CAB DEMAND PREDICTION
README

1. Problem Statement

- Given a geographical location in a city, the model developed should be capable enough to predict the number of cabs (demand for cabs) that can be placed in that given location. So that the cab drivers can move to that location and increase their chance of receiving a pick-up.

2. Problem Objective

- The main objective of the problem is to predict the number of pickups for every region as precisely as possible given a future time interval. For example, to predict the cab demand at a region "ABC" from 5:00 pm to 5:10 pm. In this problem, more than the absolute error, the percentage error becomes prominent. By this we can notify the cab driver about the error percentage in prediction of cab demand in his surrounding regions.

3. Dataset Description

- The data is available from 2009 to 2021 for various cabs such as yellow-taxi, green taxi, for-hire vehicles. The required datasets can be found here. June-2016 data of Yellow taxis is considered for this problem.
- **"dask"** is a powerful library which is used when data/file size is greater than RAM size. It breaks the entire large file into various smaller chunks, brings one/multiple chunks (depending on RAM size) into the memory and performs operations on each chunk once at a time and then replaces the existing chunks in the memory with new chunks and repeats the operations. The similar procedure is carried out until all the chunks are covered.
- The available features in the dataset with their short description can be seen below,
  - **Vendor ID :** There are 2 vendors. They are mapped to IDs 1 and 2 respectively.
  - **Pickup-datetime :** The date & time recorded when the passenger was picked up.
  - **Drop Off-datetime :** The date & time recorded when the passenger was dropped.
  - **Passenger count :** Number of passengers in the vehicle.
  - **Trip distance :** The distance traveled during the trip.
  - **Pickup-Longitude :** Longitude where the pickup was made.
  - **Pickup-Latitude :** Latitude where the pickup was made.
  - **RateCodeID :** There are 6 categories in this feature mapped to integers from 1 to 6. Each rate code depicts the fare to be applied for that trip.
  - **Store_and_forward_Flag :** It is a boolean feature. If YES, the data will be stored and forwarded to the central server whenever there is an internet connection.
  - **DropOffLongitude :** The longitude at which the cab is dropped.

  - **DropOffLatitude :** The Latitude at which the cab is dropped.

- **Payment_type :** There are 6 payment types and each of them are mapped from 1 to 6.
- **Fare_amount :** the amount for the trip.
- **Extra :** The surcharges for such as peak hours and overnight charges.
- **MTA_tax :** the tax imposed on metered rate.
- **improvement _surcharge :** extra surcharge that can be collected due to hike in rate.
- **Tip_amount :** the credit card tips given.
- **Tolls_amount :** Total amount paid for the trip
- **Total_amount :** Entire amount paid by the customer excluding tips. (vid : 5 over)

4. Mapping To a ML Problem & Possible Performance Metrics that can be used

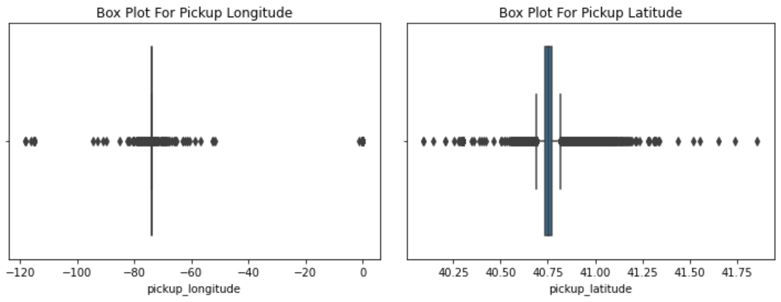
- This problem can be mapped to a time-series prediction problem. We are aware of the number of pickups happening at time “T” and we need to predict the number of pickups happening at time “T+1”.
- The performance metrics that can be used in this problem can be,
  - **Mean Squared Error (MSE)**
    - It computes the mean square of (actual value - predicted value) for all the samples.
  - **Mean/Median Absolute Percentage Error (MAPE/MDAPE)**
    - It computes the percentage error for all the predicted values and throws out the mean/median for all of them as a final value.
    - A_t : Actual Value ; F_t : Predicted Value

$$M = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|,$$

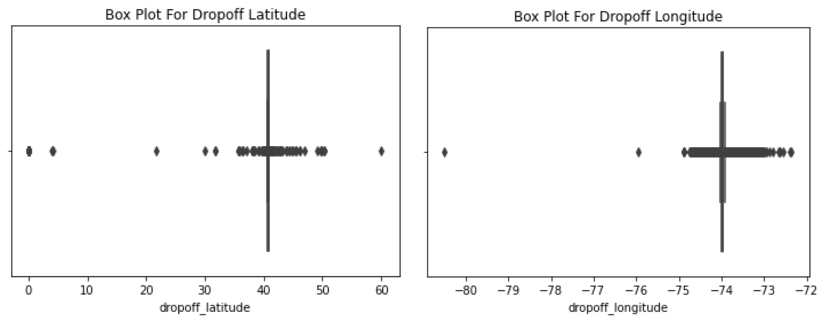
    - **MDAPE** would be less affected by outliers. Also Percentage error would be tending to be infinite when At ~ 0. That would affect MAPE So it would be better to use MDAPE.

5. Data Cleaning

- Check the raw data if there are any data points beyond latitude and longitude locations of New york city.Newyork is bounded by the location **(40.5774 lat,-74.15 long) & (40.9176 lat,-73.7004 long)**

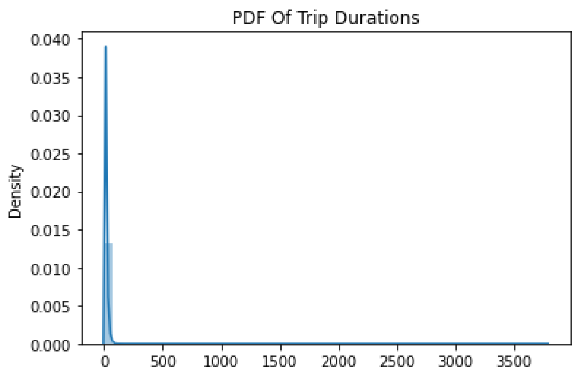


CAB DEMAND PREDICTION
README



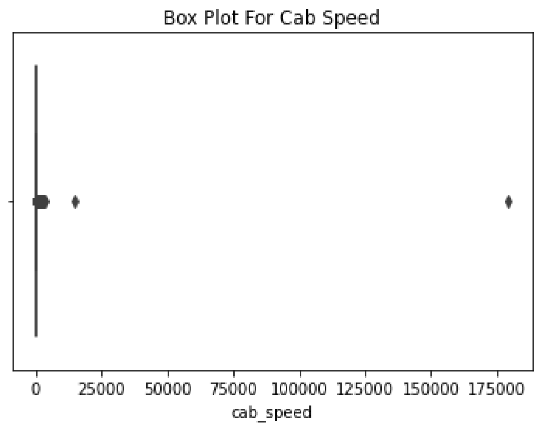
- Considering the box-plots it is clearly evident that there are some outliers corresponding to the location boundaries and they were removed.

- As per regulations in new york, the trip durations (Dropoff Time - Pickup Time) should be within 720 minutes. So we remove the trips with durations >720 minutes and negative durations. The PDF of trip times is shown below for better interpretation. This indicates there are some outliers.



```
-   Total Erroneous Records Corresponding to Trip durations :  55032
-   Percentage of Data records that are to be removed :  0.502777509531915
```

- Similarly there are erroneous values in cab speed which are obtained from distance and time columns. The impossible speeds can be cleaned from here. (The Box plot for cab speeds can be seen below which indicates that there are clearly some outliers.

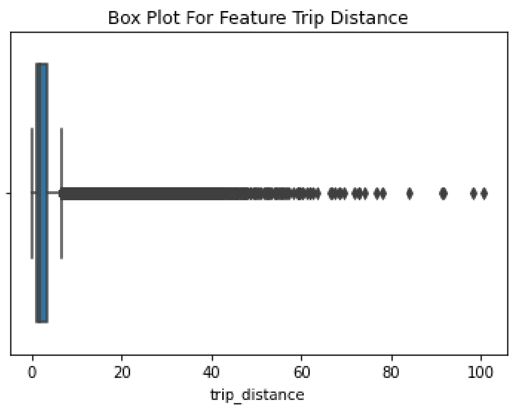


```
25th percentile value in cab_speed :: 7.38
```

```
50th percentile value in cab_speed :: 10.200000000000001
75th percentile value in cab_speed :: 14.0
90th percentile value in cab_speed :: 19.153846153846157
91th percentile value in cab_speed :: 19.8
92th percentile value in cab_speed :: 20.52
93th percentile value in cab_speed :: 21.333333333333332
94th percentile value in cab_speed :: 22.32
95th percentile value in cab_speed :: 23.542105263157897
96th percentile value in cab_speed :: 24.868965517241378
97th percentile value in cab_speed :: 26.75
98th percentile value in cab_speed :: 29.28
99th percentile value in cab_speed :: 33.31034482758621
99.1th percentile value in cab_speed :: 33.9375
99.2th percentile value in cab_speed :: 34.61538461538461
99.3th percentile value in cab_speed :: 35.4
99.4th percentile value in cab_speed :: 36.1764705882353
99.5th percentile value in cab_speed :: 37.18181818181818
99.6th percentile value in cab_speed :: 38.35
99.7th percentile value in cab_speed :: 39.77777777777778
99.8th percentile value in cab_speed :: 41.75
99.9th percentile value in cab_speed :: 44.82279215686364
99.9th percentile value in cab_speed :: 44.82279215686364
99.91th percentile value in cab_speed :: 45.3
99.92th percentile value in cab_speed :: 45.857142857142854
99.93th percentile value in cab_speed :: 46.5
99.94th percentile value in cab_speed :: 47.333333333333336
99.95th percentile value in cab_speed :: 48.24
99.96th percentile value in cab_speed :: 49.92
99.97th percentile value in cab_speed :: 54.347888144813645
99.98th percentile value in cab_speed :: 96.0 (Outliers)
99.99th percentile value in cab_speed :: 228.0
100th percentile value in cab_speed :: 179331.75,Now we remove the
records with cab speeds outside 1 and 99.97th percentile value.
```

- **Total Erroneous Records Corresponding to Cab Speeds : 29820**
- **Percentage of Data records that are to be removed : 0.27381499490614125**

- There are some huge values (outliers) in the column trip_distance as well, the box Plot below indicates the same, the percentile values are as follows,

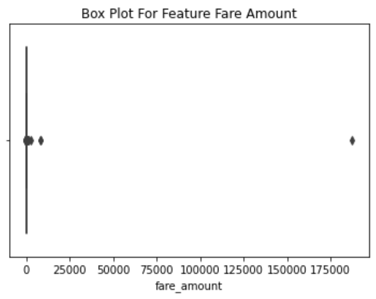


```
25th percentile value in trip_distance :: 1.02
50th percentile value in trip_distance :: 1.73
75th percentile value in trip_distance :: 3.22
90th percentile value in trip_distance :: 6.98
91th percentile value in trip_distance :: 7.62
92th percentile value in trip_distance :: 8.4
93th percentile value in trip_distance :: 9.1
94th percentile value in trip_distance :: 9.8
95th percentile value in trip_distance :: 10.6
96th percentile value in trip_distance :: 11.69
97th percentile value in trip_distance :: 13.9
98th percentile value in trip_distance :: 16.97
99th percentile value in trip_distance :: 18.5
99.1th percentile value in trip_distance :: 18.7
99.2th percentile value in trip_distance :: 18.95
99.3th percentile value in trip_distance :: 19.2
99.4th percentile value in trip_distance :: 19.51
99.5th percentile value in trip_distance :: 19.9
99.6th percentile value in trip_distance :: 20.4
99.7th percentile value in trip_distance :: 20.9
99.8th percentile value in trip_distance :: 21.53
99.9th percentile value in trip_distance :: 22.8
99.9th percentile value in trip_distance :: 22.8
99.91th percentile value in trip_distance :: 23.1
99.92th percentile value in trip_distance :: 23.47
99.93th percentile value in trip_distance :: 23.95
99.94th percentile value in trip_distance :: 24.6
99.95th percentile value in trip_distance :: 25.5
99.96th percentile value in trip_distance :: 26.5
99.97th percentile value in trip_distance :: 27.6
99.98th percentile value in trip_distance :: 28.49
99.99th percentile value in trip_distance :: 30.0
```

100th percentile value in trip_distance :: 100.63 Now we remove the records with trip distances outside 0 and 99.99th percentile value.

- **Total Erroneous Records Corresponding to Trip Distances : 1075**
- **Percentage of Data records that are to be removed : 0.009898031856930625**

- **fare_amount** also has some outliers that can be identified using box plots, percentile approach and are removed. The box plot for the same could be seen below,



```
25th percentile value in fare_amount :: 6.5
50th percentile value in fare_amount :: 10.0
75th percentile value in fare_amount :: 15.0
90th percentile value in fare_amount :: 25.5
91th percentile value in fare_amount :: 27.0
92th percentile value in fare_amount :: 28.5
93th percentile value in fare_amount :: 30.5
94th percentile value in fare_amount :: 32.5
95th percentile value in fare_amount :: 35.5
96th percentile value in fare_amount :: 39.0
97th percentile value in fare_amount :: 45.5
98th percentile value in fare_amount :: 52.0
99th percentile value in fare_amount :: 52.0
99.1th percentile value in fare_amount :: 52.0
99.2th percentile value in fare_amount :: 52.0
99.3th percentile value in fare_amount :: 52.0
99.4th percentile value in fare_amount :: 52.0
99.5th percentile value in fare_amount :: 52.0
99.6th percentile value in fare_amount :: 52.0
99.7th percentile value in fare_amount :: 53.0
99.8th percentile value in fare_amount :: 56.5
99.9th percentile value in fare_amount :: 63.0
99.9th percentile value in fare_amount :: 63.0
99.91th percentile value in fare_amount :: 64.5
99.92th percentile value in fare_amount :: 65.5
99.93th percentile value in fare_amount :: 67.0
99.94th percentile value in fare_amount :: 68.5
99.95th percentile value in fare_amount :: 70.0
99.96th percentile value in fare_amount :: 72.5
99.97th percentile value in fare_amount :: 76.0
99.98th percentile value in fare_amount :: 82.5
99.99th percentile value in fare_amount :: 100.0
100th percentile value in fare_amount :: 187440.96
```

CAB DEMAND PREDICTION
README

- Total Erroneous Records Corresponding to Fare Amount : 5115
- Percentage of Data records that are to be removed : 0.047100878755984295

- **Percentage of data removed as a result of data cleaning : 2.52% (280915 records)**

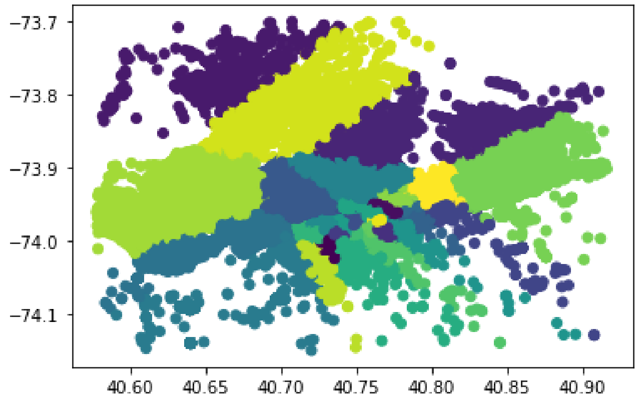
6. Data Preparation

6.1 Clustering For Determining Regions

- Given a 10 minute time bin and a region we need to predict the number of possible pickups that can be demanded from that region. So for that firstly we need to cluster the regions from the given data.
- For the purpose of clustering, the K-means algorithm has been used. The Average cab speed is found to be **12 miles/hour. So, approximately a cab travels a distance of 2 miles in 10 minutes.** So if we have an inter-cluster distance of 2 miles then the driver could travel from one region to another region in 10 minutes of time which is quite reasonable. But also the inter cluster distance should not be too low i.e the regions should not be too close (for example one side of the road could not be one region other side of the road could not be another region). So for this we can have a minimum inter-cluster distance to be 0.5 mile.
- The Optimal-K in K-Means is determined by the elbow method. The results for different K values can be seen below,

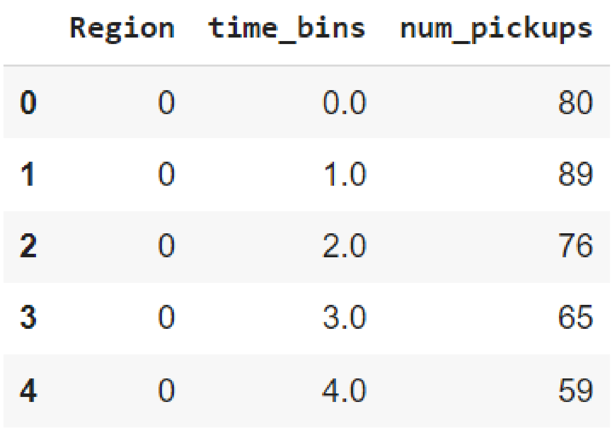
```
If Number of Clusters = 10 Then,
Minimum Inter Cluster Distance = 0.971344561613067 ; % of Clusters within the vicinity = 35.55555555555556
======================================================================
If Number of Clusters = 20 Then,
Minimum Inter Cluster Distance = 0.7218028077681672 ; % of Clusters within the vicinity = 42.10526315789474
======================================================================
If Number of Clusters = 30 Then,
Minimum Inter Cluster Distance = 0.4644664351008461 ; % of Clusters within the vicinity = 47.81609195402299
======================================================================
If Number of Clusters = 40 Then,
Minimum Inter Cluster Distance = 0.45540587415265227 ; % of Clusters within the vicinity = 43.333333333333336
======================================================================
If Number of Clusters = 50 Then,
Minimum Inter Cluster Distance = 0.42479941565475654 ; % of Clusters within the vicinity = 41.46938775510204
======================================================================
If Number of Clusters = 60 Then,
Minimum Inter Cluster Distance = 0.31294497764335827 ; % of Clusters within the vicinity = 43.61581920903955
```

- If the number of clusters are 30 the inter cluster distance is close to 0.5 and also nearly 50% of all possible pairs of clusters have distance within the range 0.5 to 2 miles (2 miles is the upper limit because it can be traveled in 10min). So, the region count of 30 would be reasonable enough to divide NYC.
- The scatter plot of 30 regions looks as follows,



6.2 Time Binning

- We first convert the pickup time stamps into unix timestamps. The unix timestamps give us the total seconds lapsed till the given date starting from Jan 1st 1970. This is a featurization technique generally used for timestamp data.
- For creating time-bins, we subtract the "unix time-stamp of 1st date of the pick-up month of pick-up year(2016-06-01)" from "pick-up unix time-stamp (timestamp from Jan 1st 1970)" then divide the result by 600 to get the 10min Bin number to which the pickup belongs to.
- Since it is June month, the total number of minutes we have are 30 days * 24 hours * 60 minutes = 43200. So the total number of ten minute bins possible will be 43200/10 = 4320 bins (i.e from 0 to 4319).
- Data after creating time bins and computing number of pickups using group-by the data looks as follows (first 5 rows),

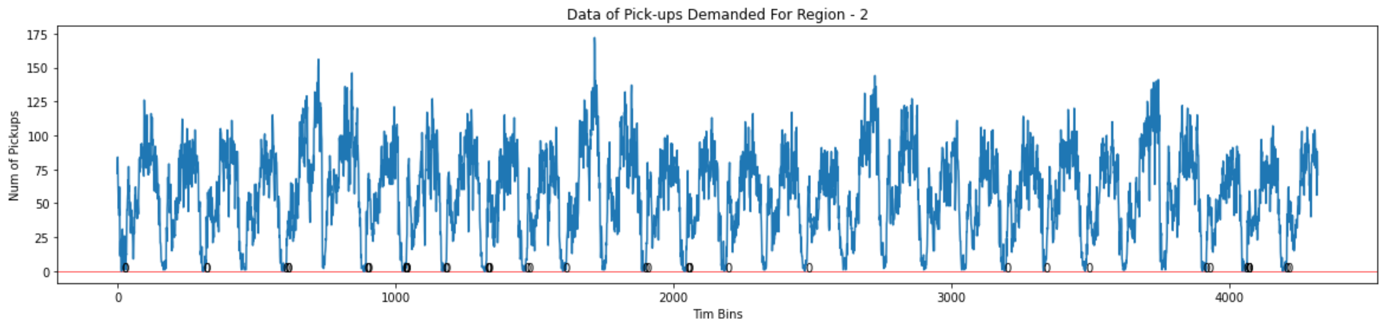


	Region	time_bins	num_pickups
0	0	0.0	80
1	0	1.0	89
2	0	2.0	76
3	0	3.0	65
4	0	4.0	59

Each row indicates the number of pickups demanded in a region/cluster for a given 10 minute bin.

6.3 Data Smoothing

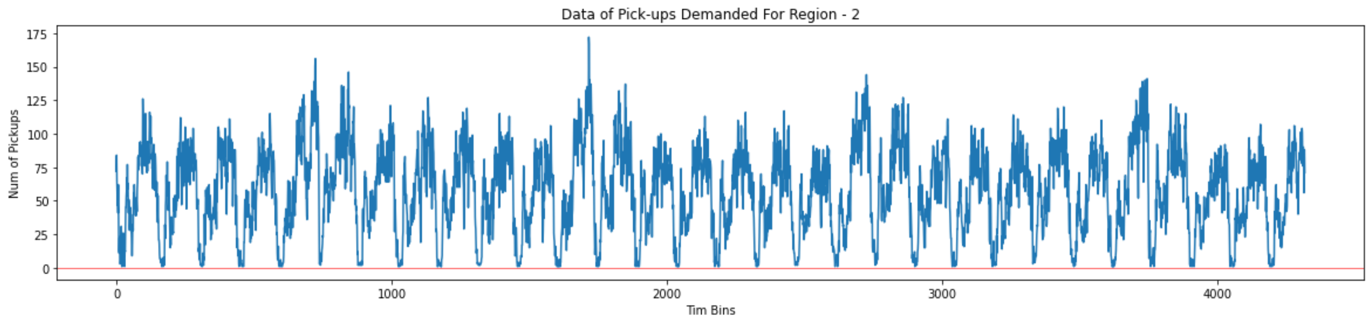
- Data has records where some regions/clusters might not have any pickups for a particular time-bin. That zero value needs to be smoothed to have a non-zero value. Because with these zero values ratio based techniques might yield division by zero error.
- A sample Region (region 2 here) time bins where numbers of pickups were 0 can be seen below, ("**0**" has been marked in the graph when the plot touches zero.



- The procedure for smoothing is as follows ("__" indicate zero value),

a __ b = ceil(a+b)/3, ceil(a+b)/3, ceil(a+b)/3
a __ __ b = ceil(a+b)/4, ceil(a+b)/4, ceil(a+b)/4, ceil(a+b)/4
a __ __ __ b = ceil(a+b)/5, ceil(a+b)/5, ceil(a+b)/5, ceil(a+b)/5, ceil(a+b)/5

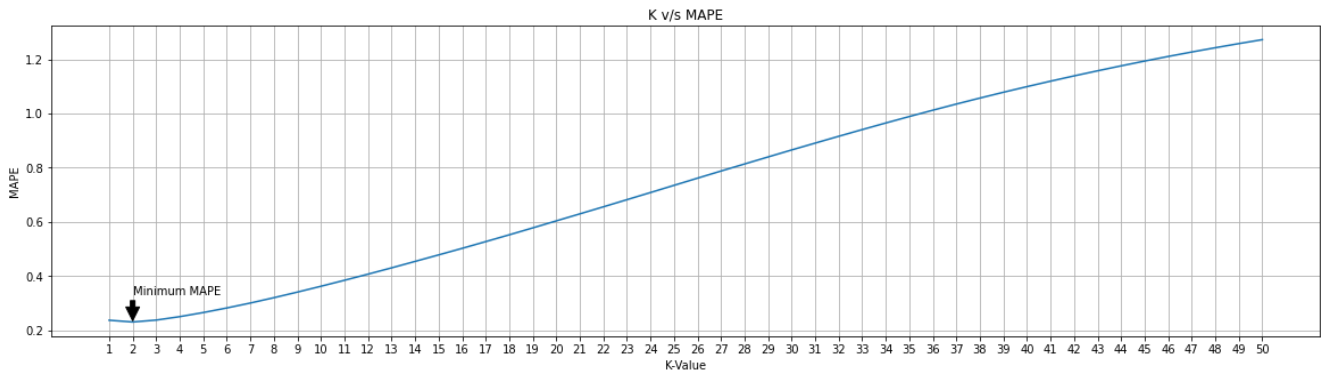
- The data plot of region-2 looks as follows after smoothing **(no zero pickups observed),**



7. Applying SMA (Simple Moving Average) technique on previous values

- The simple moving average is a technique in which we consider the average of previous 'K' values to determine the current value. The value of 'K' is a hyper parameter, which is determined by the elbow method. See the plot below,

CAB DEMAND PREDICTION README



- For the value of K=2 we are achieving the minimum MAPE. So, the optimal K would be 2. Formula for simple moving average to determine the current value is as below,

$$SMA = \frac{A_1 + A_2 + ... + A_n}{n}$$

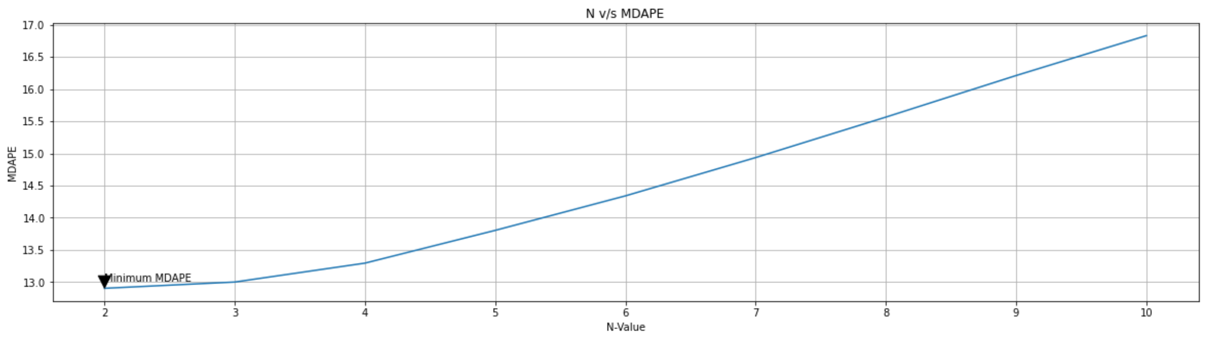
- The Mean average percentage error (MDAPE) with SMA approach is found to be : **17.92%**
- The problem with SMA is we try to give equal weightage to all the previous values to predict the current value.

8. Applying Weighted Moving Average (WMA)

- In the weighted moving average approach, we give more weightage to the most recent value which is considered for prediction and the weightage decreases as we move far from least recent prediction.
- The formula for weighted moving average is as follows,

$$P_t = (N * P_{t-1} + (N-1) * P_{t-2} + (N-2) * P_{t-3} \dots . 1 * P_{t-n})/(N * (N+1)/2)$$

- The value of 'N' is found using the elbow method. The optimal value of 'N' is found out to be 2. The plot for the same can be seen below,



- The MDAPE with N=2 using weighted moving average approach is found to be : **12.90%**

9. Applying Exponential Weighted Moving Average (EWMA)

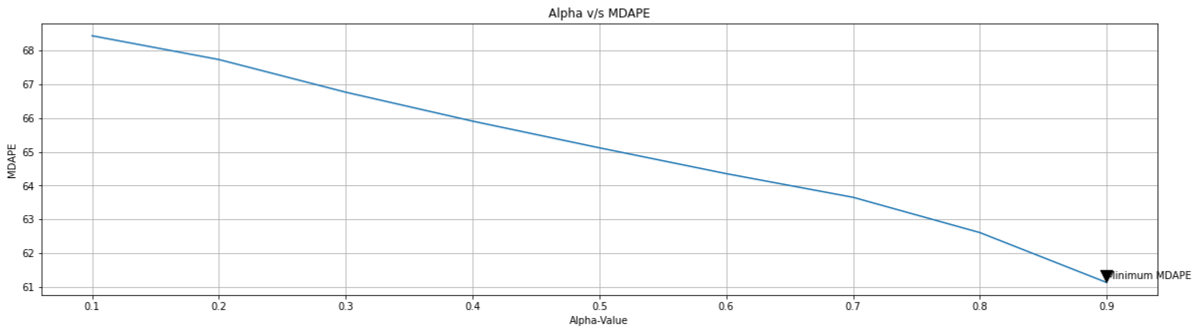
- In the EWMA approach we give complementary weightage to previous timestamp's ground truth value and predicted value, to determine the value at current timestamp.

CAB DEMAND PREDICTION
README

- The formula for Exponential weighted moving average can be seen below,

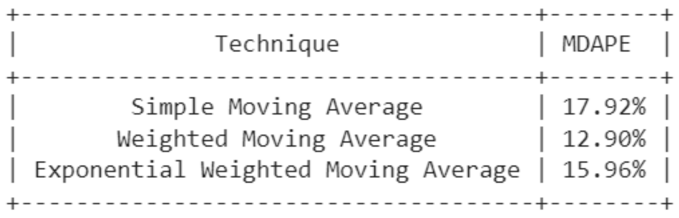
$$P'_t = \alpha * P_{t-1} + (1-\alpha) * P'_{t-1}$$

- The alpha in the above formula is the hyper parameter which ranges from 0 to 1. The optimal value of alpha is determined by using the elbow method. See the plot for the same below,



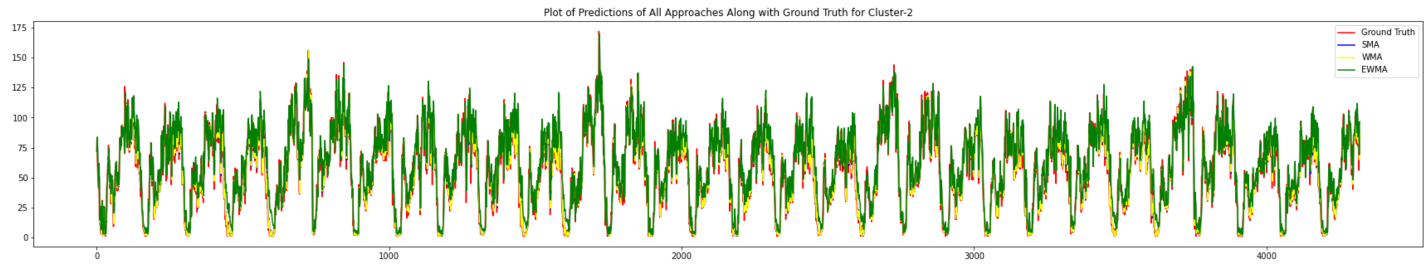
- The MDAPE with alpha=0.9 using exponential weighted moving average approach is found to be : **15.96%**

10. Final Results



```
+-------------------------------------+--------+
|              Technique              | MDAPE  |
+-------------------------------------+--------+
|        Simple Moving Average        | 17.92% |
|       Weighted Moving Average       | 12.90% |
| Exponential Weighted Moving Average | 15.96% |
+-------------------------------------+--------+
```

A plot with includes all the predictions along with ground truth for cluster-2 can be seen below,



11. Deployment link

The model has been deployed using the streamlit share cloud and the link for the same can be seen below,
https://share.streamlit.io/manikumarreddy35/cab_demand_prediction_deploy/main/app.py

12. Extensions/Future Work

- As an extension we can try building hand-crafted features and can apply machine learning techniques to predict the pick-up demand.