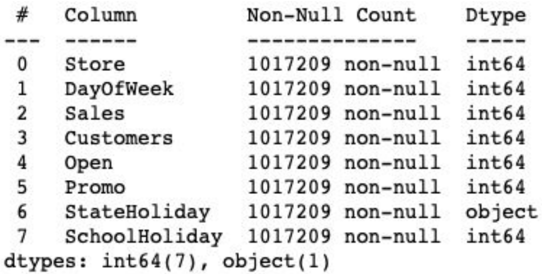
Store Sales Prediction Using Time-series Models

Ashwathy Menon (am5683)
Gefei Zhang (gz2315)
Shangzhi Liu (sl4973)
Yash Jain (ycj2103)
Yuzhao Liu (yl4897)

Data Exploration in train.csv

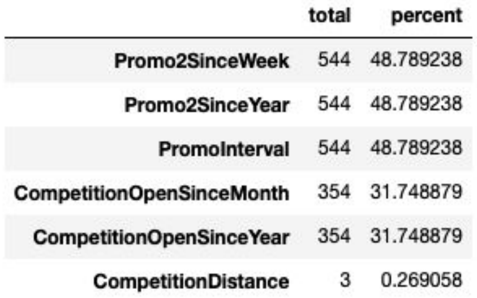
- The data set downloaded from Kaggle is split into three files train.csv, test.csv, store.csv.
- train.csv contains data about the Sales (label), Customers, Promo, School Holiday, State Holiday etc
- On Exploring the data set we found that there are no missing values in train.csv
- Since our project goal is to forecast sales, we only need to focus on the rows whose the value of ‘open’ is equal to 1. Open = 0 would imply that the store is closed resulting in zero sales for the day. Since these 172817 rows don’t add specific value to our analysis we decided to drop these rows.
- Also we notice in 40 rows, although the store was open, there was no sales, no customers, and was not affected by state holiday or school holiday. We assume these sales were incorrect or missing and we will drop these rows.



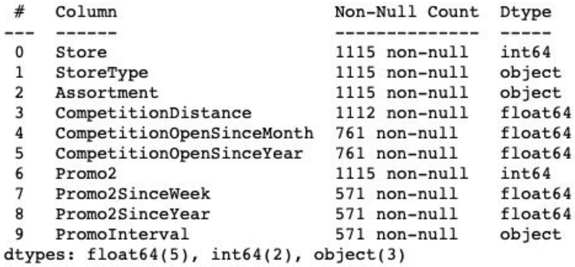
```
 #   Column         Non-Null Count    Dtype
---  ------         --------------    -----
 0   Store          1017209 non-null  int64
 1   DayOfWeek      1017209 non-null  int64
 2   Sales          1017209 non-null  int64
 3   Customers      1017209 non-null  int64
 4   Open           1017209 non-null  int64
 5   Promo          1017209 non-null  int64
 6   StateHoliday   1017209 non-null  object
 7   SchoolHoliday  1017209 non-null  int64
dtypes: int64(7), object(1)
```

Data Exploration in store.csv

- Store.csv has data about the assortment types, store type, promo 1, promo 2 and store competition.
- When Promo2 is 0, the variables derived from Promo2 such as Promo2SinceWeek and Promo2SinceYear are not applicable.
- Similarly, even though CompetitionDistance is only missing for 3 rows, the features related to the same like CompetitionOpenSinceMonth and CompetitionOpenYear are missing for approximately 40% of the dataset.
- However most of the competitor data implies that competition started since 2000s and since our train data set focuses on the years 2013, 2014 and 2015, the date when competition began does not add much value to the analysis as all the data we have is post the competition start.
- The images to the side represent the missing values and percentages and the general layout of the features.



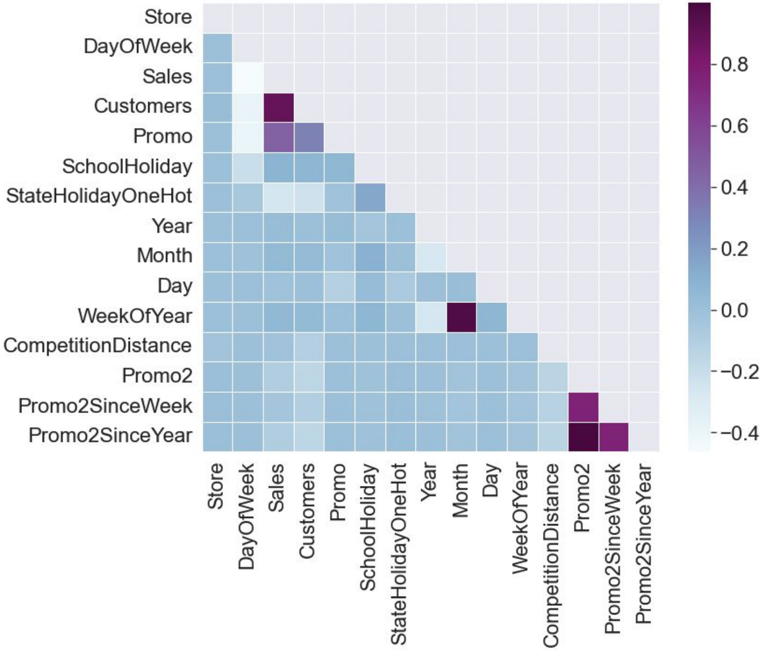
	total	percent
Promo2SinceWeek	544	48.789238
Promo2SinceYear	544	48.789238
PromoInterval	544	48.789238
CompetitionOpenSinceMonth	354	31.748879
CompetitionOpenSinceYear	354	31.748879
CompetitionDistance	3	0.269058



```
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Store                      1115 non-null   int64
 1   StoreType                  1115 non-null   object
 2   Assortment                 1115 non-null   object
 3   CompetitionDistance        1112 non-null   float64
 4   CompetitionOpenSinceMonth  761 non-null    float64
 5   CompetitionOpenSinceYear   761 non-null    float64
 6   Promo2                     1115 non-null   int64
 7   Promo2SinceWeek            571 non-null    float64
 8   Promo2SinceYear            571 non-null    float64
 9   PromoInterval              571 non-null    object
dtypes: float64(5), int64(2), object(3)
```

Correlation among various features

- After plotting the correlation matrix, we can see that sales and customers have a very high correlation which is intuitive
- Also, it can be seen that the time there is a promo on the store, the sales are positively correlated. This aligns with the tendency of humans to shop more if there is a sale/offer



Cleaning for store.csv and train.csv

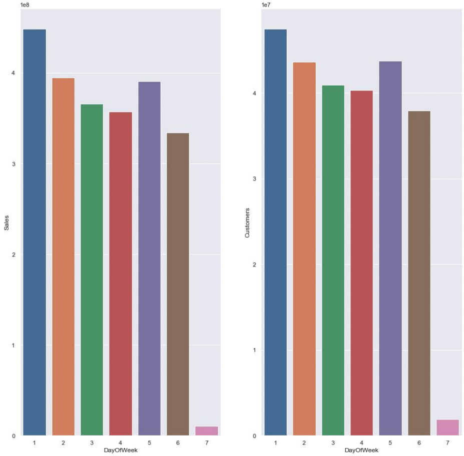
- The features competition open since month and year have 40% of the data missing and based on our current dataset, for non-missing tuples competition started as early as the year 2000. Hence, these rows don't give insight into pre and post competition analysis and considering the percentage of missing values we decided to drop these features.
- Similarly, Competition Distance is only missing for 3 rows, since the percentage of missing data is significantly less we replace the missing values with the median value of the feature.
- When Promo2 is not applicable the features derived from it have null values in the data set. Since these missing values have an underlying implication that promo2 is not applicable we replace the null values with 0.
- Categorical Features
  - StoreType: We apply one hot encoding to the same as there is no underlying "order"
  - Assortment: We apply ordinal encoding to the same as there is an order basic, extra and extended
  - PromoInterval: We apply one hot encoding to the same as there is no underlying order and it is important to capture seasonality. Promo during specific months like holidays would result in an increase in sales.
- In train.csv, the state holiday has values as 0, 0, a,b ,c we clean the data by converting both the 0 values to int and a,b,c to 1 as for our analysis we just need to know whether the state holiday was applicable on the day or not.

Sampling for store.csv and train.csv

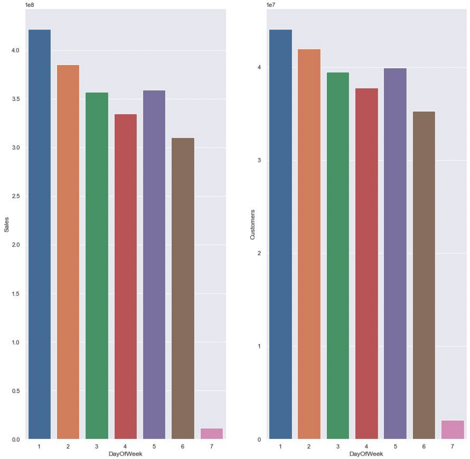
- We merge the two tables into one by combining the same based on store id, since the store data is consistent for all the train samples.
- As this is a time-series problem, it is important that we use Structured Splitting for train, validation and test.
- We sort the data set by time and then split the train.csv into train and validation set and also sort our test.csv by date which will be the test set.
- There is no need to under-sample or over sample the data set for the current problem statement of time series sales forecasting.

Sales and Customer per Day of the Week Per year

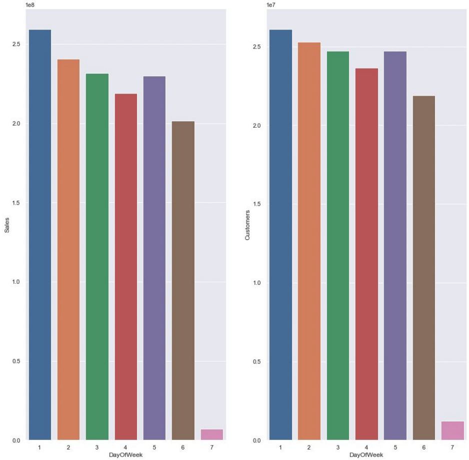
Year = 2013



Year = 2014



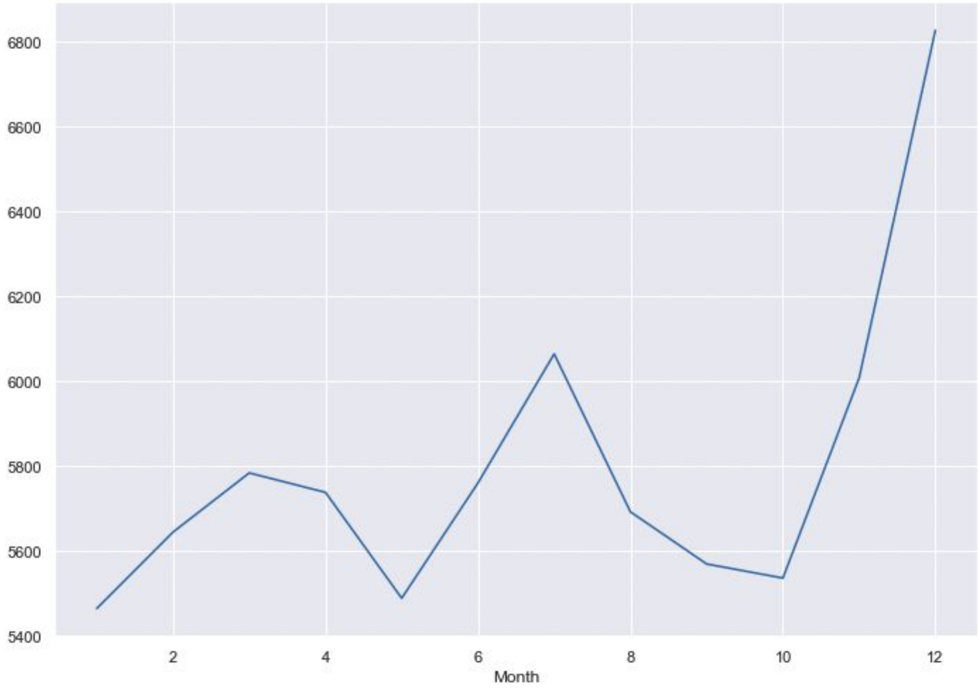
Year = 2015



- For all the years, the sales and customers have been very low on Sundays. This can be attributes to the fact that a lot of store are closed on Sundays
- Also, the sales and customers are the highest on Mondays for every year. This could also be because a lot of stores are closed on Sundays

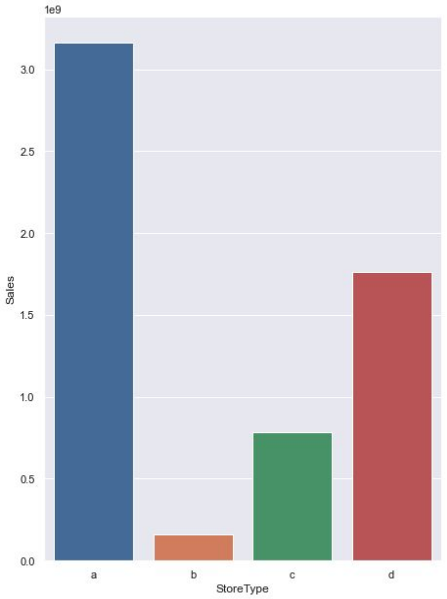
Sales per Month of the year

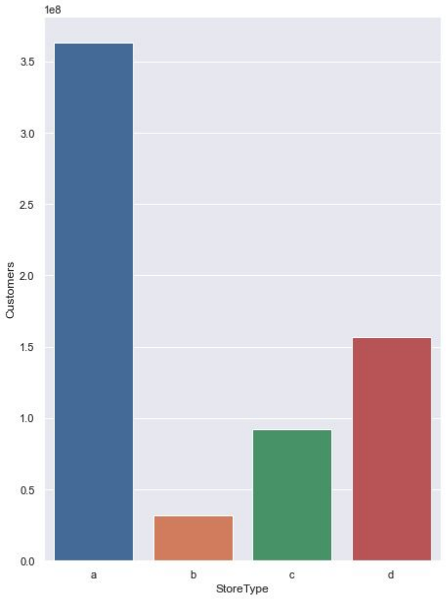
- The sales increase during the end months of the year
- This can be attributed to the fact that end of the year has a lot of festivals like christmas and thanksgiving. Hence, the sales might increase during that time



Sales and Customers per store type

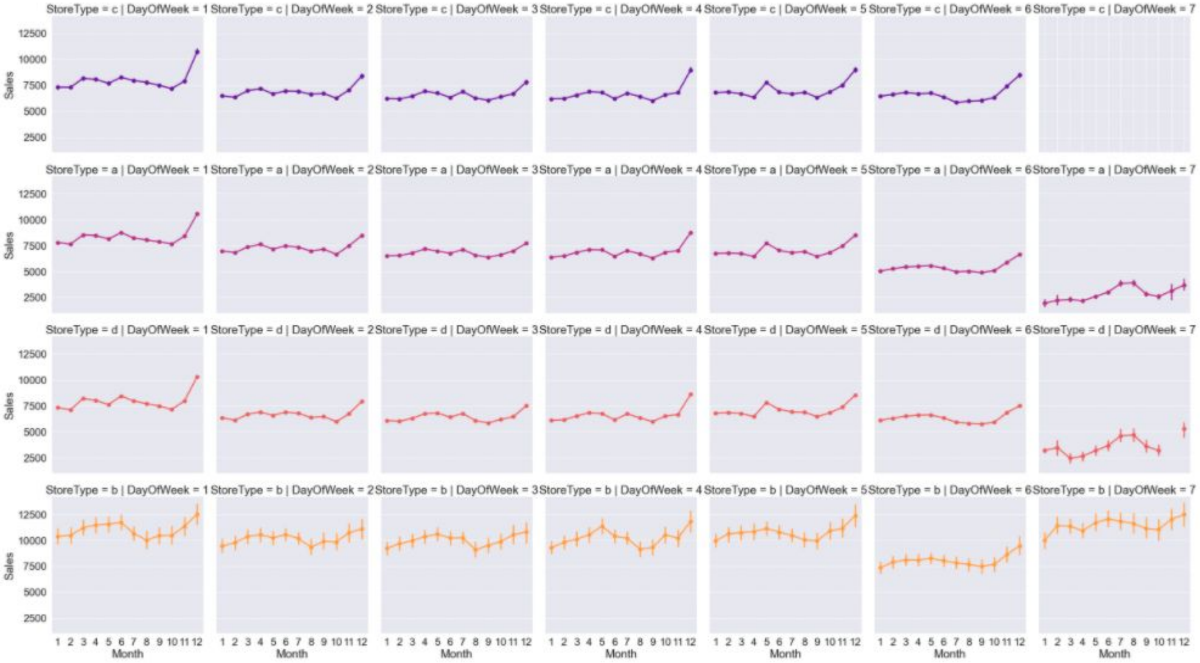
- The sales and customers are the most for type a stores compared to other store types
- This could be due to multiple factors like no competition nearby, good products, etc





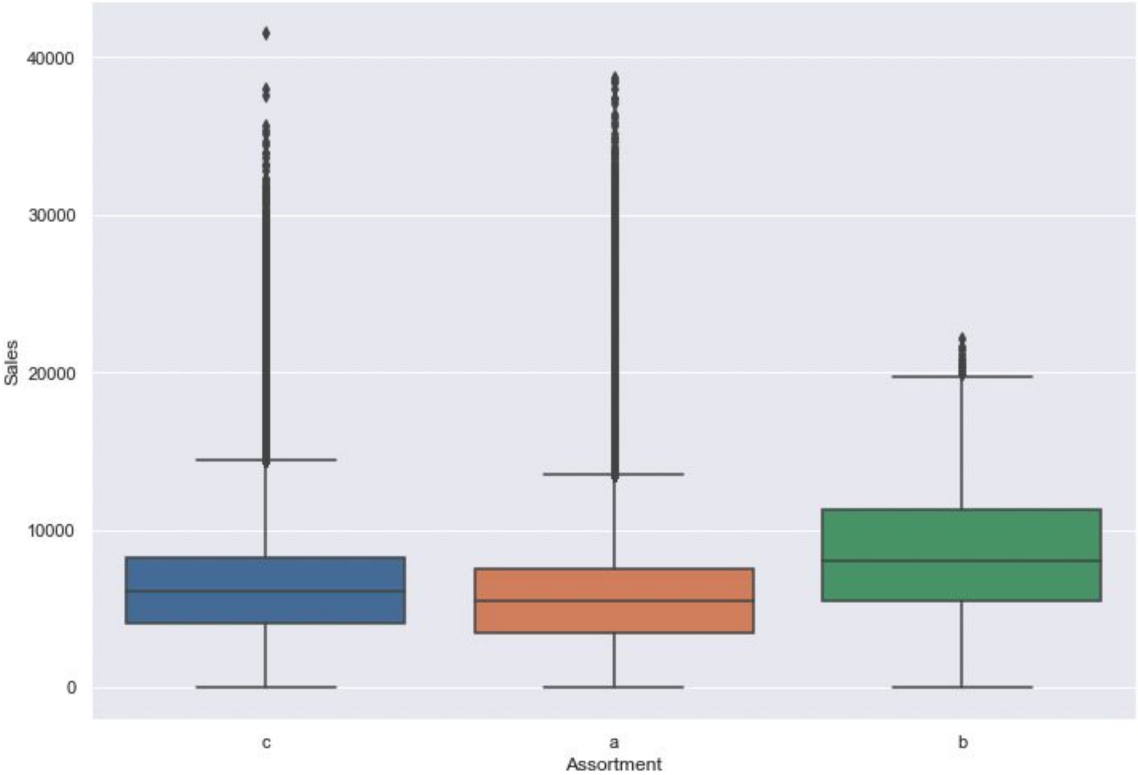
Sales per day of week per store type per month

- It is interesting to note that only store of type c is closed on every Sunday
- Also, store of type b is closed on sundays for the months of October and November



Sales per assortment type

- The stores with assortment of type extra have a higher median sales value
- However, it is interesting to note that store with assortment basic and extended have a lower median sales value but a lot of sales with a very high value



Machine Learning Models Working in Progress

- **Facebook Prophet**

Prophet is known to work best with the time series data that have strong seasonal effects and several seasons of historical data. In our project, we would also be considering seasonality and holiday effects (indicated by the StateHoliday column), hence this methodology can be resourceful. Besides, in forecasting the sales, Prophet can return the detailed components of the forecasts. This can help us to understand the forecast results and the patterns. This model will be implemented in the project by using fbprophet package.

- **Tree-Based Model**

A time series problem can be solved using a tree-based approach. It requires that the time series data be transformed to a supervised learning problem first and then a tree-based approach can be used. While using a tree-based model we need to use a special validation technique called walk-forward validation as using k-fold cross validation will give biased results.

- **ARIMA**

An ARIMA model is a class of statistical models for analyzing and forecasting time series data. It explicitly caters to a suite of standard structures in time series data, and as such provides a simple yet powerful method for making skillful time series forecasts. We aim to implement and compare both ARIMA and Prophet as they are the most popular time-series prediction models.