Applied Data Science

(16-791 – Fall'18)

project report on

Forecasting Trips for San Francisco's Bike Share Program

Team K - Bhavya Chhadva(bchhadva),
Hardit Singh(hardits),
Ravi Mittal(ravimitt)

Background:

A bicycle-sharing system, public bicycle system, or bike-share scheme is a service in which bicycles are made available for shared use to individuals on a short term basis for a price or free. Most bike share systems allow people to borrow a bike from a "dock" and return it at another dock belonging to the same system. Bikesharing systems have been increasingly popular in recent years, all around the world. In the United States alone, there are at least 119 bike-sharing systems that, in total, share about 4,800 docking stations as of January 2017.

The Issue with Bike-sharing:

With the surge in bike share usage, comes the challenge of forecasting the trips that will be made from different stations so that bikes can be effectively distributed for commuters to avail the facility. Since bike-sharing platforms suffer from cyclic travel patterns, there is a tendency for bikes to get collected at certain busy stations at rush-hours.

Currently, bikes are manually redistributed between stations to ensure availability of both bikes and docks. Unavailability of bikes can deter ridership and stagnant demand. From the service provider's point of view: proactively redistributing bikes between stations to ensure availability of both bikes and docks to return bikes, can help to solve this problem. If we can accurately forecast how many trips can we expect at any station at any hour of the day, it would help to redistribute bikes in advance to meet the demand.

Objective:

Through this project, we aim to forecast with reasonable accuracy, the number of trips that will be made from a station at a given date and hour. With the forecast, we can then evaluate whether the station can meet the forecasted demand or will it require rebalancing. Business can also evaluate its strategy to install or remove docking stands at a particular station based on the the predicted future demand.

Executive Summary

Increase in bike sharing usage has brought along an unique set of challenges. The most important being the effective rebalancing of bikes between stations. Rebalancing of bikes , if done effectively will optimize our fleet of bikes within and between stations to meet customer demand. Currently this operation is solved manually and not much is explored to data modelling. Rebalancing of bikes is a 2-part problem ,

1. Forecasting of number of trips at each station, day and hour.
2. Linearly optimization of the deficiency or surplus of bikes at that station with closest station to effectively meet the demand.

Forecasting the demand (number of trips) was done using data modelling techniques.
When we explored, the data of trips, bike status, station locations and weather at that localities, we found some patterns that was used to forecast the demand.

1. We saw that almost 70% of the trips are taken were taken between time period 7-10 am and 5pm-8pm. This was the cause to predict demand on hourly basis
2. Not all stations have the same demand, some are busiest than others. This was the reason to build location specific model.
3. We found that sparse stations usually have high dock capacity relative to bikes available. In future, we can set up dock capacity accordingly without renting much real estate.
4. In our busiest station on a day-to-day basis, some have high end capacity vs other where they have high beginning capacity of bikes. This creates our need to effectively rebalance bikes between station.

We used different regression modelling techniques along with trying out Time Series Modelling. Out of which, Random Forest Regressor at our busiest station gave on an average Mean Absolute error of 1.6 trips. Through this

we can predict the demand on hourly basis and calculate which stations should send or receive bikes to and from other stations to meet appropriate demand.

Data and Description of Features:

For our project, we have used data from Ford-GoBike, the bike-sharing service providers in the city of San Francisco and surrounding Bay Area. Bay Area Bike Share was introduced in 2013 as a pilot program for the region, with 700 bikes and 70 stations across San Francisco and San Jose. The bikes are available for use 24 hours/day, 7 days/week, 365 days/year and riders have access to all bikes in the network when they become a member or purchase a pass.
The time period of the dataset is from August 2013 to August 2015. It has the following tables

- Station: Data relating to the geographical location of 70 bike stations, their id, name, city, installation date and dock count. It has 70 observations and 7 variables.
- Status: Time series data containing minute-by-minute updates of bikes and docks available for each station. This table has 71,984,434 observations and 4 variables.
- Trip: This table has data related to each trip over the three years. This table has 669,959 observations and 11 variables such as start/end time, start/end station ID, start/end station name, bike ID, rider subscription type, and trip duration.
- Weather: This table has data relating to daily weather conditions of each city in the Bay Area over the three years. It has 3,665 observations and 25 variables.

Data Preparation:

We began by looking at the shape and the datatypes of each dataset. We converted all date columns to the datetime datatype. Further, we created separate columns for year, month, day, day of week as well as a flag for whether its a weekday or weekend to make our data more granular for accurate predictions. Using publicly available pandas packages, USFederalHolidayCalendar and CustomBusinessDay, we created flags for holidays and business days as we hypothesized for strong correlation between trips made on a holiday or working day.

Data Cleaning/Preprocessing

1. Change the zip_code in trip dataset to integer
2. We had over 5000 cases of missing values of zip_code in weather dataset which we kept as it is.
3. Changed Installation_date in station dataset to a datetime format
4. Changed time from status data to datetime format
5. Changed start_time and end_time to datetime format in trip dataset
6. Precipitation column in weather dataset is char and has value ‘T’ which stands for traces of precipitation. We substituted it with a numeric value 0.01 (~small)..
7. Blank values in events are coded as Normal Day in our weather dataset.(other events are fog, rain etc)
8. There are 2 values which mean the same “Rains” or “rains”. So made them consistent.

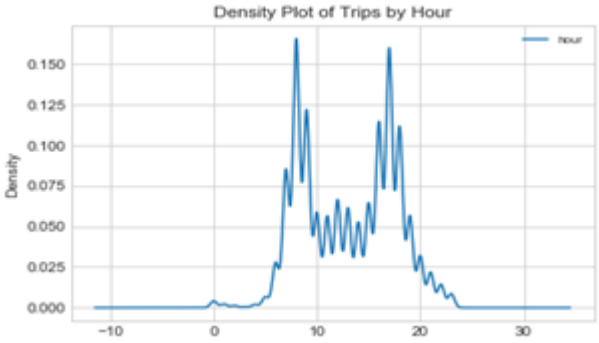
New Columns
1. Created two new columns of latitude and longitude for a station_id in the **Status** dataset to calculate nearest stations to a given station.
2. Generated hour values(0-23) from start_date in **Trip** dataset. 0 stands for timing between 12:00 am to 12:59 am and 23 indicates , 11:00 pm to 11:59pm.
3. Generate day of the week values(0-6) from start_date in the **Trip** dataset. 0 stands for Monday and 6 indicates , Sunday.
4. Grouped data on hour and day of the week to derive our output column counts which indicates number of trips taken during that day in that time period from the **Trip** dataset.

Outlier Treatment

According to our **trip** data, we found the maximum duration (end time - start time) of a trip is 287899.00 seconds(~80 hours) which is highly unlikely as according to the pricing information available on Ford Go Bike site, day passes up to 72 hours are available. It is likely that people taking a 3-day pass might extend their pass for a day or two. So, We have assumed any duration beyond that as an outlier and removed it from our trip dataset.

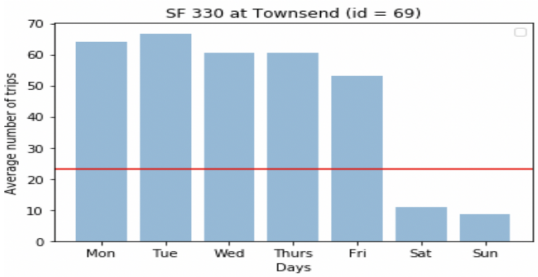
Exploratory Data Analysis

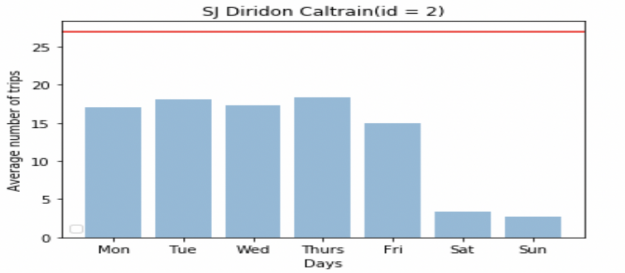
Descriptive Analysis

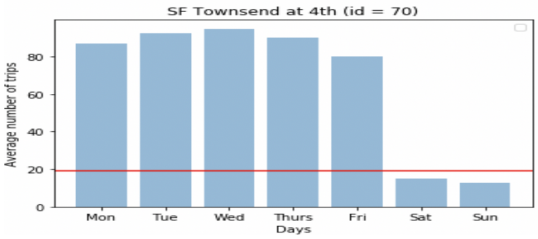


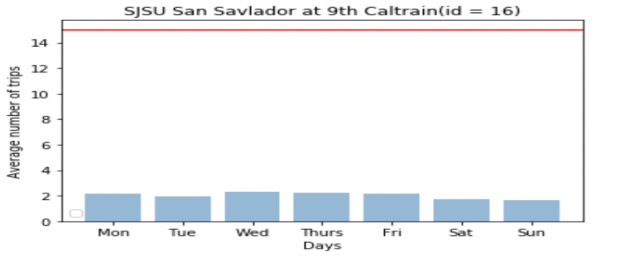
If we investigate these graphs and data, the pattern of trips on yearly basis remains consistent. The graph on the right is more intuitive and helpful for us to propagate model features. We can clearly see a pattern in them, busiest ours are as one can see is from 7am to 10 am(Morning) and 5pm to 8pm. These hours represent almost 70% of the trips taken in those hours.

Analysis to find stations with excessive docking stands









Red line represents the dock count at that station

Insights:

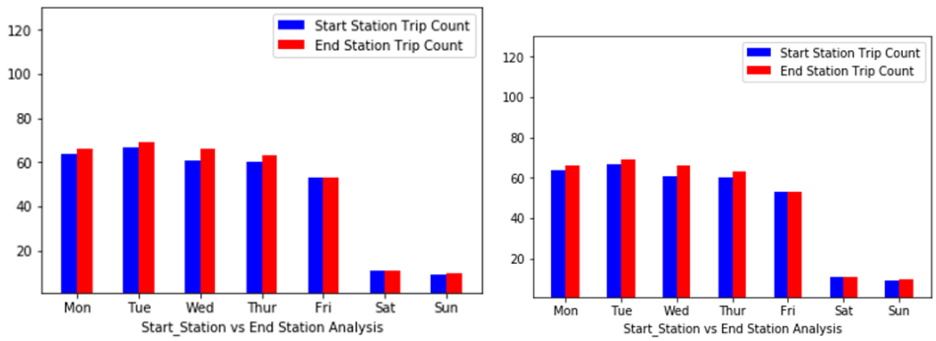
The left side graphs show the dense station plots of average number of trips on a given day of the week and red line represents the dock count at that station. The right hand side graphs are for sparse stations. Average trip per day are way higher than the dock count but average number of trips per hour can be handled with dock count and if required can be increased. We analysed such behavior to mainly identify those stations where business can free up some space occupied by removing unnecessary docks. Since in station like 2 and 16, average number of trips per day are

itself very less than(almost 10) the dock count and if we take the hour level data, then bike rotation does take place. Hence, business is bearing extra rental cost by acquiring more than necessary space of docks at these station

Start Station Trips vs End Station Trips Analysis:

Dense Stations:

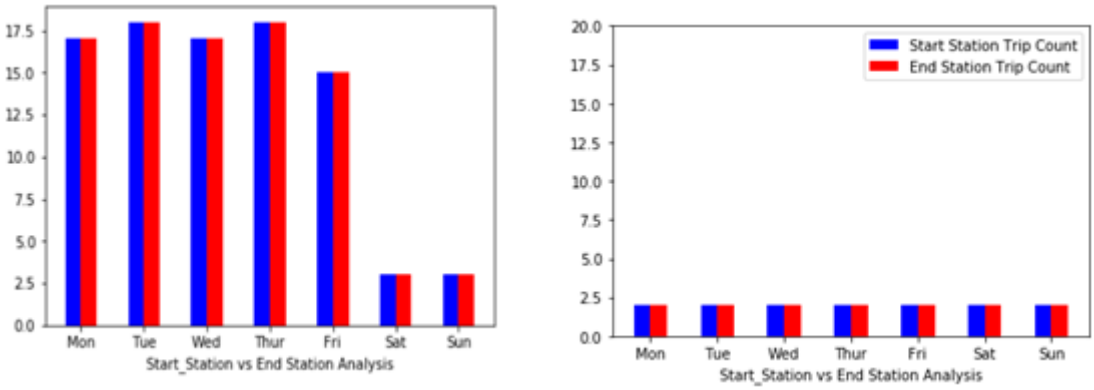
Station No:-70 SF Townsend at 4th **Station No:69 SF 330 at Townsend**



If we look into these graph, on an average over a day , stations that experience relatively high volume of trips require demand prediction and therefore data analysis.

Sparse Stations

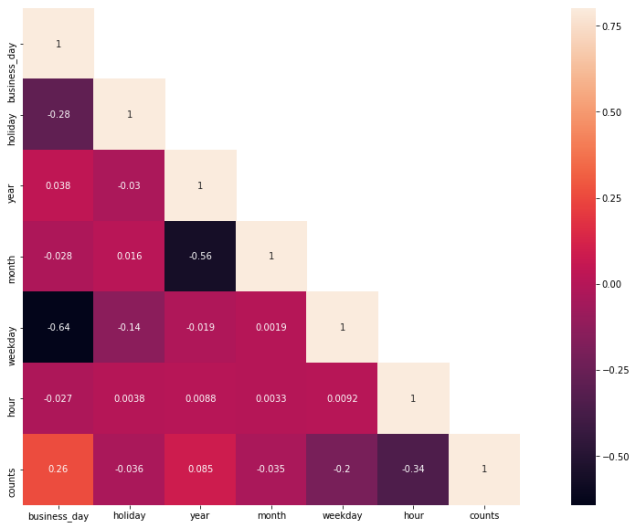
Station No: 2 SJ Diridion CalTrain **Station No 16 SJSU San Savlador at 9th CalTrain**



According to the graph, on an average over a day , stations that experience relatively low volume of trips usually settle their quota for bikes docking. That is to account that these trip are taken relatively by same group of people at least during the weekday. Therefore forecasting here may more or less have normal pattern of trips.

Feature Selection:

After we obtained trip and weather attributes at an hourly level, we headed towards identifying features that influence the count of the trips made. We computed linear correlation of attributes on the count. At an overall level, we did not find strong correlation, therefore we decided to build location specific models. We considered the busiest station in our dataset (Townsend at 4th) and plotted the correlation matrix again. It gave us some interesting findings.



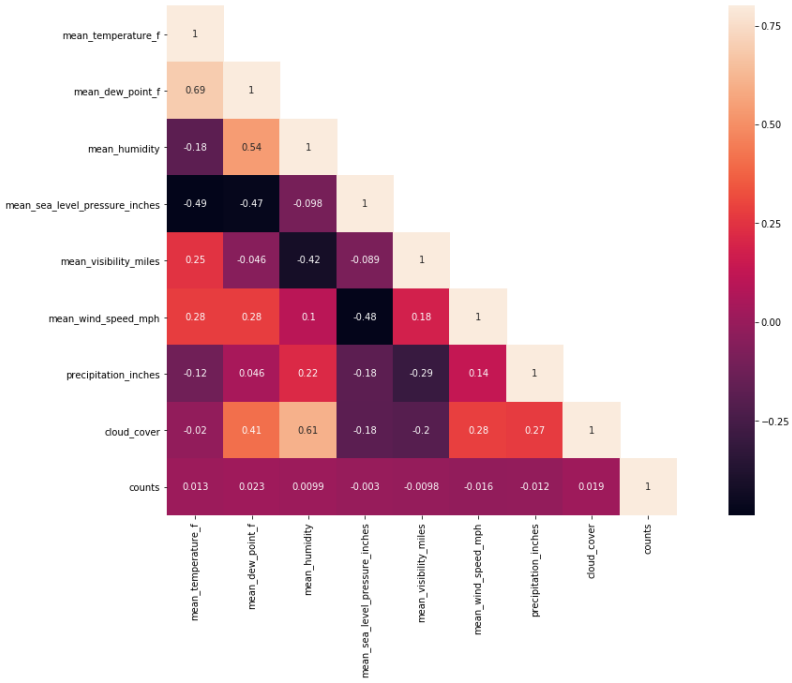
1. Attributes of the day have a strong correlation with the trips made. Specifically:

- Whether it's a working day
- Hour of the day
- Day of the week

  Since working day and day of week are strongly correlated to each other, it makes sense to use one of the two
- From our exploratory data analysis, we saw that trip demand varies for each day of the week. Therefore, we decided to use to day of the week as one of our predictor variables

2. Weather attributes do not influence the count of trips at an hourly level. Barely any linear correlation observed with the dependent variable.

   This is because our weather data is aggregated at a day level, while we are measuring trips at an hour level. So there is no real variation in weather data through the day, making the weather attributes not very useful for hourly predictions



We next performed recursive forward selection on the trip attributes and built multiple regression models. The attributes, hour and day of the week seemed to give the best results across all models.
Addition of any other attributes only marginally improved the performance of the regressor.

Error Metric selection and taking predictions to hour level:

We started with objective of predicting the number of trips being made at station for the whole day on an average, however as mentioned during our feature selection process, we interestingly, could not find any correlation overall on the given attributes. We attributed such behavior limited data availability for 2 years as static. In our hunt to increase the accuracy of the prediction and finding the feature which could significantly affect our output, we ended up in granularization of our input features and we decided to predict for a given station id on any given day(0-6) for an hour(0-23).
For our model, the error metric we have considered is Mean Absolute Error. Usually for forecasting purposes, Mean Absolute Percentage Error is used but since our output variable (count of trips) does not have high values at each hour, the MAPE will tend to be high although in absolute terms the error is small.
For example, actual trip =1, predicted trips = 2, MAPE = 100%

Default Model:

We built our default model by taking the mean number of trips. This was achieved by grouping data on day and hour of the day. The results are as follows,

	San Franciso CalTrain(Townsend at 4th)	South Van Ness at Market	San Jose Diridon CalTrain Station
RMSE(trips)	7.5	8.16	2.1
MAE(trips)	2.5	2	2

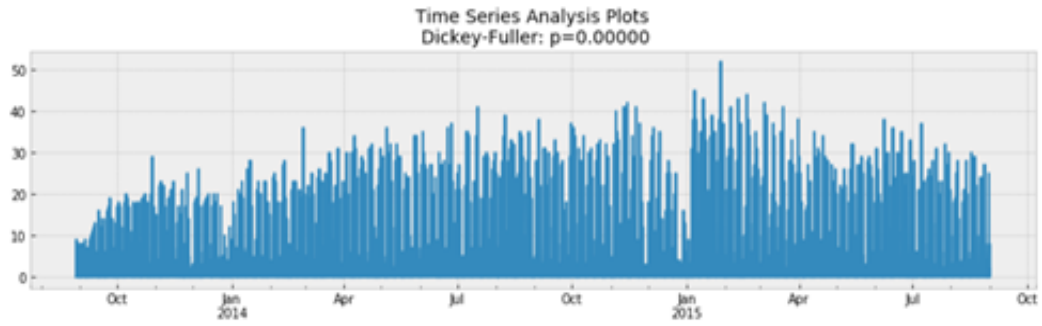
Time-Series Model (ARIMA)

Since we have minute wise status data available for the trips, we thought time series would be our best call to get the predictions. But data and the science behind it again surprised us and intrigued us with unexpected results to our assumptions.

Data Preprocessing:-

Since ARIMA time series model takes univariate data set with index being the time range we are modeling, we grouped the data on the hourly basis and count as the output column. The lag period 5 days. Before running model, just like in any time series data we decided to find seasonality(if any) in our data and validate it through null hypothesis.

Null Hypothesis Validation on data seasonality –



H0 - Seasonality is observed in the number of trips per hour.

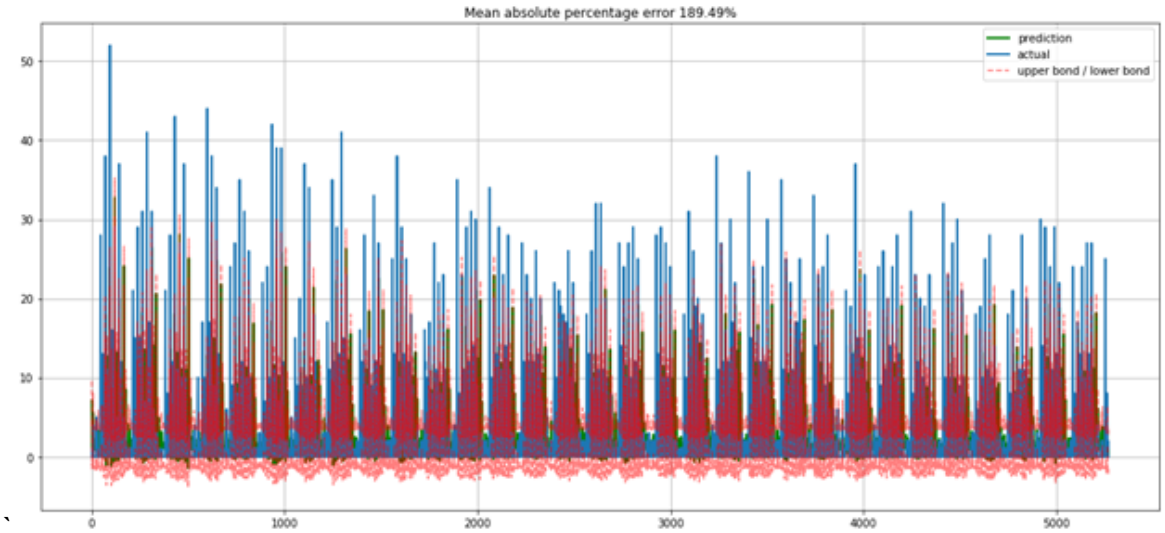
H1 - Seasonality not found

We ran ARIMA time series model and performed Dicker-Fuller p - value test at significance level of 95%.

The p- value we got from the test is 0.0000002 which is significantly smaller that 0.05. Thus concluded that with 95% confidence , we reject the null hypotheses in favour of no seasonality found in the data for no. of trips per hour.

Forecasting count of trips per hour for a specific location using Time Series[ARIMA]:

We ran the model with TimeSeriesSplit having k = 5 (keeping it equal to the lag period). As shown in the graph above, we also plotted the upper and lower bound of the predictions. We got Mean Absolute Percentage Error - 189.49% which makes the model highly inaccurate. E.g. for a station having average trip count of 40, this model will predict within bounds - [10,40].



Regression models

We ran multiple regression models by doing K fold cross validation with K = 5. The regression models were built for each station separately.

The best combination of hyperparameters were achieved by doing GridSearchCV, which runs multiple iterations of models taking different values of hyperparameters and returning the best combination that minimizes the error metric.

Model	Parameter	Value
Random Forest	N Estimators	55
	Minimum Samples Leaf	4
Gradient Boosting	Learning Rate	0.1
	N Estimators	150
	Max Depth	8
	Minimum Samples Leaf	4
Decision Tree	Minimum Samples Leaf	3
	Maximum Depth	8
AdaBoost	N Estimators	100
	Learning Rate	0.1
	Loss	Linear

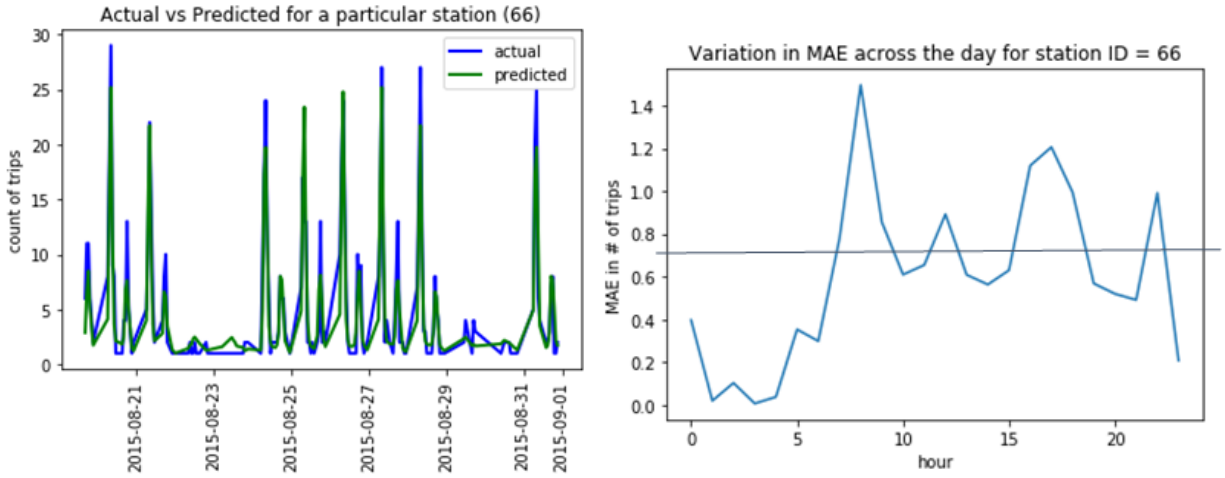
Regression Results:

We ran the model on three of the busiest stations and computed all error metrics as shown below. The units of MAE, Median Absolute Error and RMSE are number of trips.

Model	Metrics	San Francisco (Townsend at 4th)	Powell Street BART	South Van Ness at Market
Random Forest	**Mean Absolute Error (MAE)**	**1.68**	**0.95**	**0.78**
	Median Absolute Error	0.99	0.88	0.63
	Root Mean Square Error (RMSE)	2.82	1.41	1.19
	Mean Absolute Percentage Error (MAPE)	42.1%	43.9%	38.4%
Gradient Boosting	Mean Absolute Error (MAE)	1.70	0.97	0.84
	Median Absolute Error	0.93	0.84	0.59
	Root Mean Square Error (RMSE)	2.76	1.36	1.15
	Mean Absolute Percentage Error (MAPE)	49.0%	51.7%	49.1%
Decision Tree	Mean Absolute Error (MAE)	1.70	0.97	0.84
	Median Absolute Error	0.93	0.84	0.58
	Root Mean Square Error (RMSE)	2.76	1.36	1.16
	Mean Absolute Percentage Error (MAPE)	0.49	0.52	0.49
AdaBoost	Mean Absolute Error (MAE)	2.46	1.14	1.01
	Median Absolute Error	1.58	0.82	0.87
	Root Mean Square Error (RMSE)	3.35	1.52	1.24
	Mean Absolute Percentage Error (MAPE)	103.2%	72.5%	69.0%

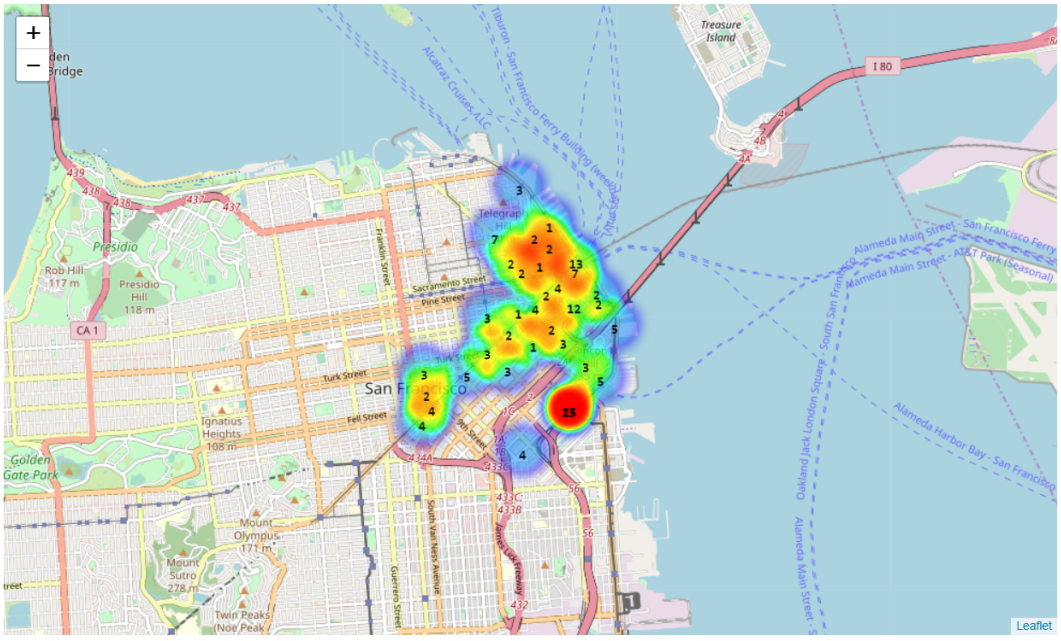
For the Powell Street BART station and the South Van Ness station, the mean absolute error is less than 1 trip. The results on the less busy stations were even better, because they would only have 0-1 trips at any hour on an average on a given day.
The chart below shows the difference of actual and predicted values at South Van Ness station for a two weeks of testing data. We also plotted the variation in Mean Absolute Hours across different hours of the day for a larger testing period.

Finding: the error in prediction is not consistent across the hours of the day. The prediction is more accurate in wee hours of the day as compared to later in the evening.



Prediction Visualization:

The map below shows the predicted values of trips at all stations of San Francisco on Wednesdays at 8am
The number represents the predicted value at that station.



Evaluation of Need for Rebalancing

As per our naive approach to figure out how rebalancing should work in such scenario, we have done one such rebalancing on station SF TownSend at 4th(Station ID 70). Here we are trying to predict deficit or excess of bikes on a Wednesday at 8 am in the morning. Through our forecasting we predicted there would be around 25 bike trips at that hour. After looking into Average Bike availability at 8am on this station and accounting for Dock capacity , we figured out that we were in shortage of 13 bikes to fulfill this trip requirement. Therefore, we would require rebalancing from nearest station after we predict demand on this stations at 8am. We calculated nearest 5 station using haversine formula.
The nearest station to this station(Station Id 70) are(in terms of Station Id):-

```
Out[154]: [61, 62, 64, 65, 69, 70]
```

After predicting demand at these station at 8am, our linear optimization algorithm works out rebalancing in following manner,

```
total deficiency :-13
get 6 from station 61
get 3 from station 62
get 3 from station 64
get 1 from station 65
```

This kind of linear optimization is naive approach, lot more research on operation side, and analysing those data points would come in handy to develop a more accurate and efficient solution.

Recommendations to the business and Future Area of Work:

1. The model can be used by the companies like Ford Go Bike or City Bike ride to reasonably redefine the threshold of number of bikes to be kept at a specific station when rebalancing. Yes there will be some buffer count taken into this threshold value based on the busyness of the station. A question that can be answered is :- Does it make sense to keep 10 bikes at a station where average trips per hour have been 2 from past one year. If not much, atleast 5 bikes can be vacated. On the contrary, considering a very busy station where average number of trips per hour has been around 30. Based on the model prediction and changing trends in the future, the model will also be helpful in devising strategy to install or remove number of docks from a particular station.

2. Identifying nearest five stations (our e naive approach can be scaled and optimized) in case there is deficiency or surplus to get the rebalancing job done efficiently in terms of time and cost.

3. With MAE of 0.78 trips at one our busiest stations, we can help the business to restructure its investment and devising a better plan for scaling their business.

Risks/Challenges associated with our model.

Question we asked to ourselves- is the solution we are presenting continuous? How then model will behave when the based-on predictions, rebalancing has been done. How much accurate future predictions can be? Can we solve the optimization task? We analyzed certain trends in our model w.r.t current scope of our project but following are the comments on aforementioned questions:

1. If there has been a live feed available about the trips, certain time series models like ARIMA or SARIMA could have been useful to cover wider range of problems faced by the bike sharing companies. Having said this, it was interesting to observe time series being unfruitful on our model when ran on static data from past 2 years.

2. The model will lay a strong foundation towards the optimization problem to correctly identify the count of bikes that can be moved from nearest stations where there are excess of bikes available.

3.The latent demand is not recorded in our current model or datasat. Having this data points will profoundly improve accuracy of our prediction of demand.

4. Thoughts about incorporating weather and geographical data - We hypothesized number of trips to be highly correlated to weather data reasoning obvious reasons for human predicaments. We validated the correlation also but it was interesting to observe only little improvement to the model when we included few important weather features. We attributed this behavior . However, provided the live feed, certain trends can be expected and which were not observed in our model such as seasonality in the number of trips being made etc.

5.This model only performs for station that have recorded respective data. This model cannot be reproducible for new stations where data has not been recorded.

References:

Bike Sharing System Overview
https://en.wikipedia.org/wiki/Bicycle-sharing_system

DataSet/Data Overview
https://www.kaggle.com/benhamner/sf-bay-area-bike-share
https://www.fordgobike.com/system-data

Time Series Modelling and Forecasting
https://medium.com/open-machine-learning-course/open-machine-learning-course-topic-9-time-series-analysis-in-python-a270cb05e0b3

Data Preprocessing and other modelling approaches
Applied Data Science Lecture Slides