Predicting net change in bikes at each station: Beachboys bike share

AIM:

The task is to predict the net change in the stock of the bikes at any selected station.

Summary:

This document describes the analysis and various steps performed in the process of predicting the net change in the bikes at any given station. The analysis is based on the station data, weather report and trip data that influence the bike changing rate. After exploring the data by calculating summary and descriptive statistics, and by creating visualizations of the data, several potential relationships between weather, station and trip features to the net change in bikes were identified. After exploring the data predictions are performed.

For predicting tensorflow - GPU model is used, model is written to make prediction for any given station.

Datasets available:

Station information,

Weather information,

Trip information

Exploratory Data Analysis (EDA):

1. Station Dataset

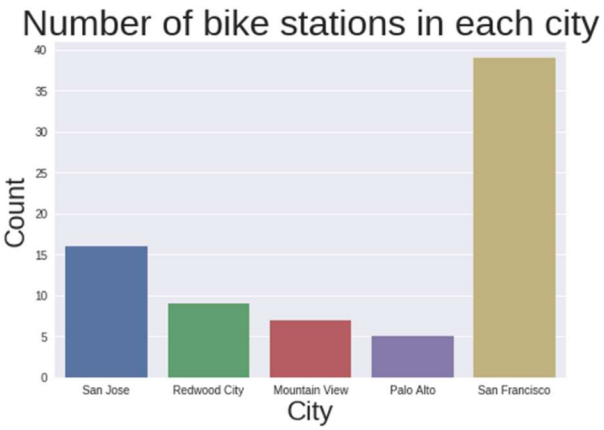
1. Checking for missing data first in this data set and found there is no missing data.

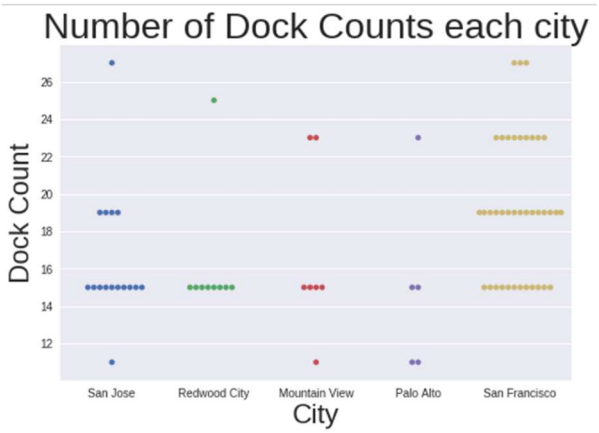
2. There are 76 unique values in the “Id” representing each station, this column seems to be very significant for locating stations.

3. The columns “Lat” and “Long” helps to locate the columns and there proximity on the map.

4.The “Dock Count” column gives information about number of docking spots for each station, more than half of the stations are having “15” docking points.

Dock Counts in each city:





5. The "City" column gives information of city in which station is located. There are 4 cities namely

Cities vs Number of stations plot:

Dropping Columns:

After exploring the data, the columns *"Name", "Lat", "Long"* are dropped.

May be in the later stages of the analysis we may need the information about "Lat" and "Long".

Weather data exploration:

1.When checked for null values, There are null values in this dataset.

By calculating the percentage of null values to the total dataset, if the null values are greater than 50% the column is deleted. In this case *"Events"* column in the dataset is dropped.

The NaN are filled with *"ffill"* option.

However later in the prediction stages that can be filled by cross comparing the columns.

To explore more about this dataset it would be more clear to merge this dataset with trip dataset.

This is a supporting dataset that makes more sense when combined with the main data.

Trip Dataset Summary:

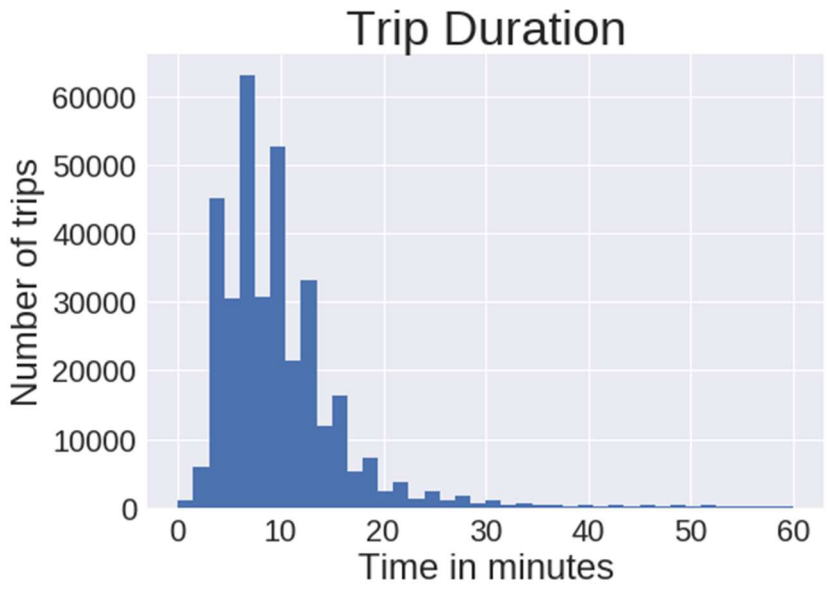
Conversion and dropping columns:

The “Trip ID” column is just the random number generated for each trip which may disturb the prediction for that reason the column dropped.

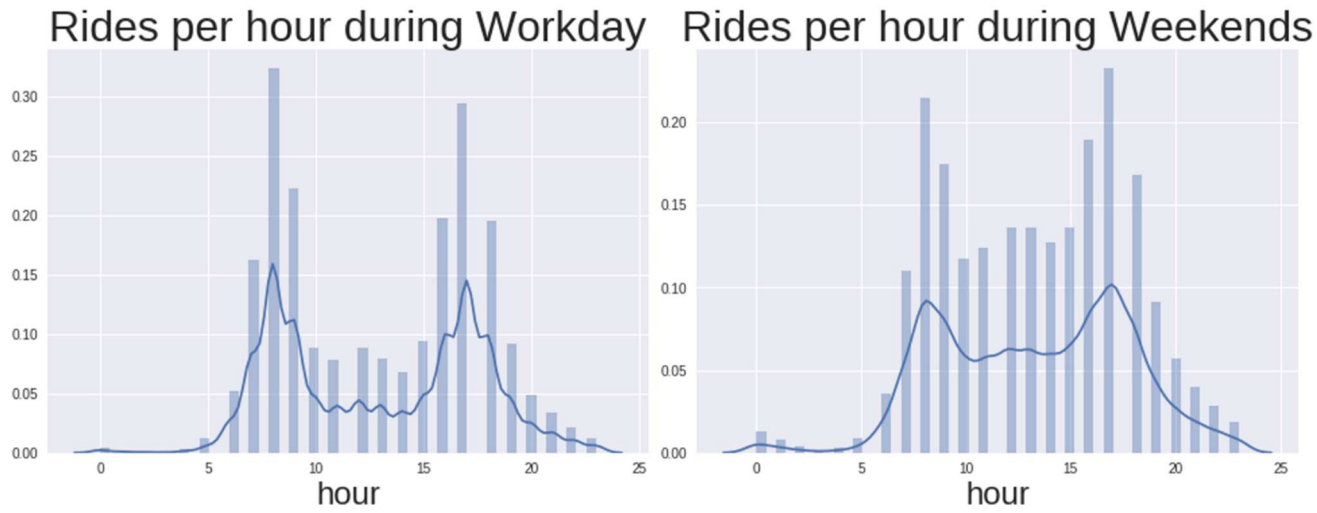
The “Subscriber Type” column is a categorical column with just two types, so hence it is converted with one hot encoding method.

Basic Analysis:

From the given trip data, the average duration of the trip is plotted by taking the time delta between start and end time of a trip.



The above plot shows the general trip durations between stations. There are very few rides that exceeds 20 minute trip duration.



Later the trips between the same stations that are less than minimum duration = 120 seconds are eliminated, considering this as outliers in the data.

After calculating the average trip duration, the information about bike rides per each hour would give the information about peak hours

For measuring busy hours, the data is separated for the weekdays and weekends because the combine result may mislead the information as shown in the plot above. From the plot it can be concluded that number of rides during “Weekends” is spread-out during entire day compared to number of rides during “Weekdays” which are high during office start hours/end hours followed by less trips during midday.

Merging the datasets:

Since there are three different datasets with information regarding to the bike data, datafames merging is performed.

First the datasets station and weather condition are merged. This is done by mapping “zip” column of the weather dataset to City Name. After changing the “Zip” column to “City” the merge is performed on “City” column of both (Station and Weather) datasets. The idea is to get the information of weather on each single station for each day listed. The new dataframe is “Weather_station_dup”.

Next step is to get the weather information on the trip data for each single day, for this to happen the merge is performed on the (“Date” and “Start Station”) columns of the two dataframes and the new dataframe created is “trip_weather_station_ID”. This new dataframe has weather report for each day at each station.

There is also one more merging operation preformed after performing analysis on trip data column.

Modelling:

By looking at the dataset It seems like a timeseries dataset, as the output expected is based on time hence time series analysis is performed.

Three new columns are created from the available trip dataset they are :

“net_change_bikes” : This columns is the subtraction between number of bikes starting at the given station and number of bikes ending at the same given station at a given hour.

“total_bikes_change”: This is cumsum of the “net_change_bikes” column.

“tota_bikes_pct_change”: This is the percentage change calculated on “total_bikes_change” column.

These three columns can be a potential targets/ label column, however after several attempts the column “total_bikes_change” column is chosen as a “target” column. This column gives information about total change in bikes and due to “cumsum” data it is also linearly dependent. This helps model to learn efficiently.

Feature columns that are input into the model are:

“Start Date, Max Temperature F, Max Wind SpeedMPH, Min TemperatureF, dayofweek, month”

Target column is : “total_bikes_change”

Model Applied: Neural Networks GRU

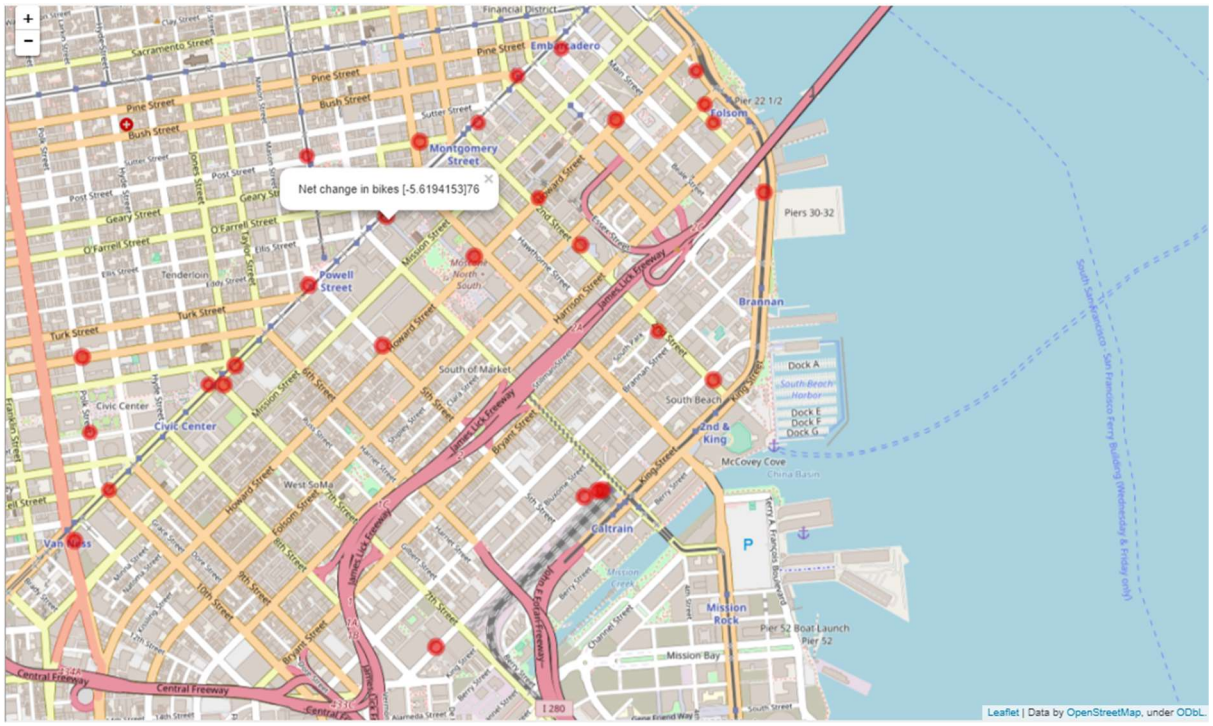
As the data is timeseries GRU model is applied which performs better for sequential data. The data is shifted by a day i.e 24 hours and fed into neural networks.

This model seems to be working well, still there is a large room of improvement provided some more time. However, the pattern and followthrough gives us the hindsight about possible improvement.

The accuracy obtained is:

```
CPU times: user 5 µs, sys: 0 ns, total: 5 µs
Wall time: 10.5 µs
Epoch 1/5
1/1 [==============================] - 2s 2s/sample - loss: 0.0759 - mean_absolute_error: 0.2702
100/100 [==============================] - 30s 300ms/step - loss: 0.0864 - mean_absolute_error: 0.2622 - val_loss: 0.0759 - val_mean_absolute_error: 0.2702
Epoch 2/5
1/1 [==============================] - 1s 1s/sample - loss: 0.1478 - mean_absolute_error: 0.3719
100/100 [==============================] - 28s 281ms/step - loss: 0.0801 - mean_absolute_error: 0.2508 - val_loss: 0.1478 - val_mean_absolute_error: 0.3719
Epoch 3/5
1/1 [==============================] - 1s 1s/sample - loss: 0.0389 - mean_absolute_error: 0.1801
100/100 [==============================] - 28s 281ms/step - loss: 0.0779 - mean_absolute_error: 0.2376 - val_loss: 0.0389 - val_mean_absolute_error: 0.1801
Epoch 4/5
1/1 [==============================] - 1s 1s/sample - loss: 0.0497 - mean_absolute_error: 0.2116
100/100 [==============================] - 28s 280ms/step - loss: 0.0733 - mean_absolute_error: 0.2269 - val_loss: 0.0497 - val_mean_absolute_error: 0.2116
Epoch 5/5
1/1 [==============================] - 1s 1s/sample - loss: 0.0156 - mean_absolute_error: 0.0859
100/100 [==============================] - 28s 280ms/step - loss: 0.0541 - mean_absolute_error: 0.1814 - val_loss: 0.0156 - val_mean_absolute_error: 0.0859
<tensorflow.python.keras.callbacks.History at 0x7fde3bc1e160>
```

The final result is that the model can predict the net change in bikes at time based on the current time in the laptop, which can be used to real time applications and easy data visualization :



By clicking on the Red circles the information of the bikes at the station is available.

Potential Improvements:

The data available is very rich and with this data the possibility of predicting the net change in bikes can be improved. Currently to give the information about all stations for each hour, the model takes each individual station and runs the model separately on each station. May be there is a way that all the stations can be data can be processed parallelly to give the output. Spectral Clustering can also be used to understand the frequent trips between the station and also connectivity.

Conclusion:

Data Analysis is performed by exploring and visualizing the data provided. From the Data Analysis conclusions are made and the new columns are created. Target column is selected based on the correlation plotting. Neural networks GRU model is used for predicting the net change in bikes at given station.