Capstone Project:
On

Retail Sales Prediction

Presented by:
Subhadip Ghosh

Journey Roadmap:

- Problem Statements
- Discussing our dataset
- Summary of our dataset
- Checking Missing values
- Exploratory Data Analysis
- Feature Engineering
- Feature Scaling
- Model Training
- Evaluation Metrics and Model Summary
- Conclusion
- Plans to improve

Problem Statement:

- Rossmann operates over 3000 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. As there are a total of 3000 stores, and we will be working with 1115 of them, some stores in the dataset were temporarily closed for refurbishment.

- We will be working with 1115 Rossmann stores. The objective of our model is to predict the 'Sales' column of our dataset.

Discussing our Dataset:

Datasets we have:

1. Rossmann Stores Data.csv
2. Store.csv

Rossmann Stores Data.csv:

This csv file has information about all the 1115 stores of our dataset, information the number of customers visiting each day and the corresponding number of Sales in a day. It has addition information about whether the store was open or not on a particular day and information about school and state holidays as well.

Store.csv

This .csv file has information about the store, such as it's Assortment level, distance from its nearest competitor, Store type and so on. Both the datasets have the "Store ID" column as the common column.

We will be merging both the databases on the common column ‘Store Id’ to get a more detailed structure of our dataset

Dataset Summary:

After merging our datasets on the common column 'Store', we get the following info:

Shape of our dataset:

Our dataset has 1017209 rows and 18 columns.

Some of the most important features of our dataset include:

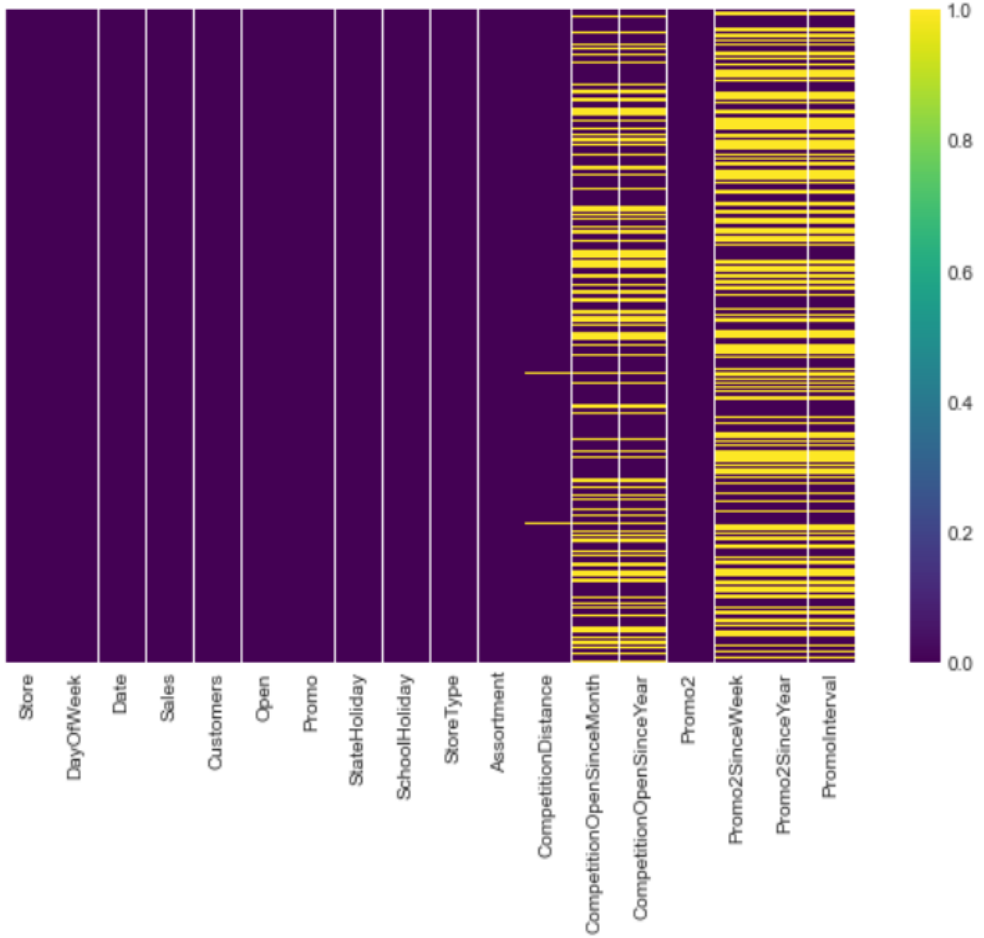
- **Store : Each store has been numbered from 1 to 1115**
- **Day of Week : Values from 1 to 7(Monday to Sunday)**
- **Date : Date of the record**
- **Customers : Number of Customers on a particular Date**
- **Competitor Distance : Nearest competitor distance**
- **There are other features like Store Type, assortment level, Promo going on or not and so on.**

Dependent Variable:

We have the Sales column which we have to predict using our model.

Checking Missing Values:

- By checking the info of our dataset, we found out that some of our columns have a large number of missing values.



Percentage of Null values in the highlighted column compared to the entire dataset:

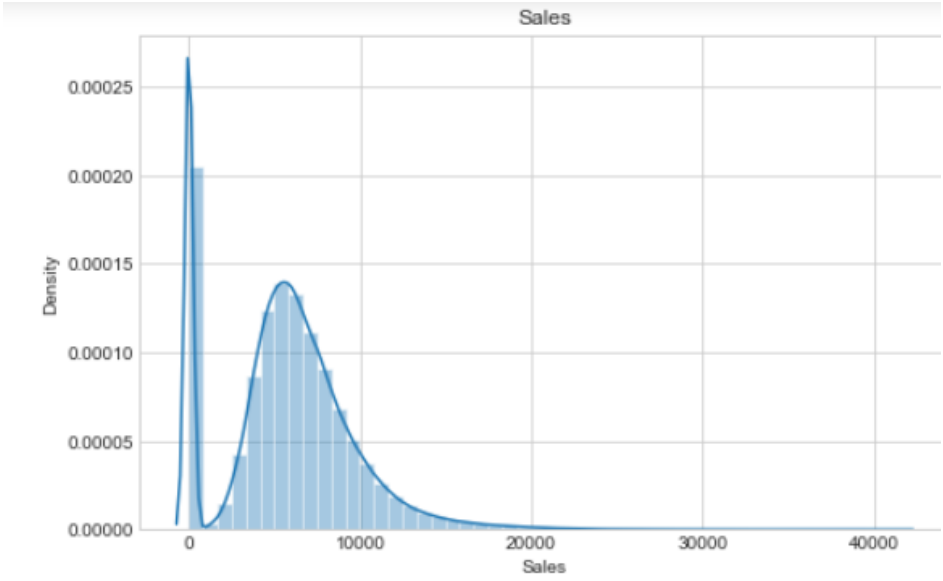
- **CompetitionDistance: 0.26%**
- **CompetitionOpenSinceMonth : 32%**
- **CompetitionOpenSinceYear : 32%**
- **Promo2SinceWeek: 50%**
- **Promo2SinceYear: 50%**
- **PromoInterval: 50%**

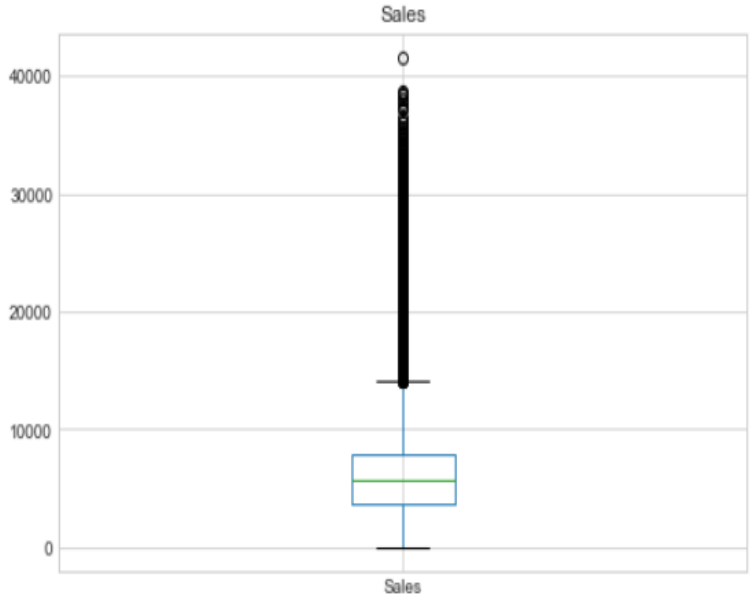
Exploratory Data Analysis:

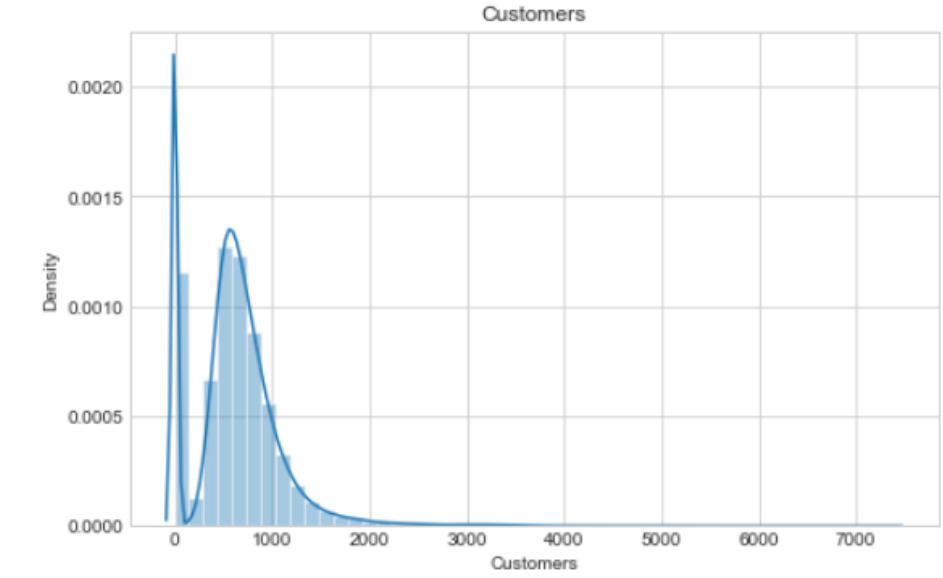
For ease of performing EDA, we divided our features into:

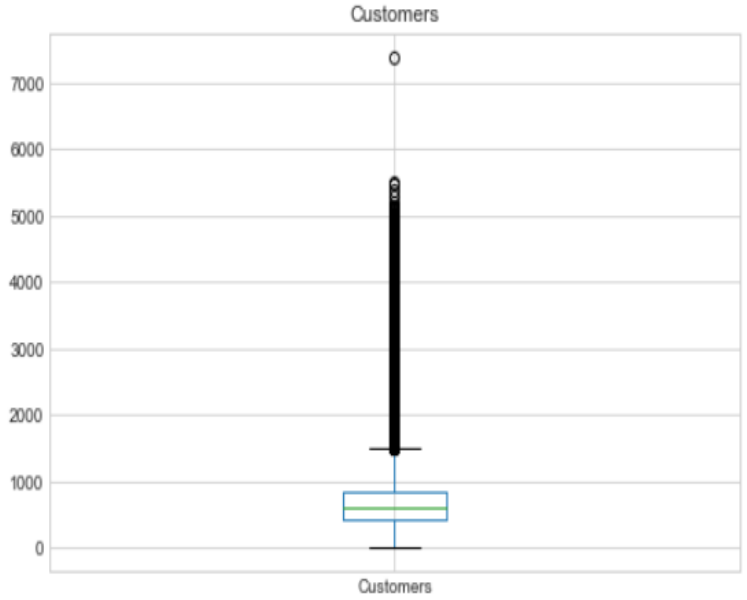
- **Numerical(continuous)**
- **Numerical(Discrete)**
- **Categorical features**
- **Temporal columns(Date/Time columns)**

EDA(Sales and Customers):



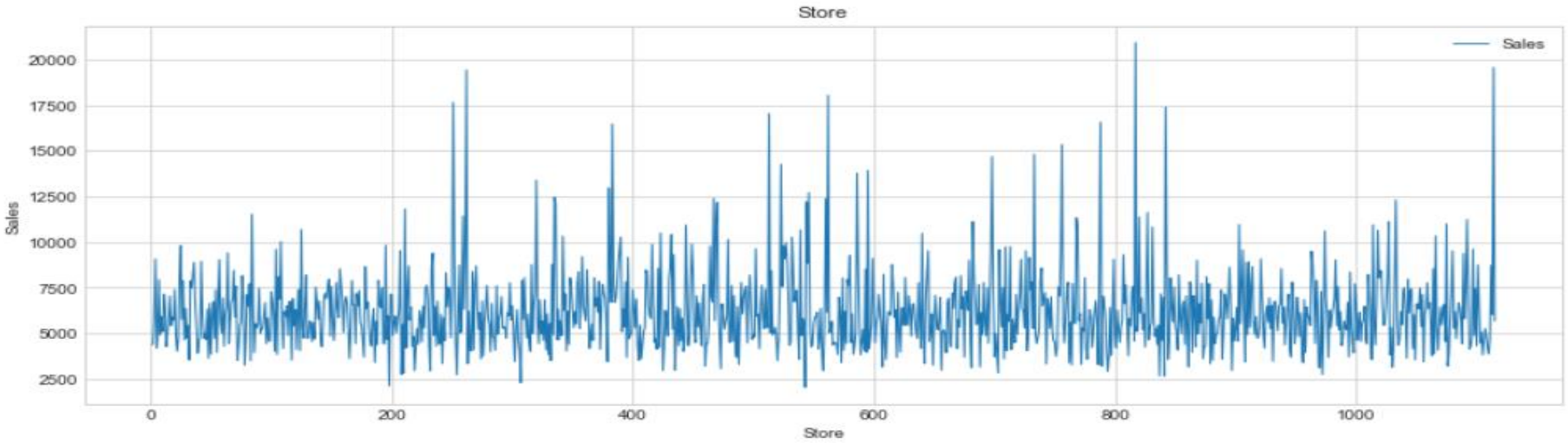


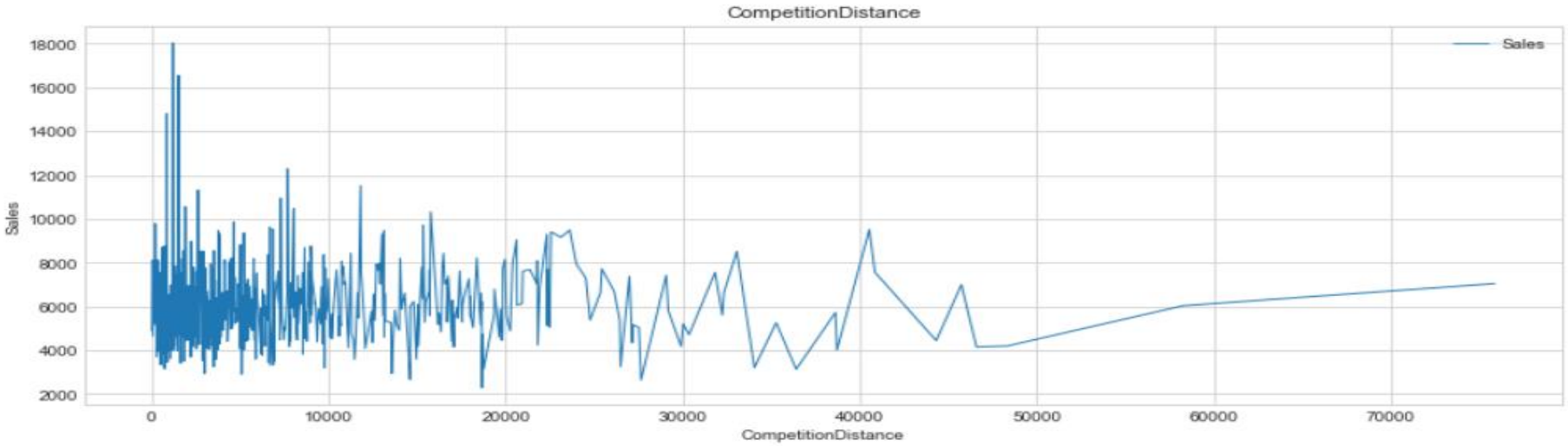




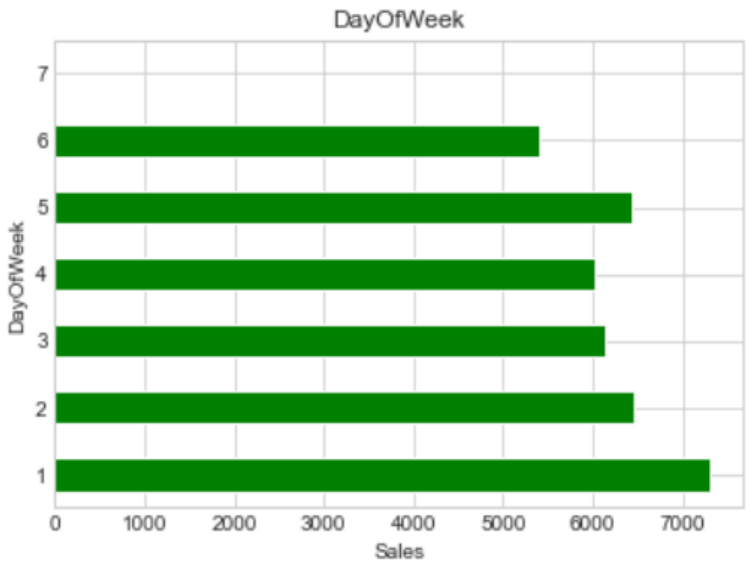
EDA(Contd.) of Discrete Features:

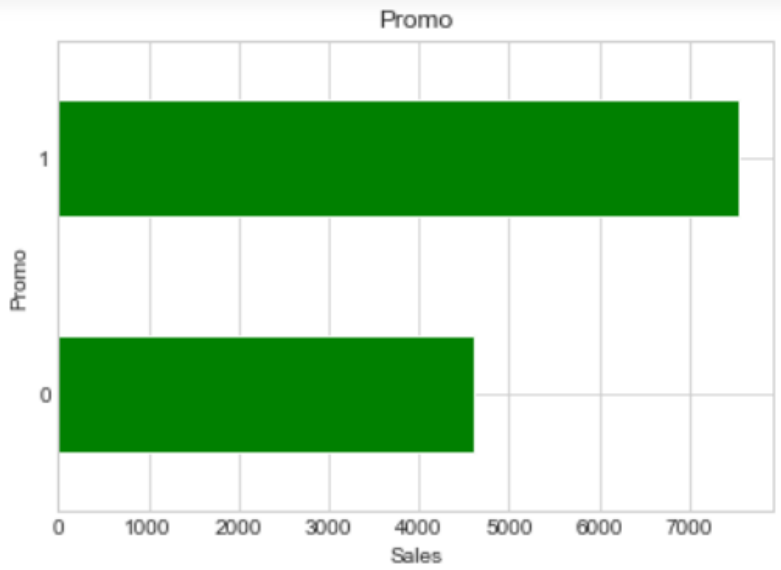
We have considered the features ‘Store’ and ‘Competitor distance’ as Discrete numeric features as we have data on 1115 stores and the distance between each store and their competitor is fixed.

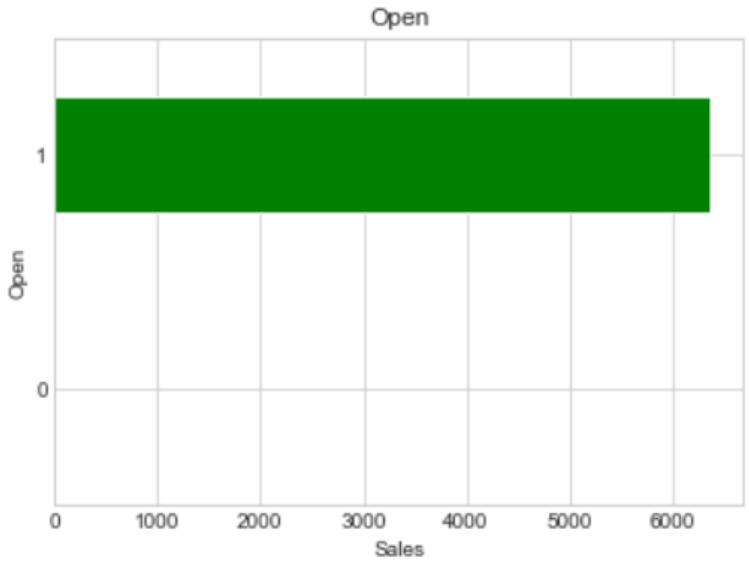


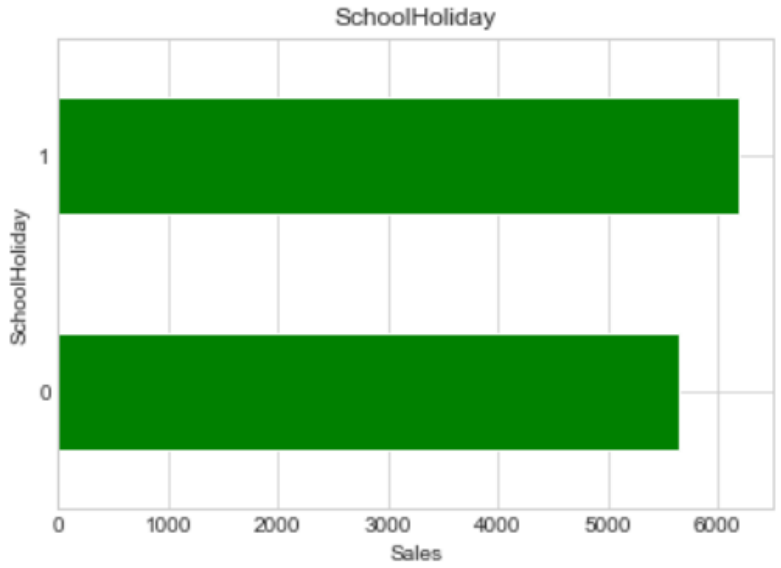


EDA(Contd.) of Discrete Features:

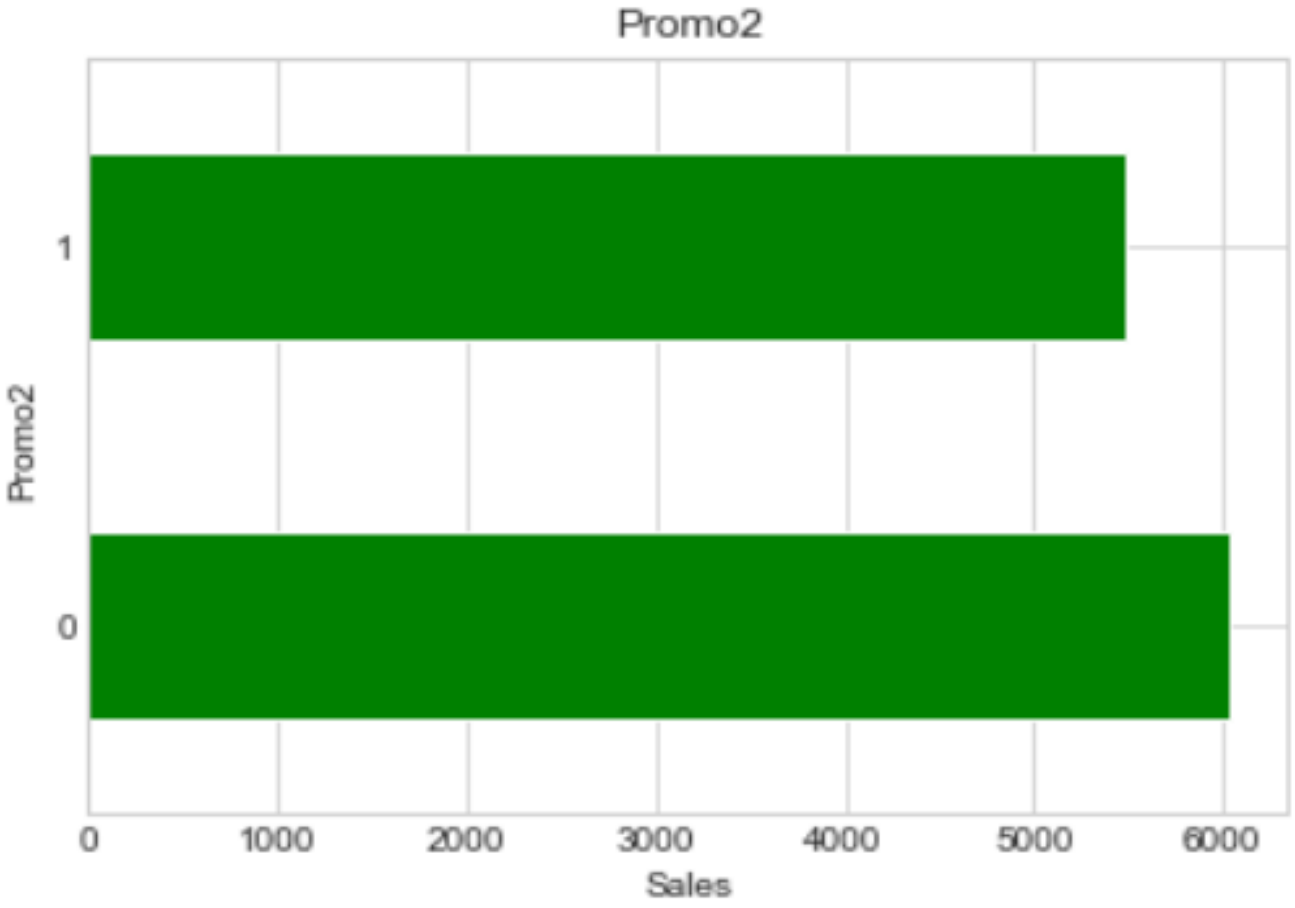




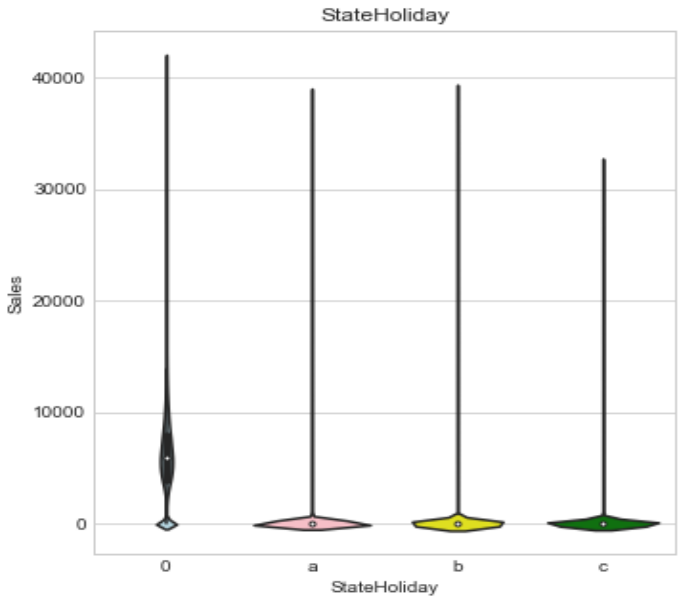


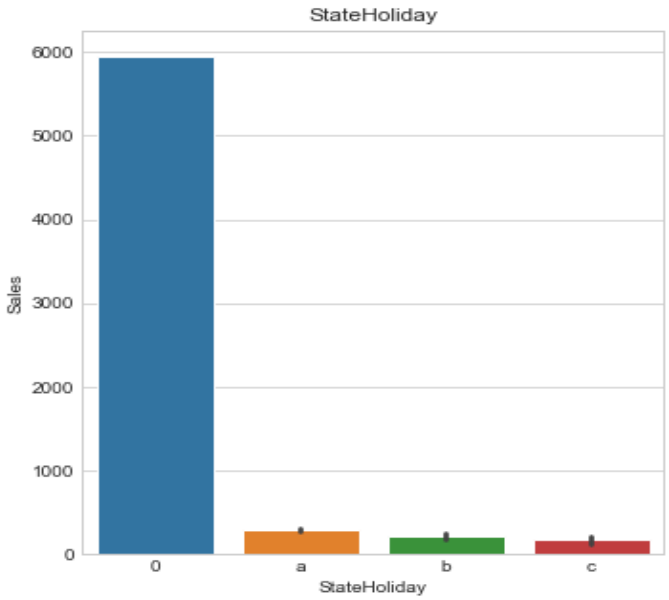


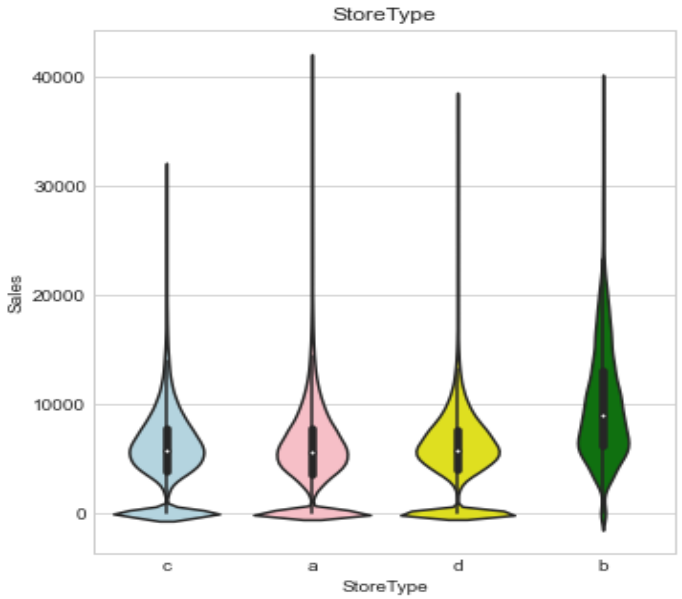
EDA(Contd.) of Discrete Features:

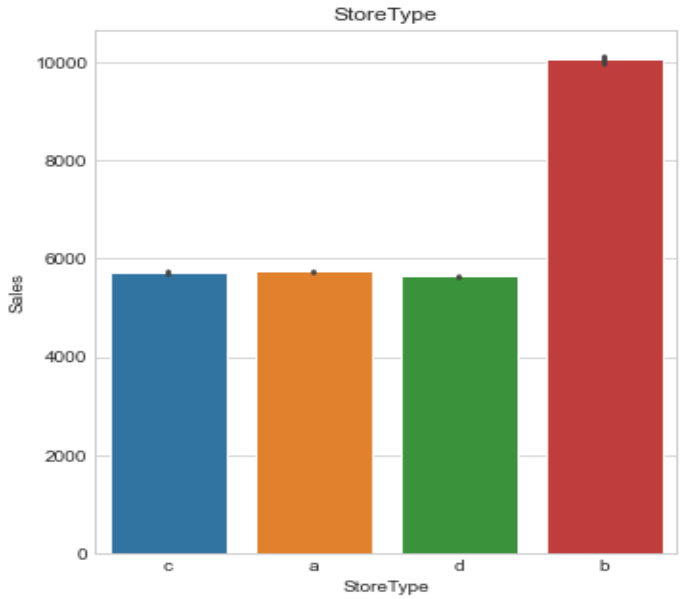


EDA(Contd.) for Categorical Features:





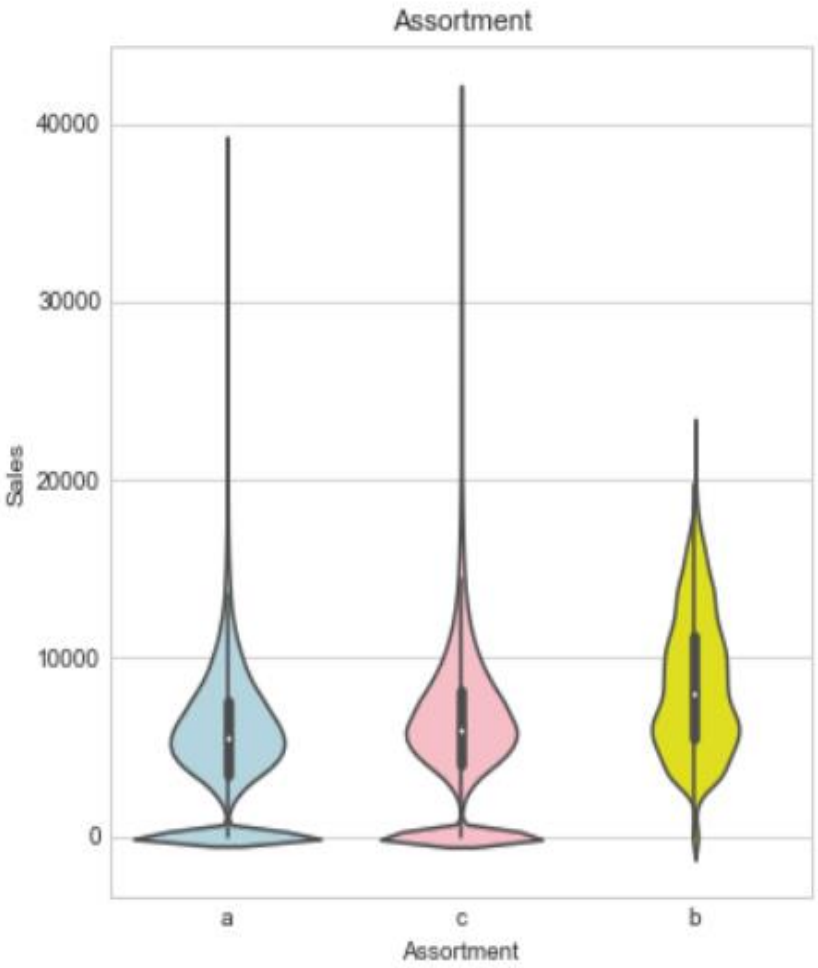


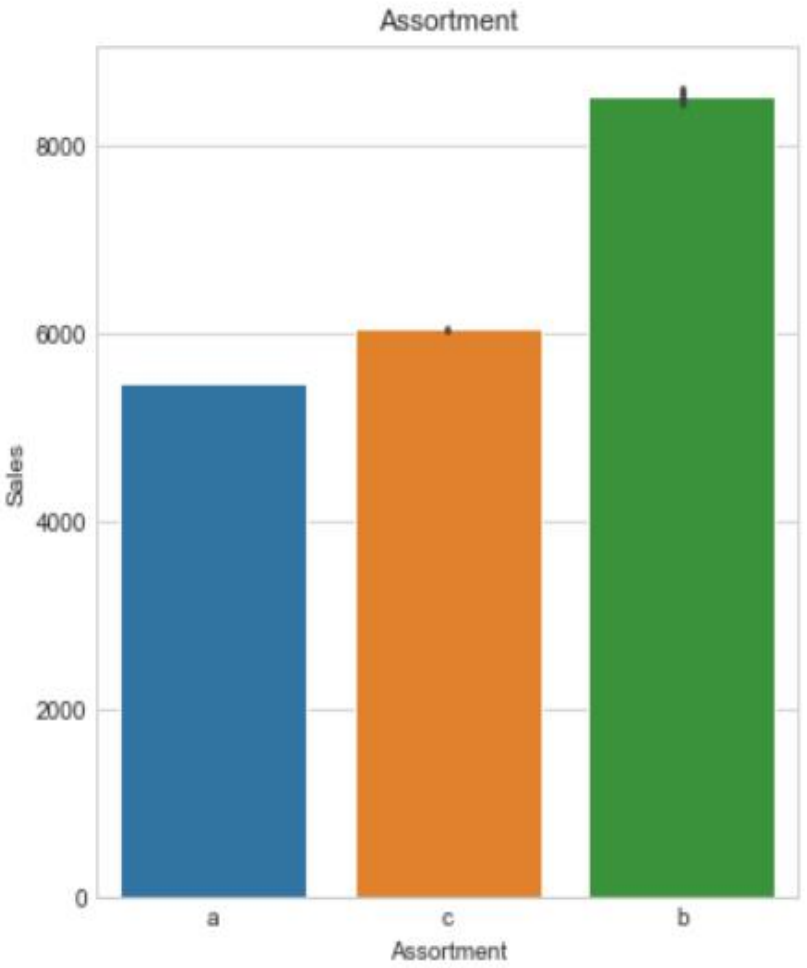


State Holidays:

- **0: No holidays**
- **a: Public Holidays**
- **b: Easter Holidays**
- **c: Christmas**

EDA(Contd.) for Categorical Features:





Assortment levels:

- **a: Basic assortment**
- **b: Extra assortment**
- **c: Extended assortment**

Feature Engineering:

These are the steps we will follow under this section:

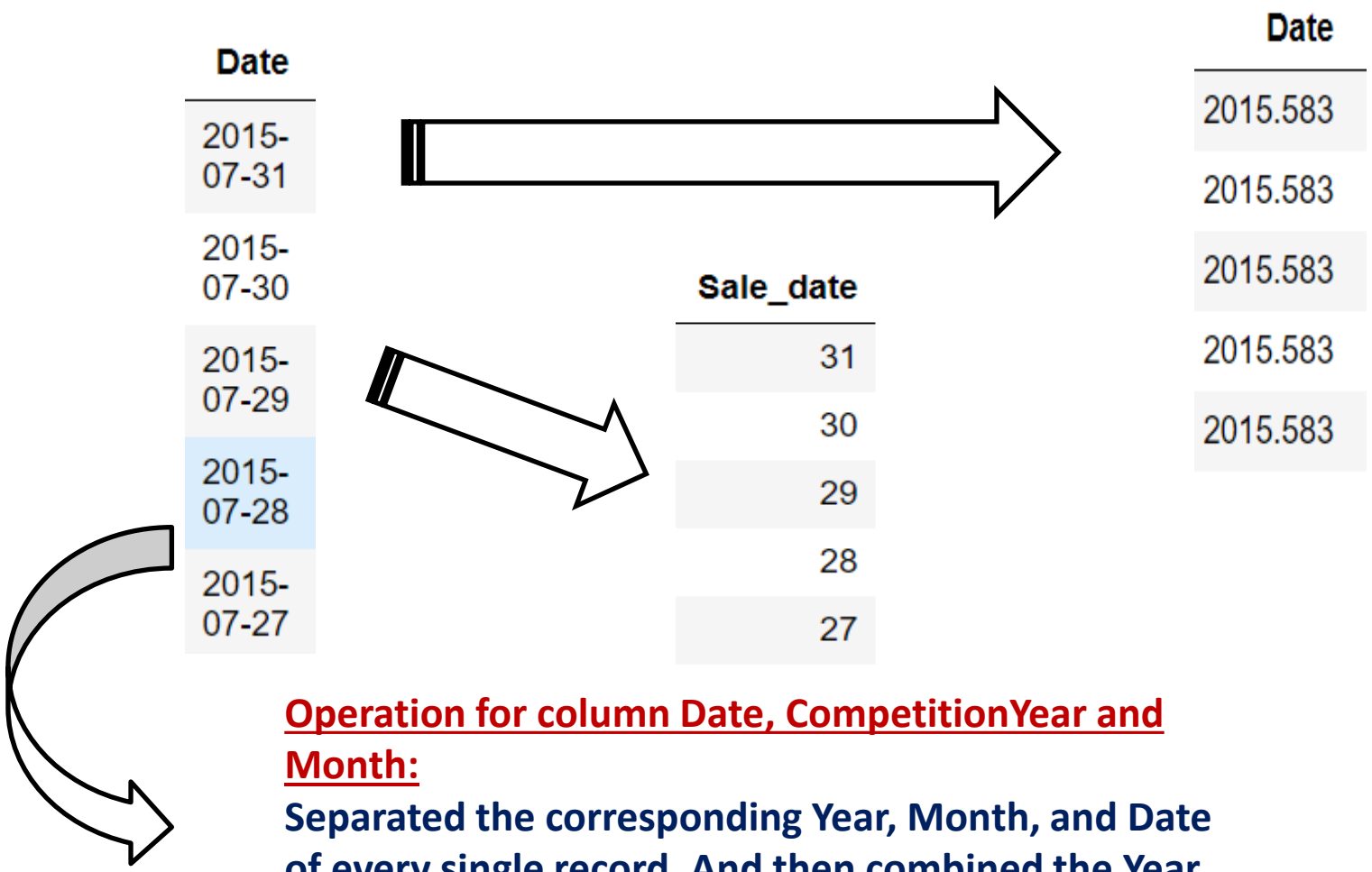
- **Data Cleaning: Will remove the columns having 50 percent of missing values.**
- **We will be making some changes to our temporal columns by trying to fit a mathematical relationship between them.**
- **Will be adjusting our discrete and categorical features. Will convert the categorical features into dummy variables and will convert some of the discrete features into dummy variables and will convert the rest by target label encoding.**
- **We will be working on the dependent variable to nullify the outliers and make it normally distributed.**
- **Outlier-Treatment: We will update the outlier values in the 'Customers' column by their median values.**
- **We will be plotting a heatmap to check the final correlation of the features in our dataset.**

Feature Engr.(Contd.)

We had previously stored our columns associated with Date/Year/Month in a separate list and named it temporal features. The features are as follows:

- Date (YY/MM/DD format – Date of the record)
- CompetitionOpenSinceYear – (Year when its nearest competitor opened their store)
- CompetitionOpenSinceMonth – (Corresponding month when its nearest competitor opened their store)

Feature Engr.(Contd.)



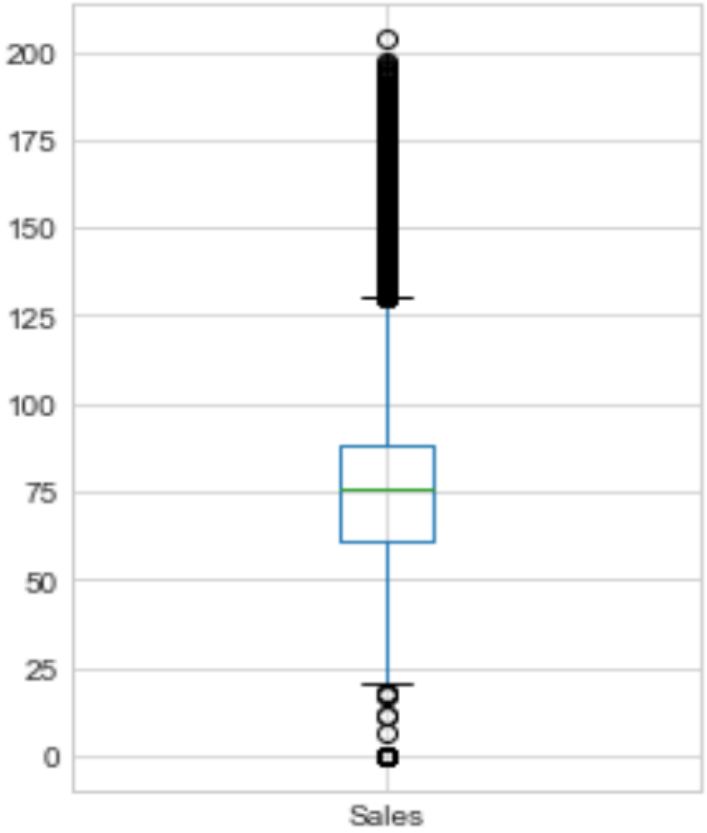
Operation for column Date, CompetitionYear and Month:

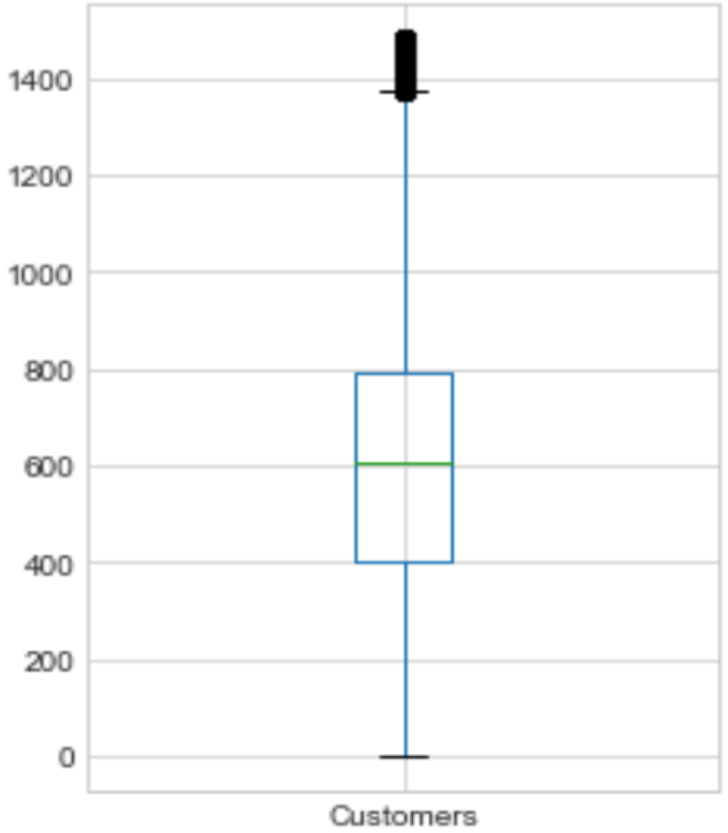
Separated the corresponding Year, Month, and Date of every single record. And then combined the Year and month columns by applying the expression:

Year-Month = Year + (Month/Total months in a year)

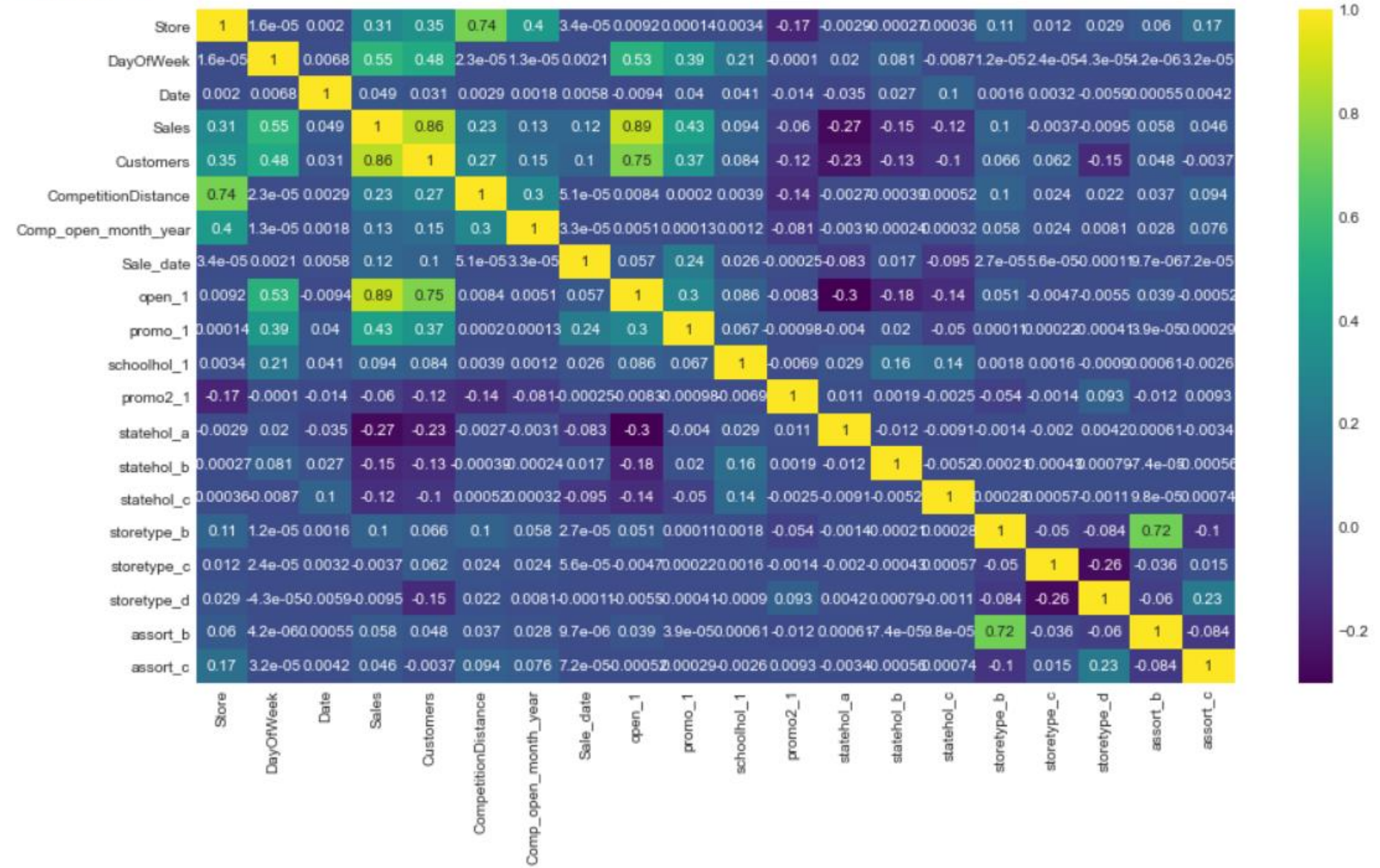
Feature Engr.(Contd.)

- Nullified Outlier Effect from our dependent Variable by applying square root on the entire ‘Sales’ column.
- Nullified Outlier from our ‘Customers’ column by replacing the extreme values with their median values.

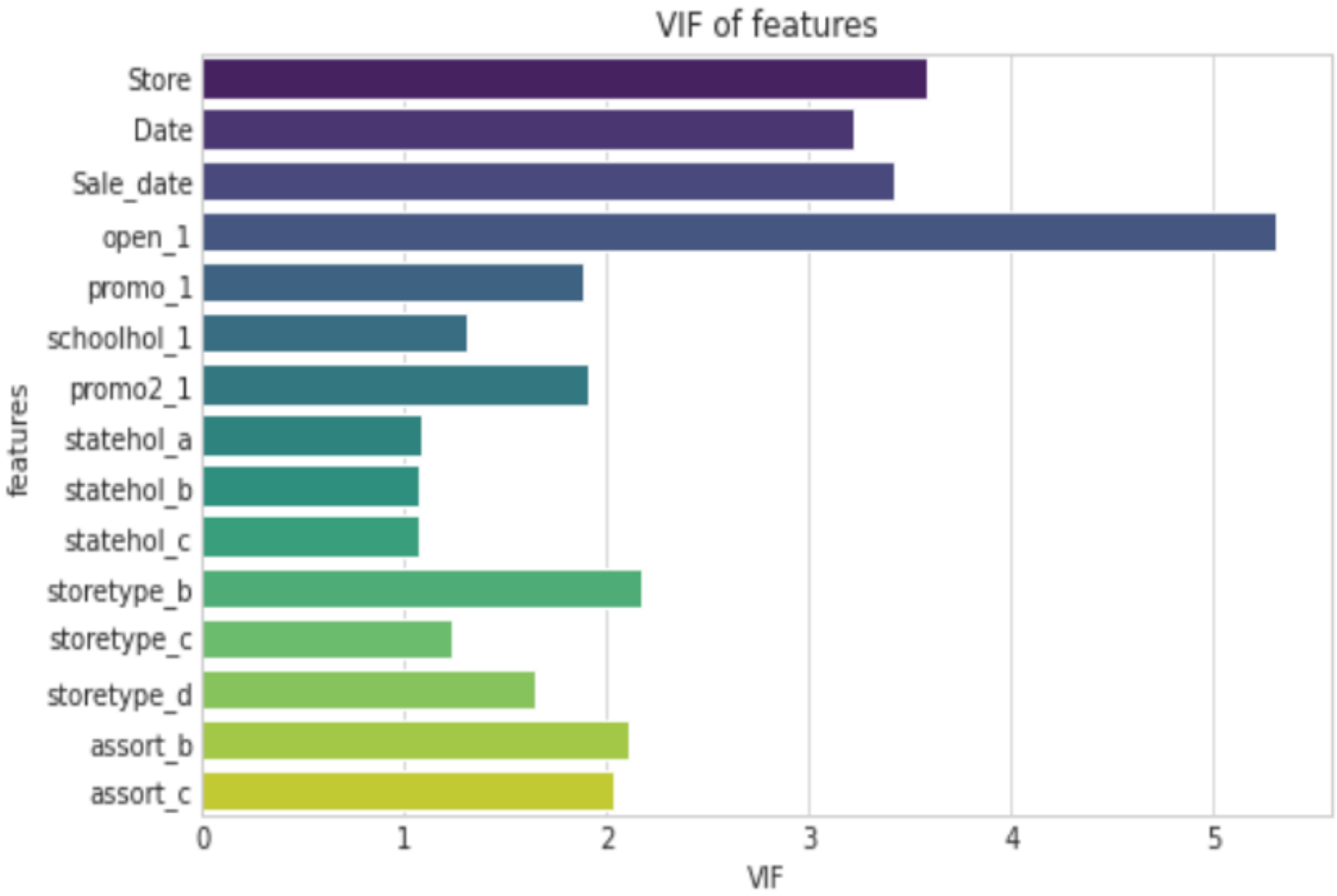




Checking Multicollinearity of the features:



Calculating Variation Inflation factor(VIF):



Conclusion from VIF:

All the features have their VIFs below 5 which says that they are not mut ually correlated with each other.
However, feature open_1 has its VIF a little more than 5, but I have decid ed to keep it, because it has a very good correlation(the highest) with the dependent feature 'Sales'. I haven't yet removed the features having low dependency with 'Sales' as I will later be applying Lasso regularization an d these features will automatically shrink to 0.

Feature Scaling:

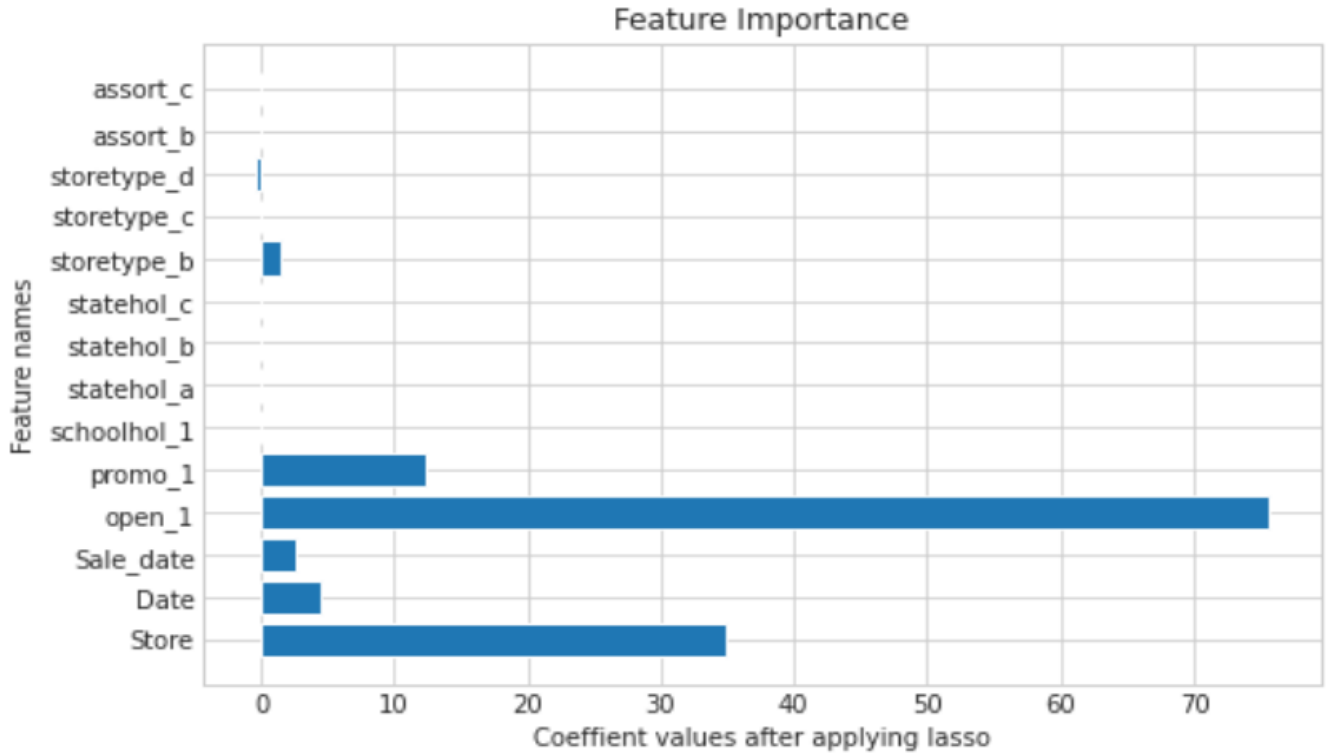
- Models like Linear Regression where it's important to bring all of our features in the same scale to get much better accuracy of our model
- It will also allow our optimizer that is Gradient Descent algorithm to find the global minima faster
- Scaled all the features using the MinMaxScaler to normalize our features, that is bringing them in the range of 0 to 1.

Using Lasso Regularization:

Reasons to use Lasso Regularization:

- Firstly, Lasso will remove all the features which will have very little impact on predicting our 'Sales' column by shrinking their coefficients to 0.
- Secondly, it will prevent the model from overfitting itself by penalising the coefficients of the most important features.
- We have cross-validated using RandomizedsearchCV and the value of alpha was found out using hyper parameter tuning.

Let's have a look at our most important features:



Evaluation metrics and Summary:

OLS Regression Results

Dep. Variable:	Sales	R-squared:	0.914
Model:	OLS	Adj. R-squared:	0.914
Method:	Least Squares	F-statistic:	3.609e+05
Date:	Sat, 24 Apr 2021	Prob (F-statistic):	0.00
Time:	16:49:53	Log-Likelihood:	-7.5906e+05
No. Observations:	203442	AIC:	1.518e+06
Df Residuals:	203435	BIC:	1.518e+06
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-22.9612	0.084	-272.793	0.000	-23.126	-22.796
Store	35.5395	0.078	455.823	0.000	35.387	35.692
Date	5.6940	0.075	75.769	0.000	5.547	5.841
Sale_date	3.5118	0.077	45.364	0.000	3.360	3.664
open_1	76.3458	0.063	1220.582	0.000	76.223	76.468
promo_1	12.5502	0.050	253.159	0.000	12.453	12.647
storetype_b	7.9179	0.182	43.455	0.000	7.561	8.275

Omnibus:	35492.239	Durbin-Watson:	1.996
Prob(Omnibus):	0.000	Jarque-Bera (JB):	139664.426
Skew:	0.833	Prob(JB):	0.00
Kurtosis:	6.702	Cond. No.	13.3

```
MAE(train set): 1036.419569135293
MAE(test set): 1032.5077789341567
RMSE(train set): 1763.8357641161526
RMSE(test set): 1757.8775984368424
r2 score(train): 0.9131466382361707
r2 score(test): 0.9133911748681841
r2_adjusted score: 0.9133852143833328
```

Conclusion:

- With a R-squared score of 0.91 the model will do well on future predictions of the ‘Sales’ column.
- The p values and the t statistic values tells us that our model is statistically significant and we can also reject the null hypothesis claiming our features were not correlated with the dependent feature ‘Sales’.
- The standard error is also minimal, which says our model has predicted the coefficients of the features with the utmost precision and accuracy.
- The metrics on both the train and test dataset are almost equal, meaning our model is not overfitting, we can say our model is optimised.

Plans to Improve

- Firstly, I have fitted the model using Linear Regression which is the simplest of ML algorithms.
- ML algorithms like Ensemble techniques such as RandomForest and Xgboost could have provided us with much better accuracy and these algorithms could also work with missing values and outliers.
- However our dataset had more than 1 million records and due to huge time complexity of ensemble algorithms, it would be computationally much more expensive than a simple Linear Regression Model.