Capstone Project

Retail Sales Prediction

Soumabha Sarkar

Points to Discuss:

- Agenda
- Data summary
- EDA and feature engineering
- Feature selection
- Preparing dataset for modelling
- Applying Model
- Model validation and selection
- Challenges
- Conclusion

Agenda

- 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.
- We 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 Summery

Rossmann Stores Data.csv - historical data including Sales

store.csv - supplemental information about the stores

Most of the fields are self-explanatory. The following are descriptions for those that aren't.

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

Data Summery (continued)

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
Competition Distance - distance in meters to the nearest competitor store

Data Summery (continued)

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

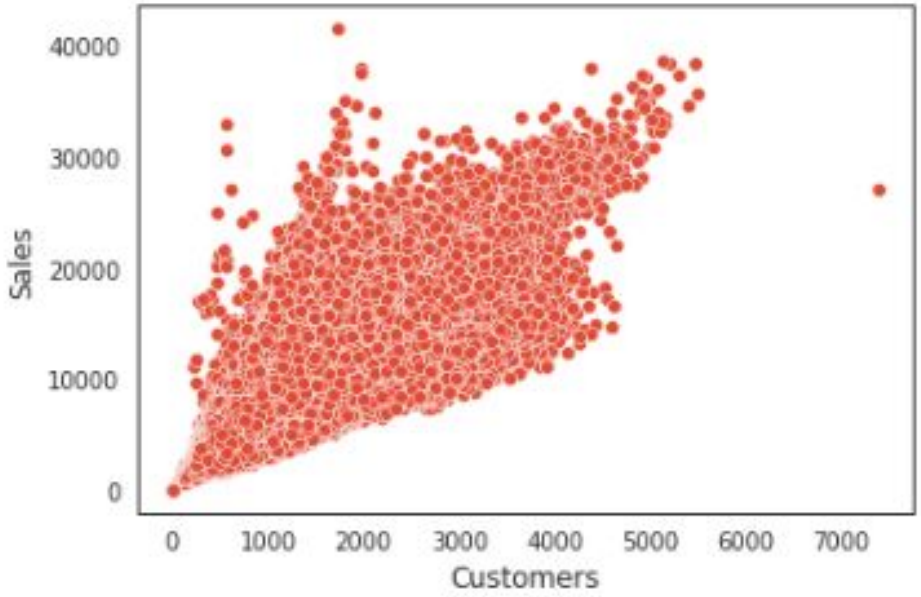
Exploratory Data Analysis

- Correlation Heatmap between features.
- Stores are mainly open on which day of the week analysis.
- Different year wise sales data analysis.
- Different year wise customer data analysis.
- Sales affected by school holidays or not.
- Distribution of different types of stores analysis.
- How store assortment type is influencing sales.
- How promotion is influencing sales.

Correlation Heatmap

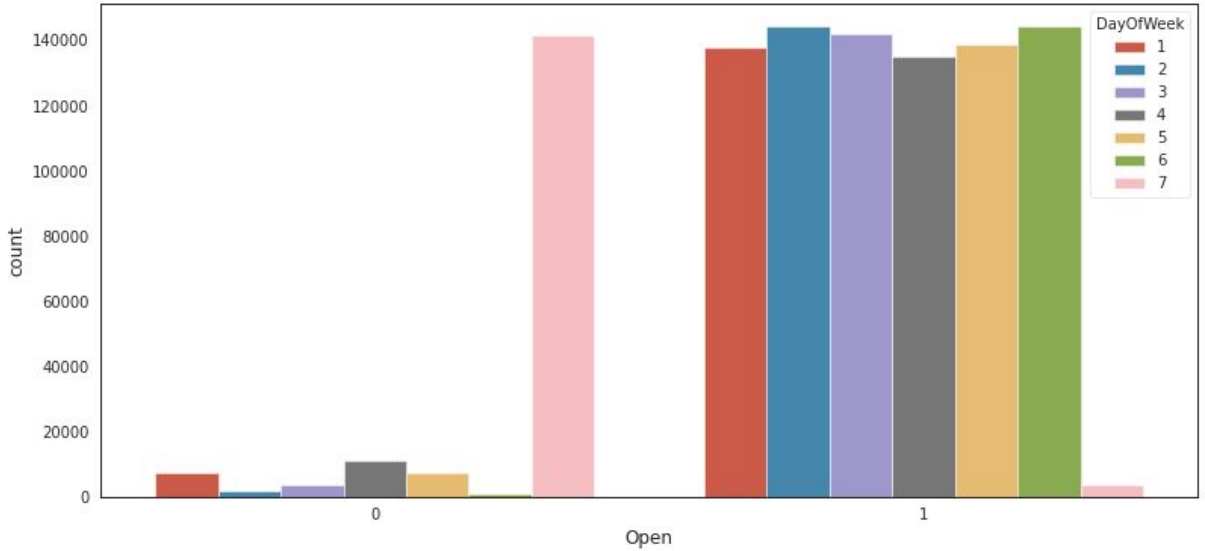
We can see that there are some columns which are correlated to the sales column.

Highly correlated feature column



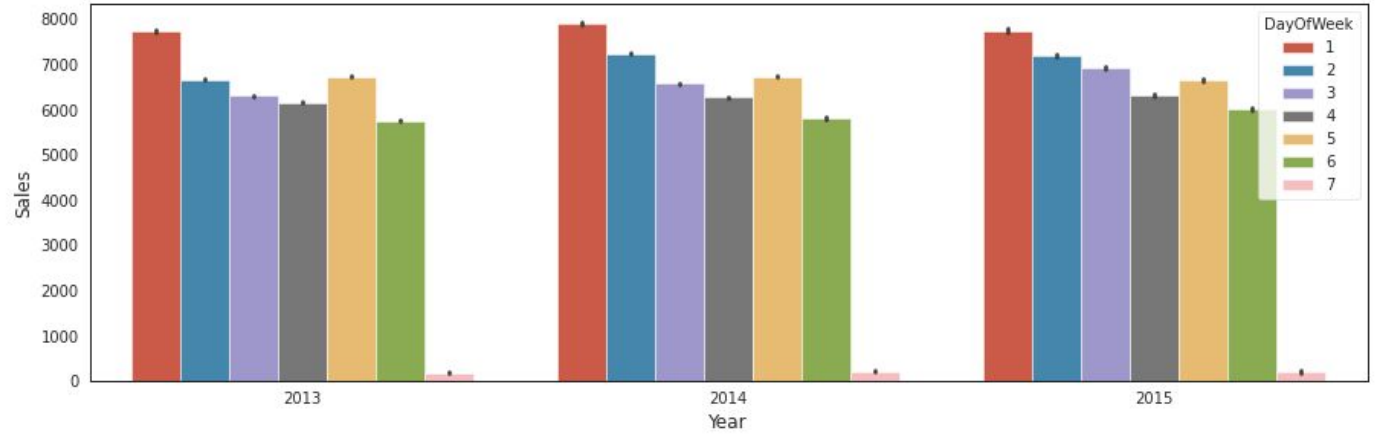
Customer column is highly correlated with Sales which is the dependent feature

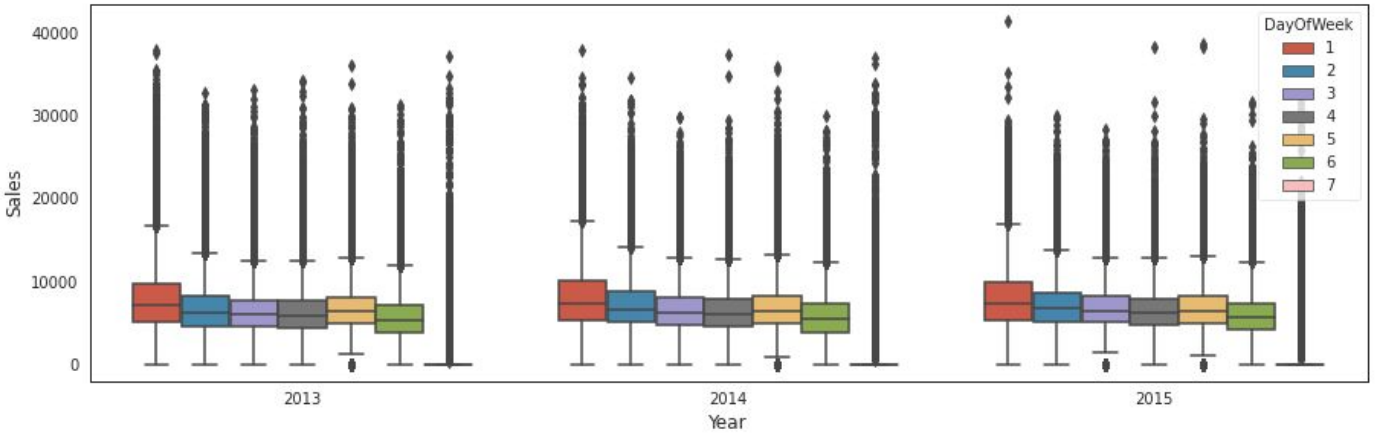
Stores are mainly open on which day of the week analysis.



From the above bar plot it is concluded that the stores are mainly open all the days of the week except Sunday.

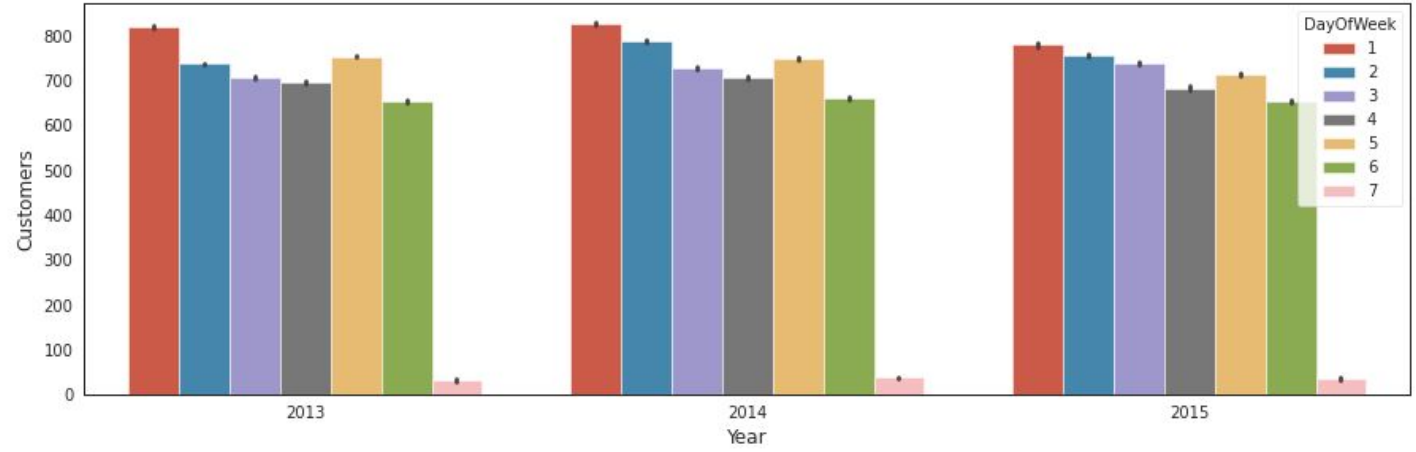
Different year wise sales data analysis.

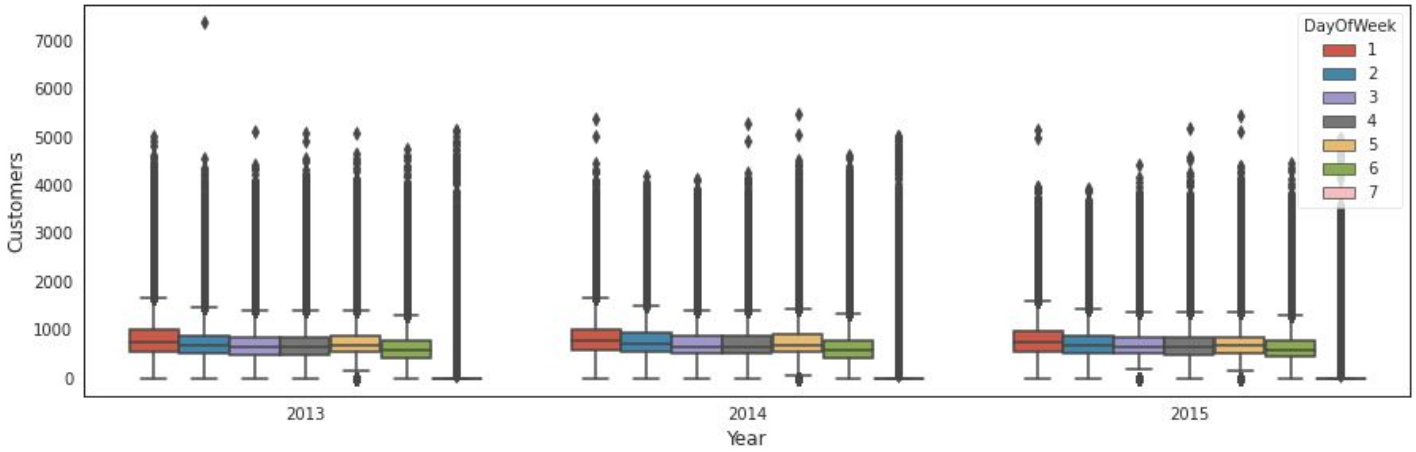




The top image is showing the mean values of sales year wise and the bottom image is showing the actual sale values year wise.

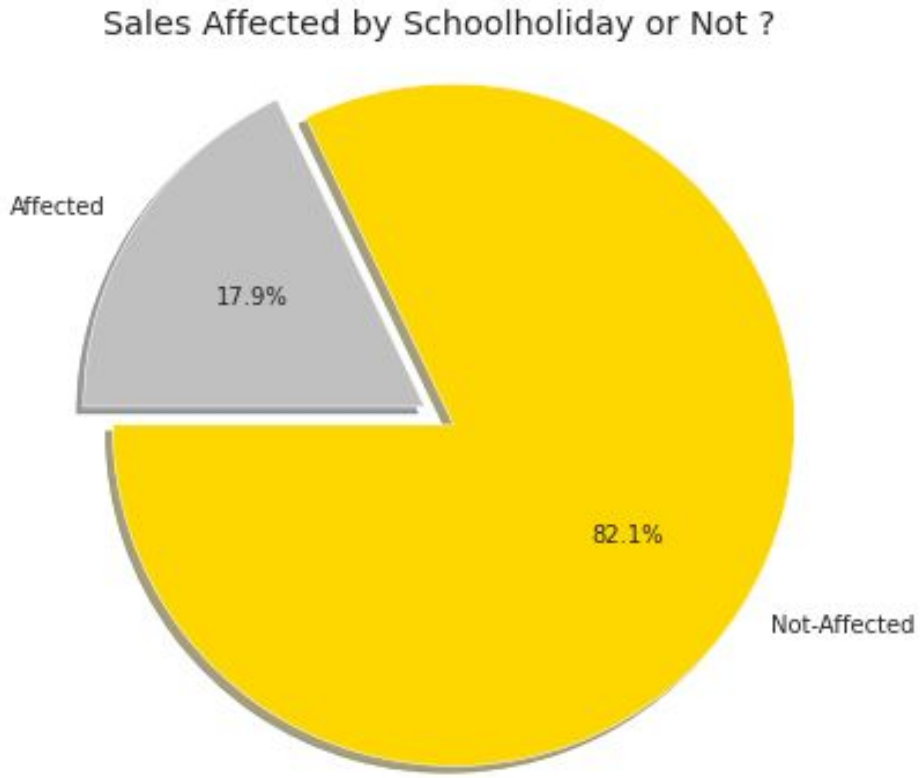
Different year wise customer data analysis.





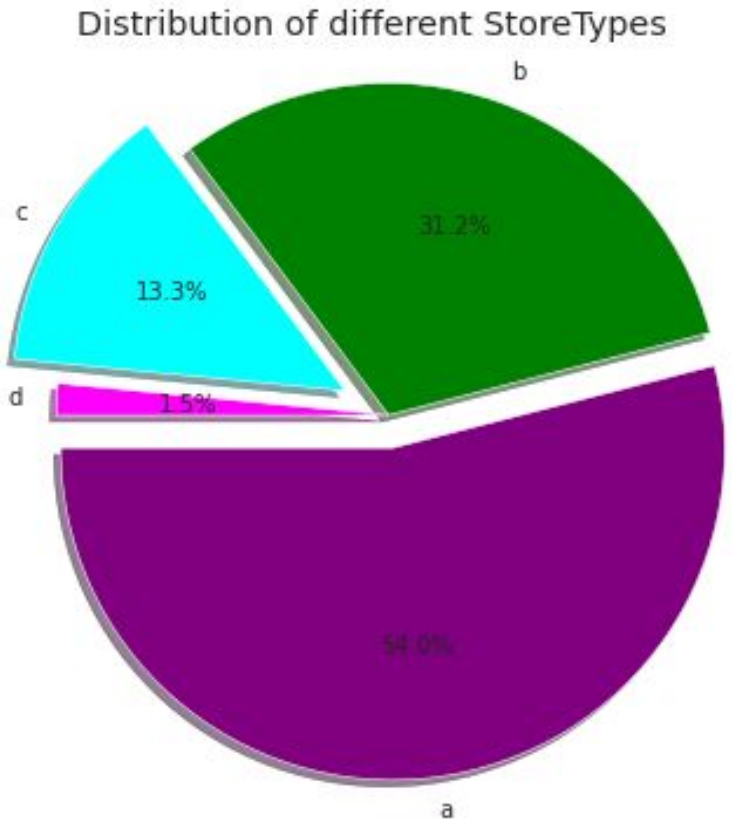
The top image is showing the mean values of number of customers year wise and the bottom image is showing the actual customer values year wise.

Sales affected by school holidays or not.



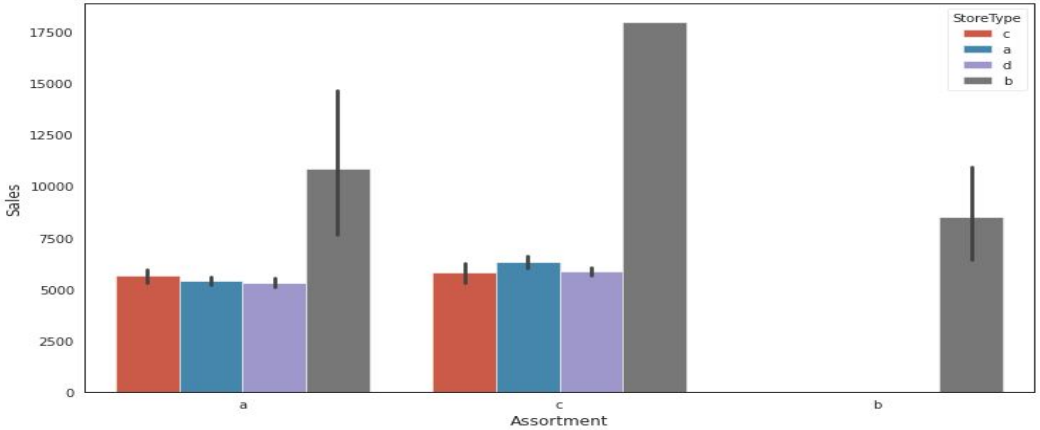
From this pie chart it is concluded that 82.1% sales are not affected by School holidays.

Distribution of different types of stores analysis.

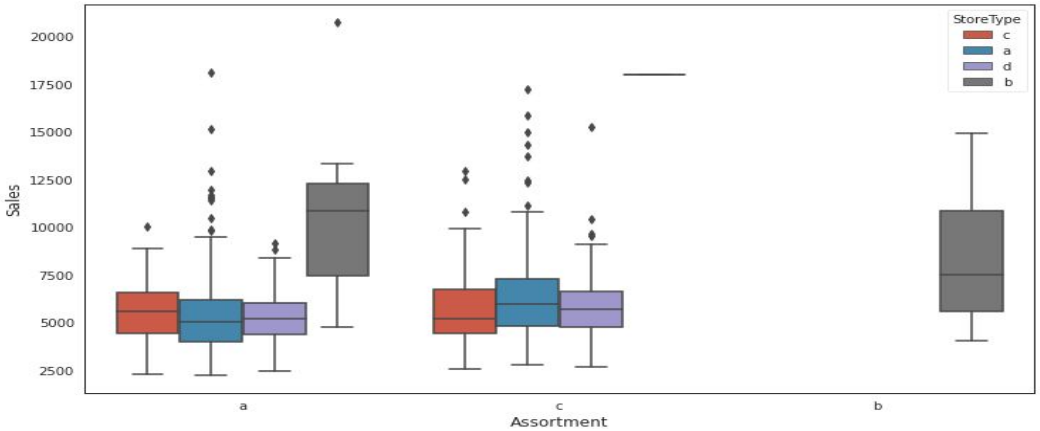


Type a stores are having 54.0% market share while Type b and Type c stores are having market share of 31.2% and 13.3% respectively.

How store assortment type is influencing sales.

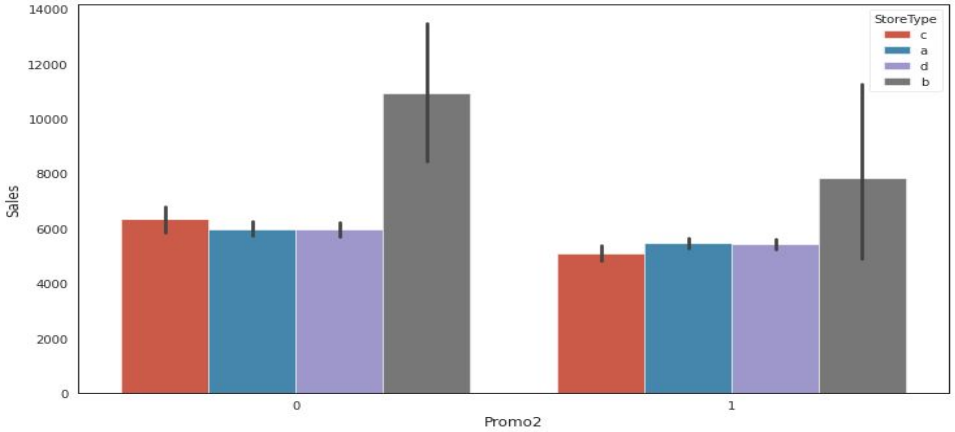


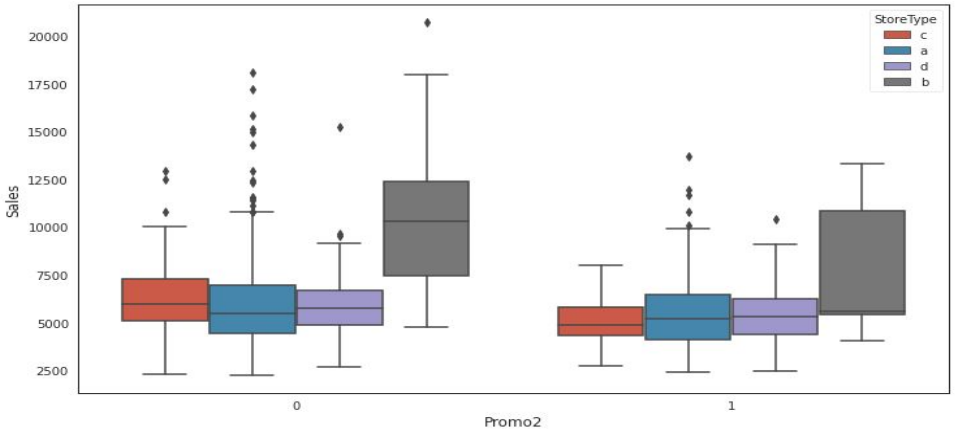
Assortment is a collection of goods or services that business provides to customers.



B type of stores are providing maximum service in every assortment type stores. Between them the extended service is highest.

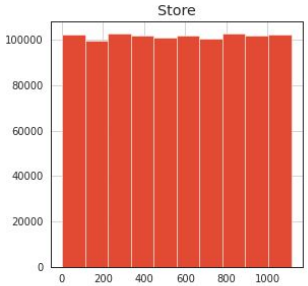
How promotion is influencing sales.

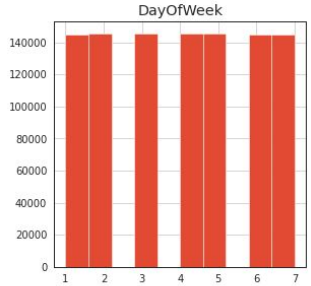


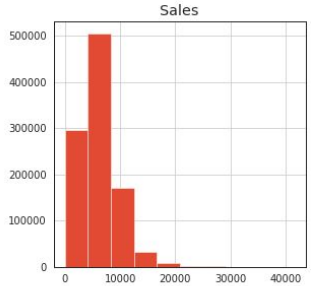


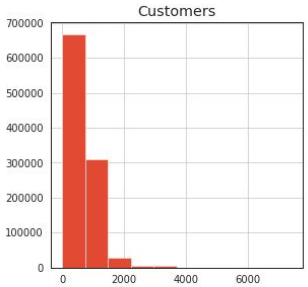
All the stores participated in promotion and among them the sales of Type b stores are influenced mostly.

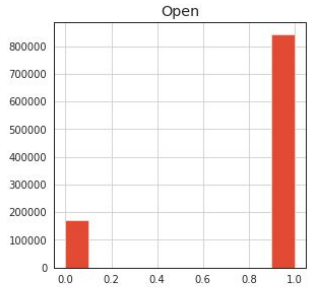
Feature Selection

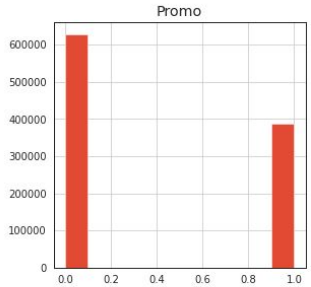


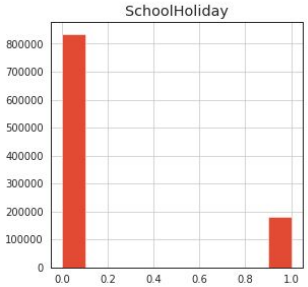








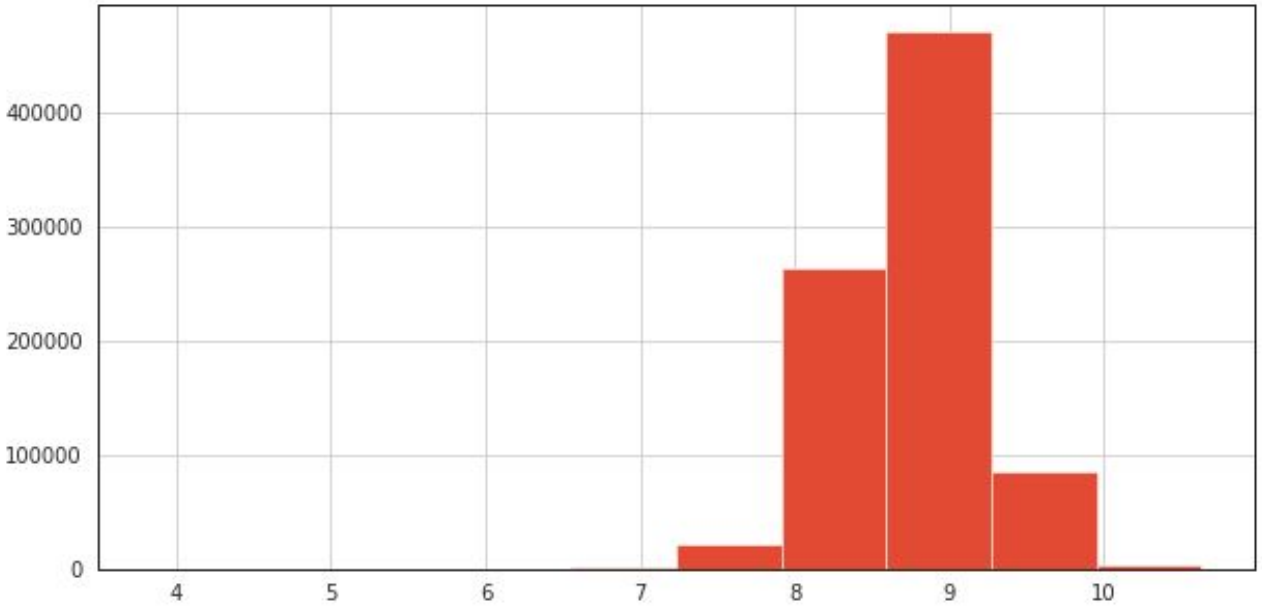




Among numerical columns Sales and Customers are heavily skewed. So we do not take Customer column at the time of feature selection.

As the Sales column is also skewed so we will apply natural logarithm to decrease the skewness.

Feature Selection(continued)



After applying natural logarithm the skewness of Sales column is decreased considerably.

Preparing dataset for modelling

```
[ ]  final = pd.merge(rossmann,store, on="Store")
```

```
[ ]  final.columns
```

```
Index(['Store', 'DayOfWeek', 'Date', 'Sales_x', 'Customers_x', 'Open', 'Promo',
       'StateHoliday', 'SchoolHoliday', 'Month', 'Year', 'log_sales',
       'StoreType', 'Assortment', 'CompetitionDistance',
       'CompetitionOpenSinceMonth', 'CompetitionOpenSinceYear', 'Promo2',
       'Promo2SinceWeek', 'Promo2SinceYear', 'PromoInterval', 'Sales_y',
       'Customers_y'],
      dtype='object')
```

Merging rossmann sales data and store data as final dataset.

```
[ ]  final=pd.get_dummies(final,columns=['StoreType','Assortment','Year'])
```

Getting dummies for “StoreType”, ”Assortment” and “Year” columns

Preparing dataset for modelling

	feature	VIF
0	Store	1.005844
1	DayOfWeek	1.144093
2	Sales_x	17.740181
3	Customers_x	32.514430
4	Open	28.882002
5	Promo	1.662012
6	StateHoliday	1.009372
7	SchoolHoliday	1.034469
8	Month	1.035269
9	CompetitionDistance	1.100646
10	CompetitionOpenSinceMonth	2.630623
11	CompetitionOpenSinceYear	2.629019
12	Promo2	791826.852532
13	Promo2SinceWeek	2.545881
14	Promo2SinceYear	791300.191221
15	Sales_y	13.523512
16	Customers_y	29.537384

	feature	VIF
0	DayOfWeek	1.113600
1	Open	0.000000
2	Promo	1.092297
3	StateHoliday	1.014546
4	SchoolHoliday	1.034374
5	Month	1.116274
6	CompetitionDistance	1.072832
7	CompetitionOpenSinceMonth	1.027565
8	Promo2	2.400977
9	Promo2SinceWeek	2.392673

After dropping some columns.

We have found some columns with multicollinearity using variance inflation factor. We dropped the columns with multicollinearity.

Preparing dataset for modelling(continued)

```
[ ] X = final.drop(['log_sales','Store','Date','Sales_x', 'Customers_x','Sales_y', 'Customers_y','CompetitionOpenSinceYear','Promo2SinceYear','PromoInterval'] , axis = 1)
    y = final['log_sales']
```

```
X.columns
```

```
Index(['DayOfWeek', 'Open', 'Promo', 'StateHoliday', 'SchoolHoliday', 'Month',
       'CompetitionDistance', 'CompetitionOpenSinceMonth', 'Promo2',
       'Promo2SinceWeek', 'StoreType_a', 'StoreType_b', 'StoreType_c',
       'StoreType_d', 'Assortment_a', 'Assortment_b', 'Assortment_c',
       'Year_2013', 'Year_2014', 'Year_2015'],
      dtype='object')
```

Preparing the dataset for train test split we have taken log_sales as dependent variable and the rest of the features as independent variables.

```
[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=7)
```

Performing Train Test split

Applying Model

We are using linear regression model so, we need to check 4 basic assumptions of linear regression.

1. There should to be linear relationship between independent and dependent variables.
2. The sum of residuals/error should be near to 0.
3. There should not be multicollinearity.
4. There should not be heteroscedasticity.

Applying Model

Linear Regression

```
rmse:  0.37317056137445515
mse:  0.13925626787652598
r2:  0.2337091859317988
```

Ridge Regression

```
[ ] grid.best_score_

    -0.14081048484182873

[ ] grid.best_params_

    {'alpha': 0.3}

[ ] rmse = np.sqrt(-grid.best_score_)
    rmse

    0.3752472316244701
```

Lasso Regression

```
[ ] grid.best_score_

    -0.14031944678933436

[ ] grid.best_params_

    {'alpha': 0.001, 'max_iter': 600}

[ ] rmse = np.sqrt(-grid.best_score_)
    rmse

    0.37459237417402713
```

Applying Model

Decision Tree

```
rmse:  0.17588256024711724
mse:  0.030934674999080824
r2_score:  0.8297745756837366
```

Without parameter tuning the decision tree regressor is giving 0.1758 as RMSE which is quite good in comparison with Lasso , Ridge , etc. But above training was not cross-validated as Lasso and Ridge were.

Random Forest

```
[ ] mse_rf = mean_squared_error(y_test,rf_pred)
    mse_rf

    0.054732583767700065

[ ] r2_rf = r2_score(y_test,rf_pred)
    r2_rf

    0.6988209090265531

[ ] rmse_rf = np.sqrt(mean_squared_error(y_test,rf_pred))
    rmse_rf

    0.23394995996516021
```

Applying Model

XGB Regressor

```
mse:  0.01676851496952229
rmse:  0.129493300867351
r2_score:  0.9077272485996585
```

XGB Regressor is giving better RMSE and R Square values compared to other previous models. It is showing higher accuracy compared to other models.

Model Validation

We have done cross validation up to 5 times on XGB Regressor model

```
array([0.20192842, 0.17939616, 0.09593488, 0.16593964, 0.15749228])
```

Cross validations up to 5 also yielded a minimum of 0.09593 which is less than any other model's RMSE.

So we trained total data set using XGB Regressor.

Final Prediction Data

Final sales data

```
[ ]  Final_data.sort_index(ascending = True).Sales.head()
```

```
3     7362.134766
4     8227.487305
6     5338.952148
18    5338.952148
29    5731.881348
Name: Sales, dtype: float64
```

The above dataframe is the predicted data form the given dataset.

We have predicted the sales using XGB Regressor

Challenges

1. Here we had to deal with the larger datasets. We had to merge two datasets into one to get the meaningful insights from data.

2. We did a lot of feature engineering in order to get useful information from multiple columns.

3. We also applied log transform to Sales column to avoid skewness which helped us to predict our values well.

4. We tried to predict future data which was the most complex part of our project.

Conclusion

The Rossmann store sales prediction is very engrossing data science problem to solve. We noticed that the problem is more concentrated towards the feature engineering and the feature selection part than on model selection. We had to spend around 60-70% of our time on analyzing data for trends in order to make our feature selection easier.

As we are building a linear regression model we emphasized on the basic 4 assumptions of a linear regression model.

Conclusion

We also have applied regularization techniques like Lasso, Ridge to Avoid Overfitting.

We also used XGBRegressor to make predictions that have better performance than any single model.

Most important feature came out to be customers, where sales is directly related to number of customers.

We performed cross validation using XGB Regressor model which gave us 0.09593 as minimum value that is lower than any other models

Thank You