Seoul Retail Case

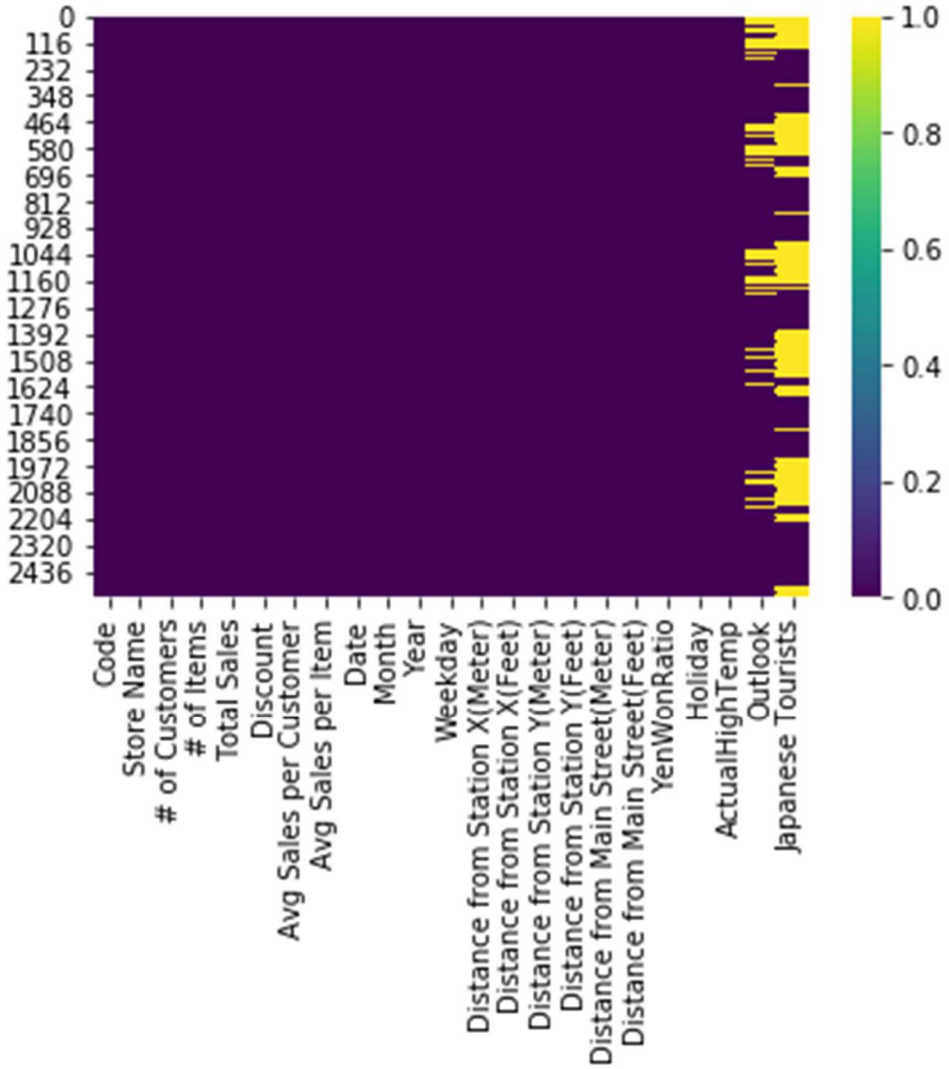
MGS616 – Predictive Analytics - Group Homework

Deepak Rajan Rishi Raj Milind Nandankar
Samadhan Sonwane Poorna Prithvi Subramaniam

5/10/18

Data Preprocessing:

- First, the dataset (Korea data) was checked for any missing values. By using a heat map (as shown below), we came to know that the variables **'Outlook'** and **'Japanese Tourists'** had many missing values.



The yellow lines above show missing (Nan) values.

- For cleaning the **'Outlook'** variable, it was checked for all its unique values. The findings were:

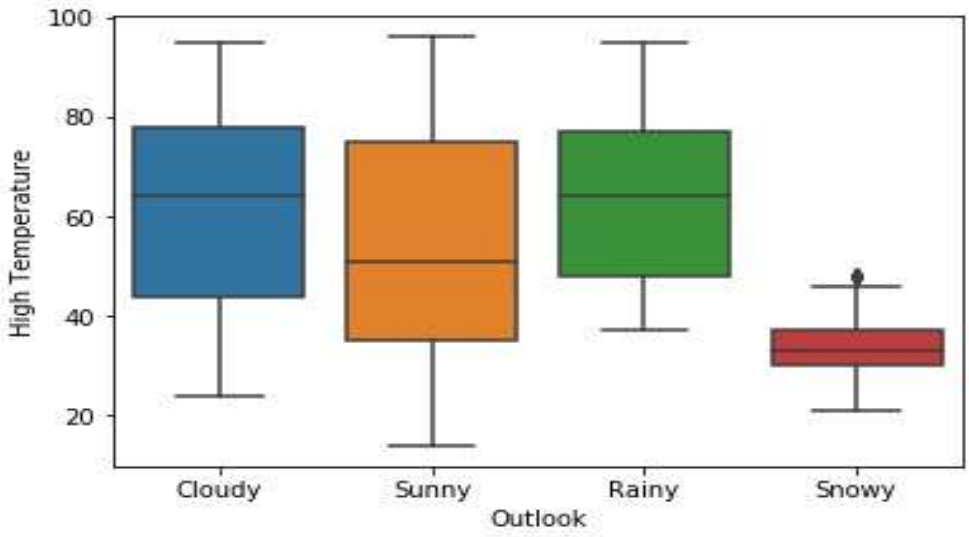
  [nan, 'Cloudy', 'Sunny', 'Rainy', 'Snowy', 'rainy', 'cloudy']

  ➔ 'nan' are the missing values which were found in above step.

  ➔ The other discrepancies in this variable include the values '**Cloudy**' and '**Rainy**' which are present as '**cloudy**' and '**rainy**' as well.

- <u>Predicting the 'nan' values in 'Outlook' variable</u>:

  ➔ We suspected that the 'Outlook' variable was dependent on the 'ActualHighTemp' variable and tried to figure out any dependencies.



  ➔ We trained **Random Forest** algorithm to predict the 'nan' values in outlook based on the ActualHighTemp variable. Below is the snapshot of the accuracy of the model:

  <u>Classification Report:</u>

	precision	recall	f1-score	support
Cloudy	0.40	0.33	0.36	458
Rainy	0.52	0.40	0.45	503
Snowy	0.75	0.14	0.24	213
Sunny	0.57	0.78	0.66	1018
avg / total	0.54	0.53	0.51	2192

Confusion Matrix:

	0	1	2	3
0	152	66	5	235
1	110	199	0	194
2	5	8	30	170
3	114	108	5	791

➔ Due to low precision and f1-score of 54 % and 51 %, respectively, we decided not to go with the predictive model, rather we classified the 'nan' values under a new value called 'missing' value.

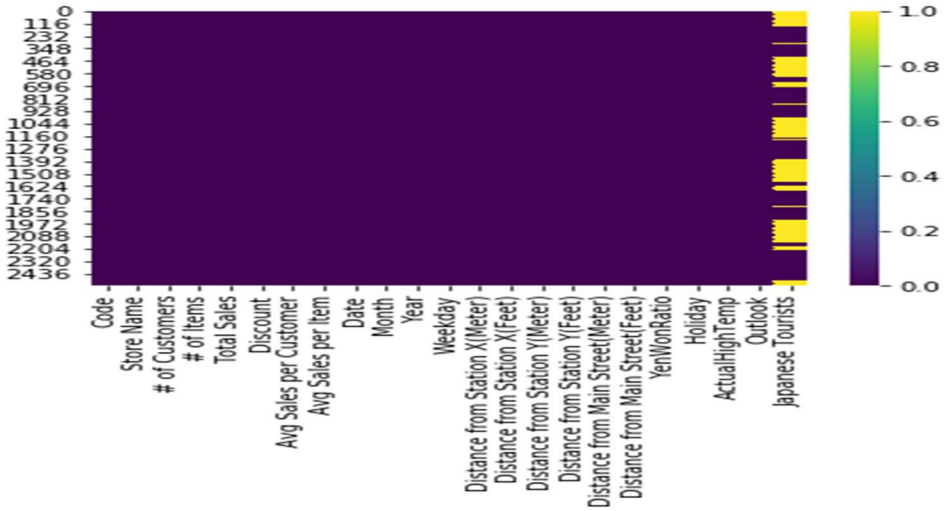
➔ Further, the 'cloudy' and 'rainy' values were corrected as 'Cloudy' and 'Rainy', respectively.

➔ The unique values for the 'Outlook' column was checked again to assure that the discrepancies were taken care of.

['Missing', 'Cloudy', 'Sunny', 'Rainy', 'Snowy']

Now, the variable has **five unique** values.

➔ The heat map below shows that the 'Outlook' variable is now clean and has no null values.



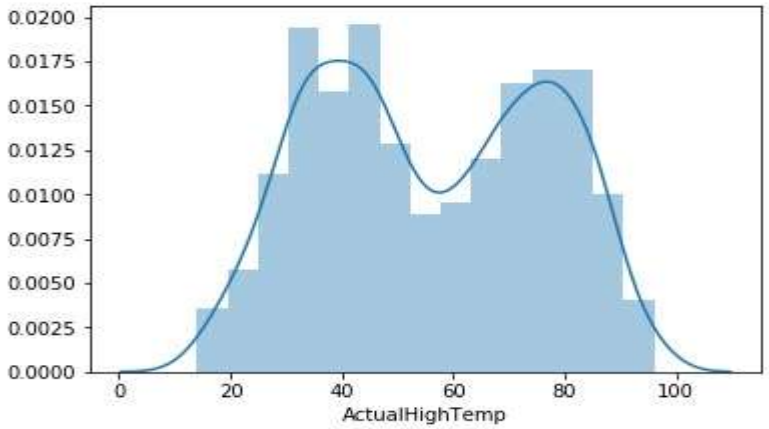
➔ The missing values in the 'Japanese Tourists' column will be predicted using other independent variable.

- **Weekday Variable**: While analyzing the 'Weekday' column, we found out that the week days were not in synchronization with the 'Date' variable. So, we used the date-time function to correct all such anomalies.
  e.g. Initial Value (1/1/2012 was a Sunday) Corrected Value

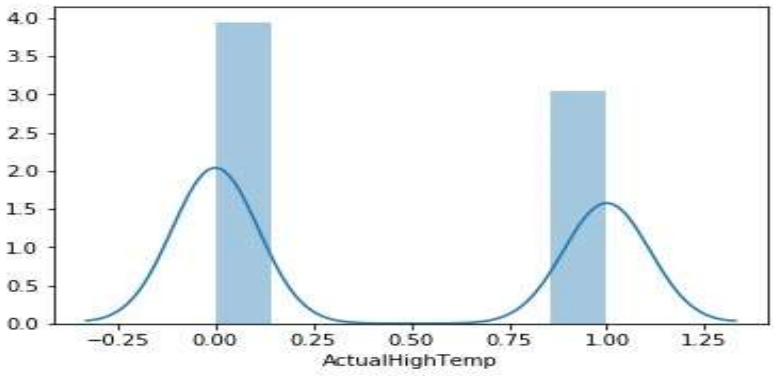
Date	Month	Year	Weekday
1/1/2012	1	2012	Monday
1/1/2012	1	2012	Monday
1/1/2012	1	2012	Monday
1/1/2012	1	2012	Monday
1/1/2012	1	2012	Sunday

Date	Month	Year	Weekday
1/1/2012	1	2012	Sunday
1/1/2012	1	2012	Sunday
1/1/2012	1	2012	Sunday
1/1/2012	1	2012	Sunday
1/1/2012	1	2012	Sunday

- **ActualHighTemp Variable:** While analyzing this column we could see that there were two categories in which the values were distributed (below and above 60 degrees)



So, we changed it into two categorical variables, namely **0(for temperature <= 60) and 1 for temperature > 60)**. The resulting plot is below:



- **Code and Store Name Variables**:

  ➔ Looking at the Code and Store Name columns we found out the following:

	Code	Store Name	count
0	20002	Store A	198
1	20036	Store C	198
2	20054	Store E	563
3	20240	Store D	559
4	20288	Store B	563
5	20488	Store A	271
6	20610	Store C	195

  ➔ Since, **Store A** was non-operational between the dates March 17 2012 and June 19 2012 and it reopened under a new ownership at the **same location**, we changed the initial Store A 'Code' (20002) with the new 'Code' (20488).

  ➔ **Store C** was reopened at a **different location**, so we are considering it as a new Store in addition to the Store C.

  The modified Code along with its corresponding Store Name is below:

	Code	Store Name	count
0	20036	Store C	198
1	20054	Store E	563
2	20240	Store D	559
3	20288	Store B	563
4	20488	Store A	469
5	20610	Store C	195

- **Discount Variable**: Some of the discount values were found to be **negative** (38 values) which is not possible in a practical setup.

```
In [27]: korea_data[korea_data['Discount']< 0]['Discount'].count()
Out[27]: 38
```

  We transformed the variable into categorical variable with below categories

  0 -> zero or negative discount

  1 -> positive discount

Data Modelling:

- **Predicting Missing Japanese Tourists**:

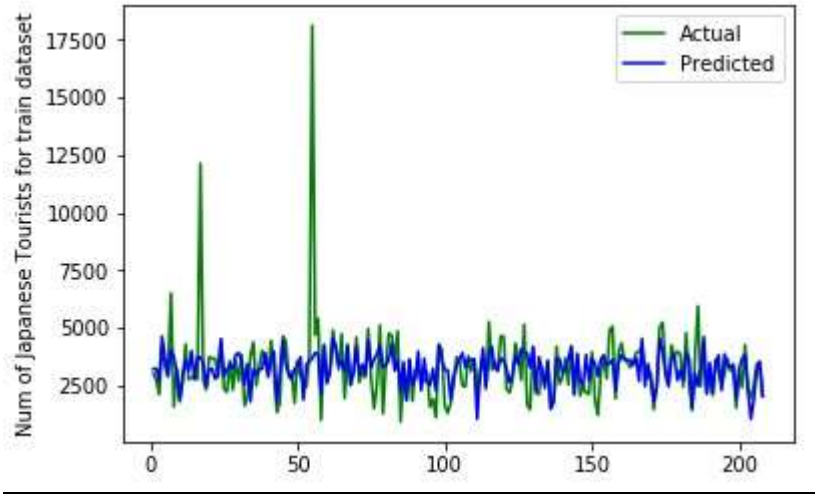
➔ We suspect that the number of Japanese Tourists visiting Seoul to be dependent on the following six variables:

1. 'Month'
2. 'Weekday'
3. 'YenWonRatio'
4. 'Holiday'
5. 'ActualHighTemp'
6. 'Outlook'

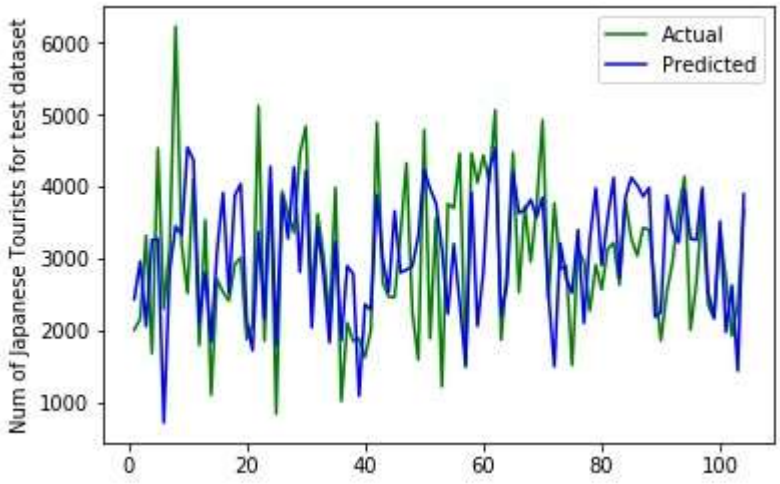
➔ The multiple Linear Regression Model was trained for the data for which Japanese Tourist were not missing to predict the cases where number of Japanese tourists were missing.

We selected 2/3 of the data for which number of Japanese Tourist available as training set and remaining as test set.

Below is the plot of the actual values of number customer verses predicted values for the training set.

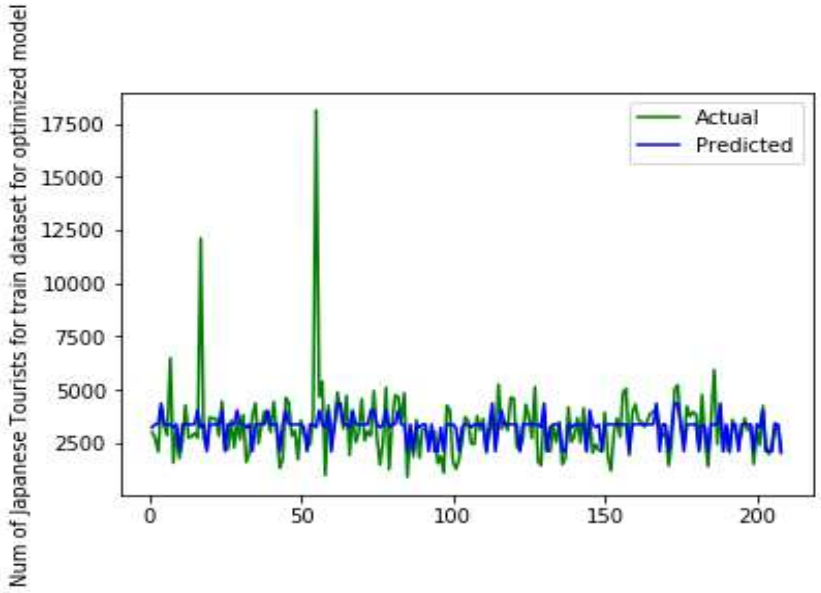


Below is the plot of the actual values of number customer verses predicted values for the test set.



We tried to optimize the predictions with backward elimination method and could understand that Number of Japanese Customer depends only on Actual high Temperature and Month.

Below is the plot of the actual values of number customer verses predicted values for the test set for optimized model. There are few spikes but very few.

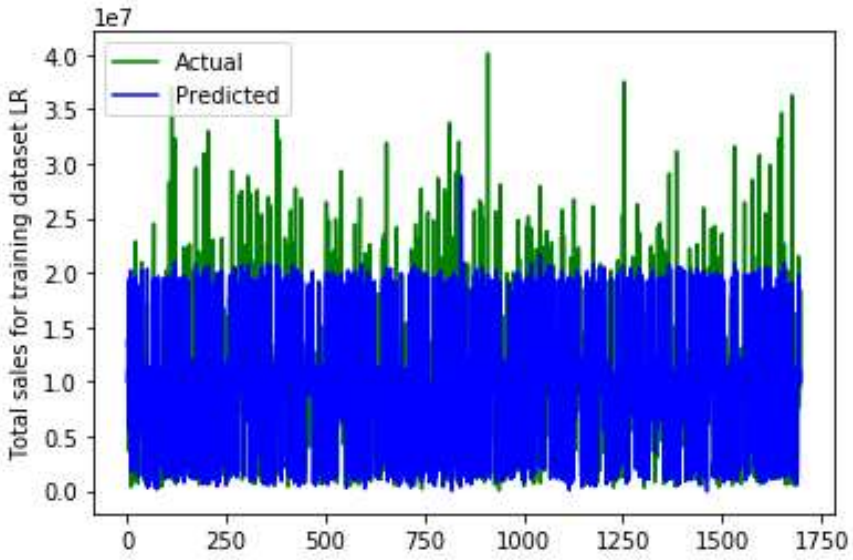


We performed the prediction for the missing number of Japanese Tourists visiting Seoul using these optimized model.

- **Predicting Total Sales pf each stores**:

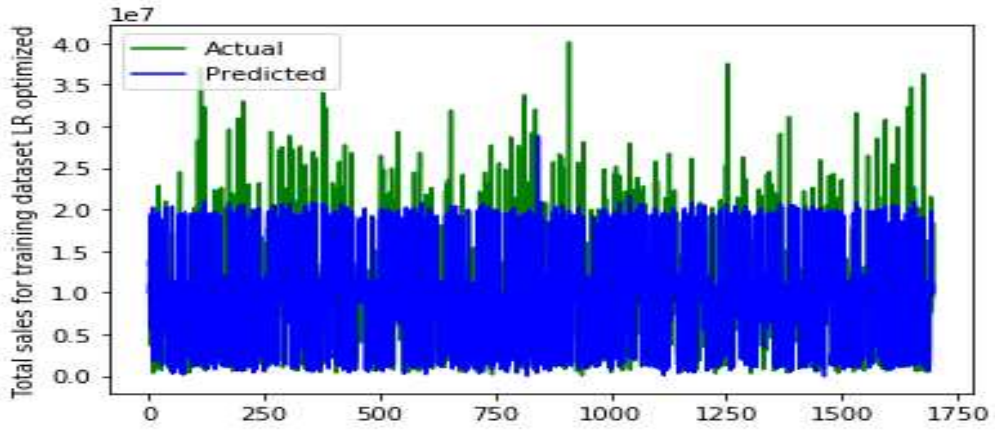
  As per our observation, we decided to design a model for prediction of total sales using Code of the stores, number of Japanese Tourists visiting Seoul and discount available, as all other variables are doesn't relate total sales.

  We used similar approach of training linear model using above three variables.



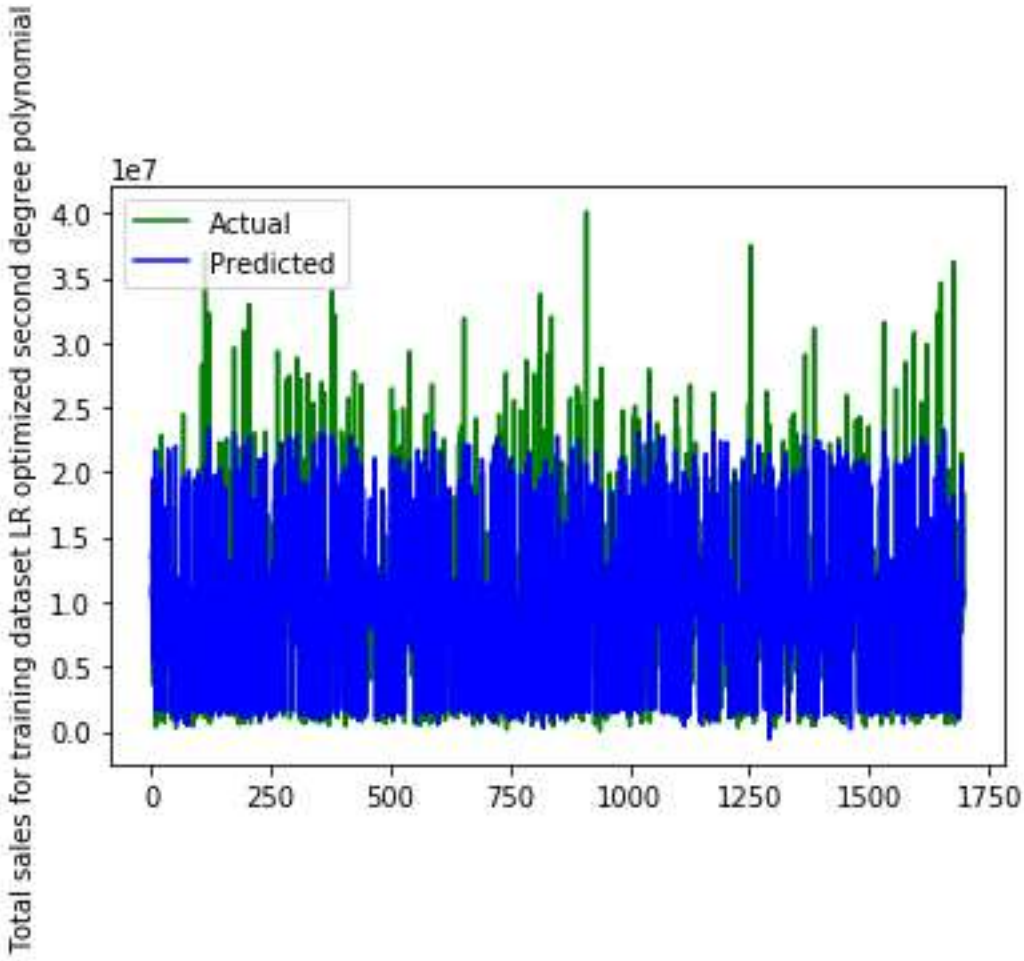
Certainly predictions doesn't look good. We then tried to optimized the model using backward elimination method and could find discount is not relevant to the total sales of each store.

We modeled the linear regression model without discount.

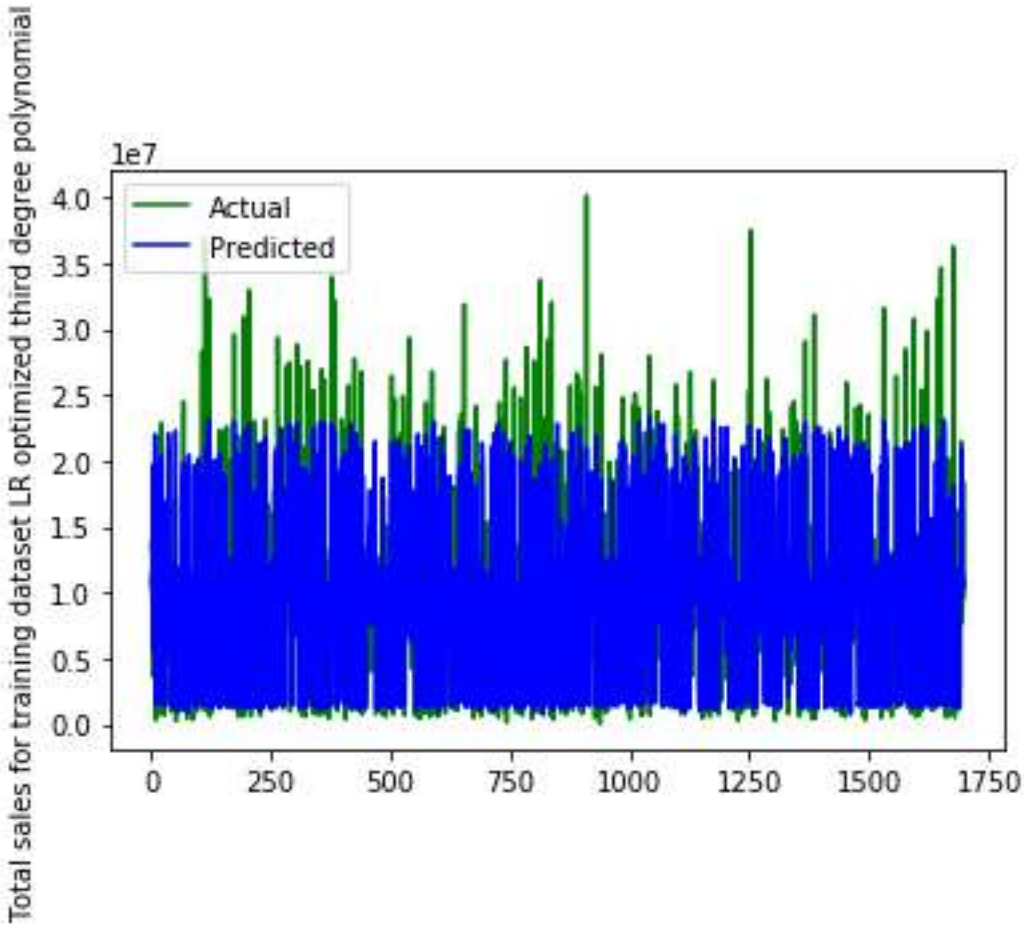


This model is also not performing to the expectations.

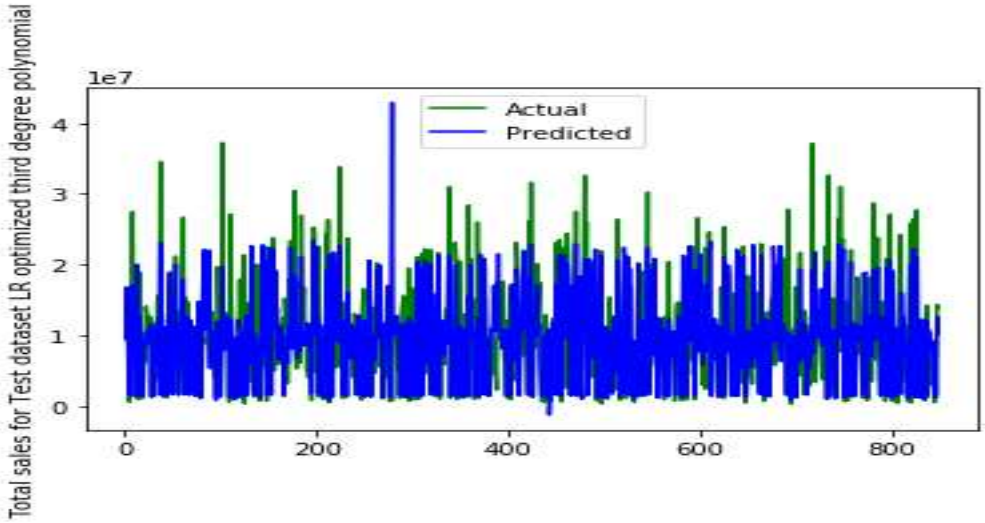
We then tried to use second degree polynomial of the training set and results improved, now we are getting better predictions but still not to the expectations.



To further improve the performance of the model, we used third degree polynomial of the training set. We are ow able to get the more correct predictions.

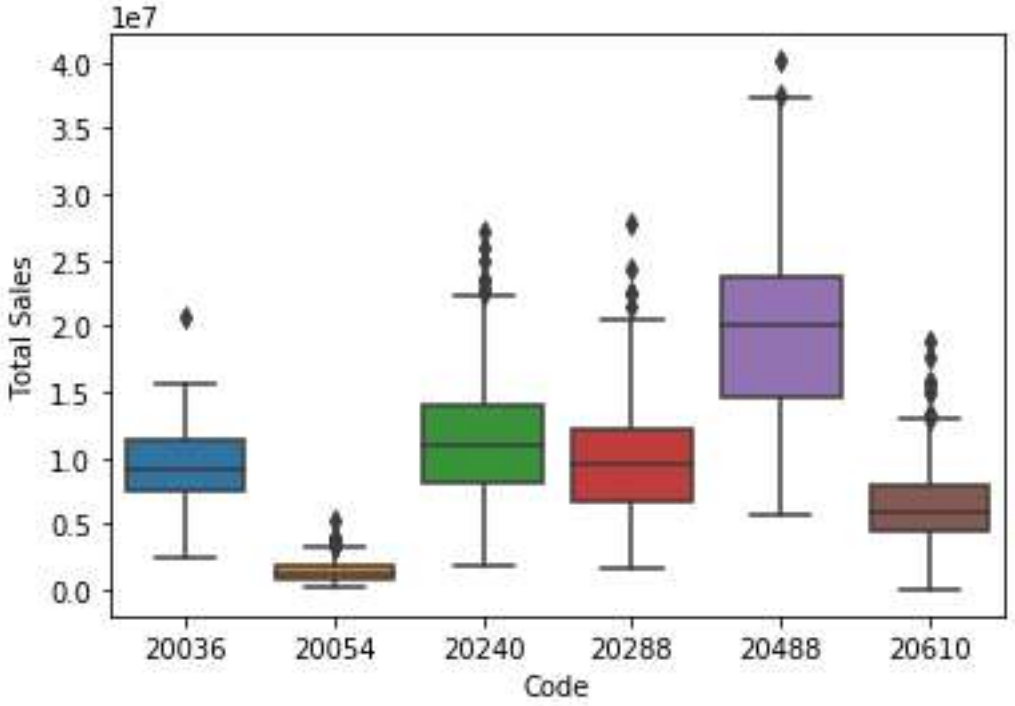


We tested the model for test set and predictions are better.

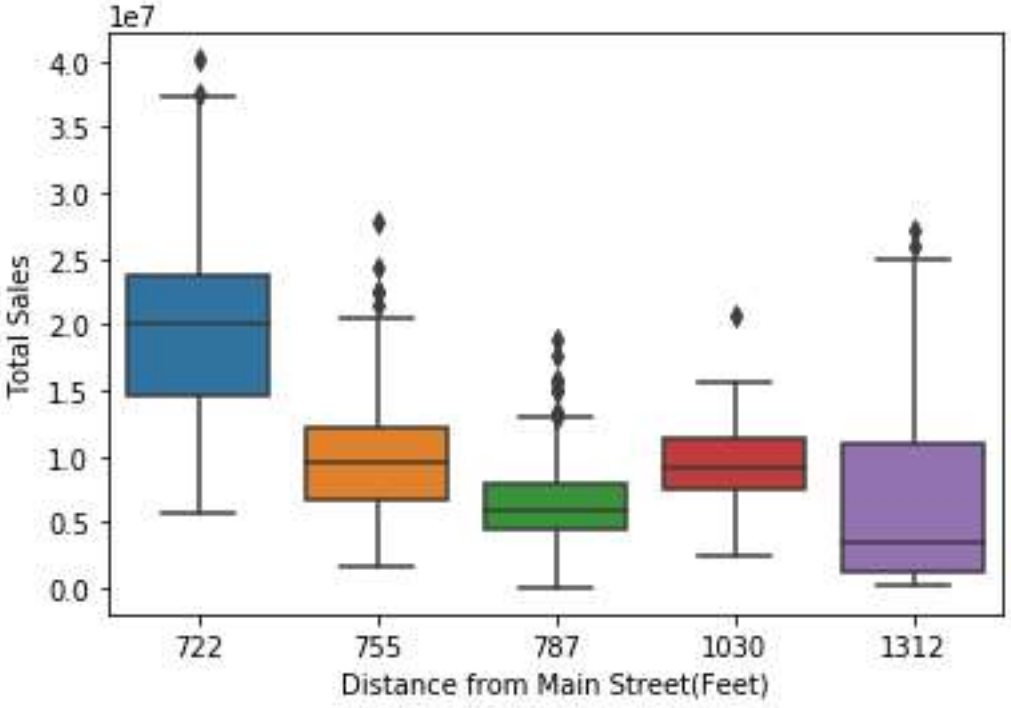


We tried to use fourth degree polynomial but it was overfitting the training set and was not performing well on test set, so we decided to go ahead third degree polynomial of the input features, which are store code (store name) and number of Japanese tourists visiting the Seoul.

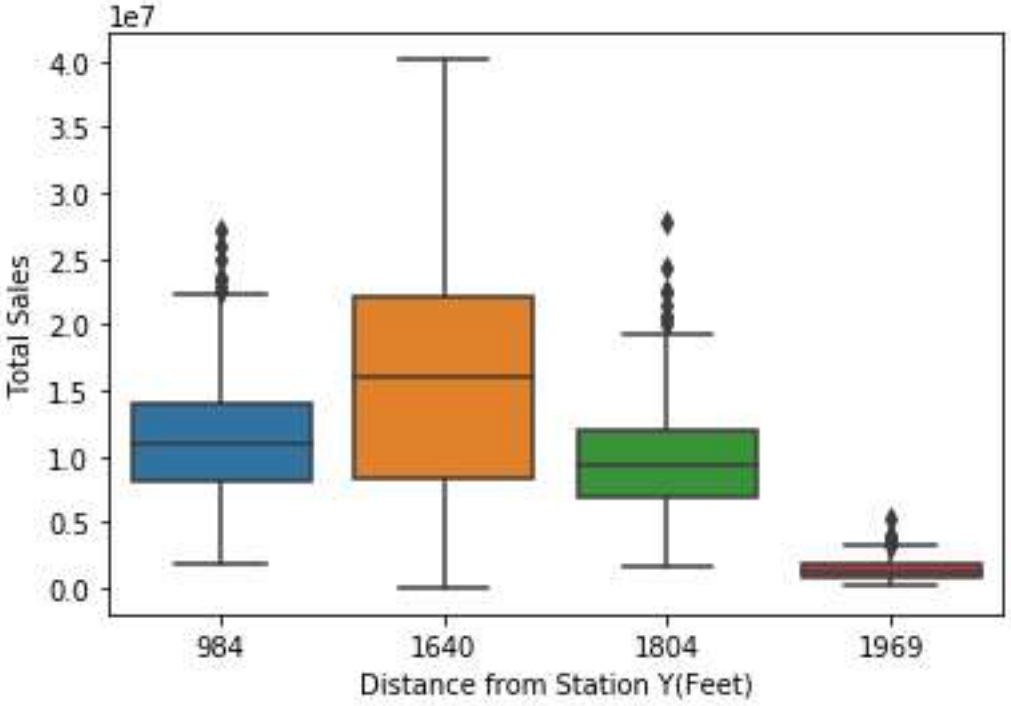
Further, we tried to understand the reason why different store have different sales on same day



With our observation, we see that main reason for this is the distance of the store from the main street. As the distance increases, sales of the store decreases.



Distance of the store from station Y also has impact on the total sales.



Conclusion:

Total sales of the each store are directly related to distance of the store from main street and station Y and number of tourists visiting Seoul on that particular day.

Number of tourists visiting the Seoul is related to Actual high temperature and month. Few months have higher visits by the tourists.