Retail Sales Prediction

Midhun R
Data Science Trainee,
AlmaBetter, Bangalore

Abstract:

Every business needs to keep up with their customer demands, or they risk losing loyal customers. And predicting demand and future sales accurately is important to financial performance.

Sales forecasting is the technique of predicting the sales or revenue your company will generate over a period of time in the future, allowing them to create effective and smart business plans. It's crucial that these projections be accurate since important decisions are driven by the revenue generated by the company.

In this project, I have attempted to analyze the retail sales dataset of Rossmann stores and build a predictive model to forecast the sales over the next 6 weeks. No personal information of customer is provided in this dataset.

Keywords: EDA, Feature Engineering, Modelling, Regression, Decision Tree, Random Forest, XGBoost

01. Problem Statement

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.

Two datasets are given: one with store data and the other with historical sales data of 1115 stores from January 2013 to July 2015.

The main objective is to understand the existing data and identify the key factors that will predict future sales, so that a predictive model can be built for making forecasts about future sales.

02. Introduction

In this day and age, when data science has transformed into an invisible force that influences the decisions of the modern world, retail trade sector has also become beneficiary of data science.

Nowadays, having a proper understanding of data and connecting all data sources effectively is paramount in generating competitive advantage, providing superior customer value, and ultimately orientating the future of any business. Over the last few years, companies have begun to deploy predictive analytics to better anticipate and meet customer needs and preferences.

From a company standpoint, these sales predictions are done on a constant basis to enhance their sales forecasting models since they have a direct influence on their decision-making process, objectives, plans, and growth strategies. Research shows that companies with accurate sales forecasts are 10% more likely to grow their revenue year-over-year and 7.3% more likely to hit their quota.

The goal of this project is to inspect, clean and analyze the given dataset containing sales history of 1115 drug stores to find out which factors affect the sales and customers in stores and how they will affect it. This information is used to build predictive models and compare them to find out the best model to forecast the sales generated by the stores in the next six weeks.

03. Approach

The sequence of steps taken to solve the task are as follows:

1. Understanding the business task.
2. Import relevant libraries and define useful functions.
3. Reading data from files given.
4. Data pre-processing, which involves inspection of both datasets and data cleaning.
5. Exploratory data analysis, to find which factors affect sales and how they affect it.
6. Feature engineering, to prepare data for modelling.
7. Modelling data and comparing the models to find out most suitable one for forecasting.
8. Conclusion and recommendations to boost sales.

04. Business Task

Build a machine learning model to forecast the sales of all Rossmann stores upto 6 weeks. This is undertaken as an individual project.

05. Import Libraries and Define Function

Some relevant libraries imported for aid are:

1. NumPy, for numerical operations
2. Pandas, for data manipulation
3. Matplotlib and Seaborn, for data visualization.
4. Math, for mathematical operations.
5. Random, to generate random values.
6. Statsmodels, for statistical data exploration.
7. Scikit Learn, for machine learning.
8. XGBoost

In addition to this, few useful functions were defined to avoid repetition of codes.

06. Reading Data

After drive was mounted, data from csv files were read and stored in pandas dataframes.

07. Data Inspection

After loading the datasets and importing relevant libraries, the datasets have been explored thoroughly by looking into its head, tail, brief summary, number of records and features, etc.

Store dataset contains 1115 records across 10 features with information about each store and sales dataset contains 1017209 records across 9 features with information regarding sales in stores from January 2013 to July 2015. The common feature among them is store ID. None of the datasets have duplicate rows but six columns in store dataset have missing values. Some columns require conversion of datatypes. Additionally, new columns need to be added from existing ones to make analysis easier.

Then merging of two datasets also needs to be done.

The features in store dataset were identified as:

1. Store: a unique Id for each store.
2. StoreType: differentiates between 4 different store models: a, b, c, d.
3. Assortment: describes an assortment level: a = basic, b = extra, c = extended.
4. CompetitionDistance: distance in meters to the nearest competitor store.
5. CompetitionOpenSinceMonth: gives the approximate month of the time the nearest competitor was opened.
6. CompetitionOpenSinceYear: gives the approximate year of the time the nearest competitor was opened.
7. Promo2: Promo2 is a continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating.
8. Promo2SinceWeek: describes the calendar week when the store started participating in Promo2.
9. Promo2SinceYear: describes the year when the store started participating in Promo2.
10. Promo2Interval: describes the consecutive intervals Promo2 is started, naming the months the promotion is started anew. Eg. "Feb,May,Aug,Nov" means each round starts in February, May, August, November of any given year for that store.

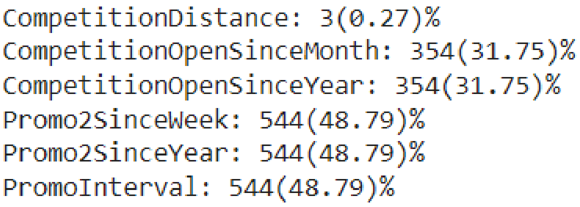
The features in store dataset were identified as:

1. Store: a unique Id for each store.
2. DayOfWeek: day of week of sale.
3. Date: date of sale
4. Sales: the turnover for any given day.
5. Customers: the number of customers on a given day.
6. Open: an indicator for whether the store was open: 0 = closed, 1 = open.
7. Promo: indicates whether a store is running a promo on that day.
8. 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.
9. SchoolHoliday: indicates if the (Store, Date) was affected by the closure of public schools.

08. Data Cleaning

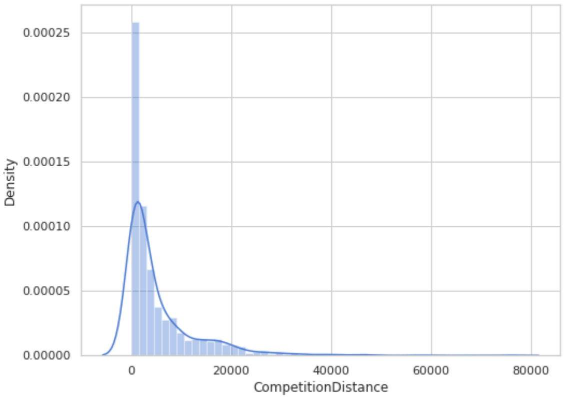
Data cleaning is done to ensure that the dataset is correct, consistent, and usable.

Six out of 10 columns in store data have missing values in them. They must be handled properly to get an accurate result.



```
CompetitionDistance: 3(0.27)%
CompetitionOpenSinceMonth: 354(31.75)%
CompetitionOpenSinceYear: 354(31.75)%
Promo2SinceWeek: 544(48.79)%
Promo2SinceYear: 544(48.79)%
PromoInterval: 544(48.79)%
```

‘CompetitionDistance’ had the least number of missing values. Since it is a numerical feature, distribution of its values was checked using a density plot.



The distribution of CompetitionDistance is positively skewed. So it is safe to impute missing values with mode or median. Since the PDF of the peak of the distribution is very low, it is better to go with median instead of mode.Both 'CompetitionOpenSinceMonth' and 'CompetitionOpenSinceYear' have the same number of missing values. After checking, it was found that both the features have missing values in the same rows. It was also checked whether the null values in both these features relate to 'CompetitionDistance'. Since no useful information was obtained from the above process, missing values were imputed with the mode of data.

'Promo2SinceWeek', 'Promo2SinceYear' and 'PromoInterval' have the same number of missing values. These features have missing values in the same rows, where the value of 'Promo2' is zero. So, the missing values were imputed with 0.

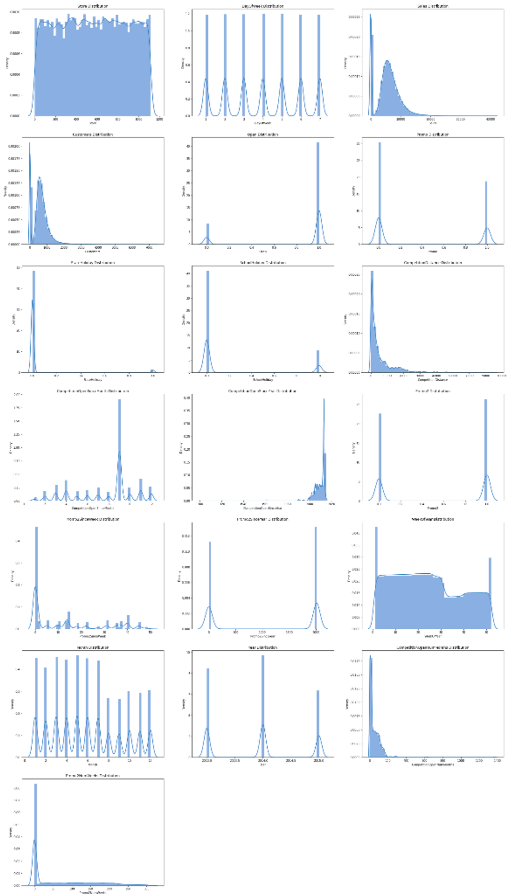
After the merging of two datasets, some columns were converted to appropriate datatypes to make analysis easier and more accurate. These columns were 'Date' (datetime), 'CompetitionOpenSinceMonth', 'CompetitionOpenSinceYear', 'Promo2SinceWeek', 'Promo2SinceYear' (int). Characters denoting different types of state holidays in 'StateHoliday' was converted to 1, thus converting it to another binary feature.

Some extra columns were also added which will be useful during analysis. These are 'WeekOfYear', 'Month', 'Year', 'CompetitionOpenNumMonths' and 'Promo2NumWeeks'.

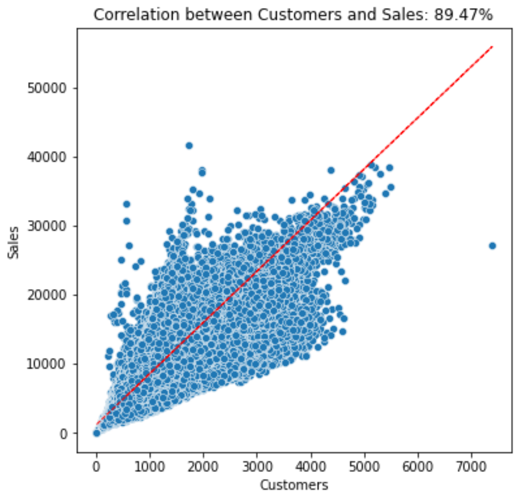
09. Exploratory Data Analysis

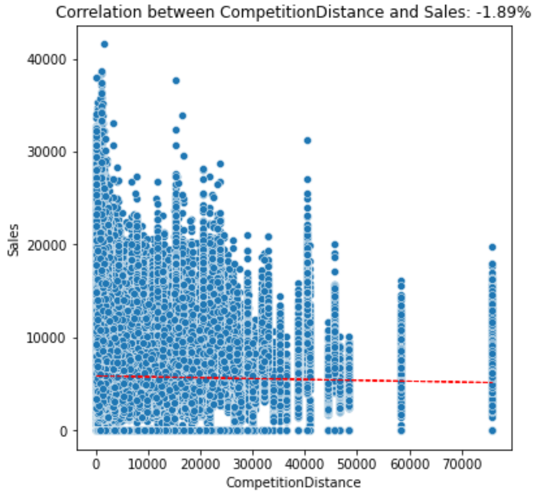
Dataset, after cleaning, is subjected to exploratory data analysis, which will visualize the data and identify trends and patterns that can be later used to increase revenue.

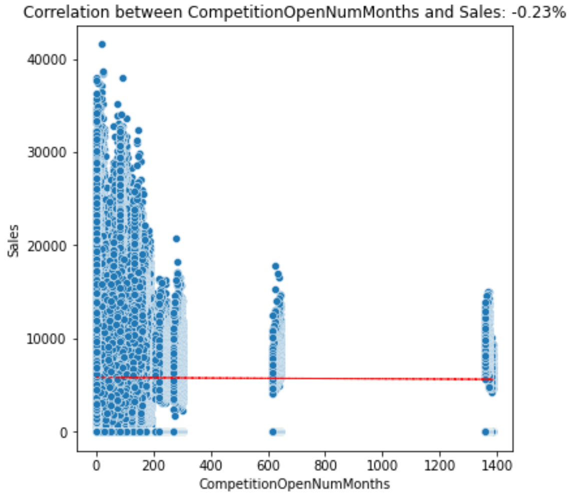
The following observations were made after EDA:

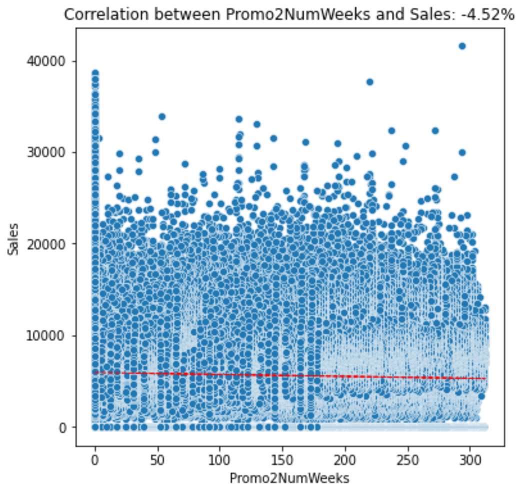


Distribution of sales, customer, competition distance, competition open number of months and promo 2 number of weeks are positively skewed. Everything else are categorial features.



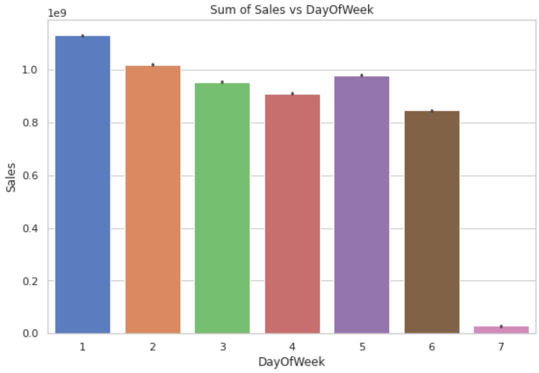


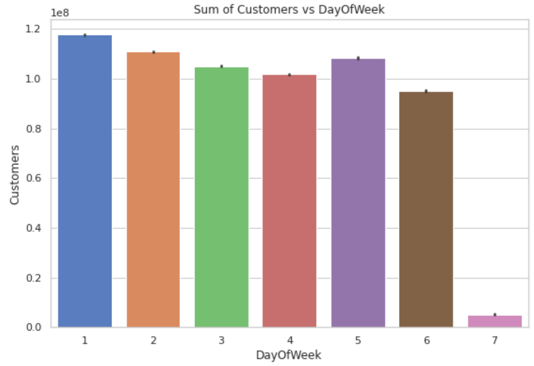




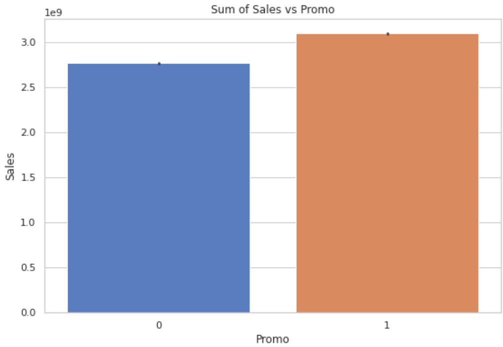
Sales decreases with increase in competition distance, competition open number of months and promo 2 number of weeks but they have very low correlation with competition distance.

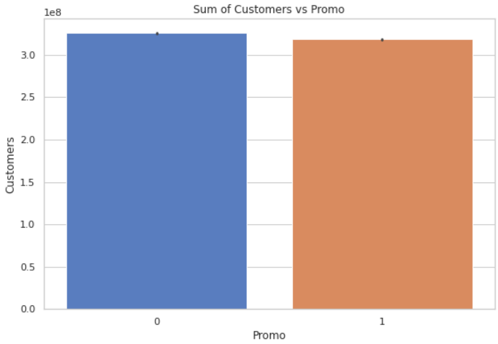
Total number of sales and customers were plotted for all categorical features using bar graphs.



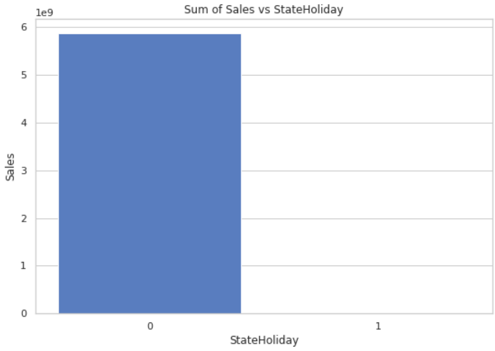


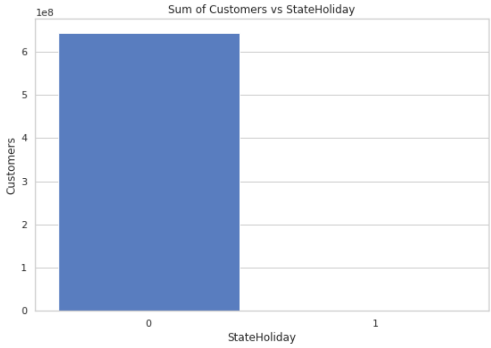
Highest sales are recorded on Mondays and lowest sales are recorded on Sundays. This may be because most of the shops are closed on Sundays and this leads to higher demand on the next day, which is Monday.

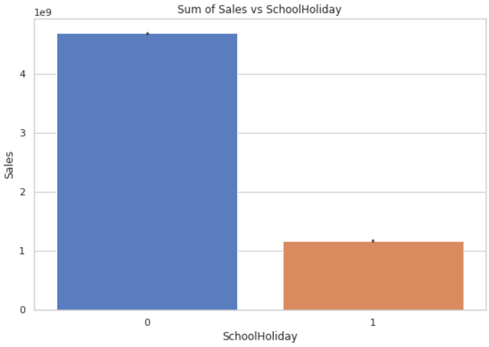


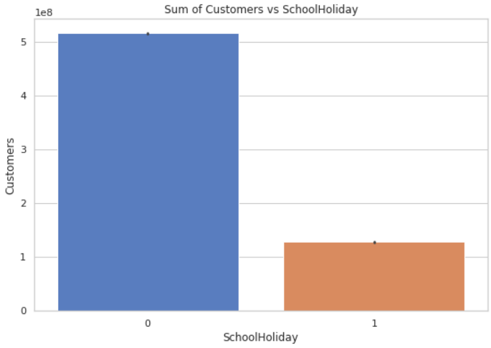


Presence of promos increases sales, but it doesn't help much in generating new customers.

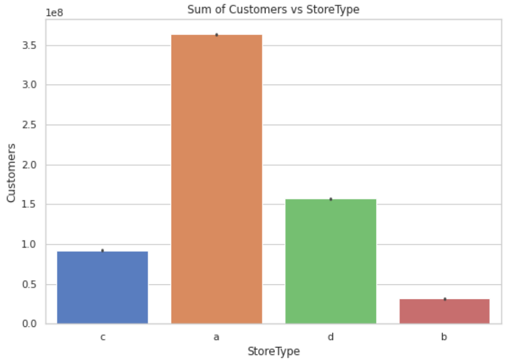


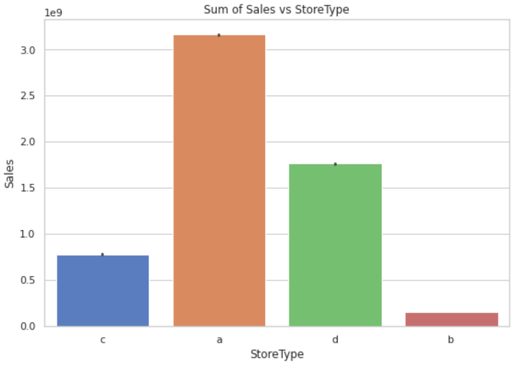




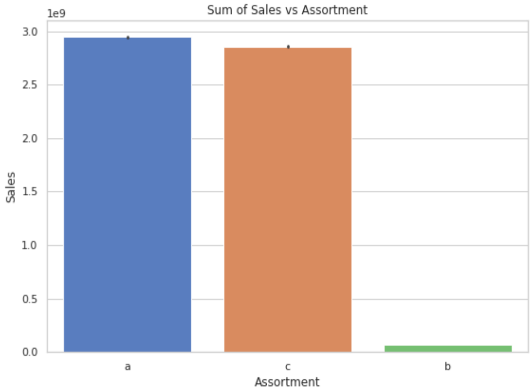


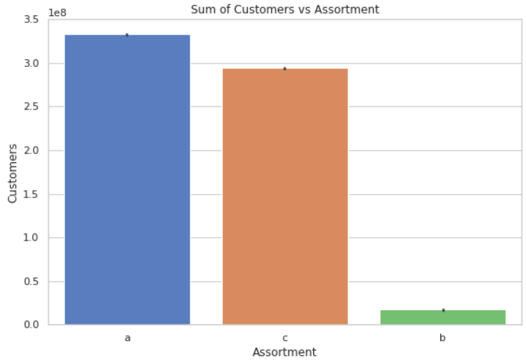
Shops are closed on all state holidays but sometimes some shops are opened during school holidays, which may be Saturdays. Sales are higher on school holidays than other days.



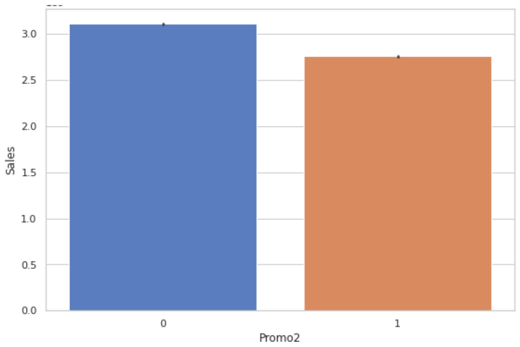


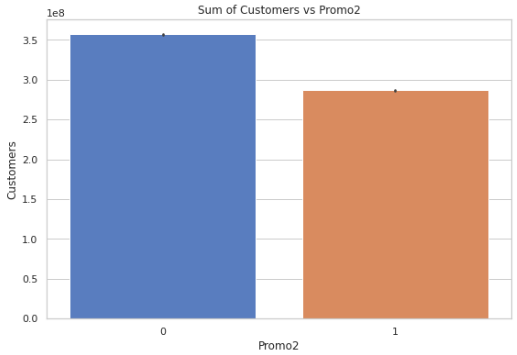
Store type a record the most amount of sales and has more customers, mostly because the majority of the shops are type a.



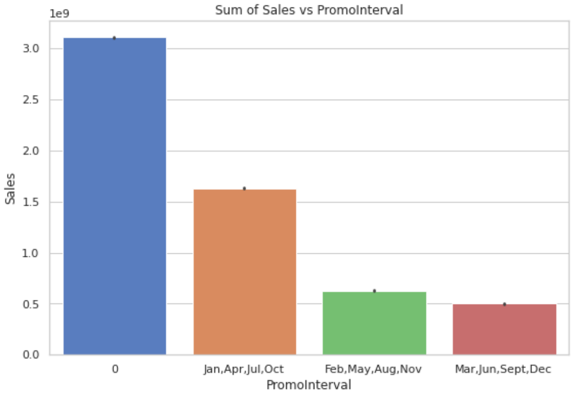


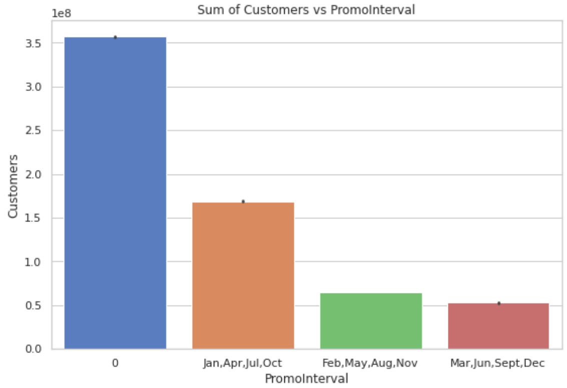
Sales of assortment b is very rare, and it has least customers.



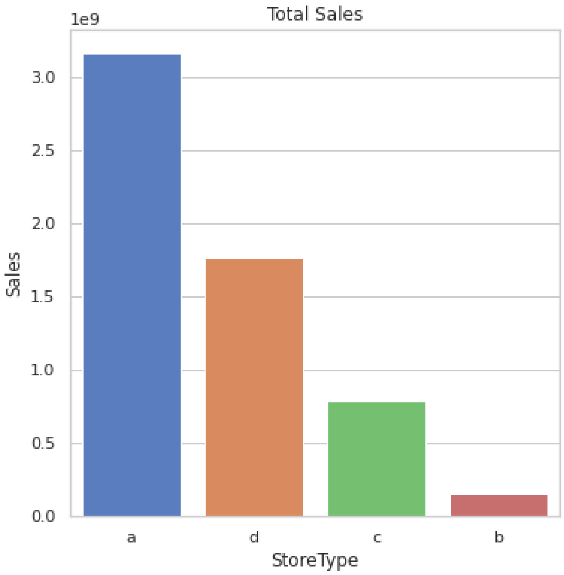


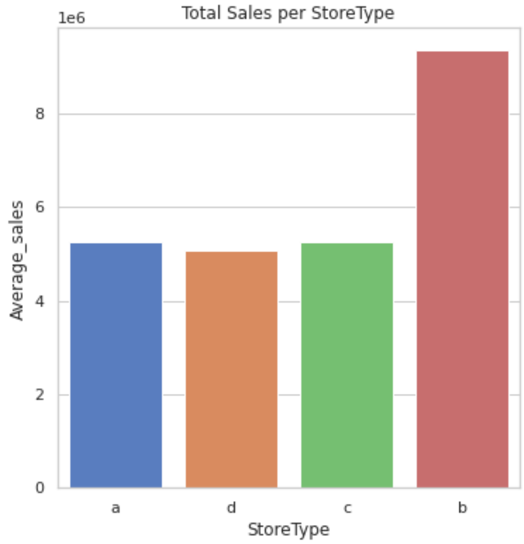
Participants of promo 2 has less sales and customers, might be because it is seasonal.

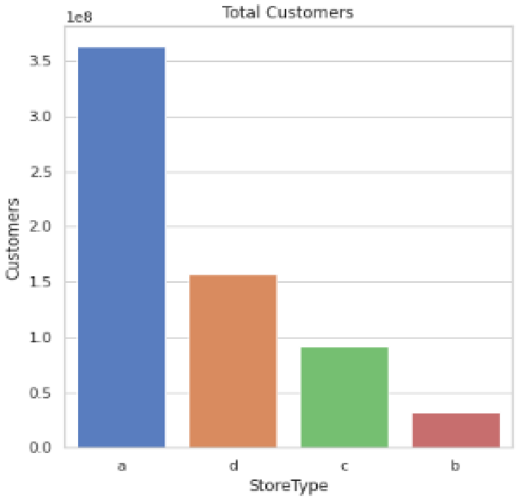


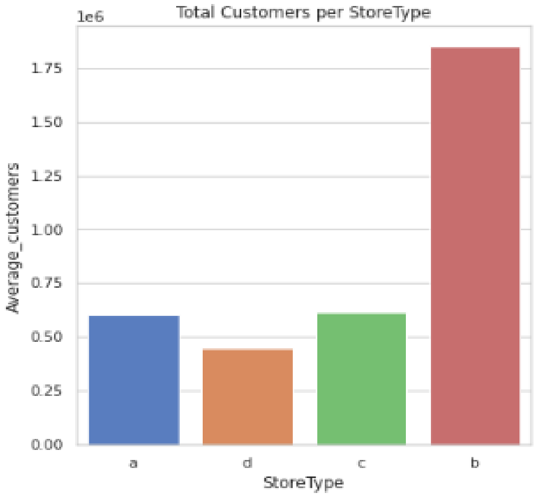


Promo interval Jan, Apr, Jul, Oct brings in more sales and customers.

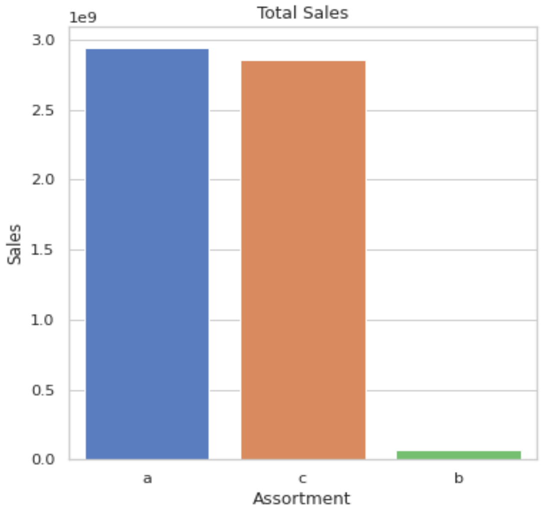


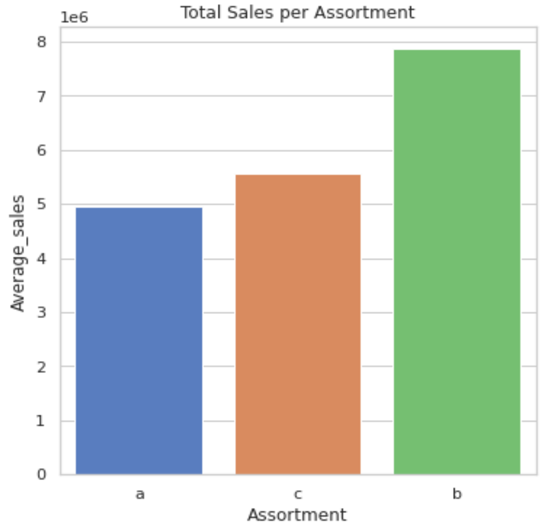


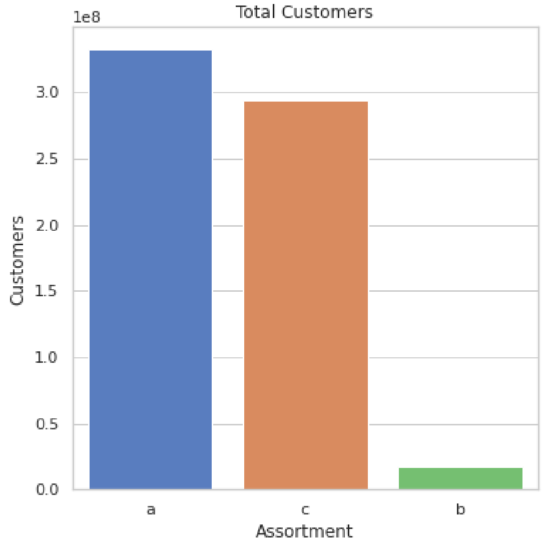


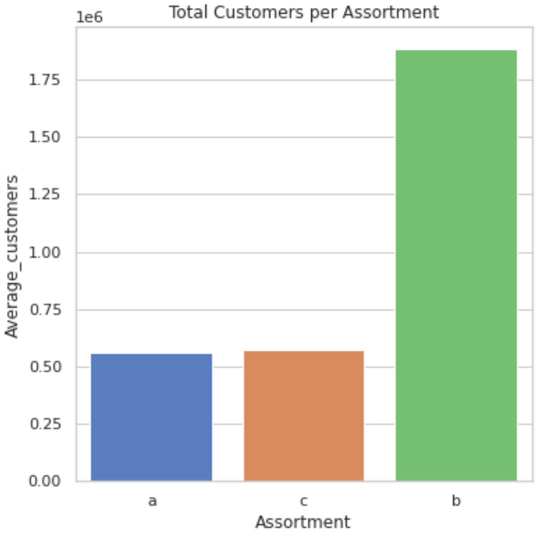


Even though the volume of sales and customers is low, store type b has the highest average sales and customers. This means store type b is more preferred by customers.

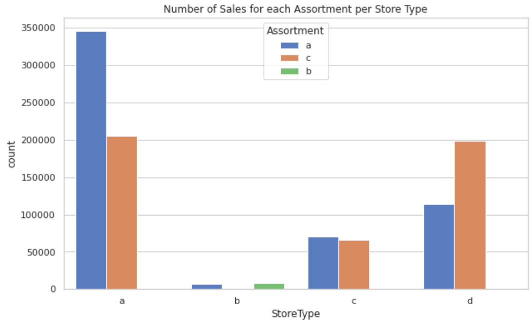




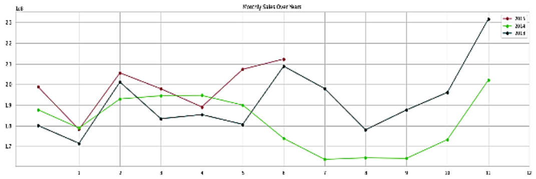


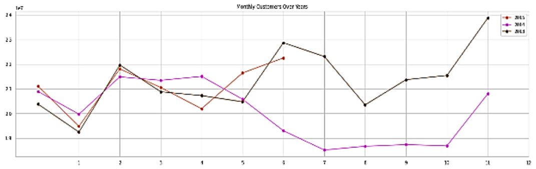


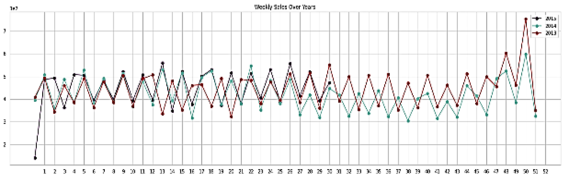
Even though the volume of sales and customers is low, assortment b has the highest average sales and customers. This means assortment b is in high demand.

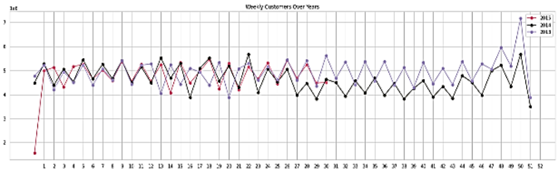


Only store type b sells assortment b, and it has higher sales than assortment a. Assortment c is not sold in store type b. Assortment a has more sales than c in store type a and d. Assortment c has more sales than a in store type c.

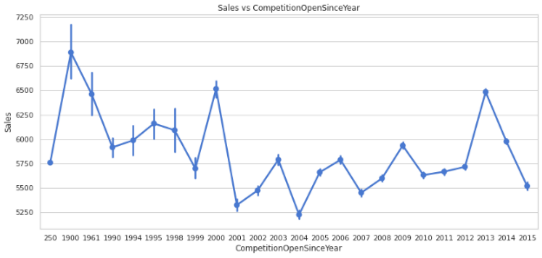


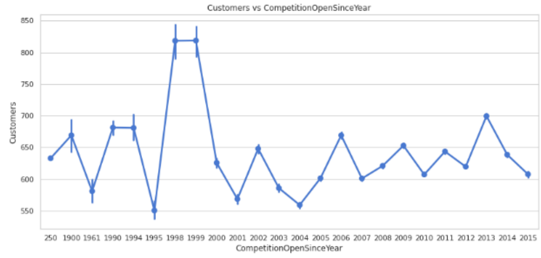


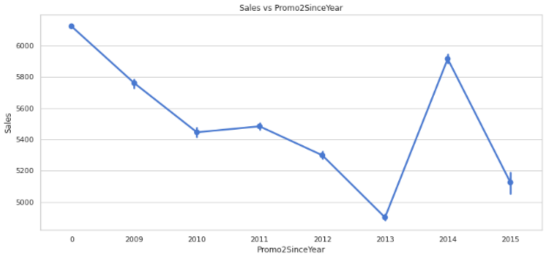


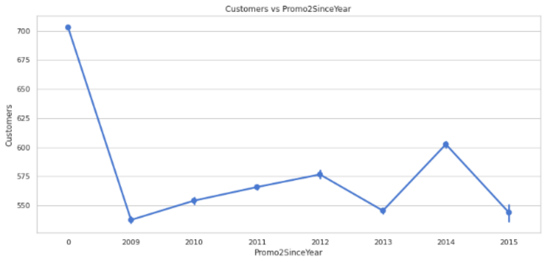


Weekly Sales and Customers are showing almost similar trends. Both peaks at mid-December. This may be because people buy drugs in advance just before the shops close for the holiday season.



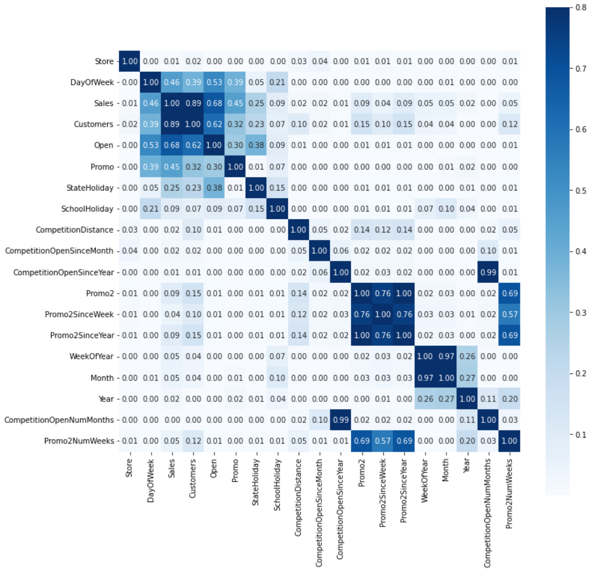






Sales peaked in 1900 when a few stores were present so there was only a little competition. As years passed sales started to decline due to the increase in competition. But number of customers has peaked during late 1990s. Sales due to promo 2 is generally decreasing over the years, only exception in 2014.

Correlation analysis of the dataset was carried out using a correlation heatmap with all numerical features.



	Store	DayOfWeek	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2	Promo2SinceWeek	Promo2SinceYear	WeekOfYear	Month	Year	CompetitionOpenNumMonths	Promo2NumWeeks
Store	1.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.03	0.04	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01
DayOfWeek	0.00	1.00	0.46	0.39	0.53	0.39	0.05	0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
Sales	0.01	0.46	1.00	0.89	0.68	0.45	0.25	0.09	0.02	0.02	0.01	0.09	0.04	0.09	0.05	0.05	0.02	0.00	0.05
Customers	0.02	0.39	0.89	1.00	0.62	0.32	0.23	0.07	0.10	0.02	0.01	0.15	0.10	0.15	0.04	0.04	0.00	0.00	0.12
Open	0.00	0.53	0.68	0.62	1.00	0.30	0.38	0.09	0.01	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01
Promo	0.00	0.39	0.45	0.32	0.30	1.00	0.01	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00
StateHoliday	0.00	0.05	0.25	0.23	0.38	0.01	1.00	0.15	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.01	0.00	0.01
SchoolHoliday	0.00	0.21	0.09	0.07	0.09	0.07	0.15	1.00	0.00	0.00	0.00	0.01	0.01	0.01	0.07	0.10	0.04	0.00	0.01
CompetitionDistance	0.03	0.00	0.02	0.10	0.01	0.00	0.00	0.00	1.00	0.05	0.02	0.14	0.12	0.14	0.00	0.00	0.00	0.02	0.05
CompetitionOpenSinceMonth	0.04	0.00	0.02	0.02	0.00	0.00	0.00	0.00	0.05	1.00	0.06	0.02	0.02	0.02	0.00	0.00	0.00	0.10	0.01
CompetitionOpenSinceYear	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.02	0.06	1.00	0.02	0.03	0.02	0.00	0.00	0.00	0.99	0.01
Promo2	0.01	0.00	0.09	0.15	0.01	0.00	0.01	0.01	0.14	0.02	0.02	1.00	0.76	1.00	0.02	0.03	0.00	0.02	0.69
Promo2SinceWeek	0.01	0.00	0.04	0.10	0.01	0.00	0.01	0.01	0.12	0.02	0.03	0.76	1.00	0.76	0.03	0.03	0.01	0.02	0.57
Promo2SinceYear	0.01	0.00	0.09	0.15	0.01	0.00	0.01	0.01	0.14	0.02	0.02	1.00	0.76	1.00	0.02	0.03	0.00	0.02	0.69
WeekOfYear	0.00	0.00	0.05	0.04	0.00	0.00	0.00	0.07	0.00	0.00	0.00	0.02	0.03	0.02	1.00	0.97	0.26	0.00	0.00
Month	0.00	0.01	0.05	0.04	0.00	0.01	0.00	0.10	0.00	0.00	0.00	0.03	0.03	0.03	0.97	1.00	0.27	0.00	0.00
Year	0.00	0.00	0.02	0.00	0.00	0.02	0.01	0.04	0.00	0.00	0.00	0.00	0.01	0.00	0.26	0.27	1.00	0.11	0.20
CompetitionOpenNumMonths	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.10	0.99	0.02	0.02	0.02	0.00	0.00	0.11	1.00	0.03
Promo2NumWeeks	0.01	0.00	0.05	0.12	0.01	0.00	0.01	0.01	0.05	0.01	0.01	0.69	0.57	0.69	0.00	0.00	0.20	0.03	1.00

‘Customers’ and ‘Sales’ are highly correlated because an increase in customers means that there must be an increase in sales too.

We can see that ‘WeekOfYear’ and ‘Month’ are also highly correlated, which is obvious since they represent points in the same timeframe but with different range.

‘Open’ is moderately correlated with ‘Sales’ and ‘Customers’ because customers, who drive sales, can access shops only when the shops are open.

‘Promo2’ is correlated with ‘Promo2SinceWeek’ and ‘Promo2SinceYear’ because Promo2 is a binary feature and ‘Promo2SinceWeek’ and ‘Promo2SinceYear’ have the value of 1 only when promo2 is 1.

‘CompetitionOpenNumMonths’ is highly correlated with ‘CompetitionOpenSinceYear’ because it is derived from the latter.

‘Promo2NumWeeks’ is moderately correlated with ‘Promo2’, ‘Promo2SinceWeek’ and ‘Promo2SinceYear’ as it is derived from these features.

10. Feature Engineering

Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in machine learning. This process can significantly affect the performance of the model.

The following processes were involved in feature engineering:

1. **Feature Selection**: ‘Store’ is dropped since sales of all the stores are needed, not a particular one and sales can be predicted through store type, assortment, etc.

   ‘Date’ is also dropped since there is already day of week and week of year features in the dataset.

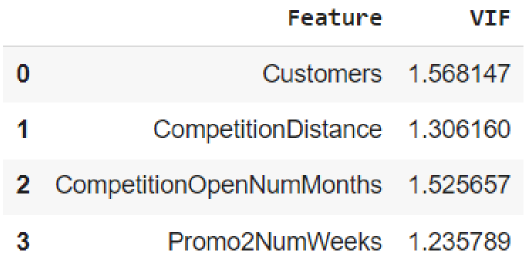
   ‘CompetitionOpenSinceMonth’ and ‘CompetitionOpenSinceYear’ are dropped as the information provided by them can be obtained from ‘CompetitionOpenNumMonths’.

   ‘Promo2’, ‘Promo2SinceWeek’ and ‘Promo2SinceYear’ are also dropped as the information provided by them can be obtained from ‘Promo2NumWeeks’.

   ‘Month’ is dropped since we get the same information from ‘WeekOfYear’.

   ‘Year’ is also dropped as we have already established in EDA that it’s not the year that influence the sales but ‘DayOfWeek’ and ‘WeekOfYear’.

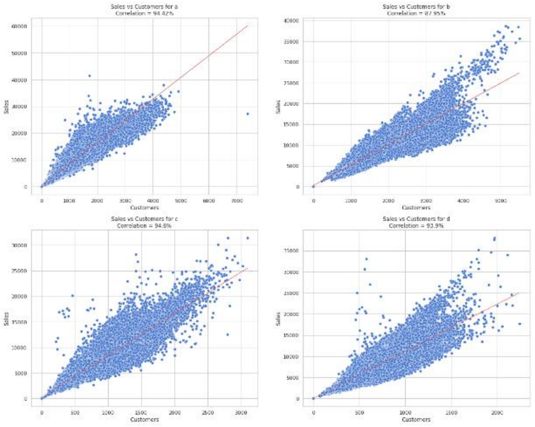
2. **Handling Multicollinearity**: The variance inflation factor (VIF) of all numerical features except 'Sales' is calculated in order to remove highly correlated features. Features having VIF greater than 5 should be eliminated.



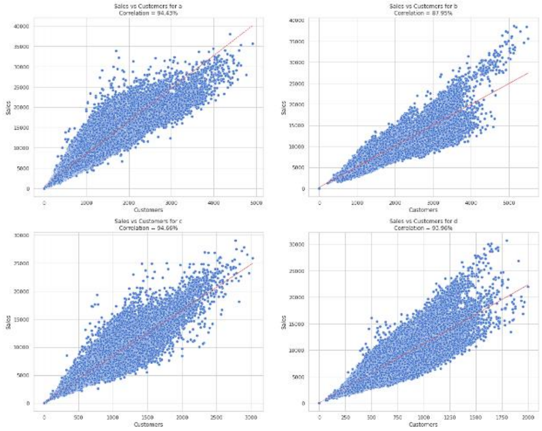
	Feature	VIF
0	Customers	1.568147
1	CompetitionDistance	1.306160
2	CompetitionOpenNumMonths	1.525657
3	Promo2NumWeeks	1.235789

All features have VIF less than 5.

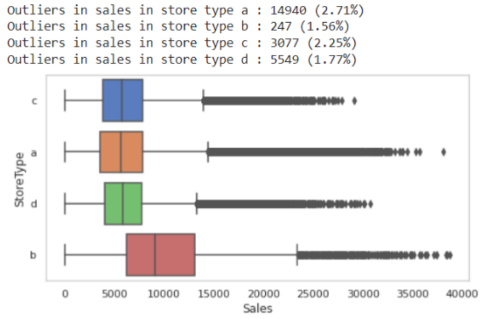
3. **Handling Outliers**: Outlier detection for ‘Sales’ and ‘Customers’ was done together since their linear relationship is already established. Removal of outliers will be more convenient and accurate. Each store type records sales in different range. So, it is more appropriate to detect outliers separately for each store type.



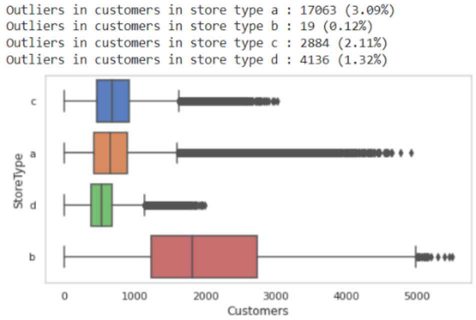
Outliers were identified from the scatter plot and removed.



Data was checked for any remaining outliers.



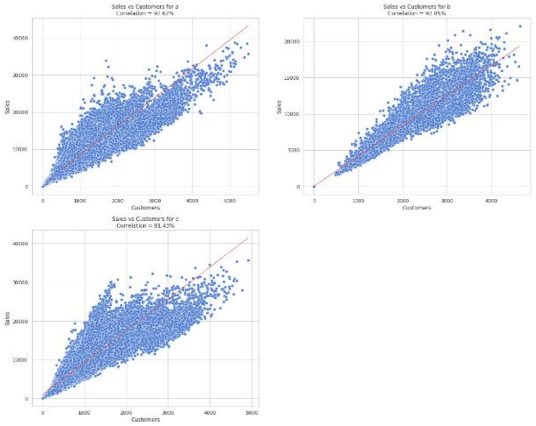
```
Outliers in sales in store type a : 14940 (2.71%)
Outliers in sales in store type b : 247 (1.56%)
Outliers in sales in store type c : 3077 (2.25%)
Outliers in sales in store type d : 5549 (1.77%)
```



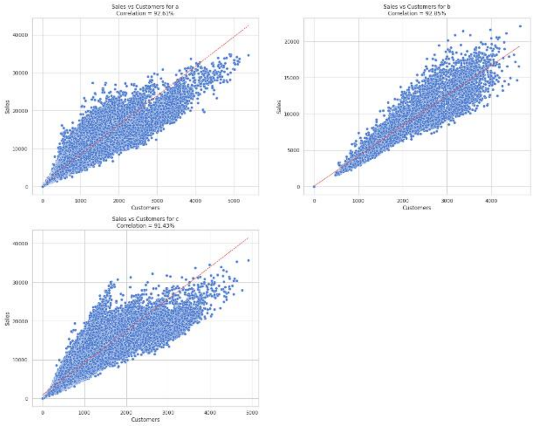
```
Outliers in customers in store type a : 17063 (3.09%)
Outliers in customers in store type b : 19 (0.12%)
Outliers in customers in store type c : 2884 (2.11%)
Outliers in customers in store type d : 4136 (1.32%)
```

There are still outliers present in the data, which are detected using the above box plots. The Sales vs Customers scatter plot has been cleared of outliers and the remaining data maintains a good relationship between sales and customers. So, these outliers are just deviation from usual values and not errors in the measurement. Eliminating them can cause overfitting, so we will keep these outliers.

Each assortment records sales in different range. So, it is more appropriate to detect outliers separately for each assortment.

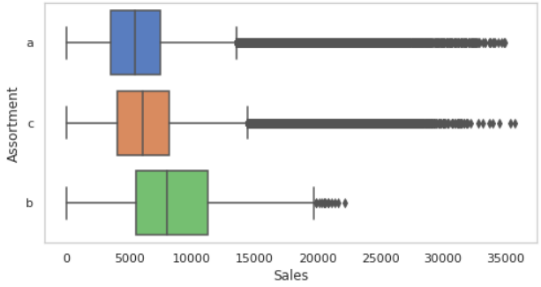


Outliers were identified from the scatter plot and removed.

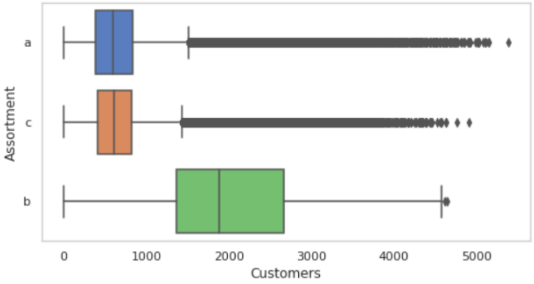


Data was checked for any remaining outliers.

```
Outliers in sales in assortment a : 12384 (2.3%)
Outliers in sales in assortment b : 13 (0.16%)
Outliers in sales in assortment c : 13735 (2.91%)
```



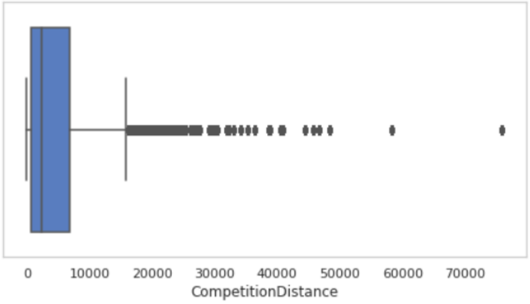
```
Outliers in customers in assortment a : 15624 (2.91%)
Outliers in customers in assortment b : 2 (0.02%)
Outliers in customers in assortment c : 17824 (3.78%)
```



There are still outliers present in the data, which are detected using the above box plots. The Sales vs Customers scatter plot has been cleared of outliers and the remaining data maintains a good relationship between sales and customers. So, these outliers are just deviation from usual values and not errors in the measurement. Eliminating them can cause overfitting, so we will keep these outliers.

Outliers in ‘CompetitionDistance’ were detected with a box plot.

```
Outliers in competition distance : 99158 (9.75%)
```

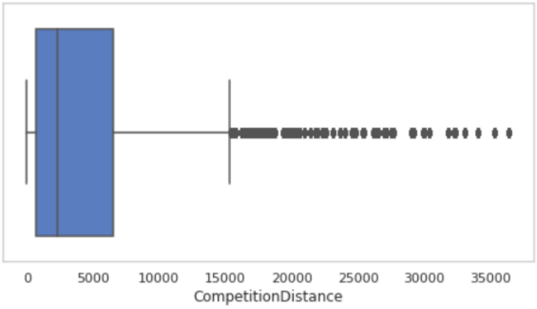


Almost 10% of data are outliers and removing them will cause the loss of useful information. What we can do is eliminate only a certain portion of outliers. There are three options for the section of outliers that we can remove:

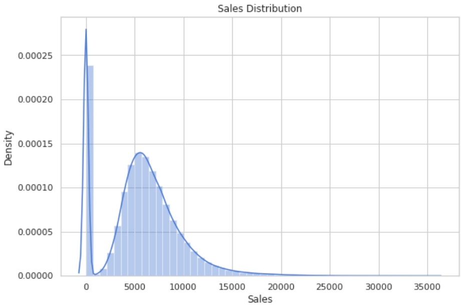
- Competition distance > 55000(0.19%)
- Competition distance > 44000(0.56%)
- Competition distance > 37500(0.93%)

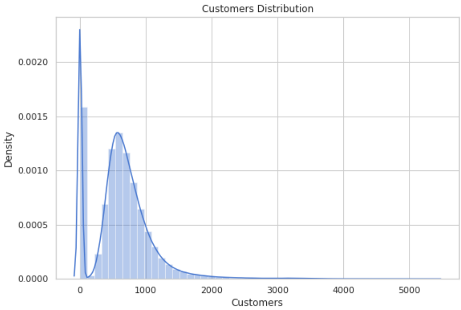
The 3rd option is chosen since it clears the most outliers without losing much data.

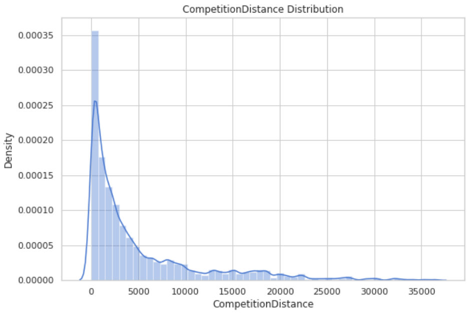
```
Outliers in competition distance : 95022 (9.43%)
```

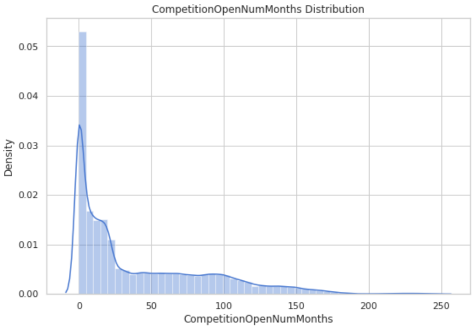


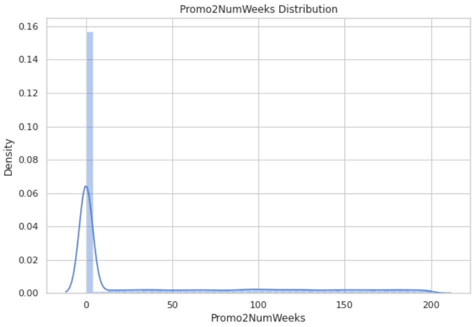
4. **Feature Transformation**: Transformation of numerical features is important because algorithms like linear regression, which assumes normal distribution, performs better when features are close to Normal Distribution.





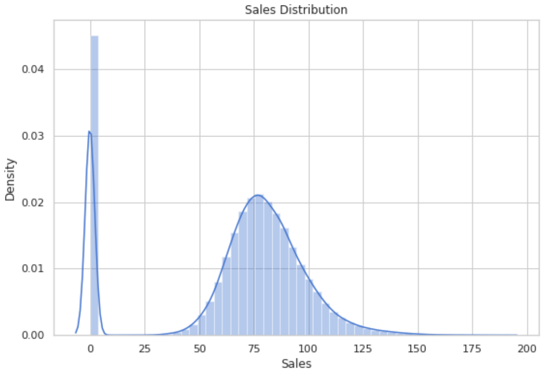


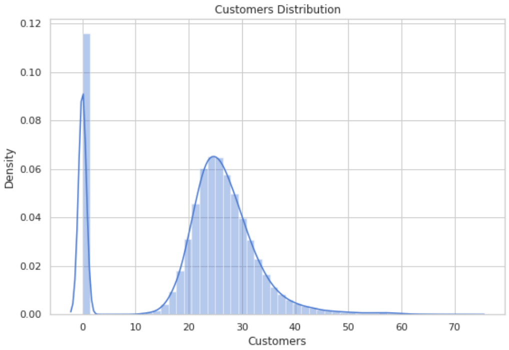


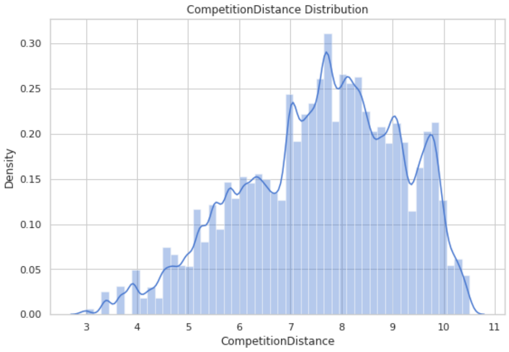


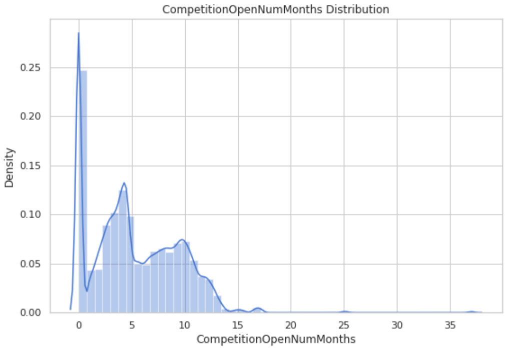
All these features are positively skewed but linear regression assumes normal distribution. So, they must be transformed to normal distribution before forecasting sales.

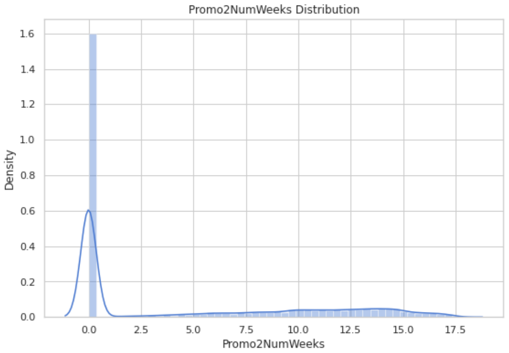
‘Sales’, ‘Customers’, ‘CompetitionOpenNumMonths’ and ‘Promo2NumWeeks’ have zero values while ‘CompetitionDistance’ has only positive values. So, ‘Sales’, ‘Customers’, ‘CompetitionOpenNumMonths’ and ‘Promo2NumWeeks’ are square root transformed and ‘CompetitionDistance’ is log transformed.











5. **Categorical Feature Encoding**: Most algorithms cannot handle the categorical variables unless they are converted into a numerical value. So, ‘StoreType’ and ‘Assortment’ were encoded using one hot encoder while ‘PromoInterval’ was dummified. To overcome dummy variable trap, one resultant feature from each encoded feature must be removed. Correlation matrix was used to decide which features to remove. ‘PromoInterval_Jan,Apr,Jul,Oct’, ‘StoreType_c’ & ‘Assortment_b’ have the least correlation to Sales and they were removed.

11. Modelling

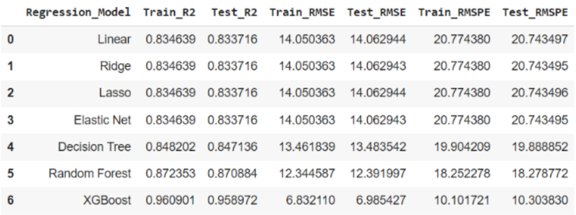
Input and target data were separated, and both were split into training and test data with 25% test data.

The feature ‘Customers’ was also removed since the number of customers for the period, which is under consideration for forecasting, won't be available until the mentioned period is over. Training and test data of independent features were scaled using standardization.

Model training was done with these data using 7 different algorithms:

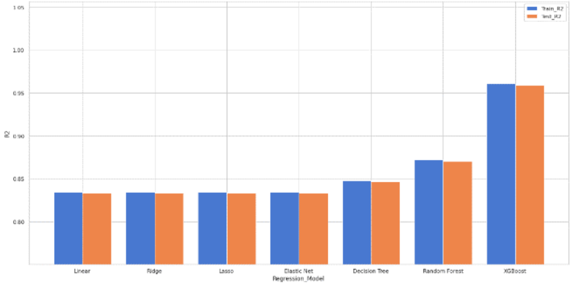
1. Linear regression
2. Ridge regression
3. Lasso regression
4. Elastic net regression
5. Decision tree regression
6. Random forest regression
7. XGBoost regression

Evaluation metrics like R-squared, RMSE and RMSPE were calculated for each model.



	Regression_Model	Train_R2	Test_R2	Train_RMSE	Test_RMSE	Train_RMSPE	Test_RMSPE
0	Linear	0.834639	0.833716	14.050363	14.062944	20.774380	20.743497
1	Ridge	0.834639	0.833716	14.050363	14.062943	20.774380	20.743495
2	Lasso	0.834639	0.833716	14.050363	14.062944	20.774380	20.743496
3	Elastic Net	0.834639	0.833716	14.050363	14.062943	20.774380	20.743495
4	Decision Tree	0.848202	0.847136	13.461839	13.483542	19.904209	19.888852
5	Random Forest	0.872353	0.870884	12.344587	12.391997	18.252278	18.278772
6	XGBoost	0.960901	0.958972	6.832110	6.985427	10.101721	10.303830

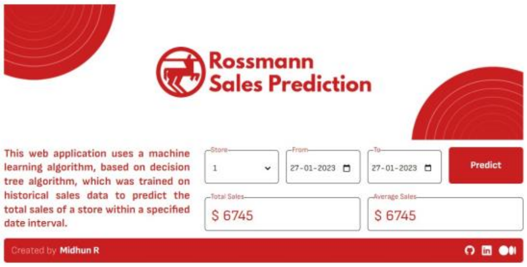
R2 score is be used to compare models and determine which one gives higher accuracy. Higher the R2 score, higher the accuracy.



The model built using XGBoost algorithm has the highest R2 score with ~96%, followed by the one using random forest and decision tree.

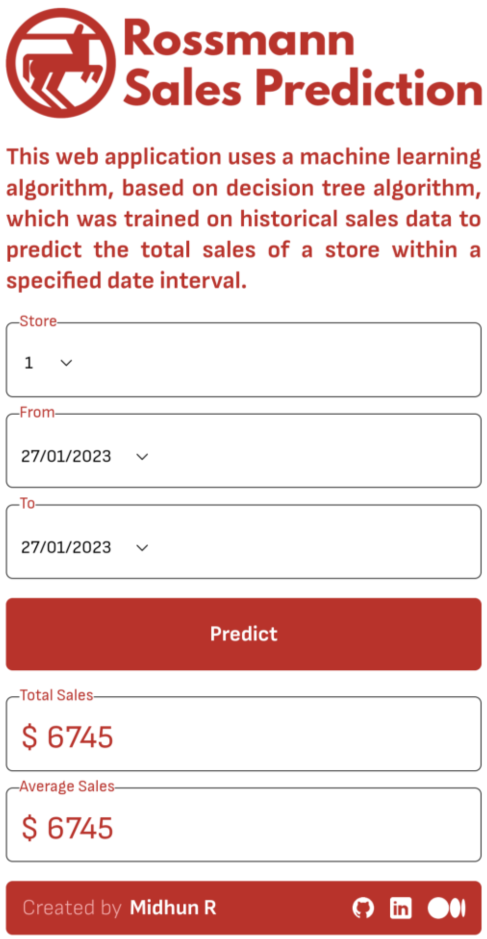
12. Deployment

- A web application is built to demonstrate the working of the trained machine learning model using a combination of HTML, CSS, and JavaScript.
- The prediction of sales using the trained ML model is carried out via a Flask API.
- This web application is dockerized and deployed with AWS Elastic Beanstalk, employing CI/CD pipeline.



This web application uses a machine learning algorithm, based on decision tree algorithm, which was trained on historical sales data to predict the total sales of a store within a specified date interval.

Store: 1
From: 27-01-2023
To: 27-01-2023
Total Sales: $ 6745
Average Sales: $ 6745



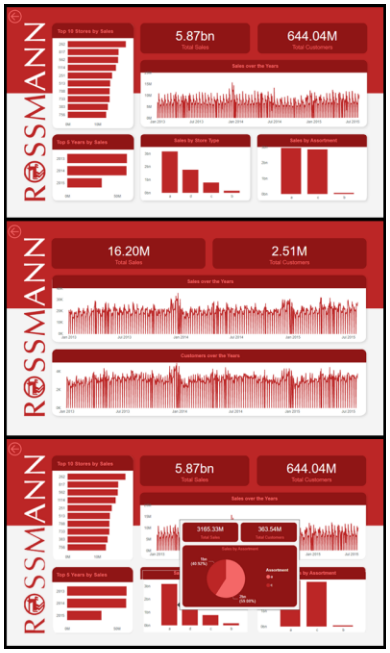
This web application uses a machine learning algorithm, based on decision tree algorithm, which was trained on historical sales data to predict the total sales of a store within a specified date interval.

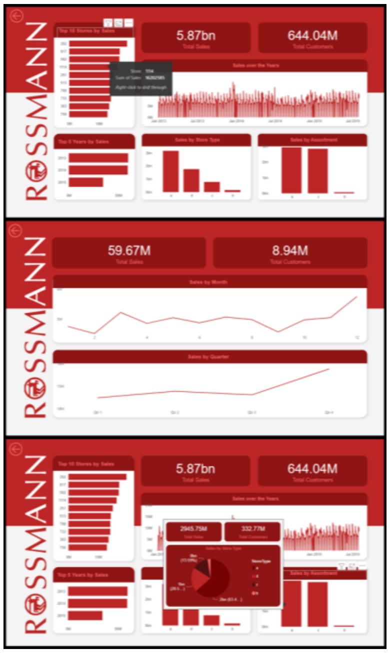
Store: 1
From: 27/01/2023
To: 27/01/2023
Total Sales: $ 6745
Average Sales: $ 6745

Link to deployed model: http://rossmannsalesprediction-env-1.eba-3wi97vqp.ap-south-1.elasticbeanstalk.com/

13. Data Visualization

An interactive dashboard was also created with Power BI to display charts associated with the analysis. It includes features like drill-through and customized tooltip.





Click here to download the data visualization.

14. Challenges

Handling 1017209 observations of data over 18 features of data were a bit difficult as a beginner. The inspection and cleaning of dataset was tedious and complicated. Visualization of data was properly carried out after providing a great amount of attention to each and every details.

Selection of the right type of transformation took a lot of research. Deciding if some features have to be dropped or not was tough. Training of models was a time-consuming process.

15. Conclusion

The following conclusions were drawn from EDA:

- Mondays have most sales since most of the Sundays are closed.
- Promotions seem to have a significant effect on sales but not for the number of customers. It is advisable to spend more on promos to get higher returns.
- Store type b has higher sales and customers per store than other store types. More Store type b must be opened.
- Assortment b is available only at store type b and it has more sales and customers than any other assortment. More assortment b must be stocked to meet the demands of customers.
- Weekly sales and customers peak at the mid-December. It may be guessed that people buy drugs in advance just before the shops close for the holiday season.

The following conclusions were drawn from Modelling:

- The model built using XGBoost algorithm is the most accurate one. This can be attributed to higher number of categorical features in the data.
- If model interpretability is more important than accuracy, model built using decision tree algorithm should be

chosen over the one using random forest algorithm.

- Decision tree-based algorithms are slightly more accurate than linear regression-based algorithms.

References-

1. Kaggle
2. GeeksforGeeks
3. Analytics Vidhya
4. Towards Data Science