Capstone Project
Rossmann Sales Prediction

- Shyam Sundar K

CONTENTS

Problem Description

Rossmann operates over 3,000 drug stores in 7 European countries. Currently, Rossmann store managers are tasked with predicting their daily sales for up to six weeks in advance. Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality. With thousands of individual managers predicting sales based on their unique circumstances, the accuracy of results can be quite varied.

You are provided with historical sales data for 1,115 Rossmann stores. The task is to forecast the "Sales" column for the test set. Note that some stores in the dataset were temporarily closed for refurbishment.

Data fields

- **Id** - an Id that represents a (Store, Date) duple within the test set
- **Store** - a unique Id for each store
- **Sales** - the turnover for any given day (this is what you are predicting)
- **Customers** - the number of customers on a given day
- **Open** - an indicator for whether the store was open: 0 = closed, 1 = open
- **StateHoliday** - indicates a state holiday. Normally all stores, with few exceptions, are closed on state holidays. Note that all schools are closed on public holidays and weekends. a = public holiday, b = Easter holiday, c = Christmas, 0 = None
- **SchoolHoliday** - indicates if the (Store, Date) was affected by the closure of public schools
- **StoreType** - differentiates between 4 different store models: a, b, c, d
- **Assortment** - describes an assortment level: a = basic, b = extra, c = extended
- **CompetitionDistance** - distance in meters to the nearest competitor store
- **CompetitionOpenSince[Month/Year]** - gives the approximate year and month of the time the nearest competitor was opened
- **Promo** - indicates whether a store is running a promo on that day
- **Promo2** - Promo2 is a continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating
- **Promo2Since[Year/Week]** - describes the year and calendar week when the store started participating in Promo2
- **PromoInterval** - describes the consecutive intervals Promo2 is started, naming the months the promotion is started anew. E.g. "Feb,May,Aug,Nov" means each round starts in February, May, August, November of any given year for that store

Data Overview

Sales Prediction is key aspect for all retail stores to maintain the right amount of production and inventory so that there is no shortage of goods and inventory costs are also optimised.

In this project we have been given with the past data of major store chain Rossmann Drug stores and we have to build a model to predict the future sales.

There were 2 datasets one with the sales data and other with metadata of the 1115 unique stores.

Shape of datasets:

- Sales Data : 1017209 rows x 9 columns
- Store Data : 1115 rows x 10 columns

Data Cleaning

There are no null values in the sales data however there are few columns with null values in store data.

- We can see there are lot of null values in competition open since Year, and for available data we can more than 80% of the competitors open before 2013, and we are doing analysis for sales after 2013, we will drop the columns CompetitionOpenSinceYear and CompetitionOpenSinceMonth
- Filling the 3 missing values in competition distance by mean value of the column.
- We are filling null of the promo2 date after the latest day of the available data. It is done as it makes it easy for us to check if the promotion was active while merging the datasets.
- We are filling null of the promoInterval with None which will be used for one hot encoding once the datasets are merged.
- 'Promo2','Promo2SinceWeek','Promo2SinceYear' are merged as one promo2_start_date. Now the data set is ready to be merged.

	No of Null Values
Store	0
StoreType	0
Assortment	0
CompetitionDistance	3
CompetitionOpenSinceMonth	354
CompetitionOpenSinceYear	354
Promo2	0
Promo2SinceWeek	544
Promo2SinceYear	544
PromoInterval	544

Merging Datasets

As there are 2 data sets, it was merged on the common column Store Id. Left join was used to merge the dataset.

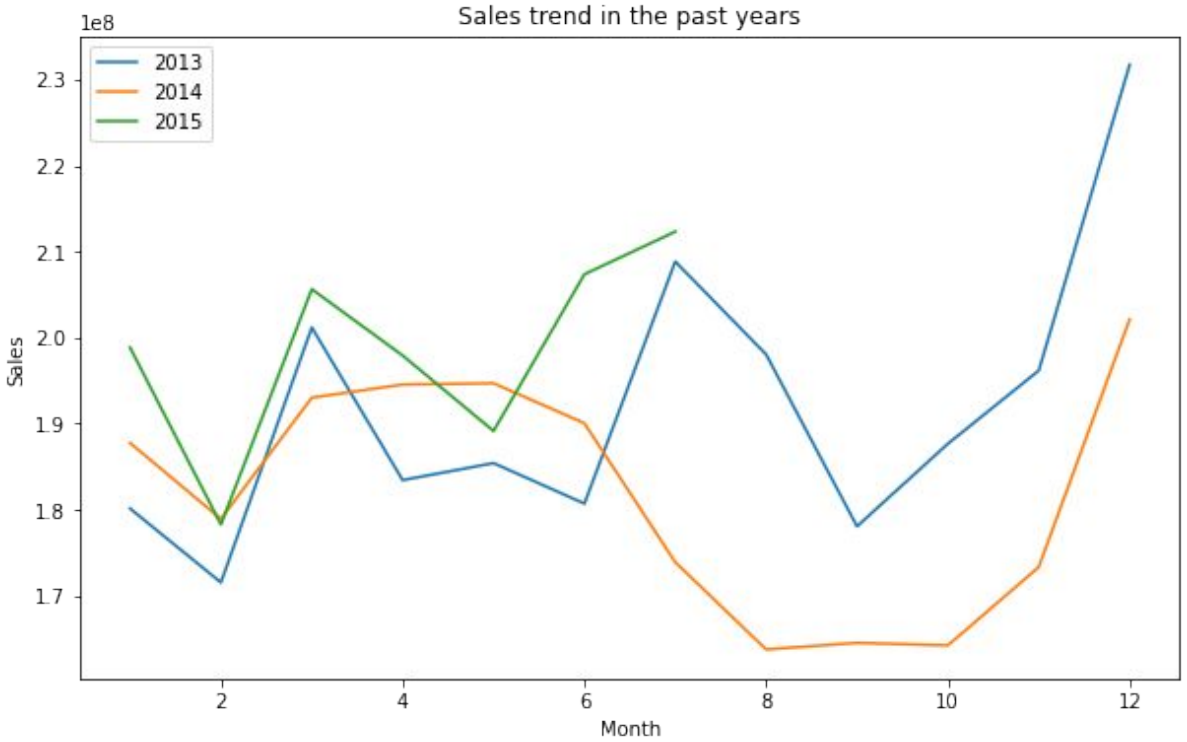
```
#merging the meta data with main dataset
df = pd.merge(df, df_meta, on='Store', how='left')
```

The Shape of our final data is now 1017209 rows x 14 columns.

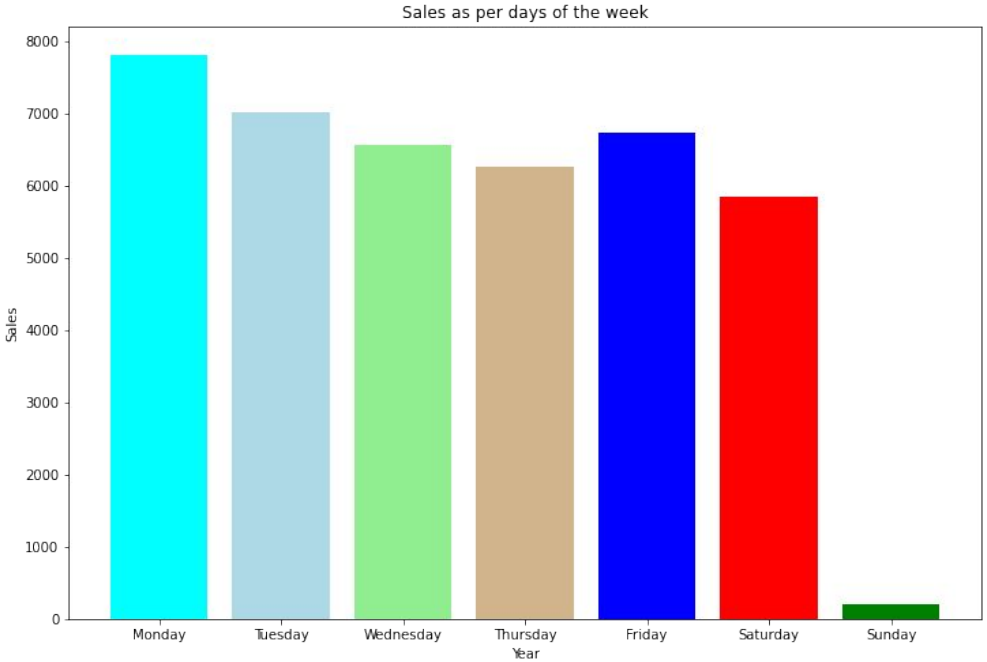
Exploratory Data Analysis

Sales trend in the past years

There is fall in sales in the month of Feb trent slowly increases and then again there is decreasing trend in the months August, September and then the sales touches the peak in December.



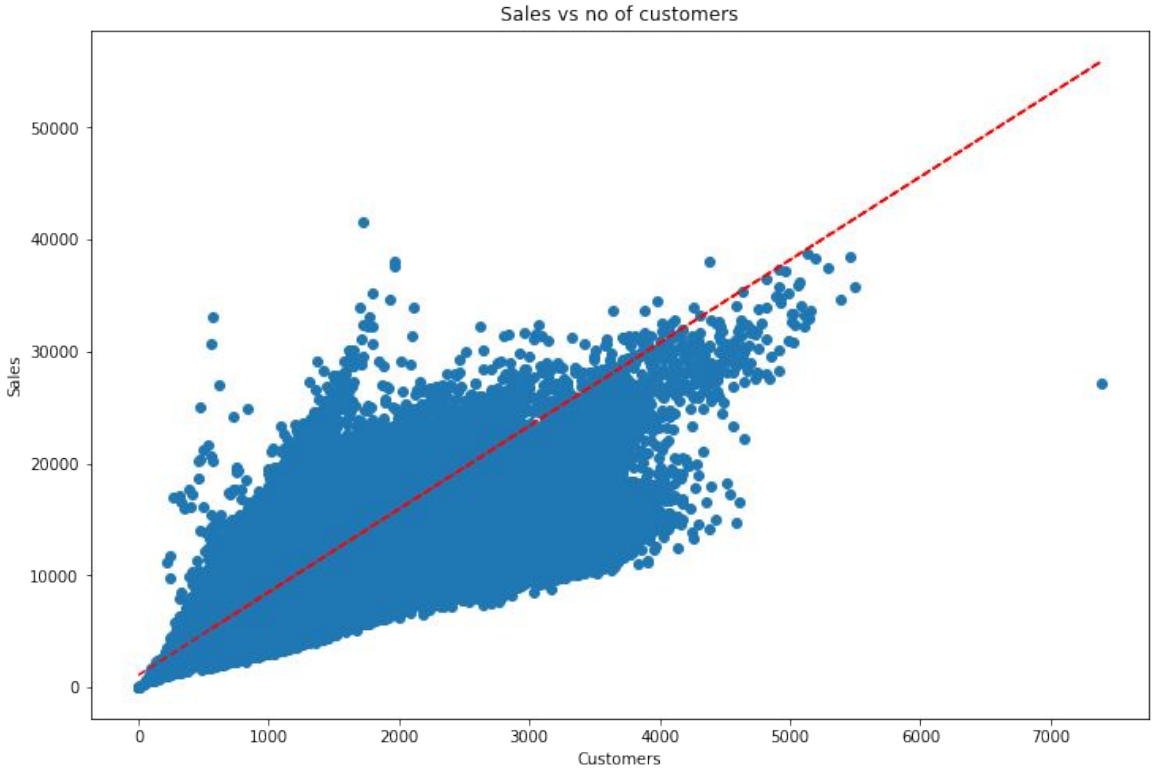
Sales as per days of the week



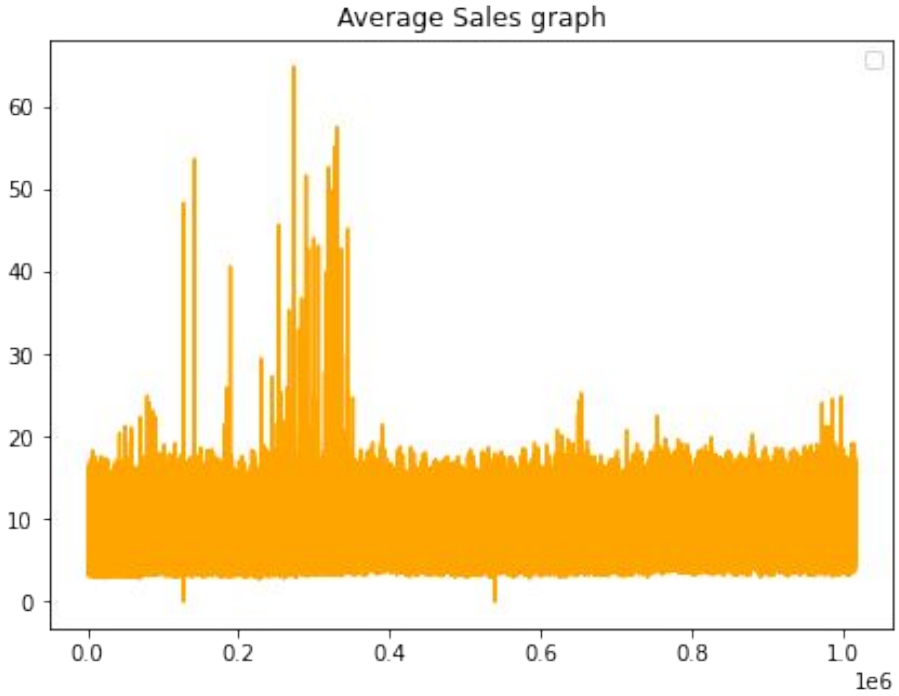
- The sales are highest on mondays and there is slight decline over the weekdays.
- The sales again increases a little towards the weeknd.
- The sales are almost NIL on sundays as most of the shops are closed on Sundays

Sales vs no of customers

It is noticed that sales is highly correlated to the no of customer visits. In the next graph, the average sales per customer is analysed.



Average Sales graph

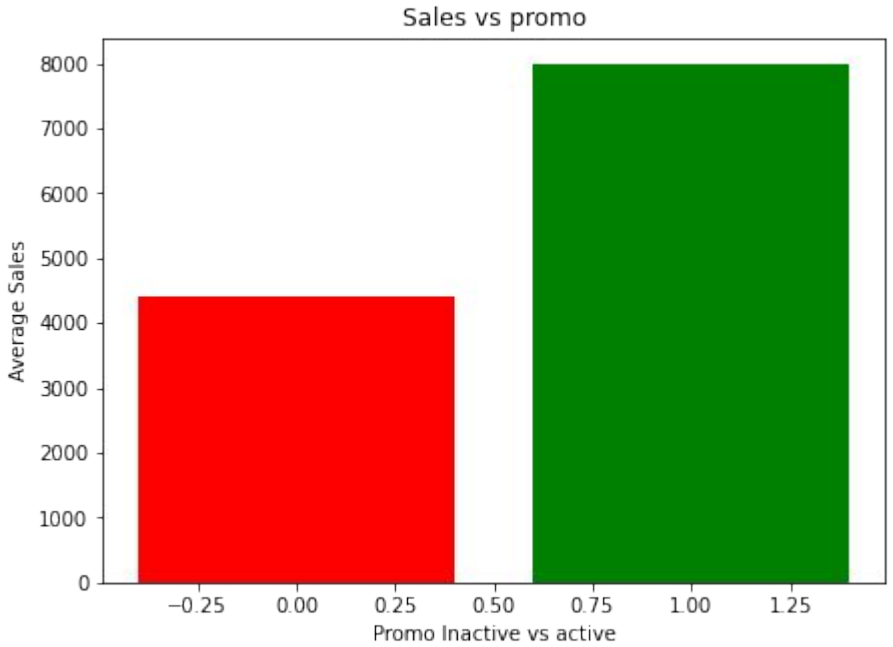


- Average sale per customer mostly lies between $4 and 2$0
- It is also noticed that there are some outlier values that goes upto $70
- The mean value is $9.

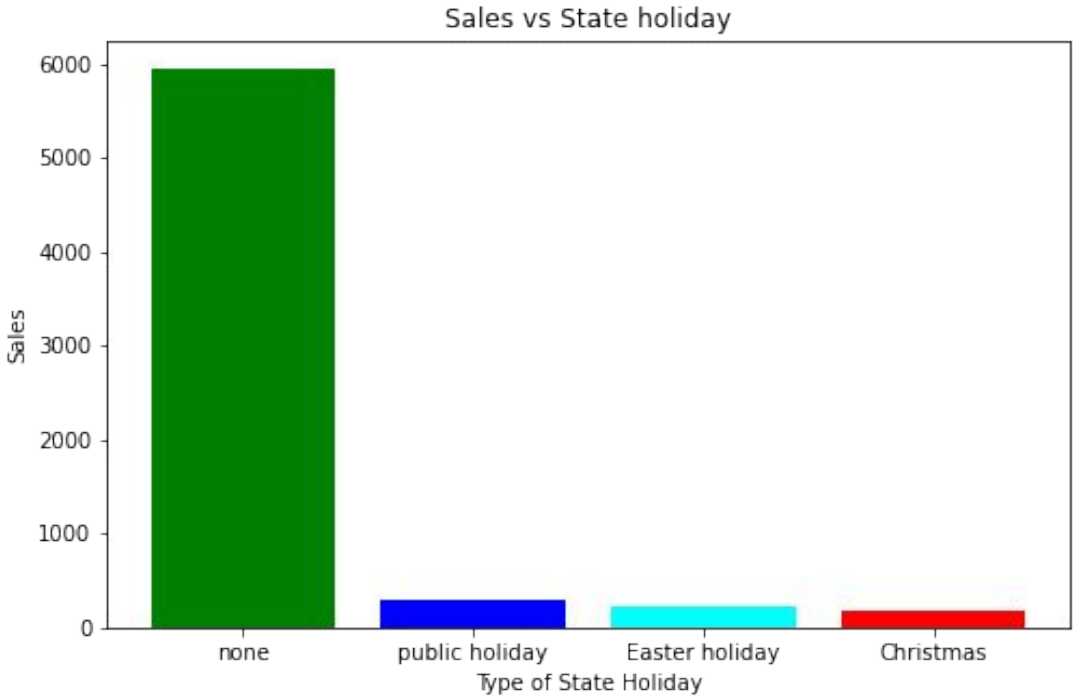
Average Sales = Total Sales/No of Customers

Sales vs promo

There is a significant increase in sales when the promo is active. Hence promotional period can be great way to attract more customers



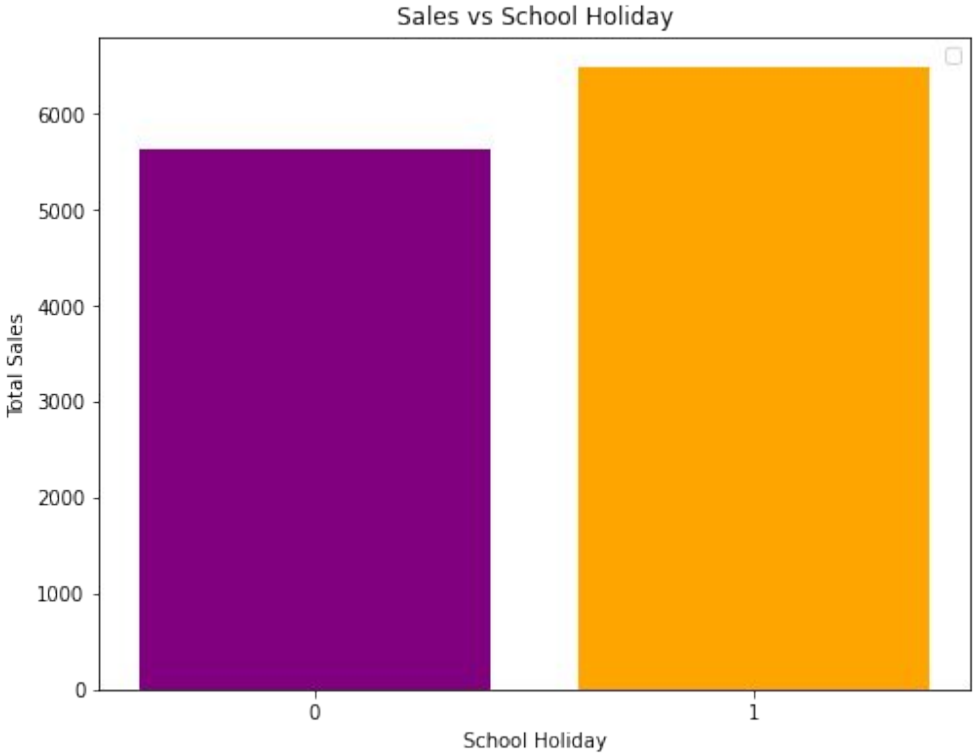
Sales During State Holidays



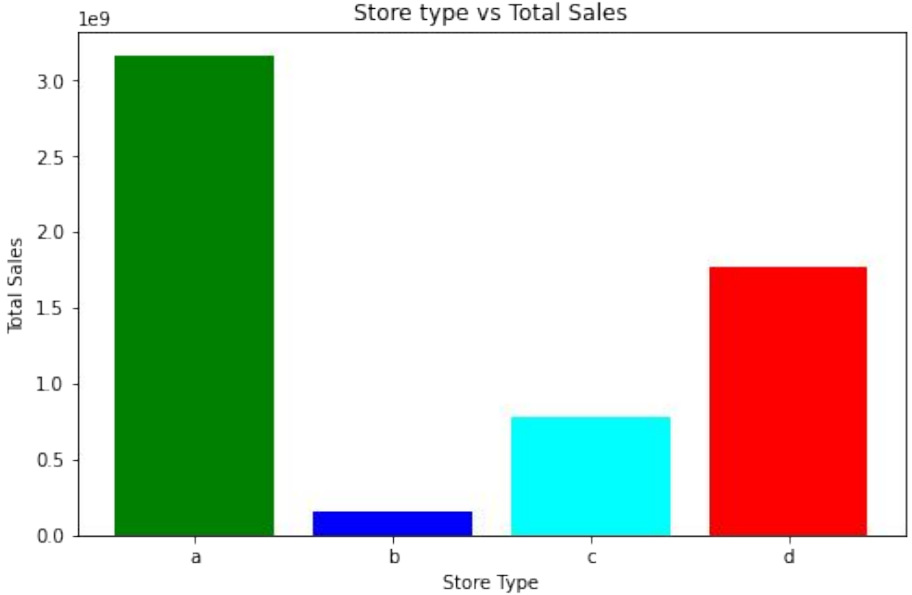
Sales reduces drastically during the State holidays as most the shops remains closed.

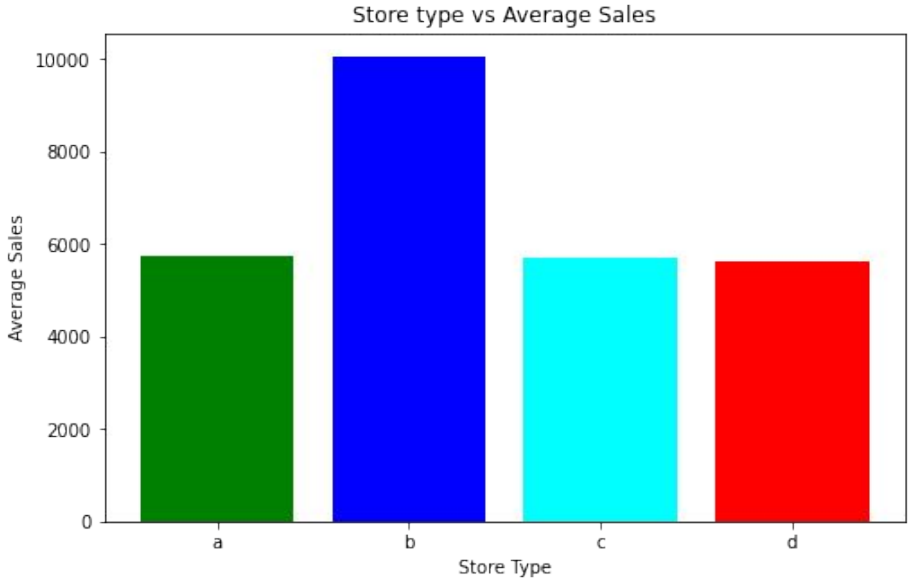
Sales During School Holidays

- There is a increase in average sales during the school holidays.
- The reason may be that there is higher requirement when the children are at home more than when there are in school days.



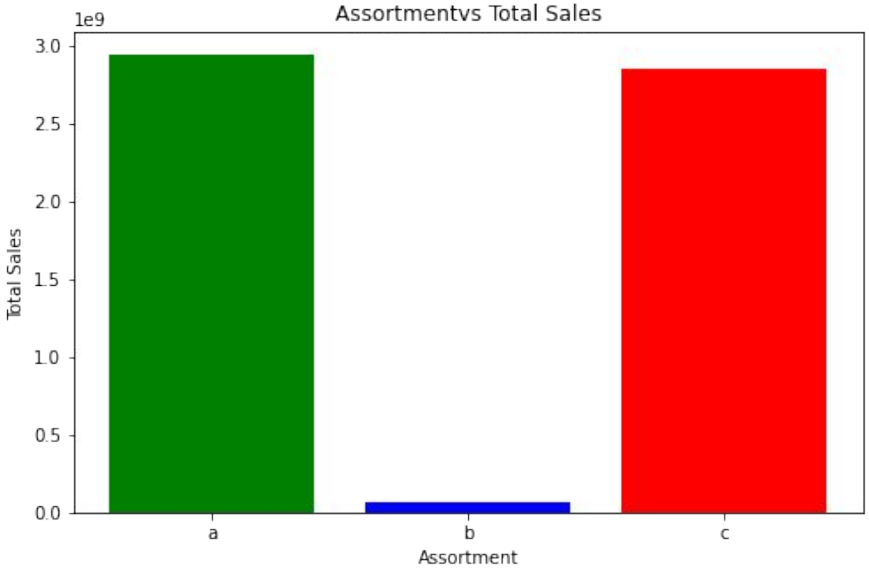
Sales (Total and Average) vs Store Types

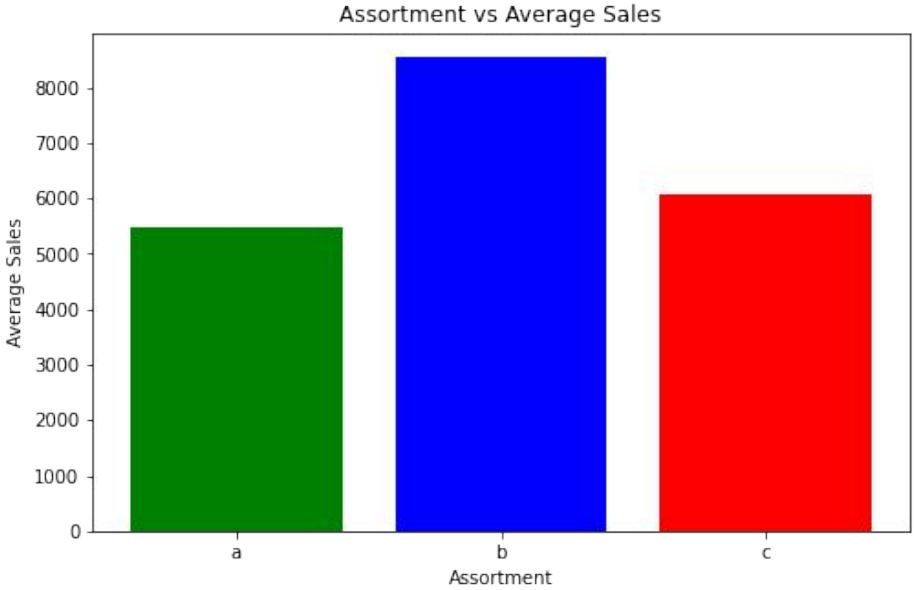




Store type A,C,D have approximately same average sales and Store type B has much higher average Sales, but the number of stores of Type B is very less.

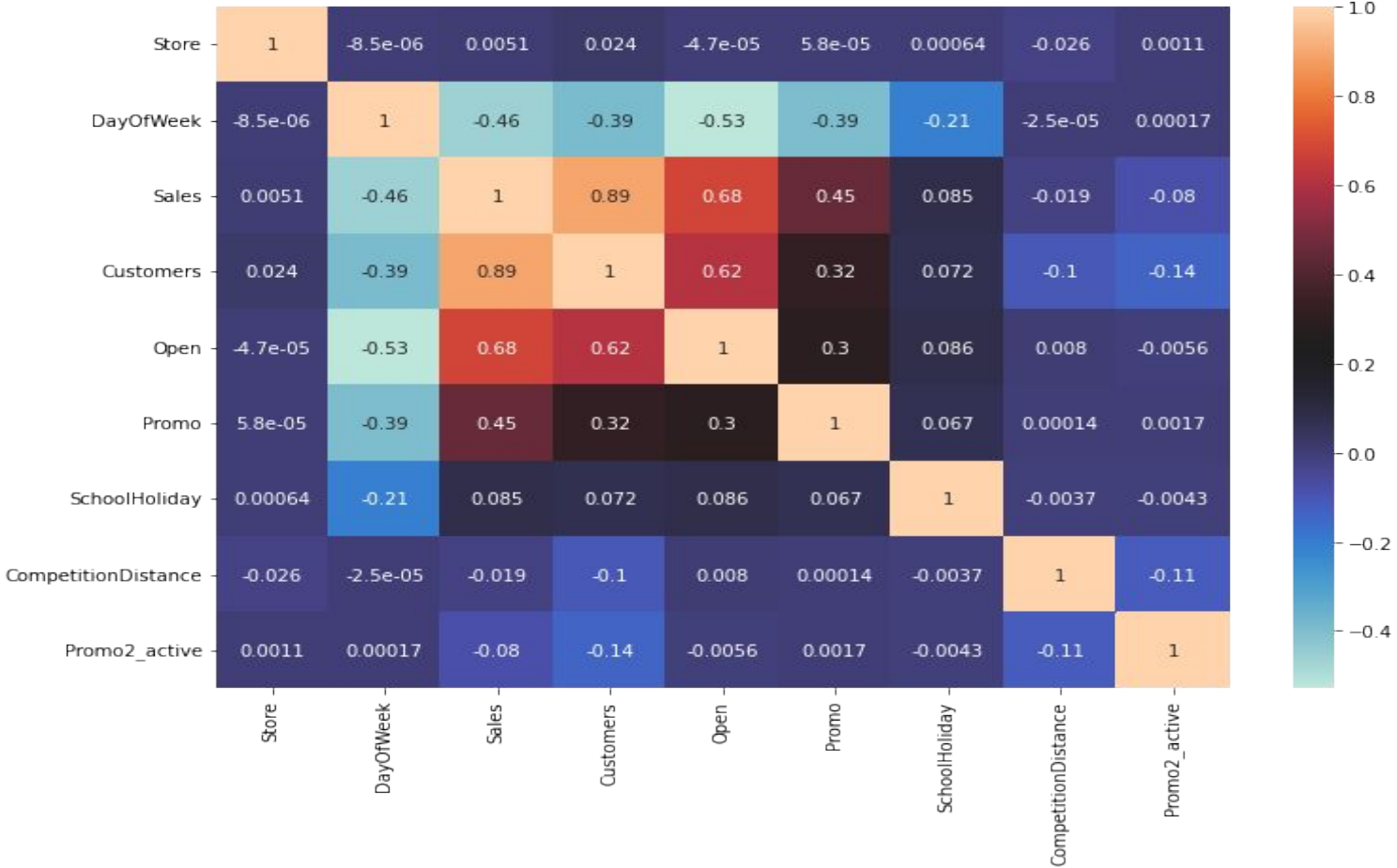
Assortment vs Sales(Total and Average)





Assortment type A and C have approximately same average sales and Assortment type B has higher average Sales, but the number of stores of Assortment Type B is very less.

Correlation

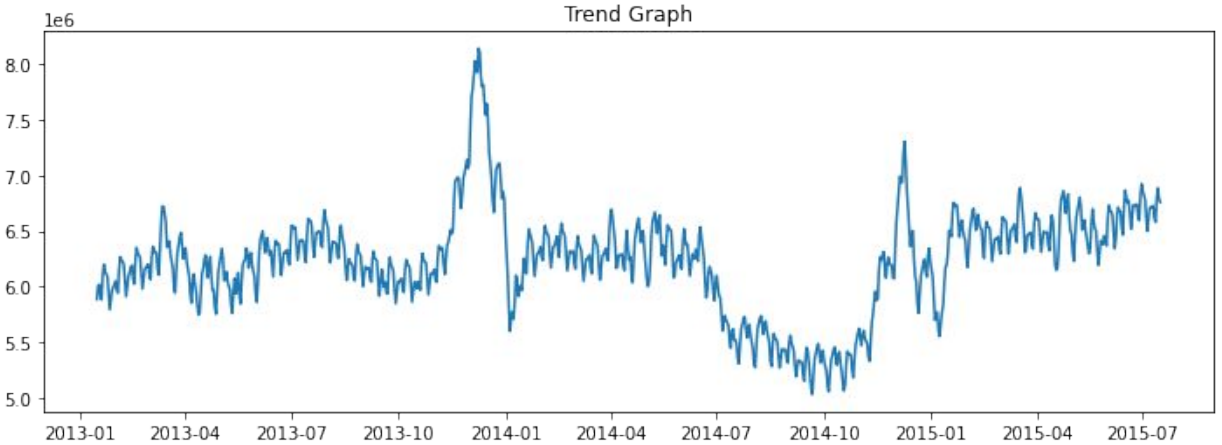


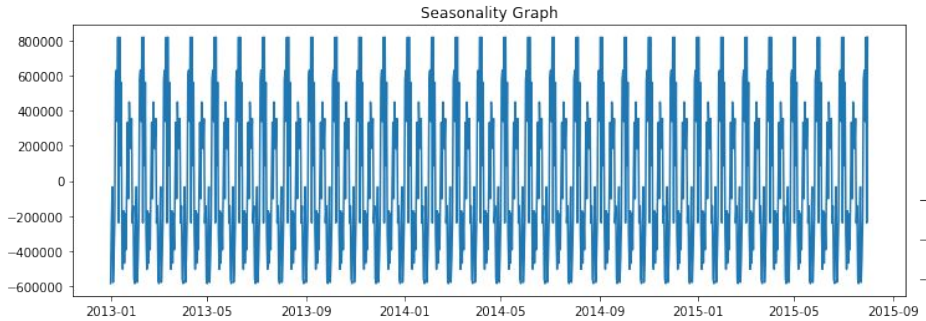
Insights From EDA

- There is fall in sales in the month of Feb and then again there is decreasing trend in the months August, September and then the sales touches the peak in December.
- The sales are highest on mondays and there is slight decline over the weekdays. The sales again increases a little towards the weeknd. The sales are almost NIL on sundays as most of the shops are closed on Sundays
- It is noticed that sales is highly correlated to the no of customer visits. Average sale per customer mostly lies between $4 and $20, however we can see some outlier values that goes upto $70 and the mean value is $9.
- There is a significant increase in sales when the promo is active.
- Sales reduces drastically during the State as most the shops remains closed.
- There is a increase in average sales during the school holidays.
- Store type A,C,D have approximately same average sales and Store type B has much higher average Sales, but the number of stores of Type B is very less.
- Assortment type A and C have approximately same average sales and Assortment type B has higher average Sales, but the number of stores of Assortment Type B is very less.

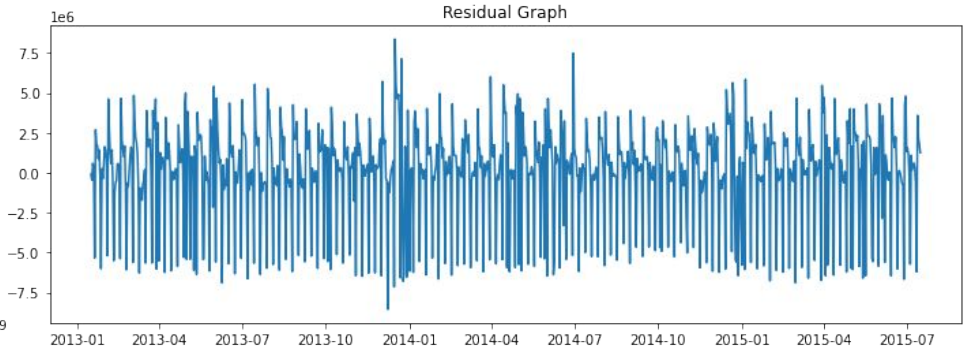
Time Series Analysis

Trend Graph





Seasonality Graph



Residual Graph

Feature Engineering

Removing records when shops were closed

We notice that Sales is 0 for all rows where Open is 0, so we can simply predict the sales to be 0 if the open column is 0 and drop the those rows for our modeling.

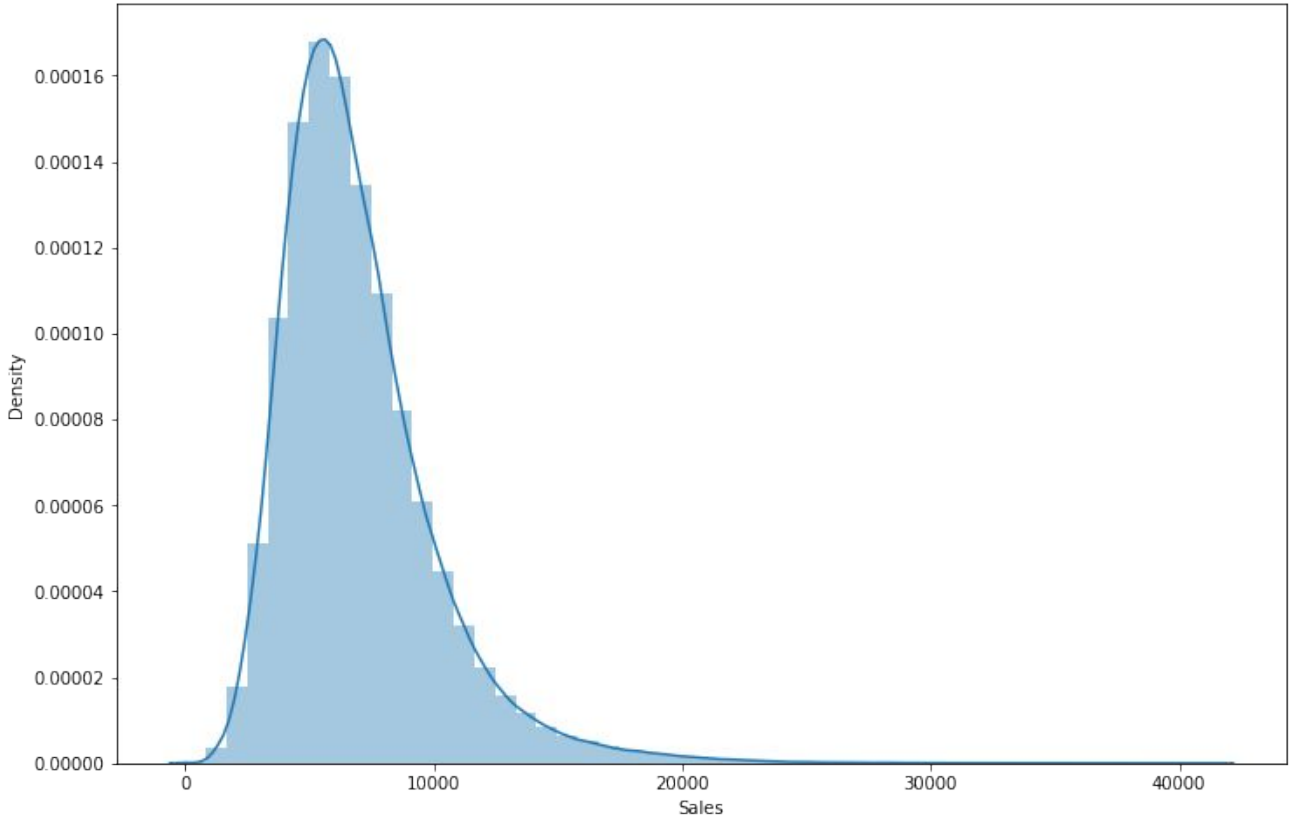
Code Snippet :

```
df[df.Open==0][df.Sales!=0].shape
```

df = Dataset

Output: `(0, 14)`

Dependent variable



df.Sales is our dependent Variable.

Encoding Independent Variables

- One Hot Encoding was applied on the variables StateHoliday, StoreType and PromoInterval

```
# One hot encoding
df = pd.get_dummies(df, columns= ["StateHoliday", "StoreType","PromoInterval"])
df.drop(columns= ['StateHoliday_0','StoreType_b', 'PromoInterval_None'],inplace = True)
```

- Label Encoding was applied on the variables

```
# Label Ecoding
encoders_nums = {"year":{"2013":1,"2014":2,"2015":3}, "Assortment":{"a":1,"b":2,"c":3} }
df = df.replace(encoders_nums)
```

Other Data Preparation

- Date column of DateType format was split into 3 columns, month, day and year.

```
# Extracting columns using date

df["month"] = df['Date'].dt.month
df["day"] = df['Date'].dt.day
df["year"] = df['Date'].map(lambda x: x.year).astype("string")
df = df.drop(['Date'],axis=1)
```

- MinMax Scaler was used to transform the independent variables

```
scaler = MinMaxScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test= scaler.transform(X_test)
```

Modelling

Linear Regression

- Linear Regression was applied and the score obtained was 0.8175 which is good score.

- Total 18 coefficients and a intercept were obtained.

```
array([[-2.46080535e+02,  4.03905094e+04,  1.14788028e+03,
         3.38682769e+01,  2.92077837e+02,  1.89305149e+03,
        -1.37014438e+02,  4.08039441e+02,  5.13679432e+01,
         3.77538350e+02, -1.45066475e+02,  2.29054917e+02,
         2.66813200e+03,  5.48728789e+03,  5.36646067e+03,
         6.63464195e+03,  3.82097876e+02,  4.65549319e+02,

         1.98334666e+02]])
```

Evaluation of Linear Regression Model

The following scores were obtained for the Linear Regression Model

MSE : 1759881.991782484
MAE : 952.3706773743044
RMSE : 1326.6054393761863

R2 : 0.8159008519416168
Adjusted R2 : 0.8158801371214821

#Base Model
The mean sales value is 6955.51
The standard deviation sales value is 3104.21

We can observe that the RMSE is on the higher side and R2 score is 0.815 which is a pretty decent score.

Lasso Regression

- Lasso Regression was applied and hyperparameter tuning for alpha value was performed along with Grid search Cross Validation.

```
### Cross validation
lasso = Lasso()
parameters = {'alpha': [1e-15,1e-10,1e-5,1e-3,1e-1,1,5,10,20,30,50,100]}
lasso_regressor = GridSearchCV(lasso, parameters, scoring='neg_mean_squared_error', cv=4)
lasso_regressor.fit(X_train, y_train)
```

- The best fit alpha value was found out to be : {'alpha': 0.001}

- Using {'alpha': 0.001} the negative mean squared error is: -1761680.6856497126

Evaluation of Lasso Regression Model

The following scores were obtained for the Lasso Regression Model with hyperparameter tuning and cross validation.

MSE : 1759878.4688100459
MAE : 952.3705660050666
RMSE : 1326.60411156081

R2 : 0.8159012204755455
Adjusted R2 : 0.8158805056968783

#Base Model
The mean sales value is 6955.51
The standard deviation sales value is 3104.21

We can observe that the RMSE is ON and R2 score is which is a pretty decent score.

Ridge Regression

- Ridge Regression was applied and the score obtained was 0.8175 which is good score. We can notice the performance is almost same as Linear Regression.

- Total 18 coefficients and a intercept were obtained.

Evaluation of Ridge Regression Model

The following scores were obtained for the Ridge Regression Model

MSE : 1759878.0643092683
MAE : 952.3748671903994
RMSE : 1326.6039591035708

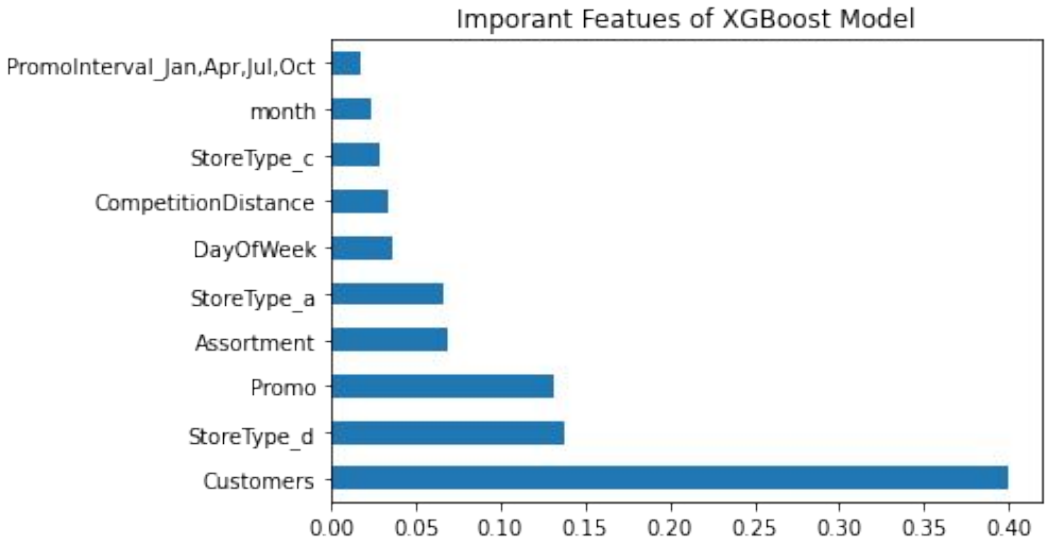
R2 : 0.8159012627898875
Adjusted R2 : 0.8158805480159815

#Base Model
The mean sales value is 6955.51
The standard deviation sales value is 3104.21

We can observe that the RMSE is on the higher side and R2 score is 0.815 which is a pretty decent score.

XGBoost Regression

- XGBoost Regression was applied and cross validation (with cv=5) was performed the score obtained was 0.879 which is good score.
- The top 10 feature important features were found to be:



Evaluation of XGB Regression Model after Cross Validation

The following scores were obtained for the XGB Regression Model after Cross Validation

MSE : 1147086.3075406377
MAE : 781.5238530666539
RMSE : 1071.0211517708872

R2 : 0.8800046747715293
Adjusted R2 : 0.8799911729079666

#Base Model
The mean sales value is 6955.51
The standard deviation sales value is 3104.21

We can observe that the RMSE is on the better and R2 score is 0.88 which is a pretty decent score.

Stochastic Gradient Descent Regression

- SGD Regression was applied and the score obtained was 0.816 which is good score.
- Evaluation of Linear Regression Model

MSE : 1763157.993953631
MAE : 956.0188575775644
RMSE : 1327.839596470007

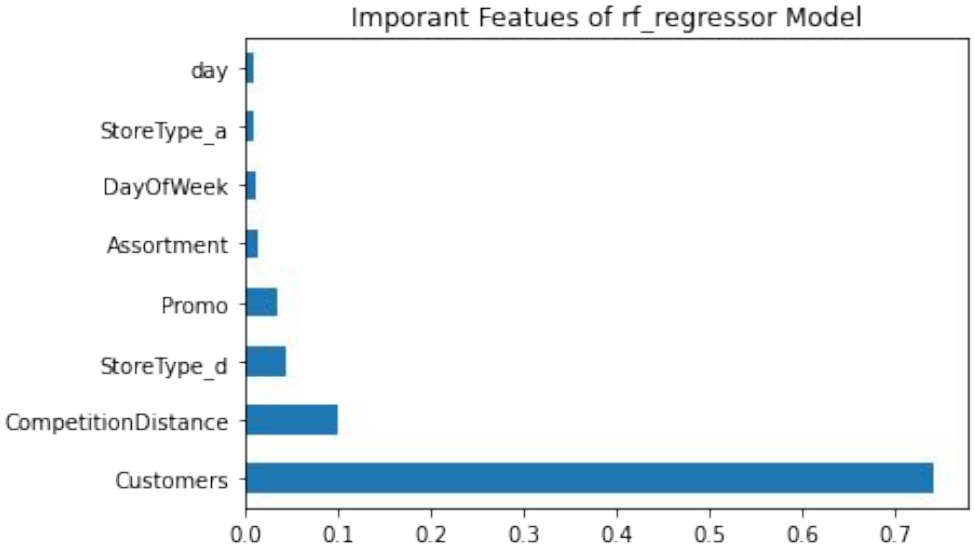
R2 : 0.8155581532768417
Adjusted R2 : 0.8155373998962832

#Base Model
The mean sales value is 6955.51
The standard deviation sales value is 3104.21

We can observe that the RMSE is on the higher side and R2 score is 0.815 which is a pretty decent score.

Random Forest Regression

- Random forest Regression was applied and hyperparameter tuning for n_estimators value was performed along with Grid search Cross Validation.
- The best fit n_estimator is found out to be : {'n_estimators': 25, 'random_state': 0}
- Using {'n_estimators': 25, 'random_state': 0} the negative mean squared error is: -339083.623
- The top 10 feature important features were found to be:



Evaluation of Random Forest Regressor Model

The following scores were obtained for the Random Forest Regressor Model

MSE : 329647.0112966973
MAE : 382.55220058937664
RMSE : 574.1489452195286

R2 : 0.9655160208337338
Adjusted R2 : 0.965512140699396

#Base Model
The mean sales value is 6955.51
The standard deviation sales value is 3104.21

We can observe RF Model has given the best results with a RMSE of 594.14 and an excellent R2 score of 0.965.

Final Results

	Model	MSE	MAE	RMSE	R2	Adjusted R2
0	Linear Regression	1759881.992	952.371	1326.605	0.816	0.816
1	Lasso Regression	1759864.770	952.370	1326.599	0.816	0.816
2	Lasso Regression (with HPT and CV)	1759878.469	952.371	1326.604	0.816	0.816
3	Ridge Regression	1759878.064	952.375	1326.604	0.816	0.816
4	XGBoost Regression	1147086.308	781.524	1071.021	0.880	0.880
5	XGBoost Regression (with CV)	1147086.308	781.524	1071.021	0.880	0.880
6	SGD Regression	1763188.298	956.992	1327.851	0.816	0.816
7	Random Forest Regression	329647.011	382.552	574.149	0.966	0.966
8	Random Forest Regression (with HPT and CV)	304084.598	366.501	551.439	0.968	0.968

HPT: Hyper Parameter Tuning.
CV: Cross Validation

Conclusion

- Random Forest Regressor model performs the best with an R2 Score of 0.968 with the best parameter of n_estimators 25. However we can us n_estimator of 10 has it almost the same accuracy but reduces the computational time.
- Next the XGBoost gives the best with the R2 Score of 0.88.
- Interestingly we notice that the models, Linear Regression, Lasso regression, Ridge Regression and Stochastic Gradient Descent Regression performs with the almost same R2 score of 0.81.

Note: Cross Validation was performed for Lasso regression, XGBoost and Random Forest Regression and Hyperparameter tuning was performed for Lasso regression and XGBoost

Thank You