NEXT HIKES

Rossmann Store Sales Prediction

Guided By: **Desmond Onam Sir**
Swati Sutar Mam

Presented by:
Walmeek Borde

Problem

Problem Statement

Predict 6 weeks of daily sales for drug stores operated by Rossmann located across Germany to enable store managers to create effective staff schedules that increase productivity and motivation.

Evaluation

Evaluation is based on the Root Mean Squared Percentage Error (RMSPE) metric, which should be minimized:

$$\text{RMSPE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2}$$

where y_i denotes the sales of a single store on a single day and $\hat{y}_i$ denotes the corresponding prediction. Any day and store with 0 sales is ignored.

Understanding the Dataset

There are two training datasets provided to us (train.csv & store.csv) and one test dataset (test.csv) whose sales we have to predict.

train.csv

This is the main training dataset which contains data about the sales figures for a store on a particular date.

Data Fields:

1. *Store*: a unique numerical store identifier (1 - 1,115)
2. *DayOfWeek*: the day of week (1 - 7)

3. *Date*: the date, ranging from 2013-01-01 to 2015-07-31
4. *Sales*: the turnover of the specified store on the specified date
5. *Customers*: the number of customers of the specified store on the specified date
6. *Open*: indicates whether the store was open (0 = Closed, 1 = Open)
7. *Promo*: indicates whether the store was running a promotion (0 = No, 1 = Yes)
8. *StateHoliday*: indicates if it was a state holiday (a = Public Holiday, b = Easter holiday, c = Christmas, 0 = None)
9. *SchoolHoliday*: indicates if it was a school holiday (0 = No, 1 = Yes)

No. of Records: 1,017,209

store.csv

This dataset contains supplementary information about each of the 1,115 stores and helps identify unique features which may (or may not) affect sales.

Data Fields:

1. *Store*: a unique numerical store identifier (1 - 1,115)
2. *StoreType*: differentiates between the 4 different types of stores (a, b, c, d)
3. *Assortment*: describes the assortment of goods carried by the store (a = Basic, b = Extra, c = Extended)
4. *CompetitionDistance*: the distance (in metres) to the nearest competitor's store
5. *CompetitionOpenSinceMonth*: the month in which the competition opened
6. *CompetitionOpenSinceYear*: the year in which the competition opened
7. *Promo2*: indicates if a store is participating in a continuing and consecutive promotion (0 = No, 1 = Yes)
8. *Promo2SinceWeek*: the week of the year in which the store began participating in *Promo2* (from 1 - 52, presumably, but some weeks are unrepresented in the data)
9. *Promo2SinceYear*: the year in which the store began participating in *Promo2* (from 2009 - 2015)
10. *PromoInterval*: describes the consecutive intervals in which *Promo2* is activated, giving the months the promotion is renewed (either "Jan,Apr,Jul,Oct", "Feb,May,Aug,Nov" or "Mar,Jun,Sept,Dec")

test.csv

This dataset is to be used for testing and evaluating the model.

Data Fields: Same as train.csv, with the exclusion of *Customers* & *Sales* (*Sales* to be predicted by the model) and an additional field *Id* which represents a (*Store* , *Date*) tuple that is used to label predictions for submission to Kaggle.

No. of Records: 41,088

Preprocessing the Data

Replacing Missing Values & Fixing Corrupted Data

- **Replacing NaN values for *Open*:** Missing values in the *Open* data field were replaced with '1' if the *DayOfWeek* was not Sunday.
- **Replacing NaN values for *CompetitionDistance*:** Since no record exists where *CompetitionDistance* is NaN and *CompetitionOpenSince[X]* is not NaN, the *Competition-Distance* was set to 0 if it was NaN.
- **Replacing NaN values for *CompetitionSince[X]* :** If *CompetitionDistance* is not 0 but the *CompetitionSince[X]* columns are missing, the earliest possible values were inserted i.e. 1900 and 01 for *CompetitionSinceYear* & *CompetitionSinceMonth* respectively.
- **Converting all 0 (number) values in *StateHoliday* to "0" (string) values:** Data in the *StateHoliday* is a mix of numerical and string values. Hence, for the sake of consistency, all values were converted to string.

One-Hot Encoding

Using the *pandas.get_dummies()* function available in the Pandas library for Python, we performed one-hot encoding on the categorical features present in our dataset, including *DayOfWeek* & *StateHoliday* in train.csv & test.csv and *StoreType* & *Assortment* in store.csv.

Additional Features

- ***DayOfMonth, Year, Month, YearMonth* & *WeekOfYear* :** These data fields were extracted from the original *Date* column to provide a better understanding of date-related trends.
- ***StateHolidayBinary*:** Apart from one-hot encoding *StateHoliday*, another column was created that simply indicated whether it was a state holiday using 0 or 1.
- ***AvgCustStore, AvgCustStoreMonth, AvgCustStoreYear*:** Stores the average no. of customers per store (overall), per store per month and per store per year respectively.
- ***IsPromoMonth*:** Indicates if the record is in the month of a running *Promo2*.

- ***CompetitionOpenYearMonthInteger*:** Stores the YYYYMM version of *Competition-OpenSince[X]*.

- ***AvgSales, AvgCustomers , AvgSalesPerCustomer* :** Stores the average sales per day, average no. of customers per day and average sales per customer per day respectively for each store.

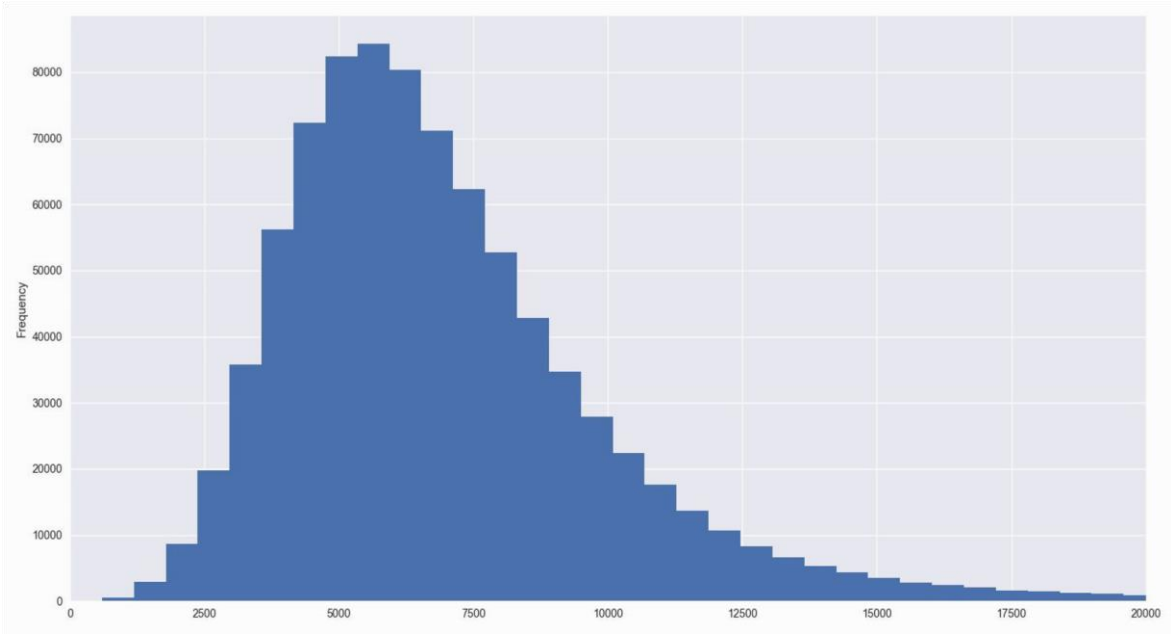
Analyzing the Dataset

Basic Analysis

Min, Max, Mean & S.D.

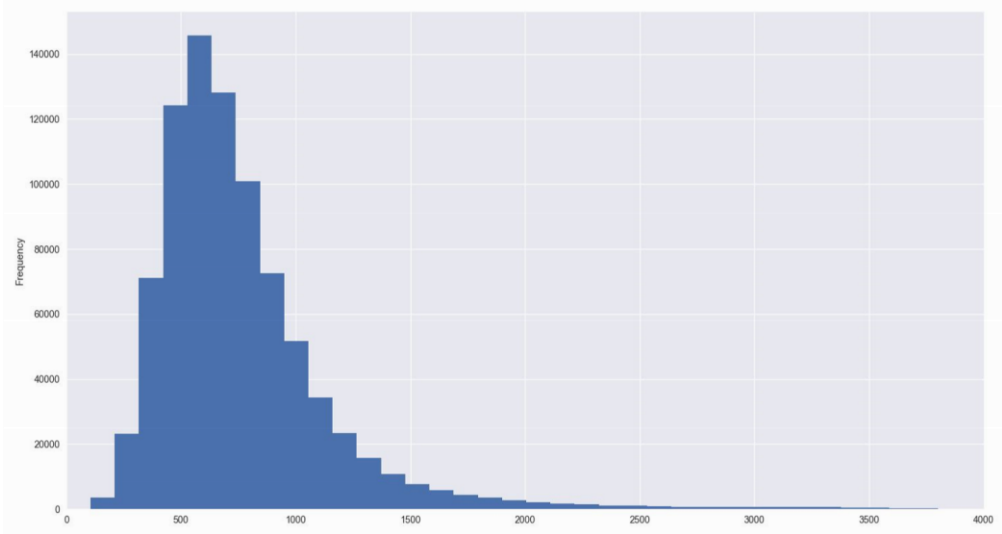
Field Name	Min	Max	Mean	Standard Deviation
Store	1	1,115	N/A	N/A
Customers	0	7,388	633.1459	464.4117
Sales	0	41,551	5,773.819	3,849.926

Frequency Distribution



Frequency of *Sales* Values for Open Stores

In the chart above, we can see that the *Sales* values forms a Poisson-like distribution, reaching its peak of frequency between 5,000 and 7,500.



Frequency of *Customers* Values for Open Stores

The distribution of *Customers* values follows a Poisson-like distribution, indicating that the number of customers commonly ranges between 500 and 1,000, while a value higher than 1,000 has relatively low frequency. The average value for *Customers* for open stores is 762.728.

Possible Outliers

When studying the data distribution, several records were highlighted as possible outliers in the data. Firstly, there are 52 records in the training data where the store is indicated to be open but *Sales* and *Customers* are both 0. This could mean that the store was mistakenly indicated as being open. Because of this, we decided to exclude all records where the store was open but *Sales* was 0 from our training data to avoid such erratic records.

Additionally, there were several records that seemed to have unusually high *Sales* values.

Sales Value	No. Of Records
> 45,000	1
> 35,000	18
> 30,000	153
> 25,000	758

This suggests that *Sales* values greater than 35,000 can be safely ignored to avoid overfitting as those records are extremely rare and only occur in a handful of stores but could affect the model's predictions for the other stores.

Trend Analysis for Date-Related Fields

Weekly Trends

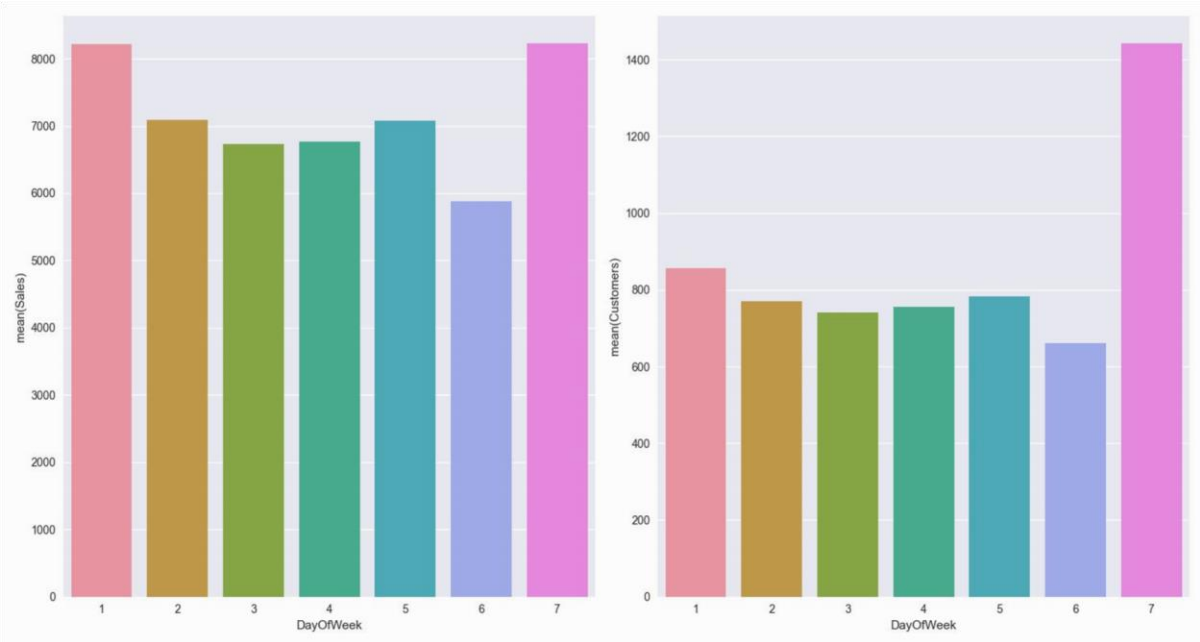


Fig. 3: Average Sales & Average Customers for Open Stores by Day of Week

As can be seen in the chart above, there are three peaks in the average sales and average no. of customers in the week. The first two are on Mondays & Fridays, which is probably due to these days being the start and end of the work week. The highest peak, however, is on Sundays, which is probably due to the fact that most other stores are closed. An interesting observation here is that there is a larger difference between the average no. of customers and average sales on Sundays as compared to other days of the week. While the average sales is approximately the same on Sundays & Mondays, there is a drastic difference in the no. of customers, which hints at a large number of window shoppers on Sundays.

Annual Trends

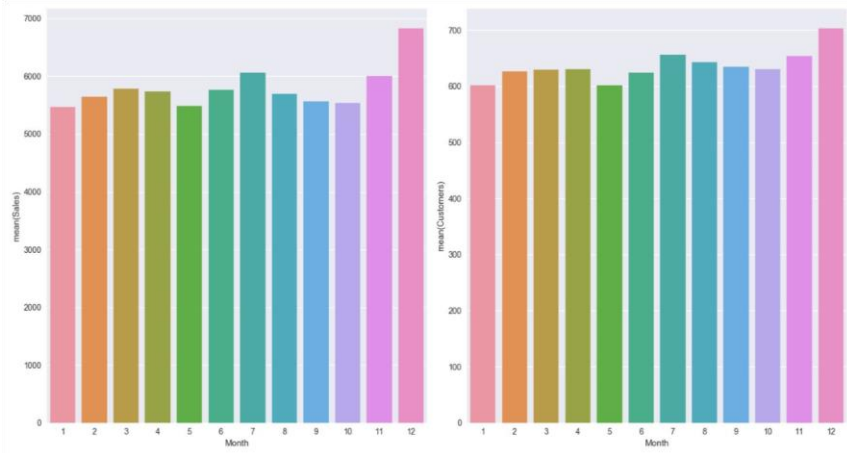
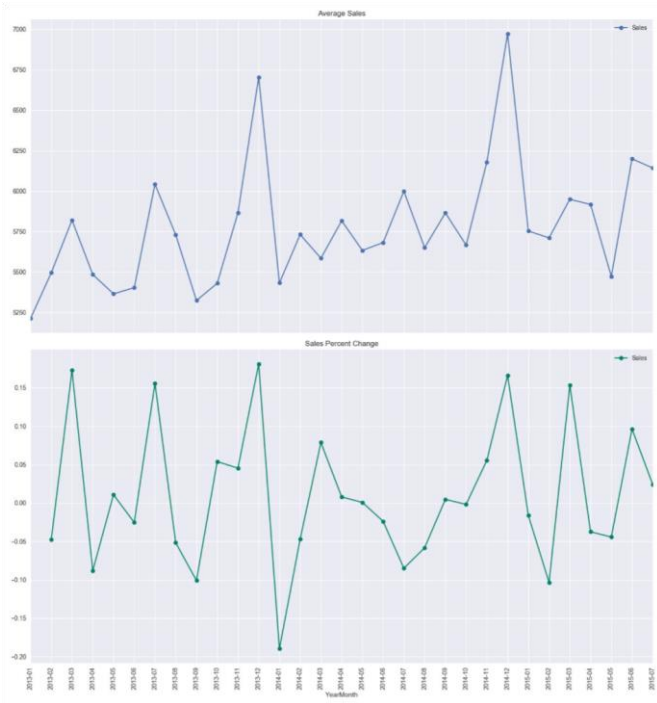


Fig. 4: Average Sales & Average Customers by Month

When plotting the average sales & no. of customers by month, we see an annual trend similar to most retail stores. There is an uptick in the months of June - July owing to summer holidays and another bigger uptick in December, which is likely due to the holiday season.



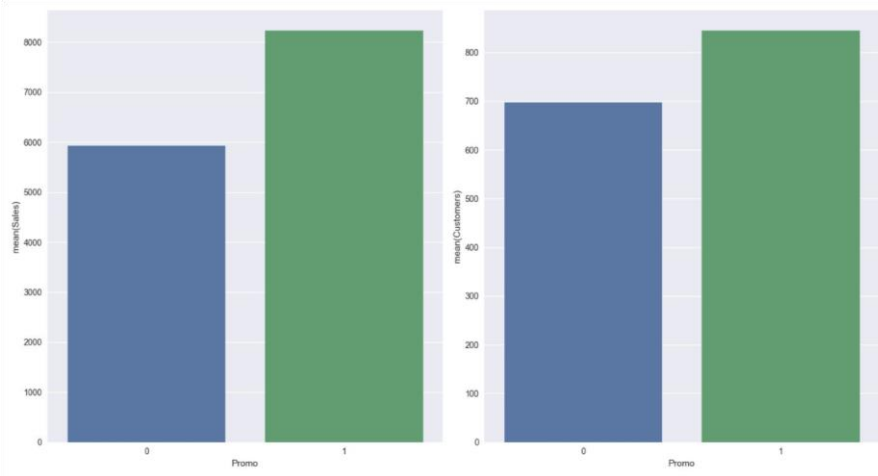
Average Sales & Percentage Change by Year-Month

The annual trends are further corroborated when plotting the average sales data per month for the entirety of the training dataset

Effect of Promotions & Holidays

Promo

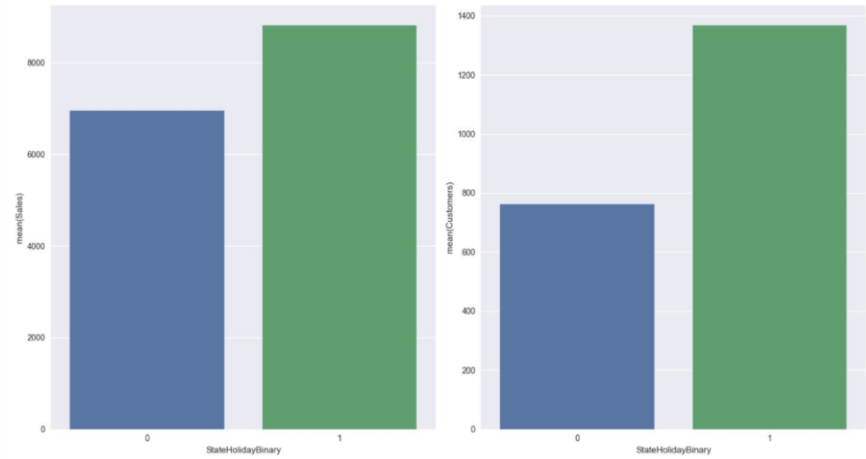
The figure below shows the average customers and average sales (across all stores) on days with and without promotions. The effect of having a promotion on sales and customers is clearly evident, with the promotion having a strong positive effect on both.



Average Sales & Average Customers for Open Days with/without Promo

State Holidays

Since most stores are closed on public holidays and closed stores can be ignored when making predictions, the average sales figures and average number of customers on state holidays is plotted, but only for open stores in the figure below. It is evident that when a store is open on state holidays, it attracts more customers and accrues more sales.



Average Sales & Average Customers for Open Stores on State Holidays

When the plot is further broken down to the individual holidays in the figure below, we see that Christmas and Easter have the best average sales, with public holidays second and non-holidays faring the worst for average sales.

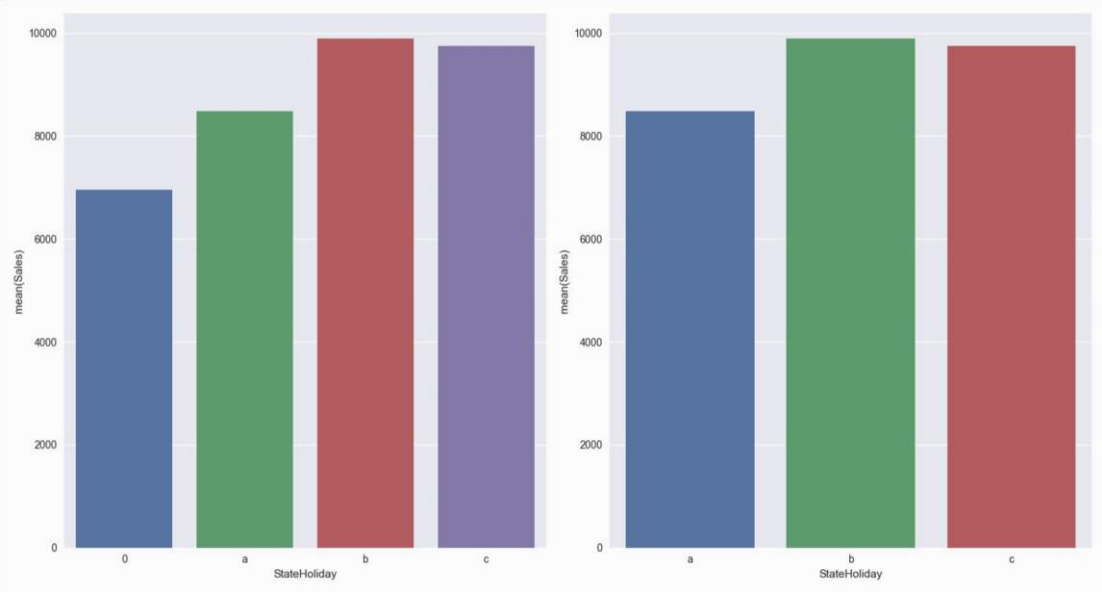
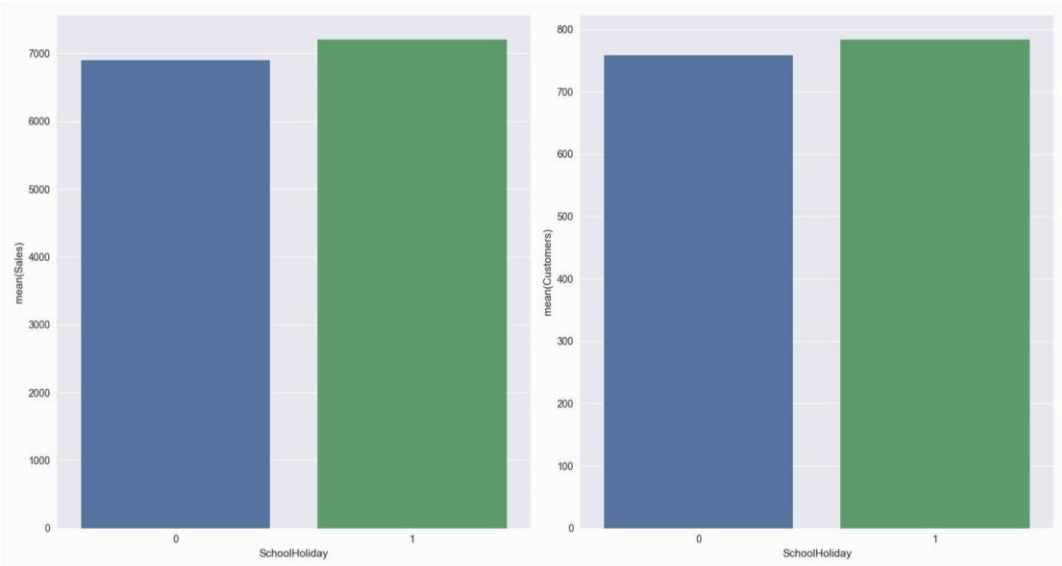


Fig. 8: Average Sales for Open Stores on Normal Days & State Holidays

School Holidays

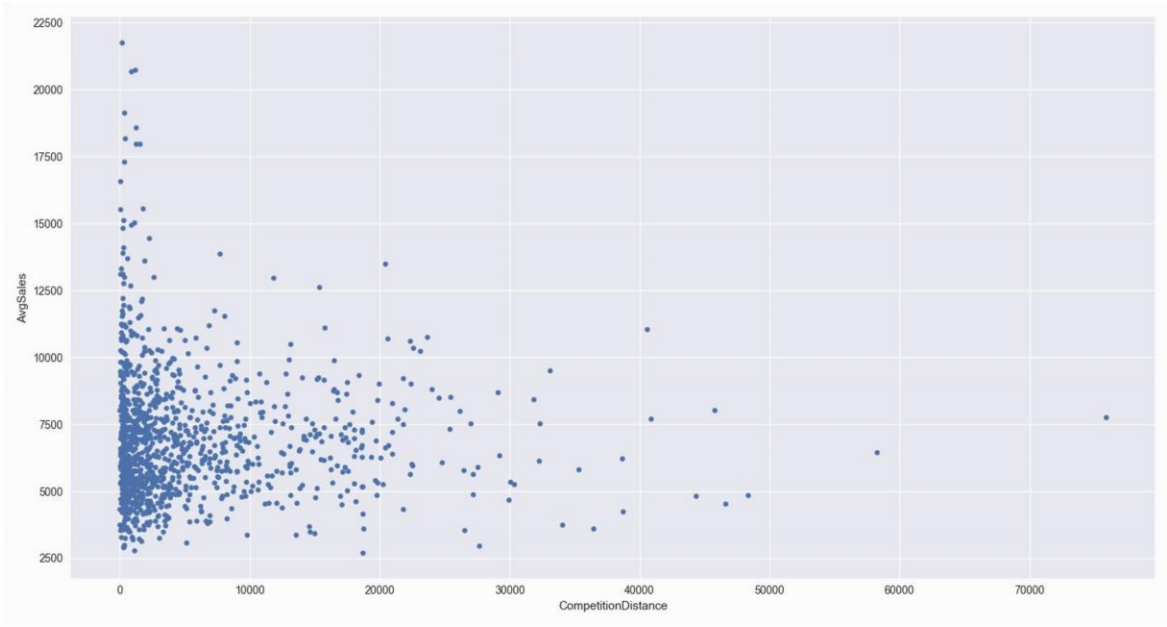
When we look at the sales averages and average number of customers in stores during school holidays, we see a small increase (~5%), which hints at a likely correlation.



Average Sales & Average Customers for Open Stores with/without School Holidays

Effect of Competition

When we study the correlation between the competition distance and sales/customers, we see that higher average sales figures are achieved when *CompetitionDistance* equals 0, or in other words, there is no nearby competition.



Competition Distance vs. Average Sales for each Store

We see a similar relationship with the *CompetitionDistance* and average no. of customers plot below.

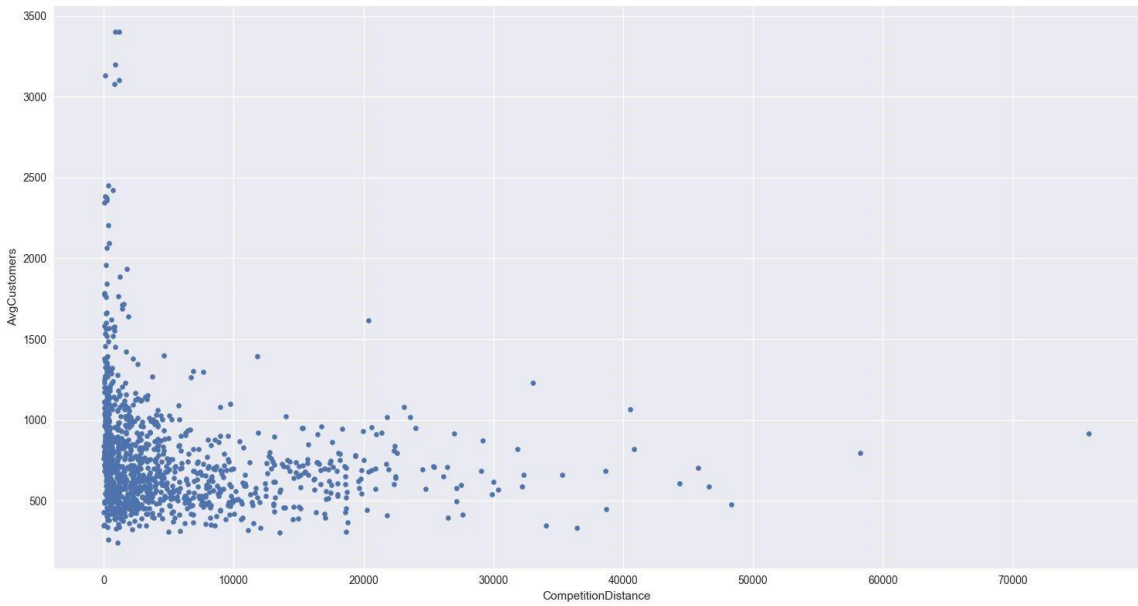


Fig. 11: *CompetitionDistance* vs. Average No. of Customers for each Store

In order to verify the effect of competition on store sales, a plot was generated selecting one particular store (Store 6) and studying its sales before and after competition opened nearby. The results are given in the plot below. The effect is immediately evident, with sales plunging right as the competition opens.

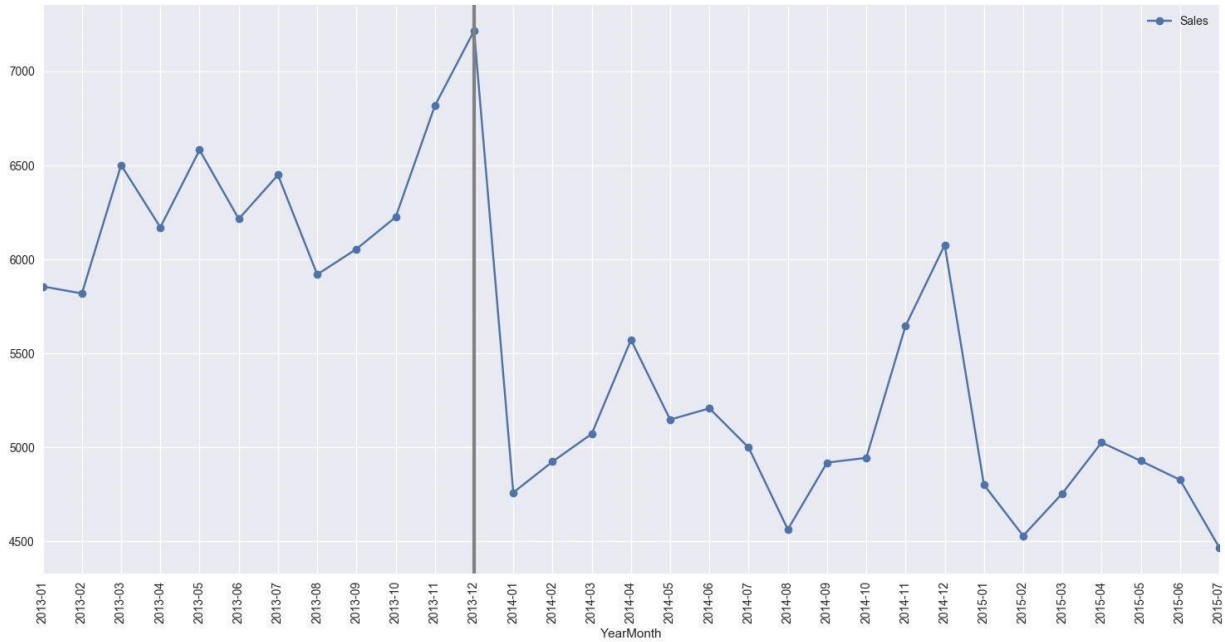


Fig. 12: Effect of Competition on Store 6 by Year-Month

Effect of Store Type & Assortment Type

Store Type

A plot of the average sales and average customers of each store type shows that there is a strong correlation between the store type and average sales.

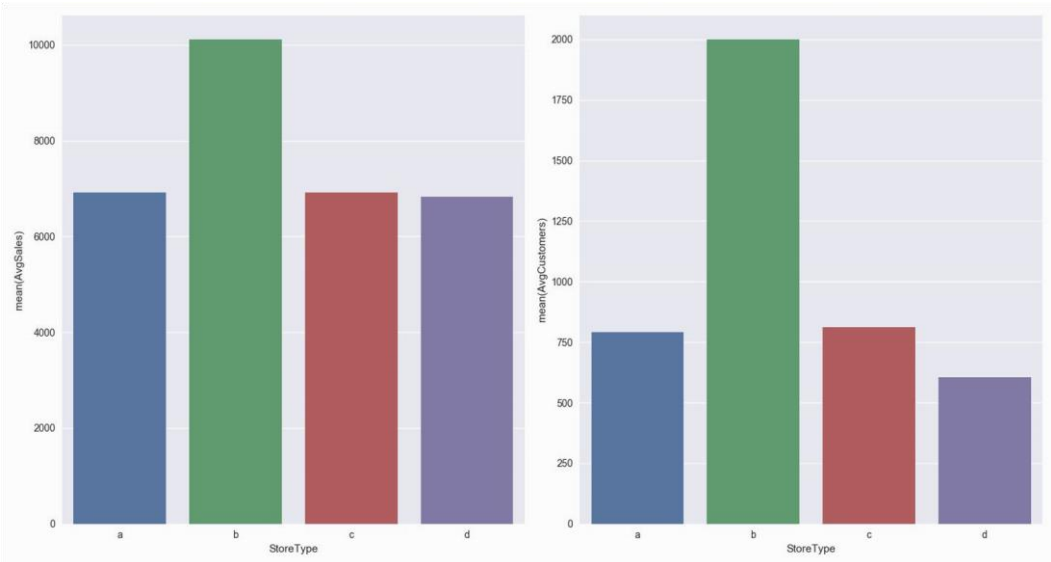
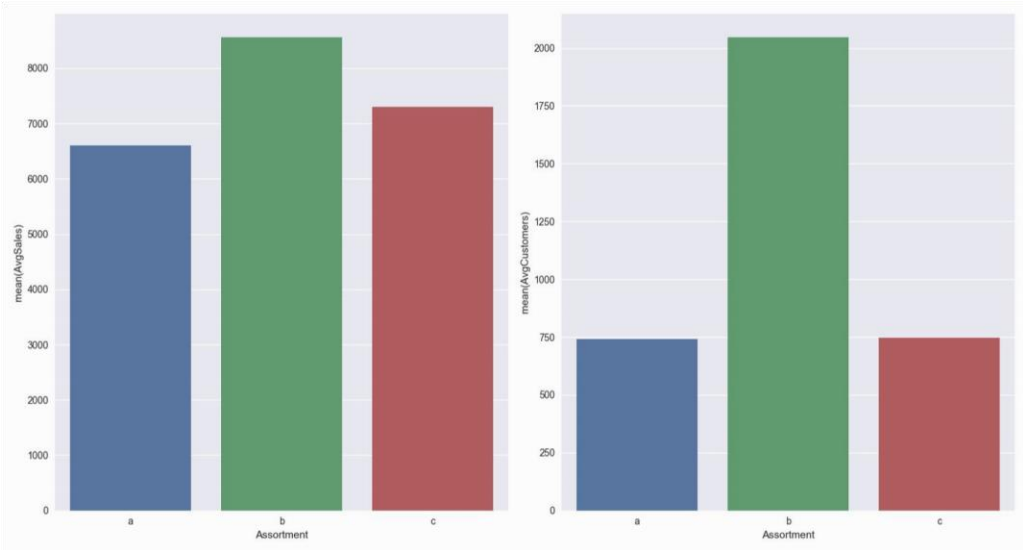


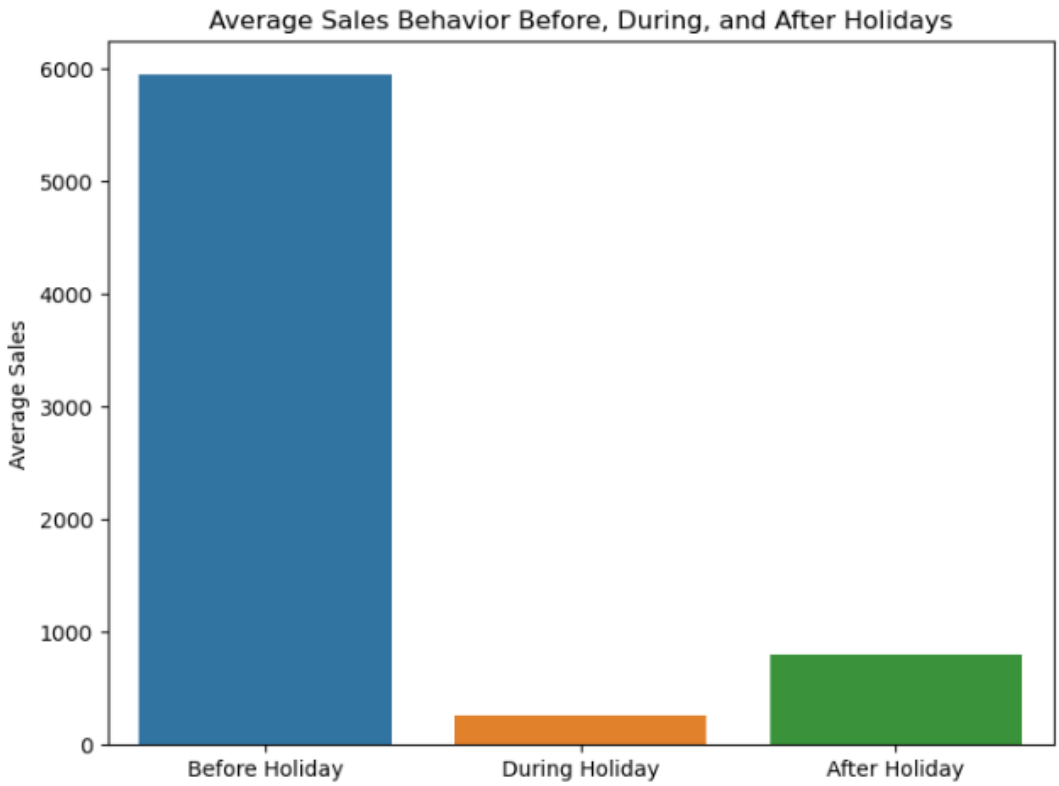
Fig. 13: Average Sales & Customers on Open Days for each Store Type

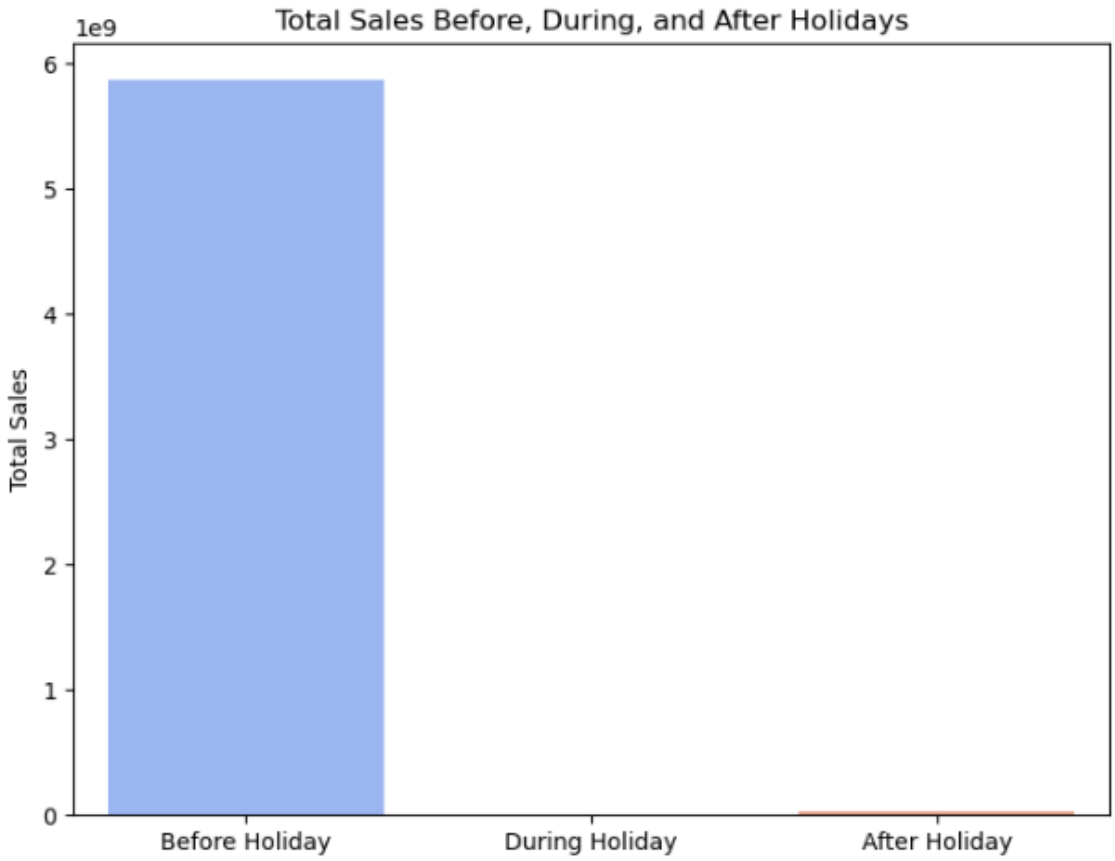
Assortment Type

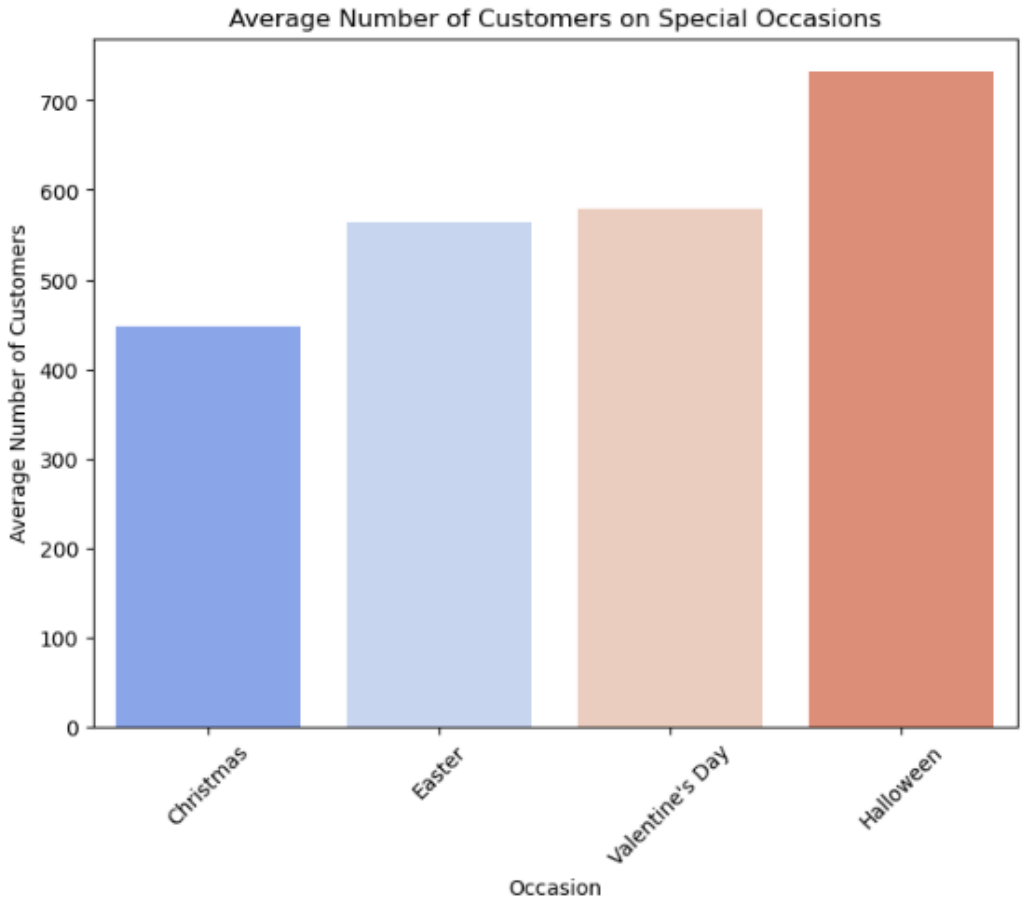
A same correlation is observed in the average sales and average customers for stores with different assortment types.



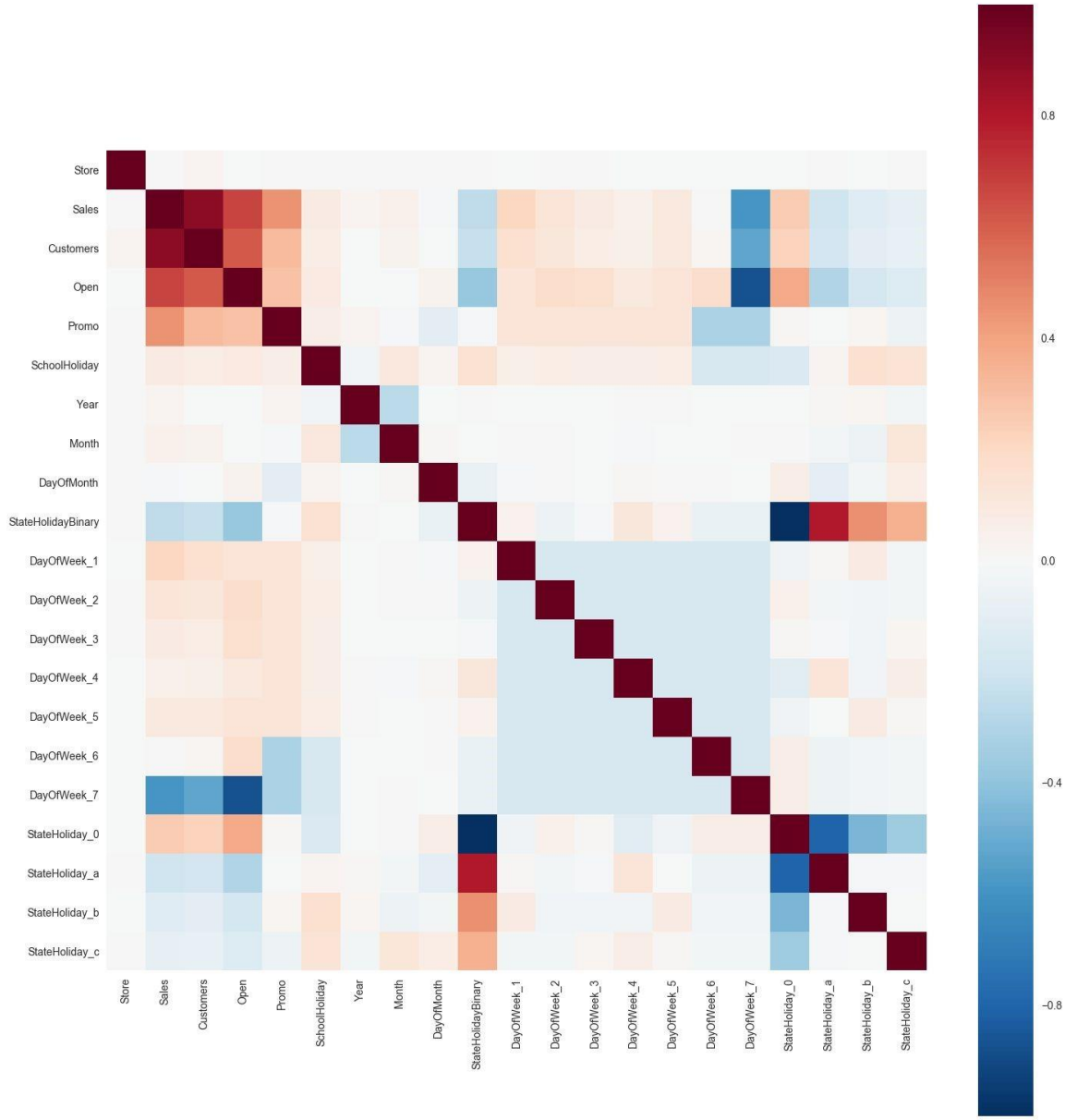
Average Sales & Customers on Open Days for each Assortment Type




Total Sales Before, During, and After Holidays
1e9
6
5
4
3
2
1
0
Total Sales
Before Holiday
During Holiday
After Holiday


Average Number of Customers on Special Occasions
700
600
500
400
300
200
100
0
Average Number of Customers
Christmas
Easter
Valentine's Day
Halloween
Occasion

Correlation Matrix of Features



Correlation Matrix of Features from train.csv

XGBoost Regressor - Cross Validation

	Model Name	MAE	MAPE	RMSE
0	XGBoost Regressor	908.750984	0.136727	1309.609772

The correlation matrix above shows the correlation between the different features in the training data, with categorical features being one-hot encoded. As can be seen, the sales is strongly affected by any running promotions, the day of the week and holidays along with the obvious correlation between the sales and the no. of customers and whether or not the store is open.

Models

We implemented a validation procedure using the last six weeks of the training data as the test dataset and the remaining data for training. This process allowed us to predict the "future" sales using "historical" data. Also, by assuming the last six weeks of training data for testing, it was possible to simulate a scenario close to the test conducted. We determined the RMSPE score for a few models using this validation procedure.

A list of some of the most significant models we implemented can be found in Appendix A.

Simple Geometric Mean Model

This model was used as a benchmark. It simply calculates the geometric mean value for every [*Store*, *DayOfWeek*, *Promo*] group and assigns that value as the prediction for every [*Store*, *DayOfWeek*, *Promo*] group in the test data.

Feature Selection

Store, *DayOfWeek* & *Promo*

Assumptions

1. The only factors that significantly affect the sales in a particular store are *DayOfWeek* & *Promo* as they have the strongest correlation in the correlation matrix (Fig. 15).
2. The geometric mean value is used instead of the arithmetic mean to normalise the different ranges in the *Sales* values.

Results

RMSPE Score

- Private Score: 0.15996
- Public Score: 0.15390

Linear Regression Model

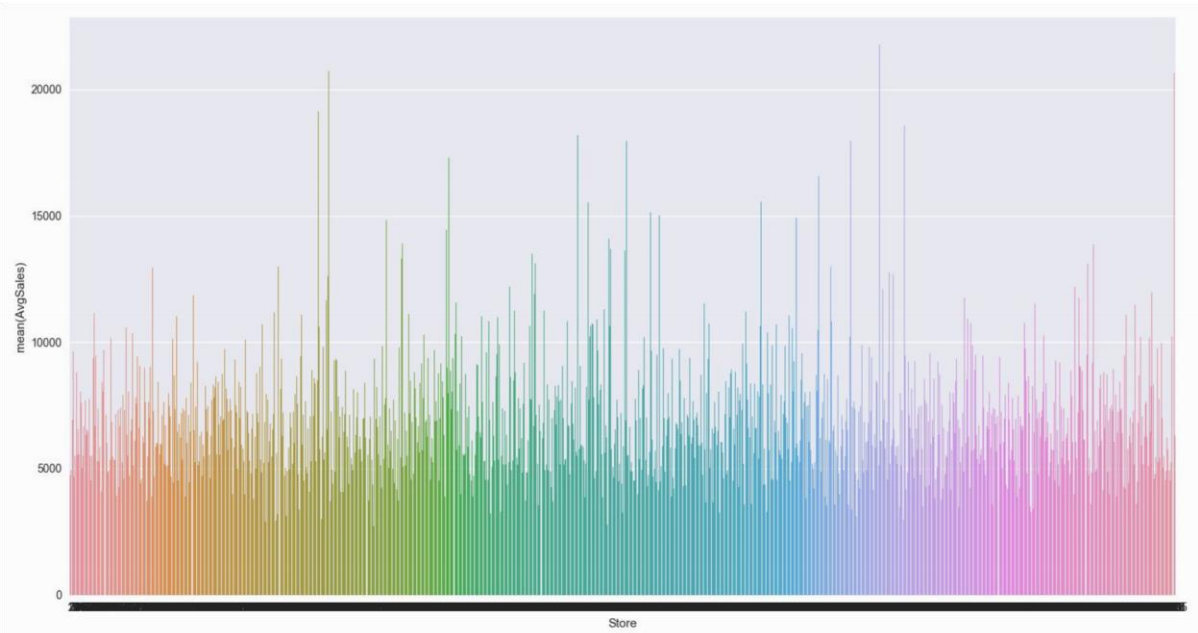
We selected a simple linear regression model as the first model type to implement as it is the most straightforward algorithm. We attempted the linear regression model using two different approaches. In the first approach, we treated each store as an independent regression problem and trained a model for each store to predict its *Sales* value. In the second approach, we treated the entire dataset as a single regression problem. However, the latter performed quite poorly in comparison to the independent regression problem approach.

Linear Regression Version 2 (linearregression-independent2.py)

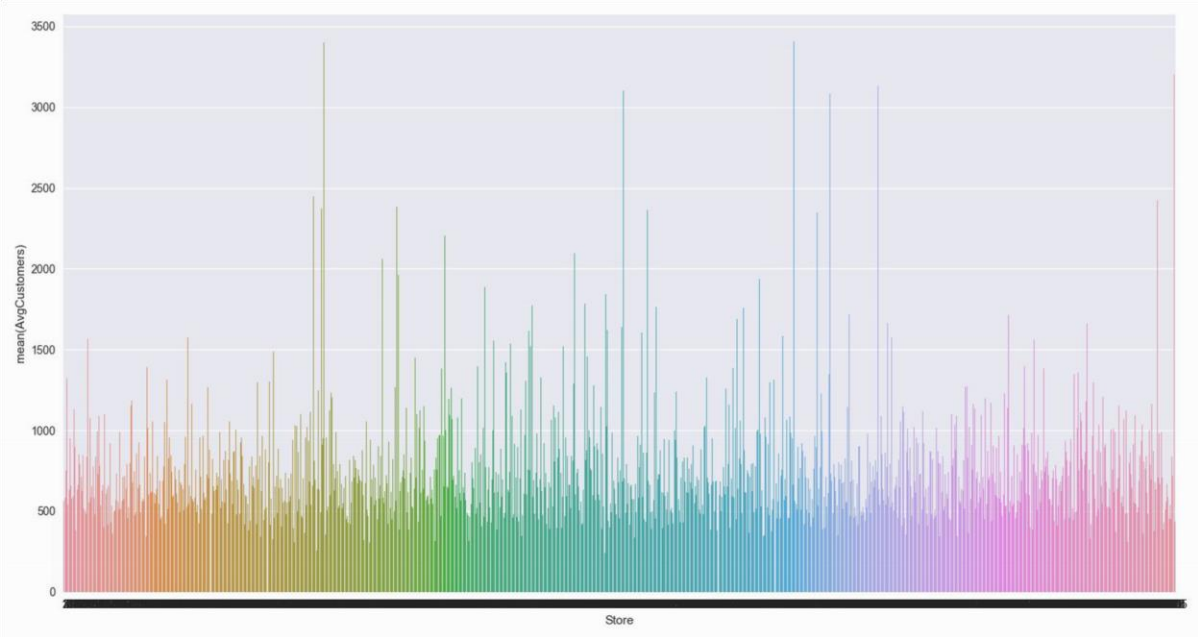
Feature Selection

Promo, DayOfWeek (one-hot encoded)

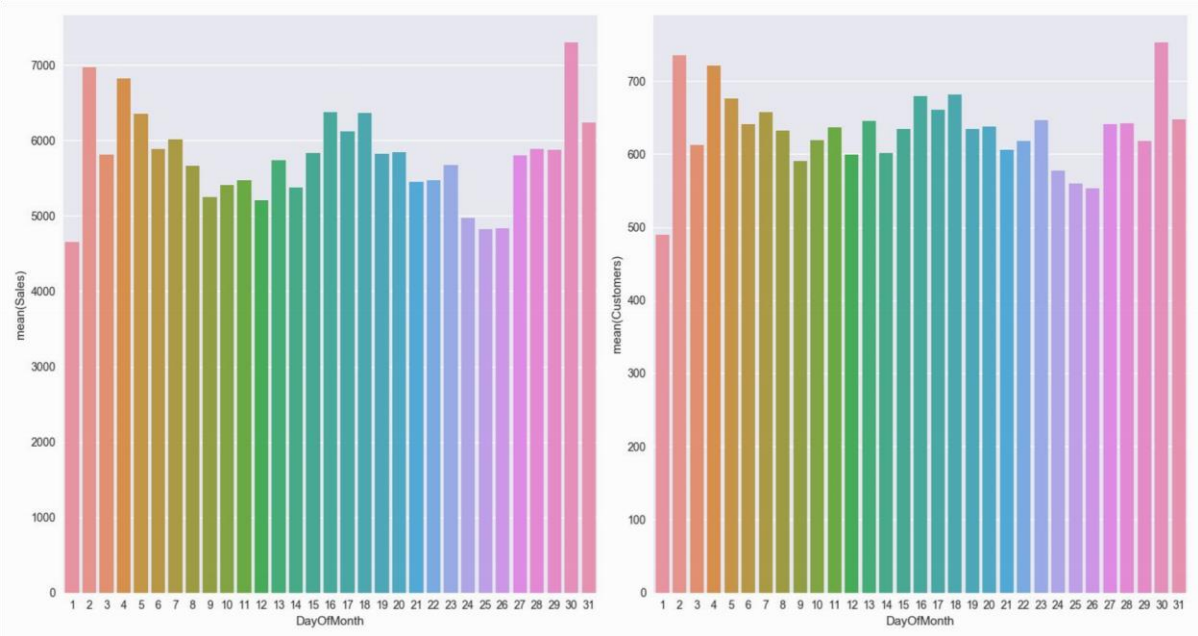
Average Sales for each Store:



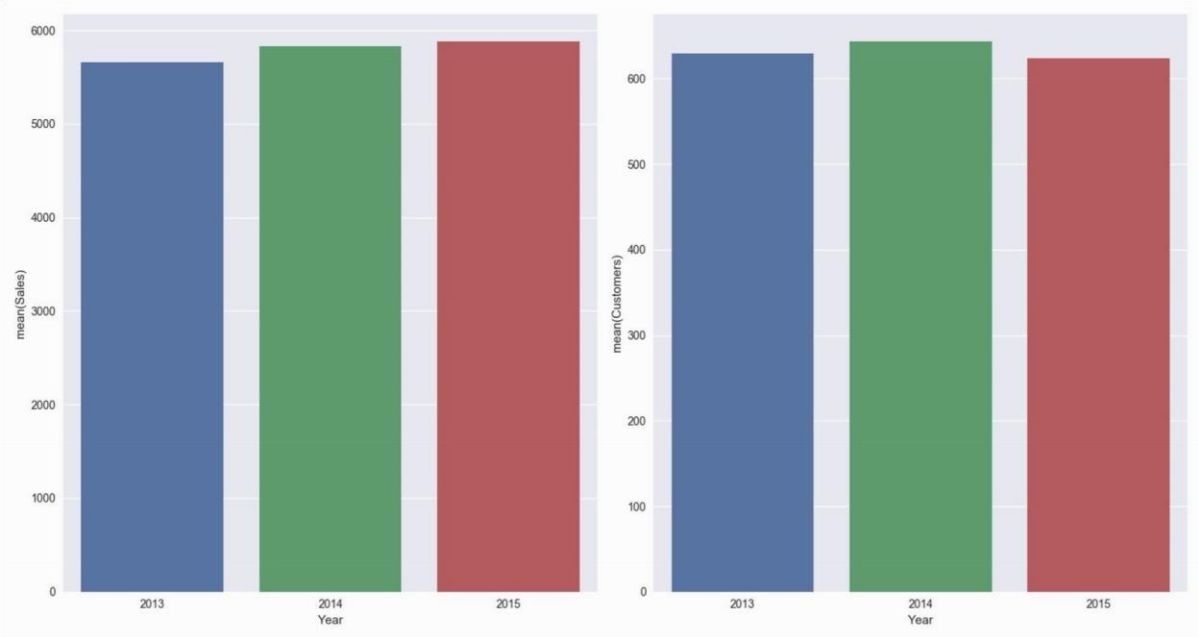
Average Customers for each Store:



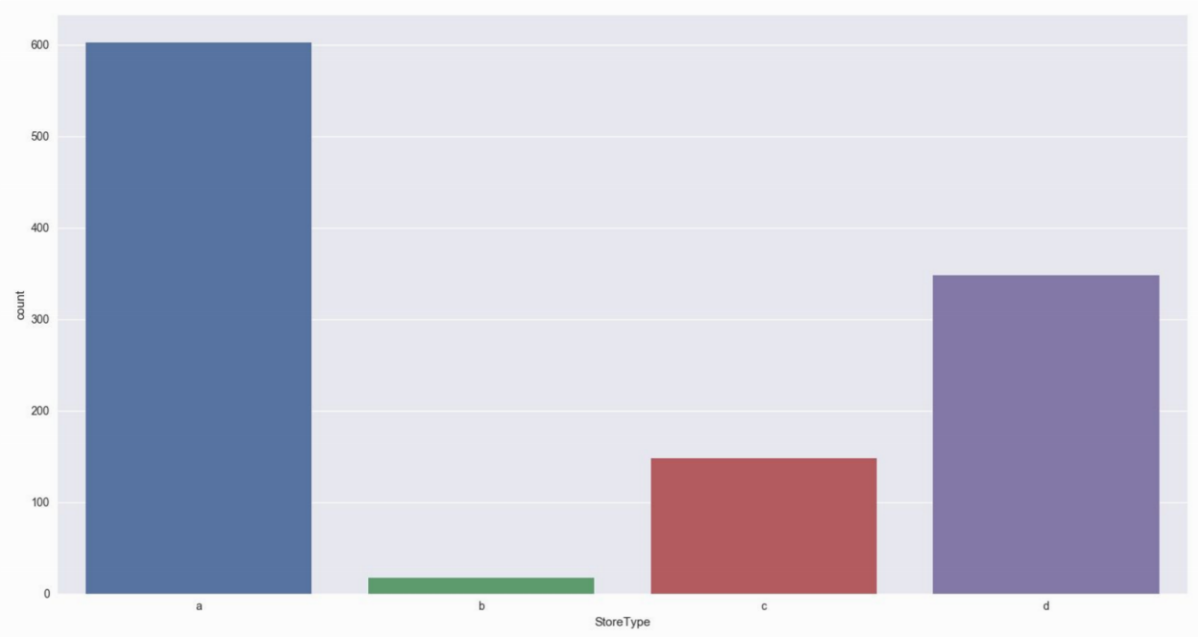
Average Sales & Average Customers by Day of Month:



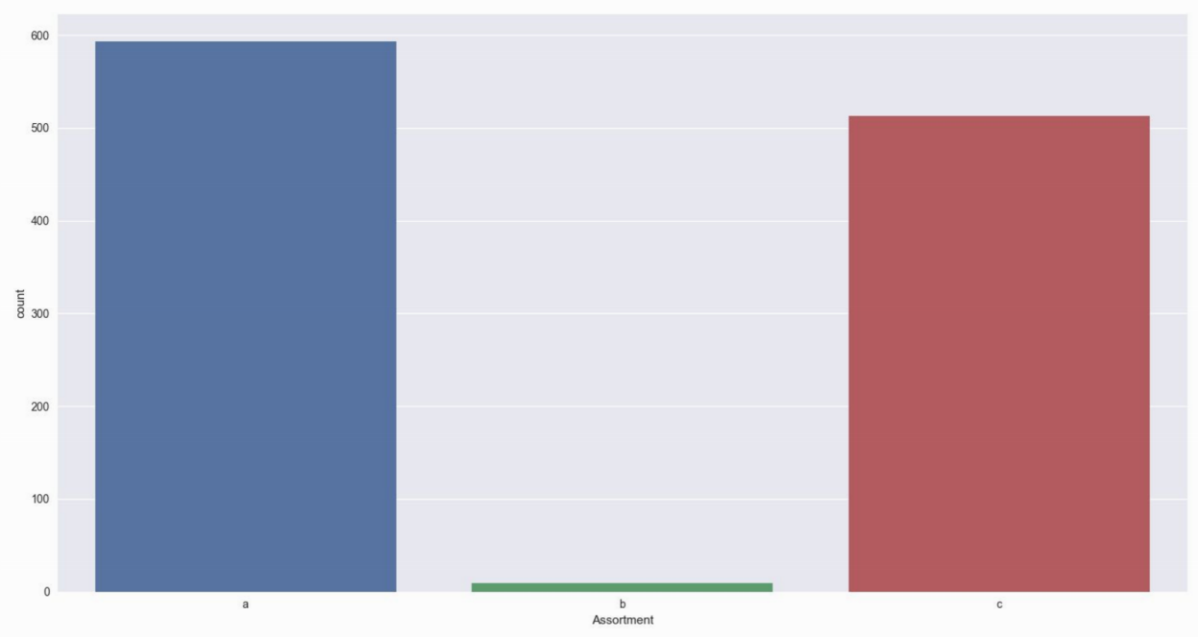
Average Sales & Average Customers by Year:



No. of Stores (by Store Type):



No. of Stores (by Assortment):



No. of Stores Closed/Open (by Day of Week):

Prediction for next 6 six week sales data

```
Store Number 10 will sell €196,208.53 in the next 6 weeks
Store Number 12 will sell €286,257.50 in the next 6 weeks
Store Number 22 will sell €181,909.70 in the next 6 weeks
Store Number 99 will sell €192,517.90 in the next 6 weeks
```

Conclusions

The Rossmann Store Sales problem is a very interesting data science problem to solve. We observed that the problem is more focused on feature engineering and feature selection than on model selection. We spent around 70% of our time analyzing the data for trends in order to make feature selection easier. We attempted several models using different features and assumptions. We concluded that ensemble learning improves the prediction accuracy considerably. We learnt the benefits of boosted trees and their applications in machine learning. However, complicated ensemble learning approaches may not be practically applicable in many scenarios as it tends to overfit the training data.

In the future, we want to implement the backward sliding window of six weeks to converge the values of the weights & *correction_factor* in our ensemble of boosted trees as opposed to