An Analysis of Pronto , A Failed Bike Share System in Seattle

By Satyam Tandon
January 7th,2018

Epilogue:

Pronto was a bicycle share system in Seattle, Washington owned and operated by Motivate from 2014 to February, 2016. Motivate , which also runs bike share programs in ten other major cities such as San Francisco, Washington DC, Portland and New York, sold Pronto to the Seattle Department of Transportation in February 2016 which then had about 50 stations across the city and owned 500 bicycles. Expansion was necessary due to lack of bicycle stations in many popular and touristy areas of the city but the realization came late and expansion was delayed by the city which led to hazardously low ridership for Pronto which ultimately led the city to shut down the bicycle sharing system in March, 2017 altogether to focus funds elsewhere. Other factors such as the topography and climate were also thought to have played a part in Pronto's demise.

For more information on the challenges faced by Pronto - https://www.citylab.com/transportation/2017/01/seattle-bike-share-pronto-goes-under/513575/

Client and Problem:

My hypothetical client would be the Seattle Department of Transportation, Motivate or some other company who is planning to restart the bicycle share system in Seattle and wants to gain better insight into the various factors, relationships ,trends and demographics that could have helped to make Pronto a more successful system in Seattle and also determine the extent of the role of speculated factors such as weather on Pronto's ridership.

Objectives:

- Analyzing customer trends and preferences to be more informed about Pronto's customer base by segmenting the customers on grounds of memberships (subscribers) and pass-holders (guest/non-subscribers)
- Determining the most dominant weather attributes and how they affect daily ridership numbers in Seattle to be able to implement measures and incentives to balance loss of ridership on days it is necessary
- Fitting a machine learning model on the data that can initially predict with some accuracy the final destination of a member rider based on features such as , the starting point, member's age, sex , day of the week, hour of the day etc.

The results of the model can be used to improve customer experience and even be used to leverage secondary income for the ridership company from sources such as targeted advertisement.

Original Datasets:

The datasets used for this projects are freely available on Kaggle - https://www.kaggle.com/pronto/cycle-share-dataset/data for anyone to use. The data is contained in 3 separate files.

- Station.csv contains data about the 58 Pronto cycle docks around Seattle
- Trip.csv contains data about every trip taken using the Pronto Cycle share system in Seattle from 2014 to 2016
- Weather.csv contains data about the weather in Seattle that corresponds to the dates of the recorded trips in Seattle in the Trips.csv file.

All three datasets can be looked at as relational databases. The station dataset which contains data about each of the 58 bicycle stations that existed in Seattle such as the latitude, longitude, number of bicycle docks at each station, modification and decommission date if there were any contains a column 'station_id' which can be used to merge the dataset with the trips dataset which contains the station_id from where a trip started and where the trip ended, depending on what we are trying to gauge. The trips dataset also contains a column for the dates on which the trips recorded in the datasets were taken, which can be used to merge the dataset with the dates column in the weather dataset which contains various weather attributes for those dates. This would require some cleaning as the dates in both the datasets was recorded in different formats.

A. Initial Wrangling and Cleaning:

Since the dataset had been previously used for analysis, they were relatively clean but still needed further wrangling and munging specific to what trends and information was being gathered.

A.1 The station dataset

- contained a column for 'modification_date' of a station which contained the date on which the station dock was modified to hold more or less cycles and null values if the station was never modified after it was initially set up. Since only 17 of the initial 58 stations were modified and it made more sense to work with the latest information about a station, this column was redundant and could be entirely dropped from the dataset.
- The dataset also contained an 'install_dockcount' column which contained the cycle holding capacity of the dock at a station when it was initially set up and the 'current_dockcount' column holds the latest cycle holding capacity of a

dock at a station. The ‘current_dockcount’ and the ‘install_dockcount’ values for a station are the same if it was never modified. Again, since we are only interested in current values for all stations, we can drop the ‘install_dockcount’ column entirely as it is redundant for our analysis.

- When the data was collected, 4 out of the 58 stations in Seattle had already been decommsioned and were no longer active. I will drop these decommisioned stations from the dataset , to work with only the active stations to keep our analysis more precise and accurate which also makes the decommissoned_date column unecessary and I will be able to drop it as well.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 54 entries, 0 to 57
Data columns (total 6 columns):
station_id           54 non-null object
name                 54 non-null object
lat                  54 non-null float64
long                 54 non-null float64
install_date         54 non-null object
current_dockcount    54 non-null int64
dtypes: float64(2), int64(1), object(3)
memory usage: 3.0+ KB
```

Information and the first 3 rows of the cleaned stations dataset

	station_id	name	lat	long	install_date	current_dockcount
0	BT-01	3rd Ave & Broad St	47.618418	-122.350964	10/13/2014	18
1	BT-03	2nd Ave & Vine St	47.615829	-122.348564	10/13/2014	16
2	BT-04	6th Ave & Blanchard St	47.616094	-122.341102	10/13/2014	16

A.2 The trips dataset

- contained some null values only in 2 columns, namely 'gender' and 'birthyear'. These null values only occurred when the trip belonged to a Short Term Pass-holder , which ranges from a 24 hour pass or a 3 day pass as offered by Pronto as compared to a trip made by someone who is a Member, which could be a monthly subscription or a yearly one to the Pronto Cycle Share.For the purpose of cleaning the null values and for a more precise analysis, I split the trip data set on the basis of the client being a Member or a Temporary Pass Holder into 2 separate datasets.

- The trip dataset contained 181,557 trips where the user was a member and 105,300 trips where the user was a Short Term Pass Holder.

A.2.1 To create the Member only sub dataset, I

- Extracted all the rows from the original trip dataset where the column usertype contained the value 'Member' and stored it in separate dataset.
- Dropped any rows with null value in gender or birthyear column if any remained.
- Dropped the usertype column entirely as it had become redundant for this dataset.

Information and the first 3 rows of the cleaned **member only** sub dataset created from the original trips dataset

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 181553 entries, 0 to 286848
Data columns (total 11 columns):
trip_id              181553 non-null int64
starttime            181553 non-null object
stoptime             181553 non-null object
bikeid               181553 non-null object
tripduration         181553 non-null float64
from_station_name    181553 non-null object
to_station_name      181553 non-null object
from_station_id      181553 non-null object
to_station_id        181553 non-null object
gender               181553 non-null object
birthyear            181553 non-null float64
dtypes: float64(2), int64(1), object(8)
memory usage: 16.6+ MB
```

	trip_id	starttime	stoptime	bikeid	tripduration	from_station_name	to_station_name	from_station_id	to_station_id	gender	birthyear
0	431	10/13/2014 10:31	10/13/2014 10:48	SEA00298	985.935	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Male	1960.0
1	432	10/13/2014 10:32	10/13/2014 10:48	SEA00195	926.375	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Male	1970.0
2	433	10/13/2014 10:33	10/13/2014 10:48	SEA00486	883.831	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Female	1988.0

Since the original trip dataset does not contain ‘gender’ and ‘birthyear' data for the trips taken by non members and so these columns only contain null values.

A.2.2 To get a clean dataset for Temporary Pass Holders, I

- Extracted all the rows from the original trip dataset where the usertype column contained the value 'Temporary Pass Holder'
- Dropped birthname and gender column completely as they only contained null values for non members
- Dropped the usertype column as it has become redundant for this dataset

Information and the first 3 rows of the cleaned **temporary pass holder only** sub dataset created from the original trips dataset

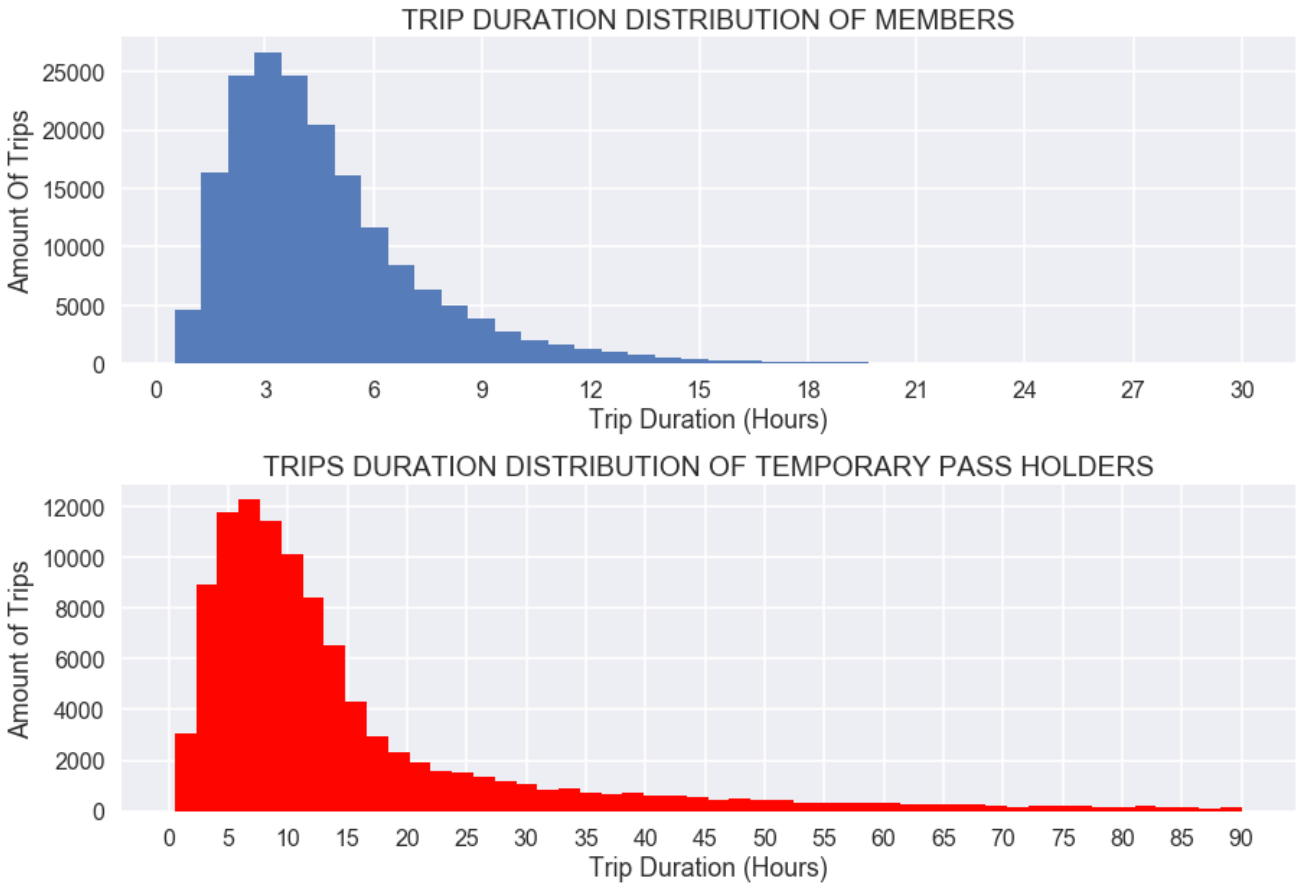
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 105300 entries, 69 to 286856
Data columns (total 9 columns):
trip_id              105300 non-null int64
starttime            105300 non-null object
stoptime             105300 non-null object
bikeid               105300 non-null object
tripduration         105300 non-null float64
from_station_name    105300 non-null object
to_station_name      105300 non-null object
from_station_id      105300 non-null object
to_station_id        105300 non-null object
dtypes: float64(1), int64(1), object(7)
memory usage: 8.0+ MB
```

trip_id	starttime	stoptime	bikeid	tripduration	from_station_name	to_station_name	from_station_id	to_station_id
507	10/13/2014 12:11	10/13/2014 12:16	SEA00321	332.457	City Hall / 4th Ave & James St	City Hall / 4th Ave & James St	CBD-07	CBD-07
518	10/13/2014 12:20	10/13/2014 12:31	SEA00321	690.793	City Hall / 4th Ave & James St	2nd Ave & Blanchard St	CBD-07	BT-05
530	10/13/2014 12:43	10/13/2014 12:48	SEA00311	278.849	King Street Station Plaza / 2nd Ave Extension ...	King Street Station Plaza / 2nd Ave Extension ...	PS-05	PS-05

B. EDA and Statistical Inference:

B.1 Analyzing Trip duration by user - type (member & pass-holders):

After removing outliers and converting trip duration from seconds to hours, I plotted the continuous distribution of trip duration on two separate histograms classified by members(blue) and temporary pass holders(red) which makes it easier to compare the distributions between these groups.



Observation:

- The histograms show that a higher frequency of trip duration for members is concentrated between **3-4 hours** whereas for temporary pass holders the concentration is between **6-11 hours** .
- Even after removing the outliers from both the sets of data, the histogram for temporary pass holders indicates **much greater variance** than the histogram for members.The standard deviation for temporary pass holders is **24.6 hours**, which is approximately **4 times greater** than the standard deviation for members at **6.2 hours.**

- This lower consistency among values for temporary pass holder indicates a lack of routine as compared to members.

B.2 Popularity of Pronto bicycle stations among members and pass - holders:

I further investigated the differences between members and temporary pass holders by finding out the most popular bike stations among members and pass holders and the differences between them or lack thereof which could provide us with more valuable insights.

B.2.1 5 Most popular starting points for trips by user type

The top 5 most popular starting points for members and the number of member trips that originated from them

```
member_trip_data['from_station_name'].value_counts().head(5)
```

```
E Pine St & 16th Ave                      9895
E Harrison St & Broadway Ave E            7736
Cal Anderson Park / 11th Ave & Pine St    7367
Westlake Ave & 6th Ave                    6795
REI / Yale Ave N & John St                6459
Name: from_station_name, dtype: int64
```

The top 5 most popular starting points for pass holders and the number of member trips that originated from them

```
pholder_trip_data['from_station_name'].value_counts().head(5)
```

```
Pier 69 / Alaskan Way & Clay St                          9163
3rd Ave & Broad St                                       6660
Seattle Aquarium / Alaskan Way S & Elliott Bay Trail     5283
2nd Ave & Pine St                                        4574
Lake Union Park / Valley St & Boren Ave N                3981
Name: from_station_name, dtype: int64
```

B.2.2 5 Most popular ending points for trips by user type

The top 5 most popular cycle drop off points for members and the number of trips that ended there

```
member_trip_data['to_station_name'].value_counts().head(5)
```

```
PATH / 9th Ave & Westlake Ave     8954
2nd Ave & Pine St                 7985
Republican St & Westlake Ave N    7836
Westlake Ave & 6th Ave            7804
Pine St & 9th Ave                 7541
Name: to_station_name, dtype: int64
```

The top 5 most popular cycle drop off points for members and the number of trips that ended there

```
pholder_trip_data['from_station_name'].value_counts().head(5)
```

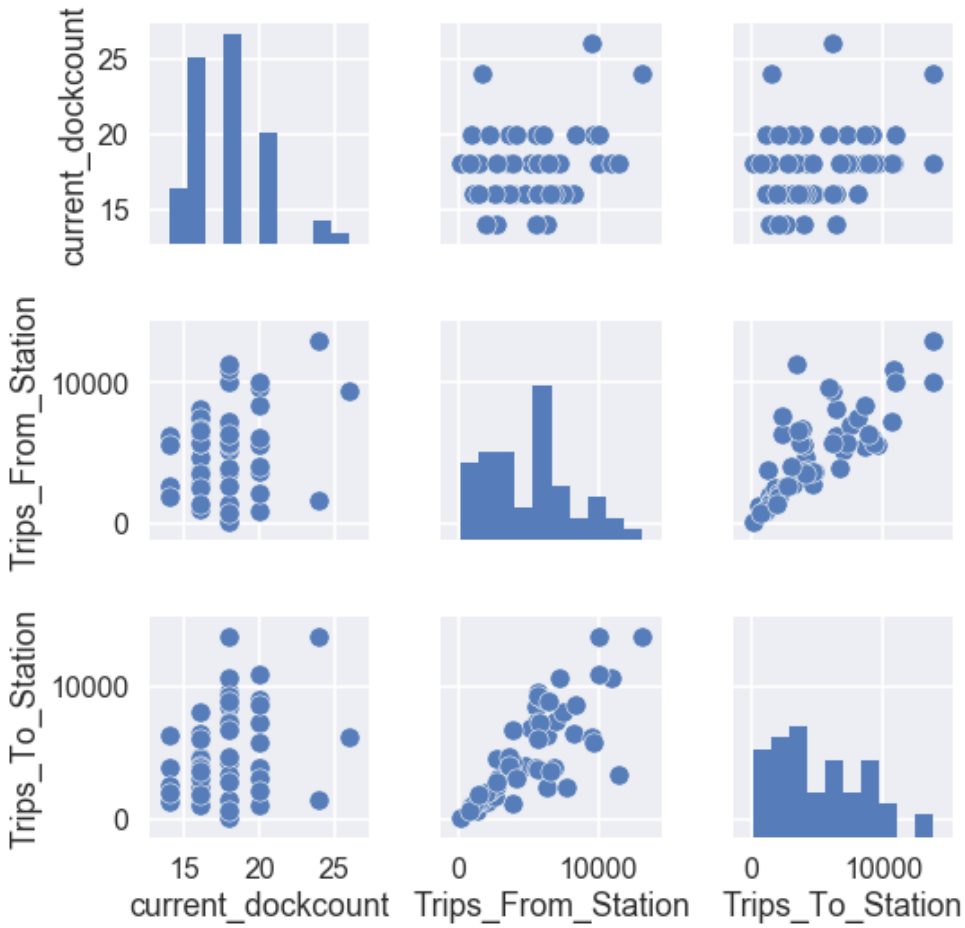
```
Pier 69 / Alaskan Way & Clay St                          9163
3rd Ave & Broad St                                       6660
Seattle Aquarium / Alaskan Way S & Elliott Bay Trail     5283
2nd Ave & Pine St                                        4574
Lake Union Park / Valley St & Boren Ave N                3981
Name: from_station_name, dtype: int64
```

Observation:

- The **most popular starting point** for trips **among members is 'E Pine St and 6th Avenue'** while the most popular starting point **among temporary passholders is 'Pier 69/Alaskan Way & Clay St'**
- The **most popular drop off point** among **members is 'PATH/ 9th Avenue & Westlake Avenue'** while the most popular drop off point **among temporary passholders is 'Pier 69/Alaskan Way and Clay St'**
- The top 5 most popular stations from where trips originate or end for members are **mostly different** from the top 5 most popular stations for temporary passholders.
- The most popular stations among passholders seem to be touristy places such as Pier 69 and the Seattle Aquarium which are not as popular among users with memberships which suggests that most of the users of temporary passes for Pronto could be tourists/visitors as compared to the users of the membership which could be mostly locals.

B.3 Relationship between the number of bicycle docks at a station and it's popularity as a starting and ending point for trips among all users:

I plotted a pair plot after further grouping the necessary data to explore the question **if there was any correlation between the number of docks at a station and it's popularity** and I computed the Pearson correlation coefficients for the same to quantify the relationship.



Statistics:

- Pearson correlation coefficient for number of docks at a station and number of trips taken from that station: **0.287489634451**
- Pearson correlation coefficient for number of docks at a station and number of trips taken to that station: **0.320496700678**
- Pearson correlation coefficient for number of trips to a station and number of trips taken from that station: **0.75048650149**

Observation:

- The correlation coefficients computed supports our initial observation. There is a **very small positive correlation between the number of docks at a station and the number of trips taken from that station**
- There is a **relatively higher correlation between the trips taken to a station and the number of docks at that station but the overall correlation is still low**
- **There is a much higher positive correlation between the trips taken from a station and the trips taken to a station** which indicates that a station's overall popularity makes it a popular starting and a popular ending point for users and **there is not a huge split between most station's popularity as a starting point and an ending point.**

B.4 Effect of various weather attributes of a given day on the total number of trips taken during that day:

To find out how different weather attributes affected the number of trips taken on a given day I computed the correlation matrix of the various weather factors and total number of trips taken per day and plotted it on a Seaborn heatmap.

Further wrangling and processing the data:

- Since the data for the trips taken and the weather on each day for the period of time when the data for the trips was collected (Oct 2014 - Oct 2016) was contained in two different datasets, it required further processing the data and merging both the datasets accurately to get a correlation between weather factors such as mean temperature, maximum temperature, mean humidity, mean wind speed and the number of trips taken on a given day.
- For this I converted the 'starttime' and 'date' columns of the trips dataset and weather dataset respectively to datetime objects and then merged both the datasets on these columns to get accurate observations of weather attributes and grouped the dataset on these attributes to get the trip_count column.

	Date	Mean_Temp(F)	Mean_Humid	Mean_Wind(MPH)	Max_Temp(F)	Trip_Count
0	2014-10-13	62.0	68	4	71	818
1	2014-10-14	59.0	78	5	63	982
2	2014-10-15	58.0	77	7	62	626

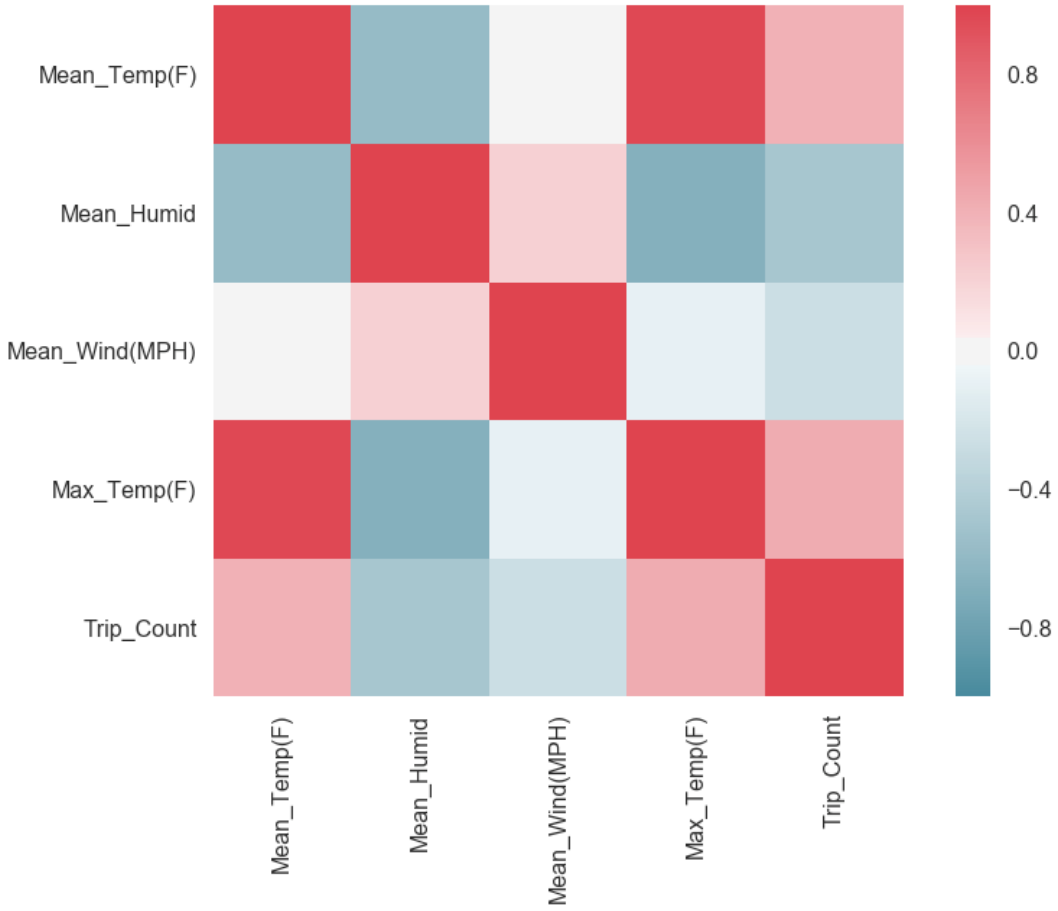
Below is the correlation matrix and heat map of the observed values:

	Mean_Temp(F)	Mean_Humid	Mean_Wind(MPH)	Max_Temp(F)	Trip_Count
Mean_Temp(F)	1.000000	-0.584973	-0.018131	0.970604	0.399741
Mean_Humid	-0.584973	1.000000	0.218338	-0.673048	-0.483360
Mean_Wind(MPH)	-0.018131	0.218338	1.000000	-0.100926	-0.266897
Max_Temp(F)	0.970604	-0.673048	-0.100926	1.000000	0.432901
Trip_Count	0.399741	-0.483360	-0.266897	0.432901	1.000000

Observation:

- The **mean temperature** and the **maximum temperature** are positively correlated with the amount of trips taken on a given day. The positive correlation between maximum temperature and number of trips is relatively higher than the positive correlation between mean temperature and number of trips, in this case they both mean that **higher temperature makes for favorable conditions for riding using Pronto's bikes for it's users.**

- The **mean humidity and mean wind speed are negatively correlated** with the **amount of trips taken during a day**. The mean humidity negatively affects the amount of trips taken during a day more than the mean wind speed does, but we can assume that they both make for unfavorable conditions for Pronto's users.

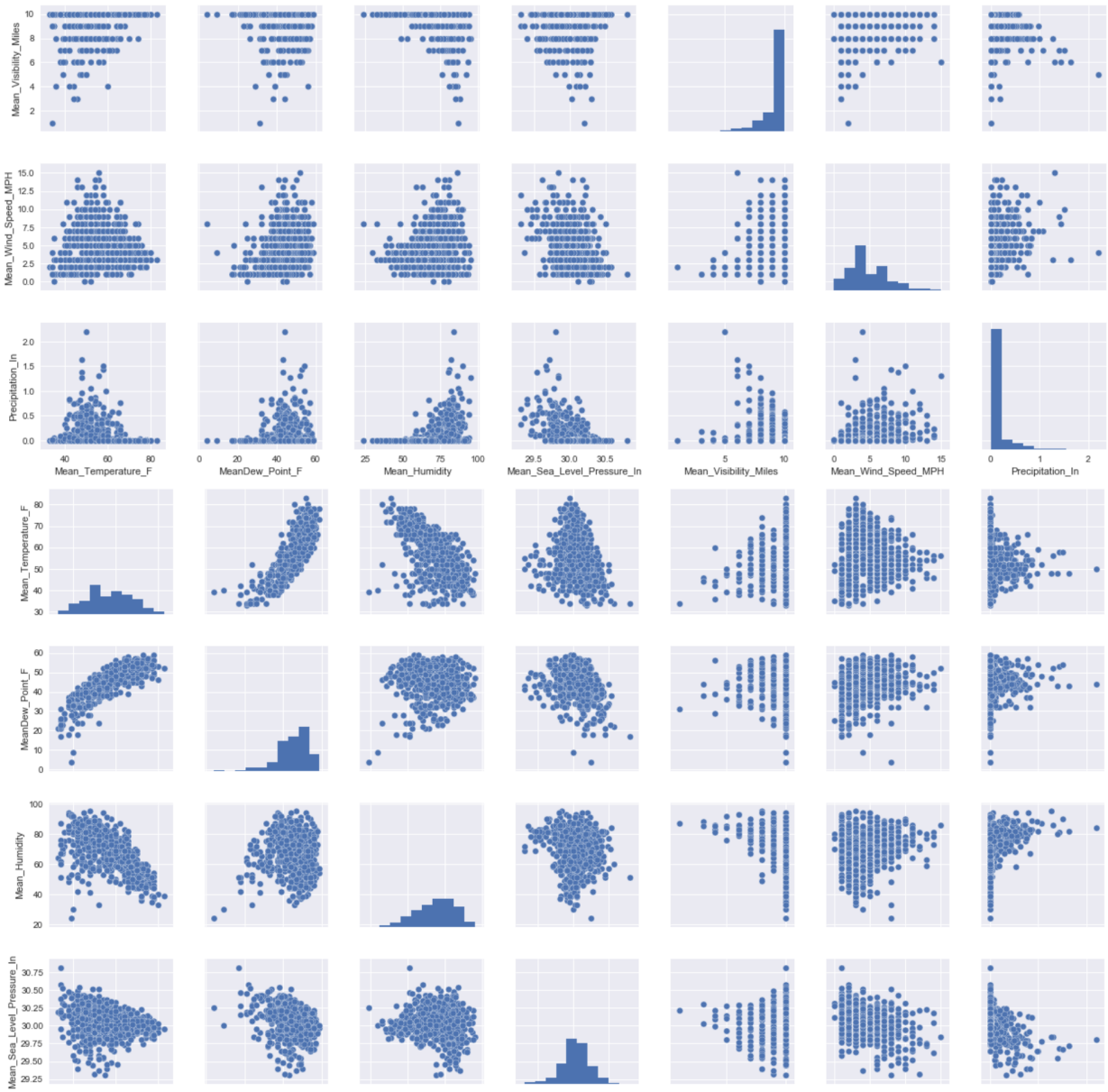


C. Regression model selection:

C.1 Eliminating features with hight collinearity and removing outliers

Pair plot showing collinearity between various weather attributes

- There is relatively higher collinearity between Mean Dew Point and Mean Temperature features and since we know from our previous analysis, Mean temperature has a high correlation with total trips so I will go ahead and drop Mean Dew Point feature from our dataset to make our model fit better.



C.2 Feature Engineering:

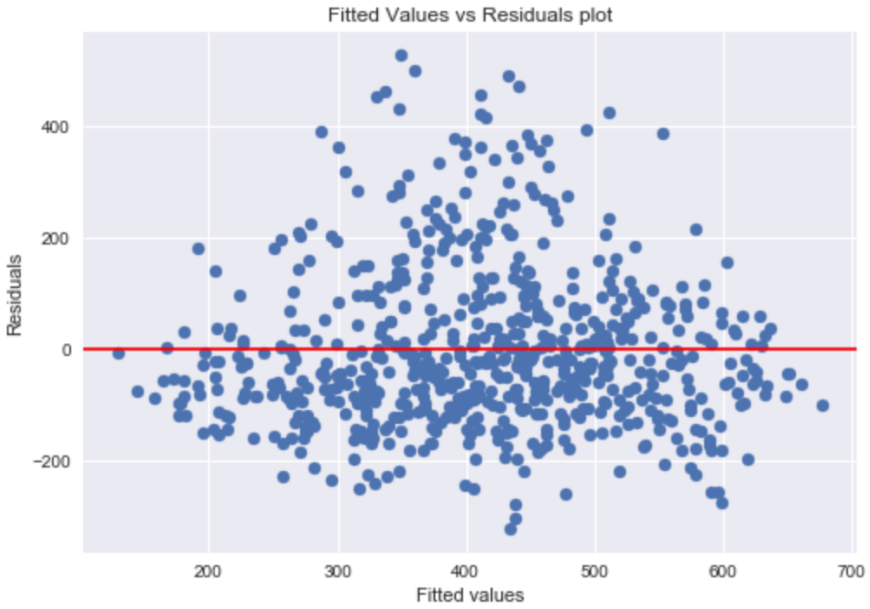
- Another feature that should help explain the variance in daily total trips for Pronto cycle share is the day of the week. The fact wether the day is a weekday or a weekend could have huge impact on the number of trips taken on that day.
- I will extract the day as a continuous variable (0 to 6, where 0 is a Monday and 6 is Sunday) using the date time column and add it to the features we will be using to fit various regression models on our data.

	Date	Mean_Temp	Mean_Humid	Mean_Sea_Press	Mean_Visib	Mean_Wind	Percepitation	Trip_Count	Day
0	2014-10-13	62.0	68	29.79	10	4	0.00	818	0
1	2014-10-14	59.0	78	29.75	9	5	0.11	982	1
2	2014-10-15	58.0	77	29.71	9	7	0.45	626	2

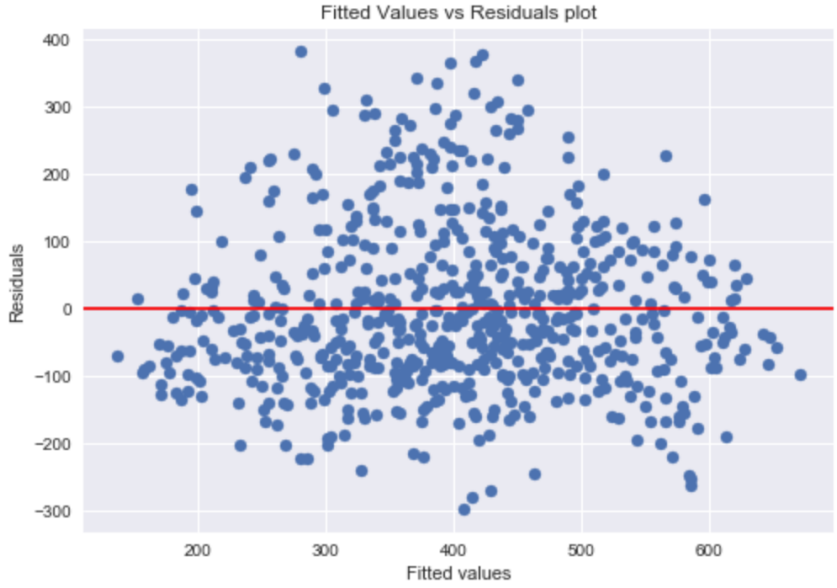
C.3 Dealing with Outliers:

Initially to deal with outliers, I used statsmodel.api to fit a linear regression model on our data and plotted a residuals vs fitted values plot and Leverage vs Normalized residuals squared plot to see if the data is homoscedastic , before and after removing outliers.

Before removing outliers with higher leverage



After removing outliers and fitting the model again

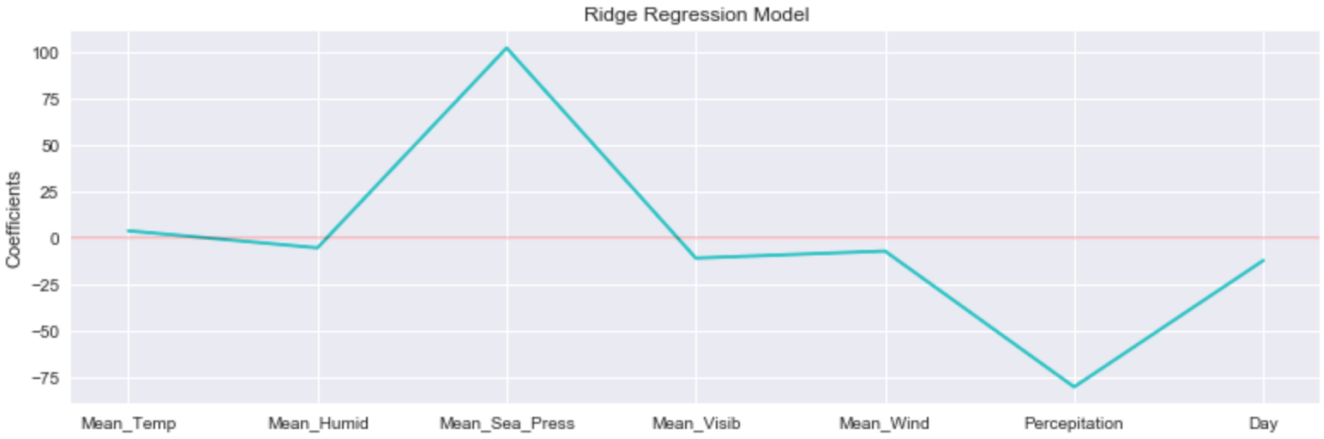


- After removing most of the ceiling values our data looks mostly homoscedastic with a few ceiling values. The **R-squared** for our model jumped from the initial **36.4%** to **44.7%** from just taking care of the outliers. A slight improvement.
- Our primary objective for our linear model is not to have an extremely high R-squared value on any one model , but use various models to determine the weather attributes that have the highest impact on ridership and have a low MRSE.

C.4 Fitting a Scikit Learn Ridge Regression model on the data

I fitted a Ridge regression model on our data using cross validation to tune alpha to determine the coefficients assigned to all the features after accounting for the L2 regularization hyper parameter. The R-squared for the ridge regression model with optimal alpha 1.0 was the same as the linear regression model **44.68%** and **MRSE was 123.07.**

Coefficients assigned to features by the ridge regression model



Observation:

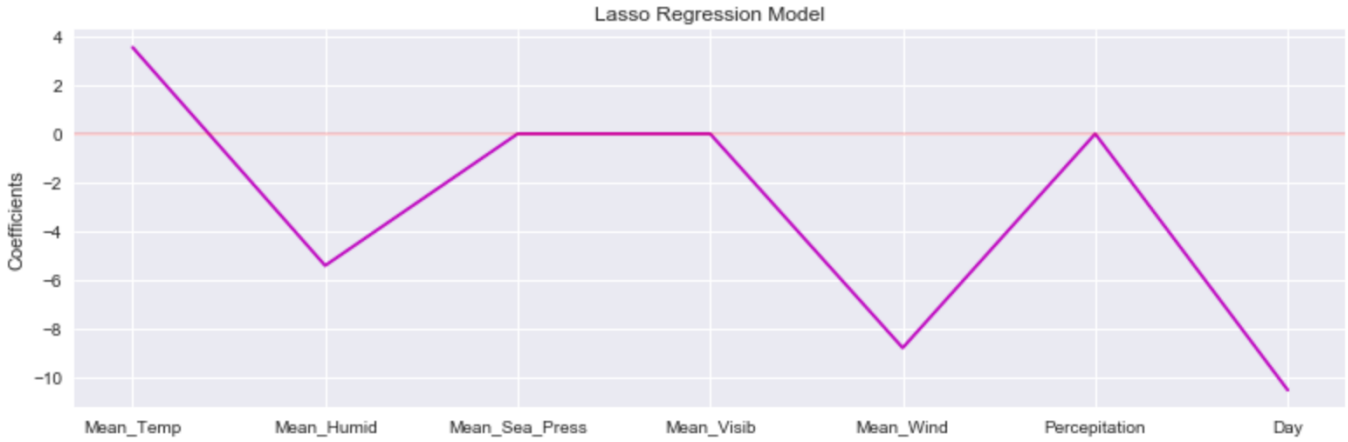
- Ridge regression assigned the highest coefficient to Mean_Sea_Pressure feature at 102.7, which seems a bit odd, while features with initially high correlation coefficients with trip counts such as mean humidity and mean temperature were assigned extremely small coefficients of -4.8 and 4.2 respectively.

C.5 Fitting a Lasso Regression Model on our data

To see what coefficients will be assigned to our features with lasso regression which has the added benefit of killing off features it deems unimportant with it's L1 regularization hyper parameter , I used cross validation to find the optimum value of alpha which came out to be 10. The R-squared for lasso regression was

slightly lower than the R-squared for ridge regression at **41.9%** while the MRSE was also slightly higher at **126.05.**

Coefficients assigned to features by the lasso regression model

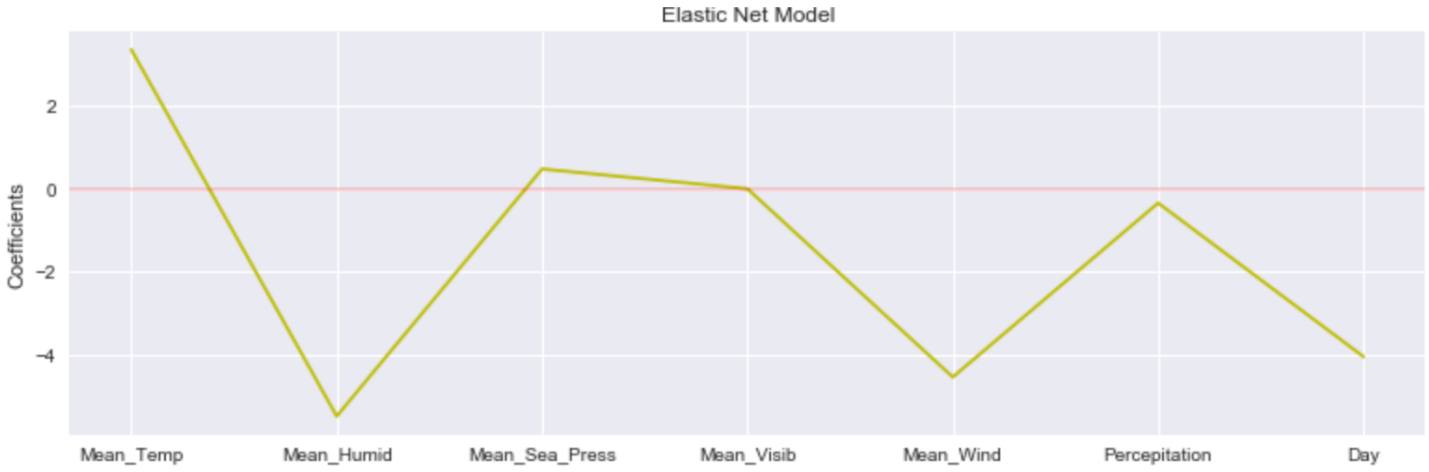


Observation:

- Even though our lasso yielded an R-squared value and MRSE value more or less the same as our ridge regression model, the coefficient value assigned to feature Mean Sea Pressure is in sharp contrast to the ridge regression model. Our Lasso model killed off Mean Sea Pressure feature, assigning it a coefficient of 0 and assigned higher values to Mean Humidity and Mean Temperature of -5.4 and 3.54. The lasso model assigned the strongest coefficient to Day, i.e -10.51.

C.6 Fitting an Elastic Net Model on our data

Since some of the coefficients assigned by our regression model and lasso regression were in sharp contrast to each other, it seems like a good idea to fit an elastic net model on our data that uses both L2 and L1 regularization parameters using the L1 ratio. I again used cross validation to determine alpha and the



optimal L1 ratio which turned out be 10.0 and 0.1 respectively. The **R-squared** computed from elastic net model was slightly lower than the lasso regression model at **40.06%** and the **MRSE** was slightly higher at **128.11**.

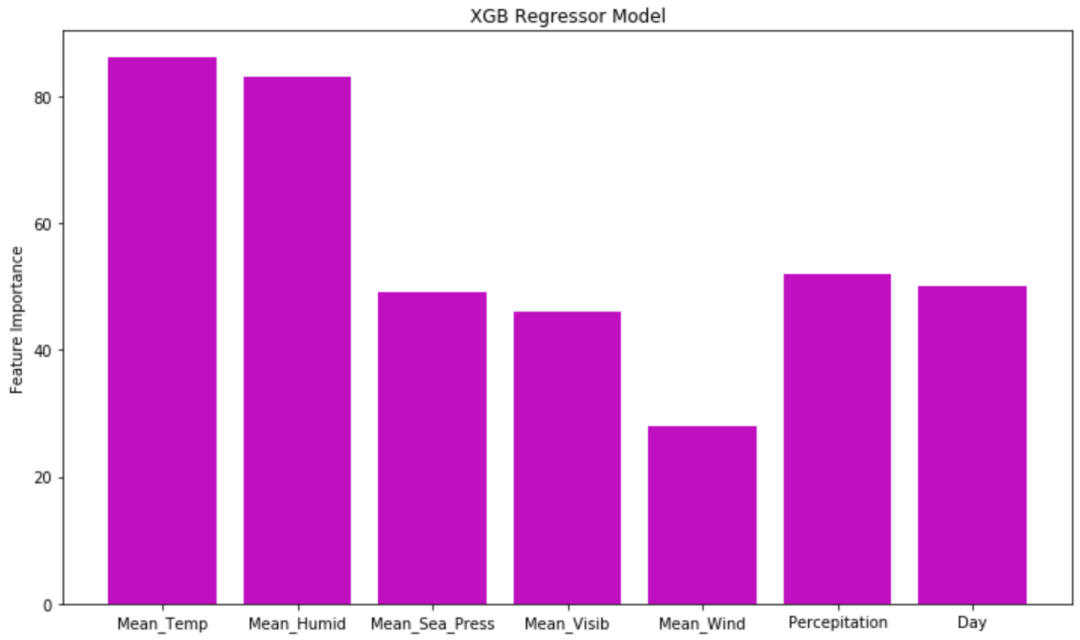
Observation:

- The L1 parameter at the optimal L1 ratio seems to be overpowering our Elastic Net model. The coefficients assigned by our elastic net model resemble our lasso model closely. Our Elastic Net model assigned an extremely low coefficient to Mean Sea Pressure at 0.47, Mean Humidity at -5.47, Mean Temperature at 3.34 and Day at -4.04.

C.7 XGB Regressor Model

It still seems that no strong conclusions can be made about the importance of various features in determining the total ridership on a given day. We need a stronger model to help us get more concrete and usable results. A good option is to use an Extreme Gradient Booster Regressor Model.

I used a scikit learn wrapper to tune and run an XGB Regressor model. I tuned max_depth, min_child_weight, gamma, subsample, colsample_bytree using cross validation and RMSE as my test metric. At the optimal number of learners which came out to be 524, the **test MRSE was 100.15** and **train MRSE was 118.55**. The **R-squared** value with the XGB Regressor model jumped quiet a bit at **61.69%**, a significant improvement from our previous models.



Observation:

- According to our XGB regressor model which is the best fit model on our data, Mean_Temp and Mean_Humid have been assigned the highest feature importance , followed by precipitation. As we know from our above analysis, Mean Temperature has a positive effect on the number of trips taken on a given day while Mean Humidity has a negative effect.

D. Classification Model Selection:

Objective:

The objective of this section is to see if it is possible to fit a classification model on the member's trip sub dataset that can given features such as age, sex , day, hour and starting point of a member's trip predict the ending point for the trip with some accuracy as further on the model can always be trained in real time to improve accuracy.

D.1 Further Data Wrangling and Feature Engineering:

Most of the features required to fit a classification model can be extracted simply from the members only sub dataset assembled from the initial data wrangling, however to extract the hour and day when each trip started requires further converting the observations in the 'starttime' column to date-time objects and then extracting the hour and day attributes from the date-time objects.

My initial approach was to convert the observations in the start-time column which included the month,day,year,hour and minute for when the trip started to epochs but the model was not able to generalize at all with an accuracy score of a mere 7%. Using only the hour and day let's the model generalize much better.

To convert the names of the stations from where the trips originated to continuous variables to be able to fit the model, I used pandas get dummy variables method which makes a separate column for each station with a binary value of 0 if the trip did not originate from that station and 1 if the trip did originate from that station. I repeated the process to convert the gender column to dummy variables as well.

	age	from_station_name_12th Ave & E Mercer St	from_station_name_12th Ave & E Yesler Way	from_station_name_12th Ave & NE Campus Pkwy	from_station_name_15th Ave E & E Thomas St	fro
0	56.0	0	0	0	0	
1	46.0	0	0	0	0	

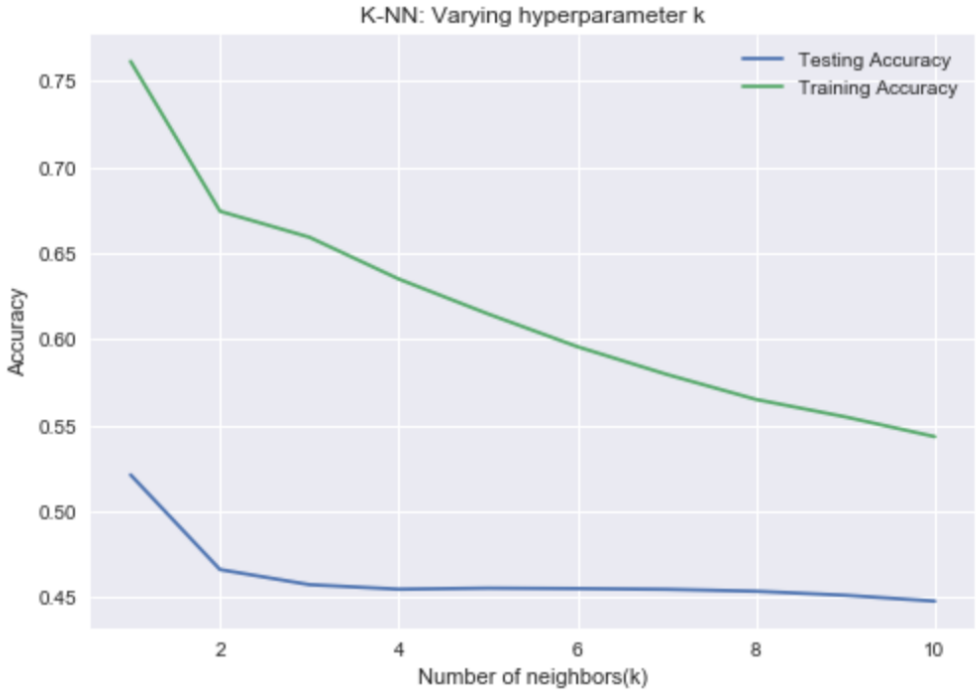
2 rows × 64 columns

ation_name_Westlake Ave & 6th Ave	gender_Female	gender_Male	day	hour
0	0	1	0	10
0	0	1	0	10

The data contains a total of 60 target variables , i.e 60 different stations where the trip can stop and a total of 181, 533 observations or trips.

D.2 K-Nearest Neighbor Classification Algorithm

My first choice of the model to be fit on our data was a KNN model. I split the data into test and training sets and used cross validation to tune K. I used a large range of numbers for K.



Observation:

- 2 and 7 are both good values to use for hyper parameter K for our model. 2 yields a value of 66.5% accuracy on our training set and 46.6% on the hold out set, which seems a bit overfitted and using 7 as K yields a value of **54.8% on the training set and 45.4% on the hold out set**, which is the value I chose as the optimal K for our model as I believe it will generalize better on unseen data.

D.3 XGB Classifier

The second model I fit on our data was an extreme gradient booster classifier to see if we can get a better accuracy score for our data. I used cross validation again to tune for tree and regularization parameters. With the XGB classifier , and tuned parameters the accuracy on the **training set** jumped to **68.4%** and accuracy on the **test set** jumped to **53.9%** with learning rate 0.05 and 375 optimal n_learners.

D.4 AWS:

Fitting an XGB classifier model on our dataset which has more than 181,500 observations was a mammoth task and required much more computational power than was available on my Mac, as a result I had to use Amazon Web Services, Elastic Compute Cloud (EC2) instance with 64 Vcpu capacity as I needed enough computational power to be able to run up to 12 parallel jobs when I was tuning the tree based hyper - parameters for our model.

E. Recommendations

I. About 36.7% of Pronto's users were pass-holders who are most likely made of tourists and people visiting Seattle, yet Pronto did not have stations at or nearby a lot of famous tourist spots and parks. Expanding in those areas for the next bicycle share system would prove fundamental in succeeding where Pronto could not

II. Advertising Promotions specific to pass holders and tourists at stations mostly popular among pass holders instead of all the stations could help cut costs for a bicycle share company and reach target audience more effectively

III. The bicycle dock capacity of a station has no concrete effects on it's popularity as a starting or ending point for users and resources are best used to match higher demand when it arises by increasing the supply of bicycles at a station then but the other way around i.e increasing supply of bicycles at a station to increase the demand for that station will most likely not work.

IV. Like any other geographical location, Seattle's topography and climate play a role in the daily trips bicycle trips taken, however some factors effect ridership more than others. In my analysis, overall mean temperature and humidity during a day play a larger role in swaying ridership in comparison to other weather factors. An overall higher temperature makes for optimal conditions while high humidity affects ridership negatively. Using weather forecasts and accounting for days where ridership could be hit hard due to weather can prove useful for the next bicycle share company.

V. In order to bring in additional revenue to fund expansion to necessary areas, other sources of income , such as targeted advertisement can be made possible by being able to predict the flow of it's user. The classifier model fit in this analysis can be improved by training it in real time by either asking users when they start a trip their final destination using an app system or simply recording their destination when their trip ends to make the model more

accurate and then using it's prediction and it's users demographics to generate revenue from targeted advertisement for itself or it's clients.