Code Rush

Financial Data Analytics Apprenticeship

Project Analysis Report

of

Rossmann Sales Forecasting

Submitted to : Code Rush

Submitted by : Pragya Shakya

October, 2022

Introduction

Sales forecasting is the process of estimating sales of a particular product over a specific period of time. It deals with forecasting which is an important part of modern business intelligence for the decision making process. It enables businesses to create effective strategies that increase productivity and motivation. Predicting the future sales for a certain time using machine learning is the main goal of the project. The historical data for the sales time series can be used to make sales predictions. The project will be done to improve performance of predictive models for sales forecasting.One of the central aspects of sales forecasting is that accuracy is key:

- If the forecast is too high it may lead to **over-investing** and therefore **losing money**.
- If the forecast is too low it may lead to **under-investing** and therefore **losing opportunity**.

The Rossmann Store Sales Dataset is used for the project. The store sales is influenced by several factors such as store type, date, promotion etc. Linear Regression(LR) and Random Forest Regressor were used to train models and predict sales. Both the models are regression models. The random forest regressor model gives better results compared to the linear regression.

Data

The dataset collected is the Rossmann Store Sales from kaggle Competition(https://www.kaggle.com/c/rossmann-store-sales). It contains the historical sales data for 1115 Rossmann stores in Germany. Rossmann is a chain drug store that operates in 7 European countries. The goal of this project is to have reliable sales predictions for the store.The files provided are train.csv and store.csv.

S.No.	Dataset	Variables	No of Variables	No of Observations
1	Train	Store, DayOfWeek, Date, Sales, Customers, Open, Promo, StateHoliday, SchoolHoliday	9	1012170
2	Store	Store, StoreType, Assortment, CompetitionDistance, CompetitionOpenSinceMonth, CompetitionOpenSinceYear, Promo2, Promo2SinceWeek, Promo2SinceYear, PromoInterval	10	1115

Variables Description:

S.No.	Variables	Measurement Scale	Possible values
1	Store	Nominal	1 to 1115
2	DayOfWeek	Nominal	1,2,3,4,5,6,7
3	Date	Interval	1/1/2013 to 7/31/2015
4	Sales	Ratio	0 to 41551
5	Customers	Ratio	0 to 7338
6	Open	Nominal	0(Closed) 1(Open)

7	StateHoliday	Nominal	a: Public Holiday b:Easter Holiday c:Christmas Holiday 0:None
8	SchoolHoliday	Nominal	0: No Holiday 1:Holiday
9	StoreType	Nominal	a, b, c, d
10	Assortment	Nominal	a:basic b:extra c:extended
11	CompetitionDistance	Ratio	20-75860
12	CompetitionOpenSince[Month]	Interval	1(Jan) - 12(Dec)
13	CompetitionOpenSince[Month/Year]	Interval	1900-2005
14	Promo	Nominal	0 : No Promotion 1: Promotion
15	Promo2	Nominal	0, 1
16	Promo2Since[Week]	Nominal	1-50
17	Promo2Since[Year/Week]	Nominal	2009-2015
18	PromoInterval	Ordinal	{jan, apr, jul,oct} {feb, may, aug, nov} {mar, jun, sept, dec}

Analysis

The two dataset : train.csv and store.csv are loaded. The train dataset consists of 1017209 rows and 9 columns whereas the store dataset consists of 1115 rows and 10 columns.

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

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
0	1	5	2015-07-31	5263	555	1	1	0	1
1	2	5	2015-07-31	6064	625	1	1	0	1
2	3	5	2015-07-31	8314	821	1	1	0	1
3	4	5	2015-07-31	13995	1498	1	1	0	1
4	5	5	2015-07-31	4822	559	1	1	0	1

```
store_data.head(5)
```

	Store	StoreType	Assortment	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2	Promo2SinceWeek	Promo2SinceYear
0	1	c	a	1270.0	9.0	2008.0	0	NaN	NaN
1	2	a	a	570.0	11.0	2007.0	1	13.0	2010.0
2	3	a	a	14130.0	12.0	2006.0	1	14.0	2011.0
3	4	c	c	620.0	9.0	2009.0	0	NaN	NaN
4	5	a	a	29910.0	4.0	2015.0	0	NaN	NaN

Data Cleaning and Preprocessing

After the data is loaded, the data needs to be cleaned. The train data does not contain null as well as duplicate values. There are several null values present in the store data.

```
#check null values
print("Store data null values:\n",store_data.isnull().sum())
```

```
Store data null values:
 Store                        0
StoreType                     0
Assortment                    0
CompetitionDistance           3
CompetitionOpenSinceMonth   354
CompetitionOpenSinceYear    354
Promo2                        0
Promo2SinceWeek             544
Promo2SinceYear             544
PromoInterval               544
dtype: int64
```

The variable CompetitionDistance has 3 null values and is right skewed. The null values are replaced with the median. The null values of CompetitionOpenSinceMonth and CompetitionOpenSinceYear are replaced with 0. While the Promo2SinceWeek, Promo2SinceYear and PromoInterval are removed as the number of null values is higher.

After dealing with null values, the train data and store data is merged into a single dataframe.

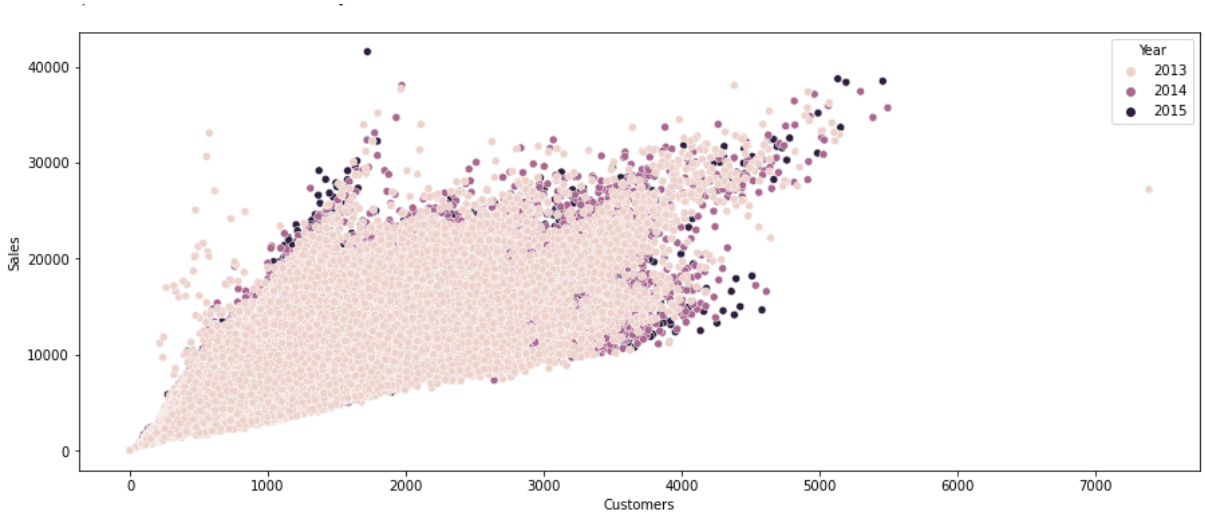
Data Visualization and Analysis

After the pre-processing of data, the data is visualized with the help of different graphs such as box plot, line graph, histogram, bar chart etc. In the analysis process, the analysis of the variables in the data is done. The univariate and bivariate analysis is conducted. In univariate analysis, the single variable is analyzed. In bivariate analysis, the relationship between two variables is represented.

The target variable is Sales. It is found that rossmann has 13.45% of the time sales over 10,000 euros while 0.02777% sales under 1000 euros. The maximum turnover in the data is 41551 euros. The maximum number of customers in the store was 7388 on 22nd january 2013.

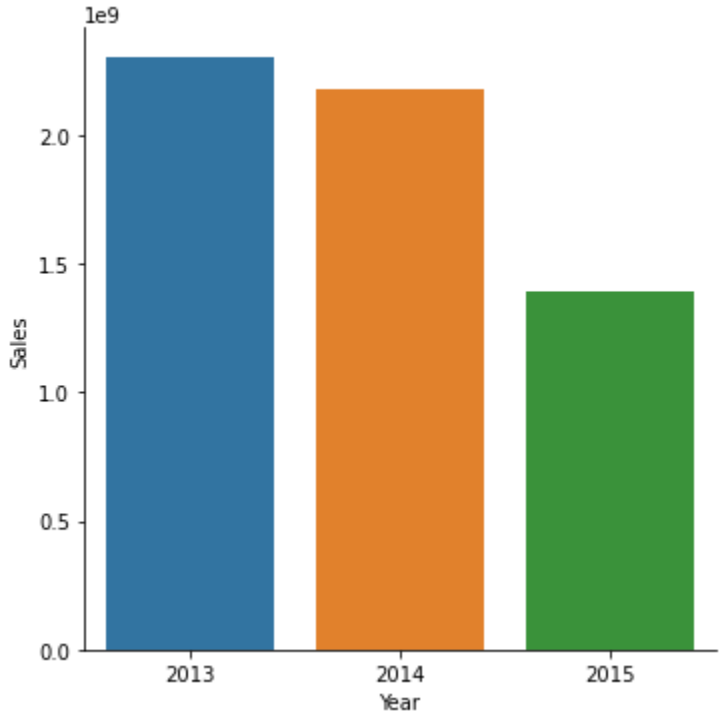
Sales Vs. Customers

The relationship between the sales and the customers is shown by the scatterplot below. The sales and customer shows strong linear relation. When the no of customer increases the sales also increases and vice-versa.



The sum of sales per year is given below:

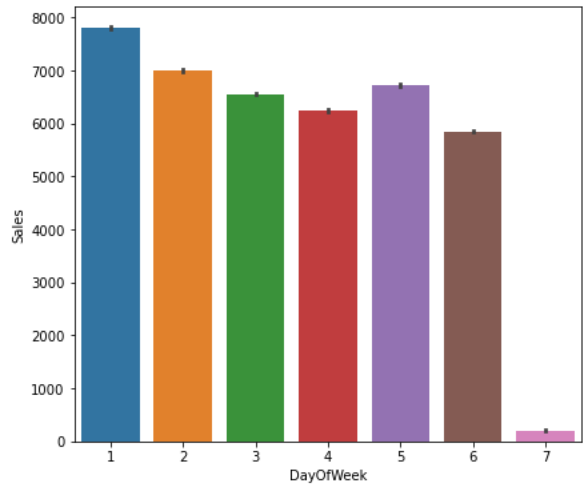
	Year	Sales
0	2013	2302876084
1	2014	2180804896
2	2015	1389499643

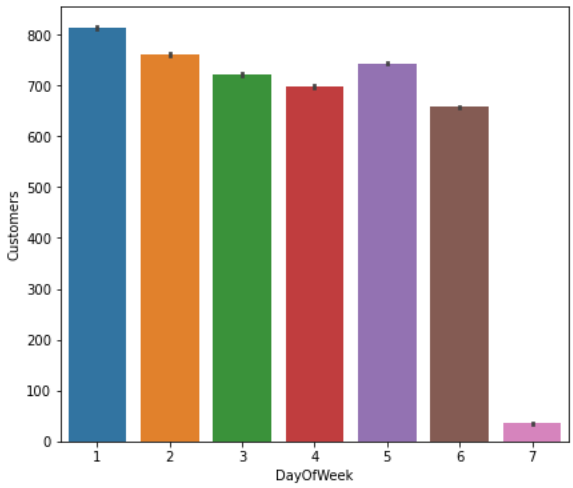


The year 2013 has maximum sales while 2015 has the minimum sales as seen in the bar graph.

DayOf Week Vs Customer and Sales

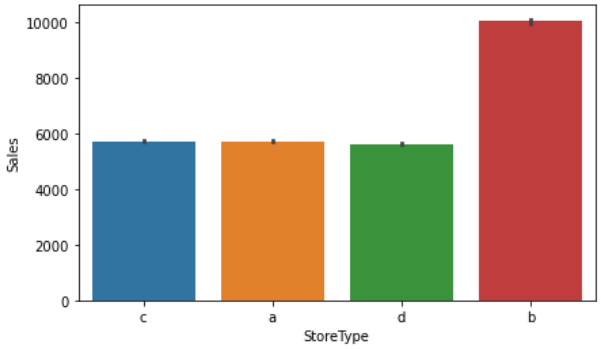
As shown in the graph below, the highest number of customers is on monday as well as maximum turnover is also on monday.

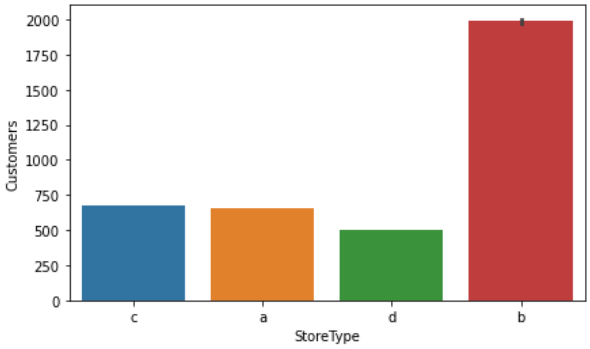




Sales and Store Type

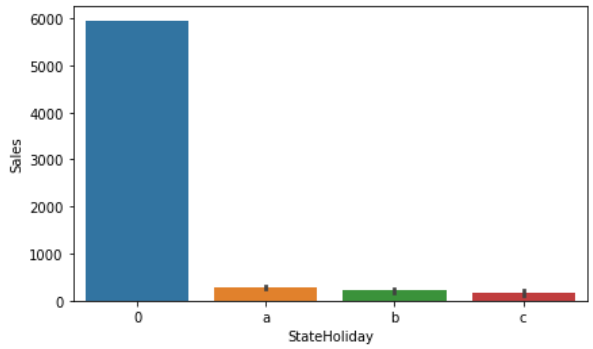
The store type b has maximum number of the customers as well as maximum turnover as well. While the store type d has a good amount of sales despite the less number of customers.

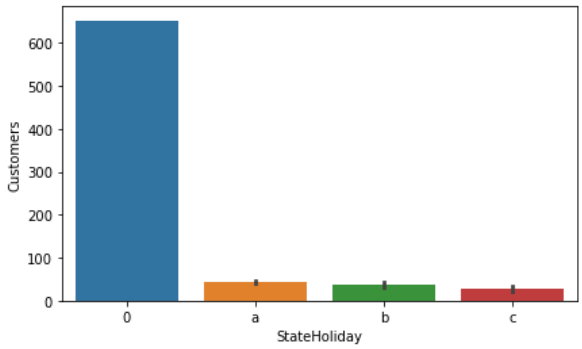




Stateholiday Vs Sales and Customers

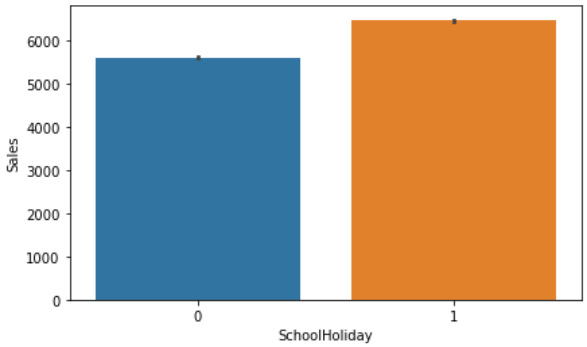
Here, the a is the public holiday, b is the easter holiday, c is christmas holiday and 0 is no holiday. The graph shows the maximum sales and customer when there is no any kind of holiday.

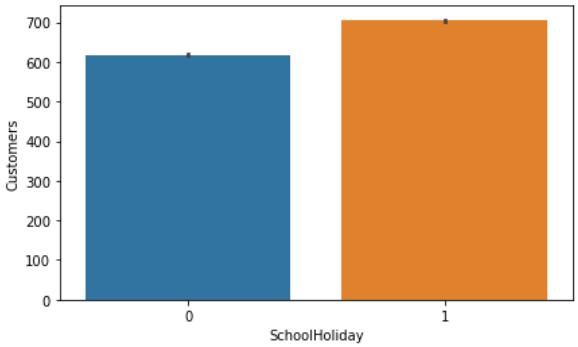




SchoolHoliday Vs Sales and Customers

The number of sales when there is a school holiday is higher than that of the days when there is no school holiday as seen from the graph below.

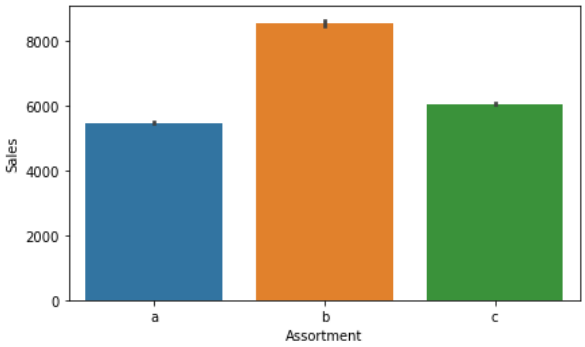


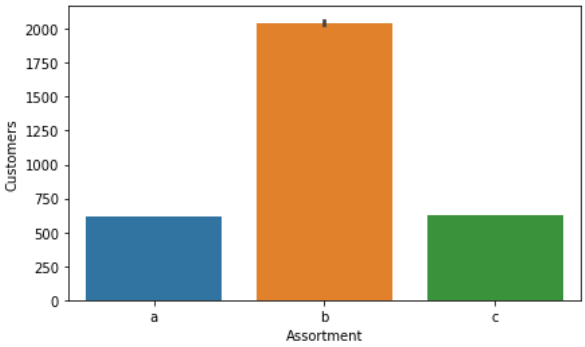


Also even when there is no school and state holidays the store is closed for 139610 days. The reason is not mentioned for this.

Assortment vs Sales and Customers

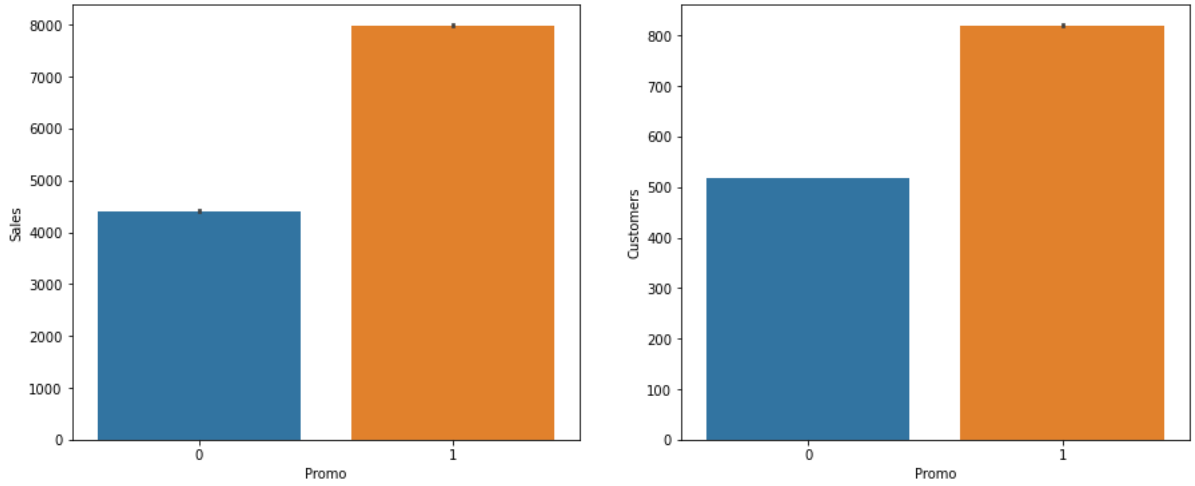
The number in sales as well as customers is higher when the assortment provided is extra than compared to basic and extended.



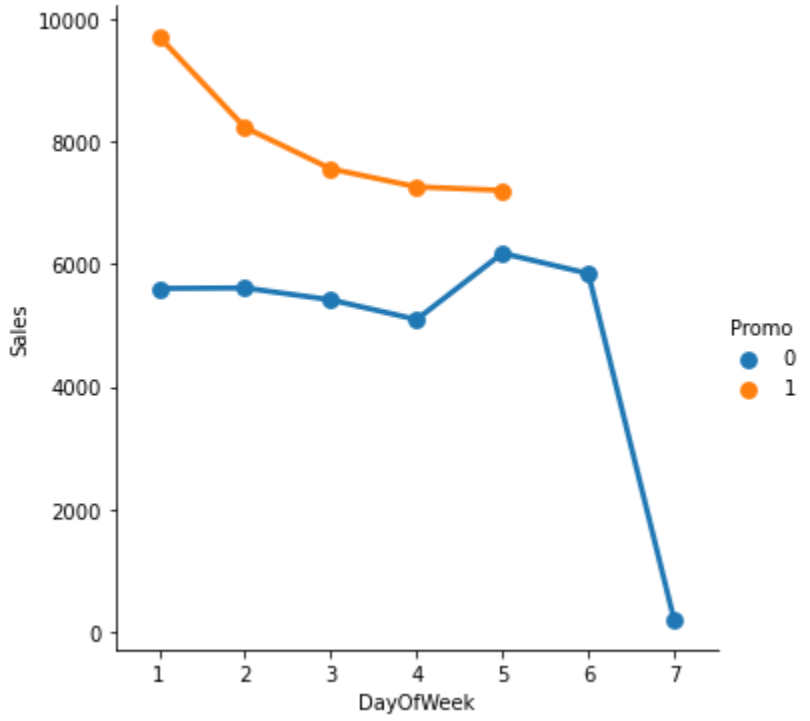


Promo vs Sales and Customers

The number of sales and customers is higher when there is a promotion in the store.

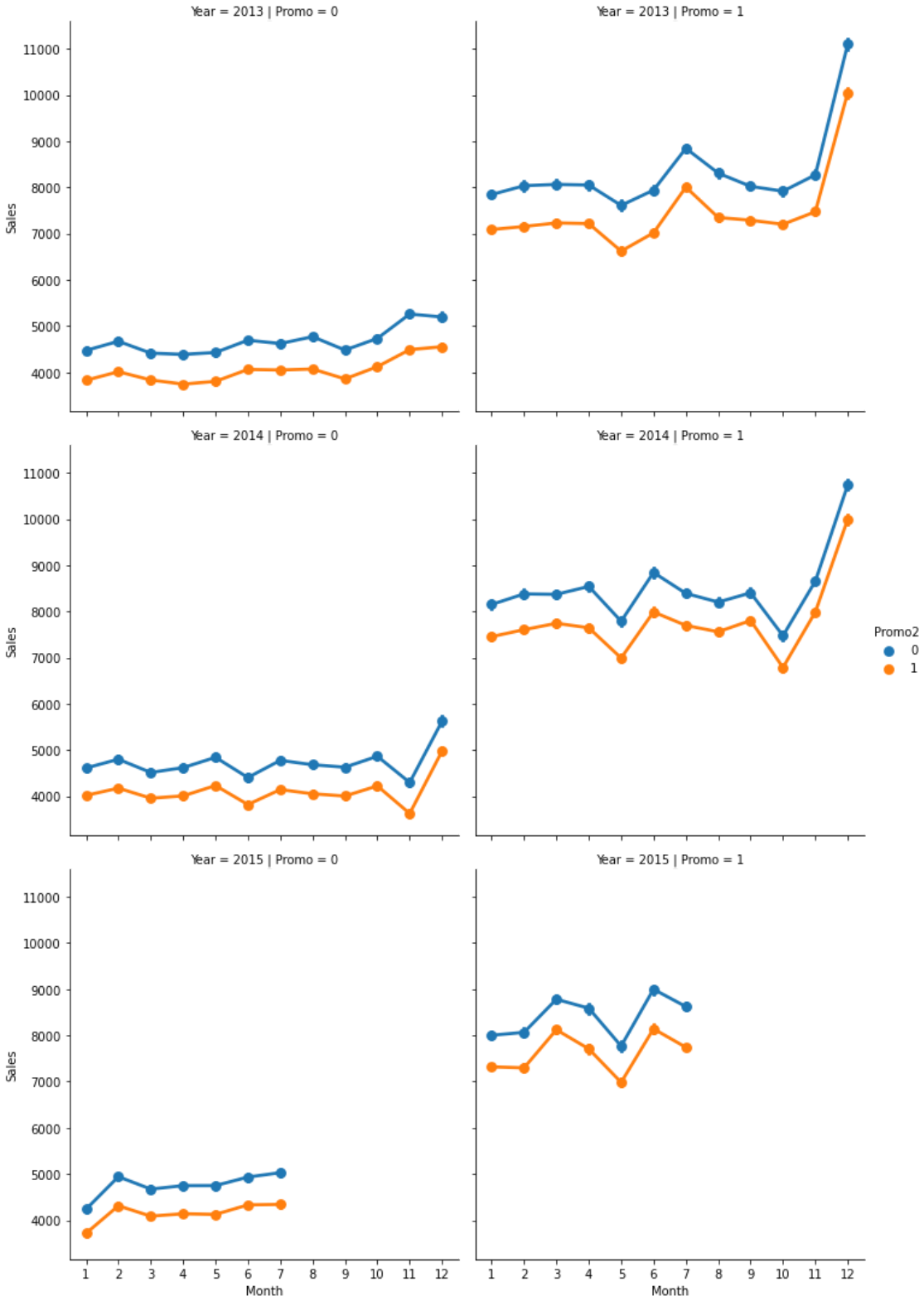


It is seen that the sales are the highest on monday. The promotion on monday seems to have maximum sales.



.

There is a significant increase in the number of sales when there is a promotion in the store every year as seen in the figure below



Predictive Modeling

For the model building, the feature extraction is performed. Dummy variables are created for the variables such as StoreType and Assortment. A number of features are extracted for model building which are 'Store', 'DayOfWeek', 'Date', 'Customers', 'Open', 'Promo', 'StateHoliday', 'SchoolHoliday', 'Year', 'Month', 'CompetitionDistance', 'Promo2', 'CompetitionOpenSince', 'a', 'b', 'c', 'd', 'basic', 'extended', 'extra'.

After the features are extracted, the data is split into training set and testing set in the ratio of 80 and 20 respectively. Then the training data is fitted into the model with the help of linear regression and random forest regressor. The model used to predict the future sales.

Results

Following results is obtained from the models:

S.No.	Algorithm	R2 Score	MSE	RMSE
1	Linear Regression	0.8073	2847072.872	1687.327
2	Random Forest Regressor	0.9845	228841.465	478.374

Conclusion and Discussion

It can be concluded that the model built using the random forest regressor gives better results as compared to the linear regression. The R2 Score of Linear regression is 0.8073 whereas the random forest regressor is 0.9845 which shows that the model fit of random forest is better than the linear regressor. The root mean square error is used as the evaluating metric. The Root Mean Square Error of Linear regression is 1687.327 and random forest regressor is 478.374. The random forest regressor has lower error compared to the linear regression. The importance of feature customers is seen higher than the other features.

Appendices

Linear Regression Model

In [81]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import *
```

In [82]:
```
x_train, x_test, y_train, y_test= train_test_split(x,y, test_size=0.2, random_state=0)
```

In [83]:
```
model= LinearRegression()
model = model.fit(x_train, y_train)
```

In [84]:
```
pred = model.predict(x_test)
```

In [85]:
```
mse = mean_squared_error(y_test, pred)
print(mse)
```

```
2847072.872523209
```

In [86]:
```
rmse = np.sqrt(mse)
print("rmse=", rmse)
```

```
rmse= 1687.3271385606317
```

In [87]:
```
r2 = r2_score(y_test, pred)
print("R square=", r2)
```

```
R square= 0.8073273561164827
```

Random Forest Regressor

In [88]:
```
from sklearn.ensemble import RandomForestRegressor

rand_model = RandomForestRegressor(n_estimators=10)
rand_model.fit(x_train, y_train)
rand_pred = rand_model.predict(x_test)
```

In [89]:
```
mse = mean_squared_error(y_test, rand_pred)
print(mse)
```

```
227970.19465926403
```

In [90]:
```
rmse = np.sqrt(mse)
print("rmse=", rmse)
```

```
rmse= 477.4622442238381
```

In [91]:
```
r2 = r2_score(y_test, rand_pred)
print("R square=", r2)
```

```
R square= 0.9845723583138519
```

References

https://medium.com/analytics-vidhya/rossmann-store-sales-prediction-998161027abf#d726

Bohdan M. Pavlyshenko," Machine-Learning Models for Sales Time Series Forecasting", 2019

Yasaman Ensafi, Saman Hassanzadeh Amin, Guoqing Zhang and Bharat Shah, "Time-series forecasting of seasonal items sales using machine learning – A comparative analysis", 2022