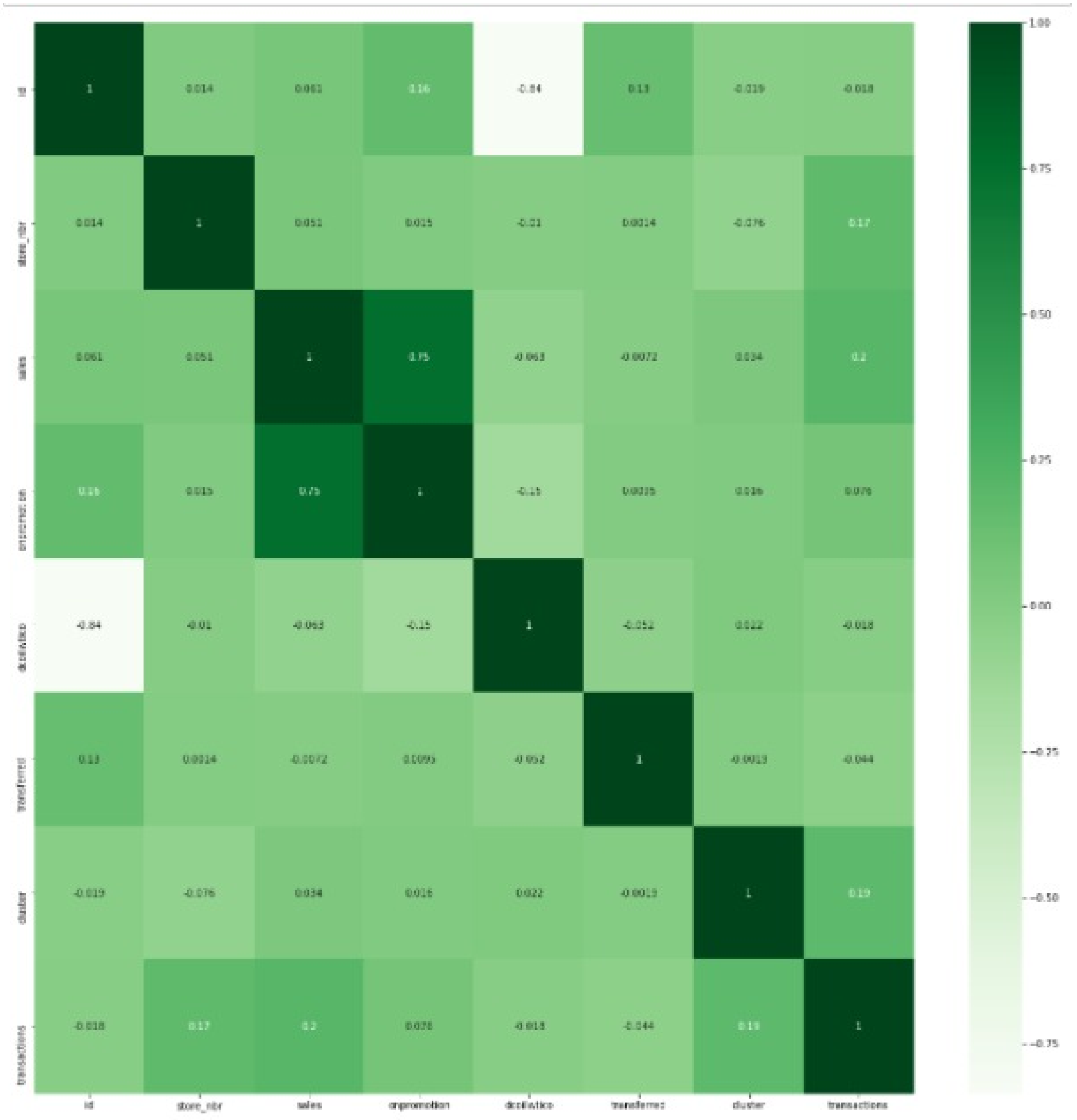


Figure 3: Correlation matrix

We build some visualizations to see the different trends in data.

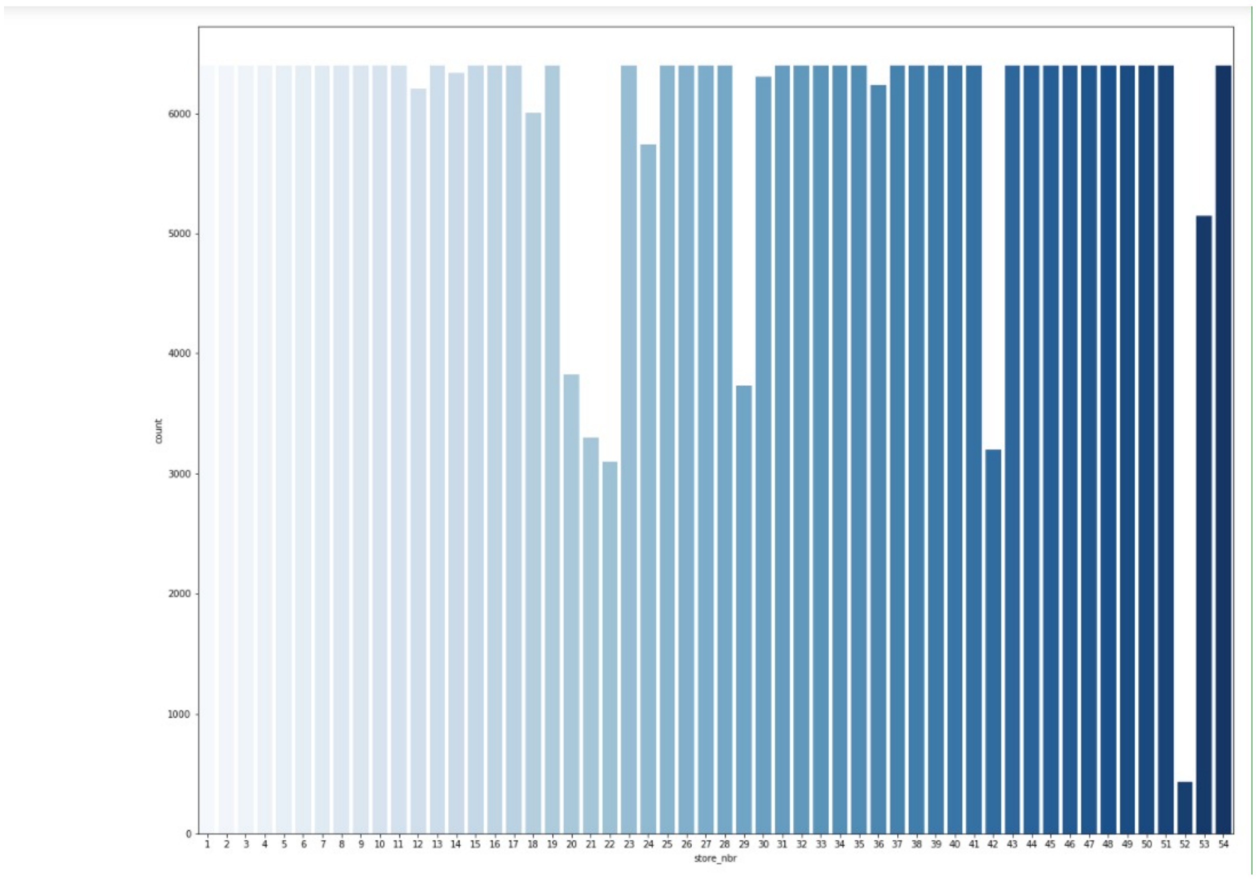


Figure 4: store vs sales

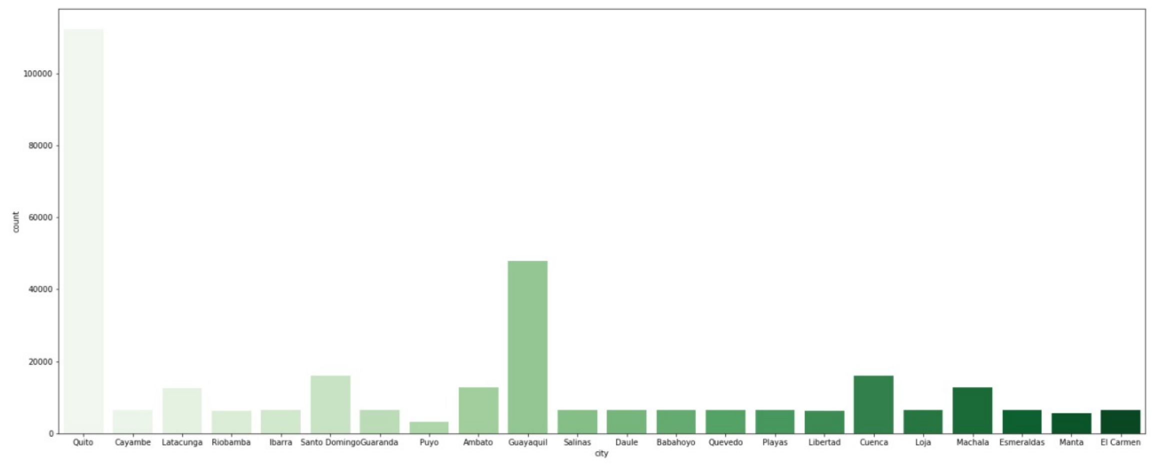


Figure 5: city vs sales

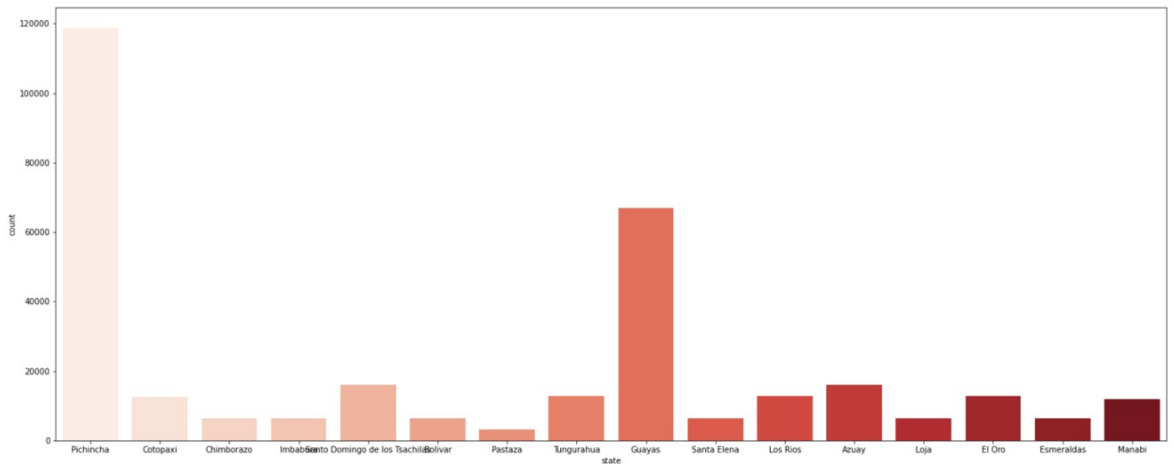


Figure 6: state vs sales

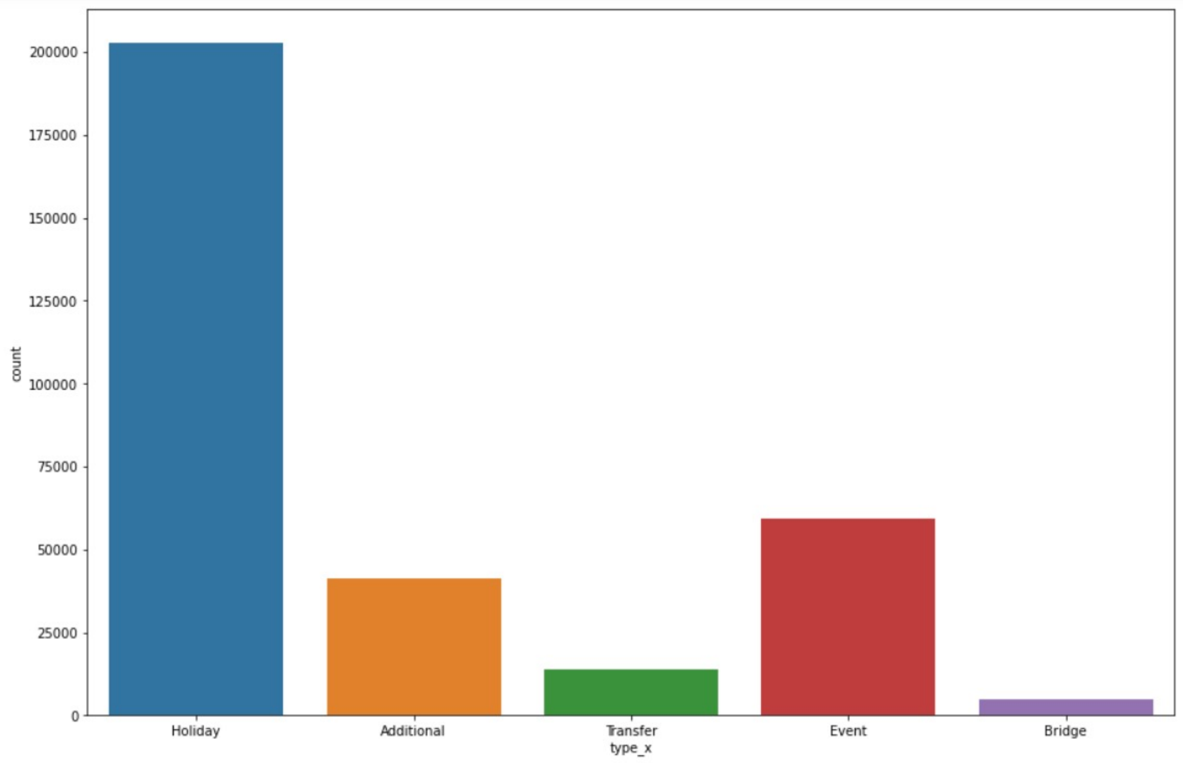
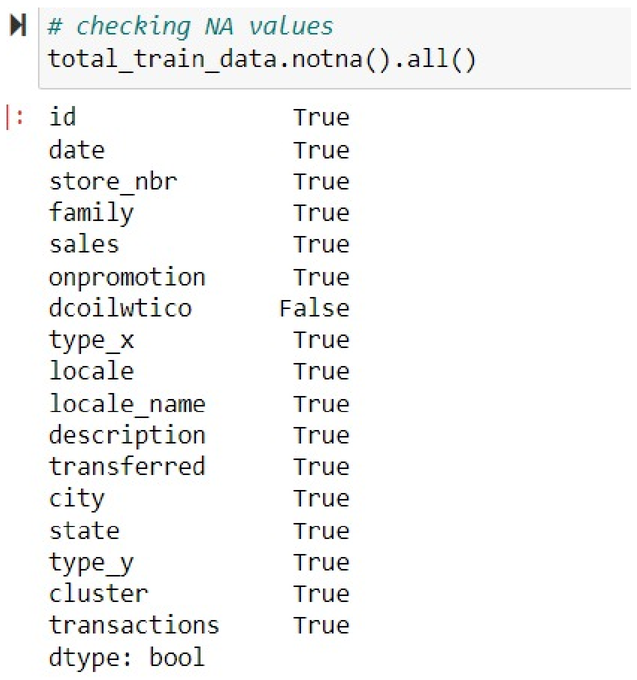


Figure 7: type of event vs sales

We have checked if there are NA values in the data set and we have dropped them.


```
# checking NA values
total_train_data.notna().all()
```

```
id               True
date             True
store_nbr        True
family           True
sales            True
onpromotion      True
dcoilwtico      False
type_x           True
locale           True
locale_name      True
description      True
transferred      True
city             True
state            True
type_y           True
cluster          True
transactions     True
dtype: bool
```

Figure 8: Finding NA values

4.2 Feature Engineering

In general, all the Machine learning algorithms use some input data to obtain some outputs.The input data has some variables called features. For the features to be used in modelling the Machine learning algorithms, they have to be in a particular format. The main goals f feature engineering are to prepare a proper input data set that would be useful to run the Machine learning algorithms and to improve the performance of the models. Date formatting is one of the most important feature engineering technique. We have converted the date format into month, year and date of the week as the major columns.

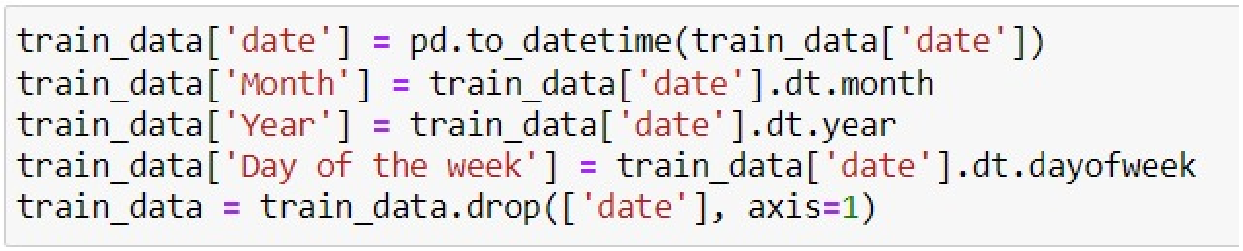

```
train_data['date'] = pd.to_datetime(train_data['date'])
train_data['Month'] = train_data['date'].dt.month
train_data['Year'] = train_data['date'].dt.year
train_data['Day of the week'] = train_data['date'].dt.dayofweek
train_data = train_data.drop(['date'], axis=1)
```

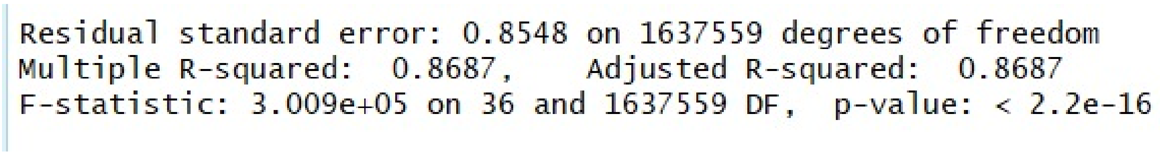
Figure 9: Date and Time formating

5 Training the Model

After we finished the Exploratory Data Analysis and feature engineering, we were all set to make hands dirty on building the Machine Learning Models. As mentioned in the data set, we already had the train data and test data. In real world scenarios, the test data is the real time data. We use the data we already have to fit and predict the model to check the proper working of the model. We have implemented the same by giving a split ratio.

5.1 Linear Regression Model in R studio

The first target was to build a basic linear model on the data. We built the first linear model in R by actually using the train data and test data by fitting on the train data and then testing on the test data. We were good to move ahead with the first model. We used all the dependent variables in the train data which are on promotion, store nbr, family and date. We did the log transformations to achieve better results. The model was then used to predict the sales of the test data. With the results obtained by converting the results into the CSV file, we were confident that we would get a decent Kaggle score. We have then uploaded our predictions to the Kaggle. Predict() was always used to predict the sales on the test set. We have obtained a Kaggle Score of 1.983 which was not our expected outcome. But we were satisfied as we have not taken complete use of all the excel sheets given in the data. This score could improve drastically if we can merge all the excel sheets and make proper use of all the data given



```
Residual standard error: 0.8548 on 1637559 degrees of freedom
Multiple R-squared:  0.8687,    Adjusted R-squared:  0.8687
F-statistic: 3.009e+05 on 36 and 1637559 DF,  p-value: < 2.2e-16
```

Figure 10: Linear in R

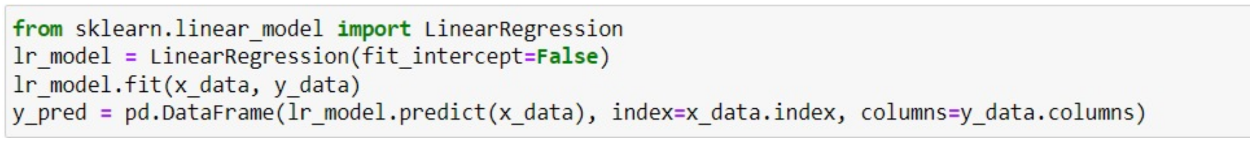
5.2 Random Forest Model in R Studio

We have learned in our curriculum that random forest is better in reducing the RMSE as compared to the linear model. With the capabilities that the Random Forest has over Linear regression, we were expecting the model to perform well with low RMSE and a good Kaggle score. To our surprise, the Random Forest model led to the crash of R studio due to capability issues and was throwing errors related to memory issues.

With the issues we were facing for Random Forest in R, we felt that there would be greater issues in the future models especially ARIMA and Neural Networks. We have then decided to move on to python.

5.3 Linear Regression Model in Python

We have implemented the exploratory data analysis and we were ready to start building our models in python which is much more efficient than R. We have created a function which is a Deterministic process and have given certain fixed parameters in the process (code was taken as a reference from exercises in Kaggle course of Time series). We have trained the data set and tested by giving values to the parameters defined in the deterministic process. We got the CSV file and we have uploaded the CSV into the Kaggle submission. We got a score of 0.5010 which was way better than the score that we have achieved earlier.

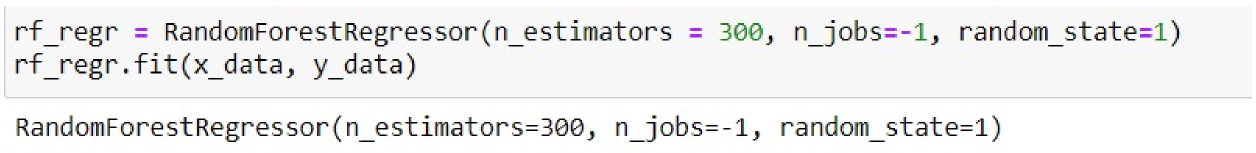


```
from sklearn.linear_model import LinearRegression
lr_model = LinearRegression(fit_intercept=False)
lr_model.fit(x_data, y_data)
y_pred = pd.DataFrame(lr_model.predict(x_data), index=x_data.index, columns=y_data.columns)
```

Figure 11: Linear Regression in Python

5.4 Random Forest Model in Python

We already know that the Random Forest works better than the Linear Regression and reduces the RMSE value. We built a Random Forest model with specific parameters such as n estimators, number of jobs and random state. We have tried to tweak these parameters just to check how our model reacts to different parameters. Each time we tweak the parameters we were getting different results and different run times.



```
rf_regr = RandomForestRegressor(n_estimators = 300, n_jobs=-1, random_state=1)
rf_regr.fit(x_data, y_data)
```

```
RandomForestRegressor(n_estimators=300, n_jobs=-1, random_state=1)
```

Figure 12: Random Forest in Python

5.5 Ridge Model

We have started to research on the internet and Kaggle discussion boards to find out more ways to get a better accuracy and reduce the RMSE value. We found out that Ridge regression is one of the techniques that could be helpful in getting a better score in the competition. From the correlation matrix, we could observe that there is multi collinearity. From the extensive research from the internet, we found that Ridge Regression could help as it works when there is multi collinearity. Moreover, it can penalize the large coefficients by almost shrinking them close to zero. We have used Robust Scaler in this process to reduce the effect of outliers and it scales the data as per the inter quantile range. Various numbers of hyperparameters were used such as max iterations, alpha etc. The values of the hyper parameters have been changed to check the results for different hyperparameters. We have created a csv file and have uploaded it into Kaggle expecting a better score. For our surprise we got a Kaggle score 0.42414 which was way more than expected.

5.6 Lasso Regression

This model is said to have a similar kind of behavior to that of Ridge regression due to similarity in most properties. We have used the robust scaler for Lasso model as well. This scaler would be so helpful in removing the outliers. We have calculated the Mean Absolute Error and the Root Mean Square Error for the Lasso Model. We have observed that the Mean Absolute error for this model is more than that of Ridge model. It is quite evident that the Kaggle upload would not give us a score that is better than of the Ridge model.

5.7 ARIMA Model

The research kept continuing every week and Google has been of a huge help. The research always revealed that for Time Series forecasting models, ARIMA works best. But it had its own disadvantages as well. There had to be a lot of feature engineering to be done in order to get the ARIMA to its best and efficient way. ARIMA does not work well for the too long seasonal cycles and for too short seasons as well. We have taken 12 months seasonal which led to the bad running of ARIMA model. Moreover, the years mentioned in the data set is also too less. The ARIMA model did not run as per the expectations. There had to be much more analysis to be held before diving deep into the ARIMA modelling.

5.8 Recurring Neural Networking Model

In most of the Machine learning scenarios, neural networks (be it simple or complex depending on the scenario) have always been the ones that have given the best prediction due to their innate and in-built properties. For the Neural Nets, Min Max scalers were the ones that have been used frequently due to their numerous advantages. We built Neural Networks with 4 hidden layers and an output layer. Each of them had different hyperparameters. We tried tweaking those hyper parameters to just check how the RNN was working for different values of the hyperparameters. But to our sudden surprise, the RNN did not perform as expected. We have used many hyper parameters such as batch size, epochs, loss, optimizer, activation, dropout and many other.

5.9 XGBoost Model

We noticed while rechecking the codes that we have overfitted the data while running the ridge model. The split ratio was read wrongly which led to getting a high score on Kaggle. The Ridge model was a good model for this data, but overfitting is not good at any case. The research to solve this problem started to take part. We have found out that XG Boost model has an inbuilt L1(Lasso Regression) and L2(Ridge Regression) regularization which prevents the model from overfitting. XG Boost is also called regularized form of GBM. We have run the XG Boost model on the complete data. But we were getting an error that stated bad memory allocation. We understood that the model was unable to run on the entire data. So we have used 10

6 Conclusion

Numerous models were run during this project. Every time a model was being run; we always were expecting a better accuracy compared to the previous attempts. We were successful in most of the attempts and sometimes we were not. We have not only learned how to build Machine Learning models but also to do various preprocessing steps including Visualizing the data, cleaning it, tackling the outliers, imputing the missing values, merging the tables. After the preprocessing and feature engineering, we have built various Machine Learning models mentioned to gradually decrease the error and increase the accuracy. Different approaches were taken to model the data and preprocess whenever required. This project has given us an opportunity to encounter real life circumstances where we can use the knowledge we have gained from the coursework as well as few concepts that we have learned from the research conducted from the internet.

7 Reference

https://www.kaggle.com/learn/feature-engineering

https://www.kaggle.com/learn/intermediate-machine-learning

https://www.kaggle.com/learn/time-series

https://towardsdatascience.com/random-forest-in-python

https://machinelearningmastery.com/ridge-regression-with-python/

https://scikit-learn.org/

https://www.datacamp.com/community/tutorials/xgboost-in-python

https://www.geeksforgeeks.org

https://www.freecodecamp.org/

https://stackoverflow.com/