Analyzing Chicago Taxi Trips Behaviour via Data Mining

Shaurya Sahai, Abishek Krishnan, and Sushmitha Sunkurdi Nataraj

Introduction

According to kaggle.com *"There are about seven thousand licensed cabs operating within the city limits."* Our project focuses on taxi trips within the city of Chicago. We aimed at finding interesting insights about one of the major components of any trip - the average speed. We were able to correlate speed with a number of varying factors including but not limited to pickup and dropoff locations, time of the day, peak hours and others. The primary techniques used in our project are *multivariate regression* and *4D clustering*.

Motivation

We started off by trying to find insights that could help taxi aggregators interpret passenger behavior to reduce cancellations, also improve their service by deploying more taxis in areas of high demand during particular hours to reduce the response time and understand other interesting patterns observed.

Although, during the exploration phase - we concluded that the varying factors were way too many to handle and make successful demand predictions. The main questions we focused on later on were primarily related to speed. We wanted to answer questions like - *"If I travel at 5:30 PM on Monday, what would be my expected average speed?"* or *"What can I expect my average speed to be while I take a ride on Wednesday at 4 PM in Chicago downtown?"* In addition to this, we also tried answering questions like - *"What would be the cost of a trip if I am traveling for 5 miles"?* Or *"How long would it take to travel for 5 miles?"*.

Such analysis is used by various web mapping services which can predict the estimated time required for going from point A to point B. Using taxi rides' data, we could also generalize the prediction since most of the vehicles running on the streets are cars or in general "taxi-like". Hence average speed prediction is our focal point for this project.

Data

Our data was collected from Kaggle which in turn is hosted by Google BigQuery.
Kaggle Source - Chicago Taxi Trips
Google BigQuery Table - bigquery-public-data:chicago_taxi_trips.taxi_trips
This is being collected by the City of Chicago (the regulatory agency) from 2013 to the present. The agency does so through periodic reporting by two major payment processors believed to cover most taxis in Chicago. Currently, it has information about over 100 million Chicago taxi rides.

At the time when we began our project, the table had over 192,029,239 rows (Table size: ~69.6 GB). We extracted relevant data at each point using the BigQuery WebUI. One of our motivations for selecting this dataset was the curiosity to learn more about Google BigQuery. The web UI made it seamless for us to extract relevant data, especially during the exploration phase when we were not completely sure of what to look at.

Data Exploration

The dataset contains data from 2013 and is constantly updated. We decided to only experiment with data for 2016 since we didn't want to be bogged down by the huge size of data. This gave us a chance to invest our time in trying out more interesting things instead.

At each step, we cleaned the data of any outliers which might have resulted in skewed models. To achieve this, we used techniques like computing the z-score and only taking values that had z-score less than (*commonly used value*) 3. We also used boxplots to visualize and understand the outliers before cleaning them out. For example, during our data exploration, we came across a use case to find the average speed.

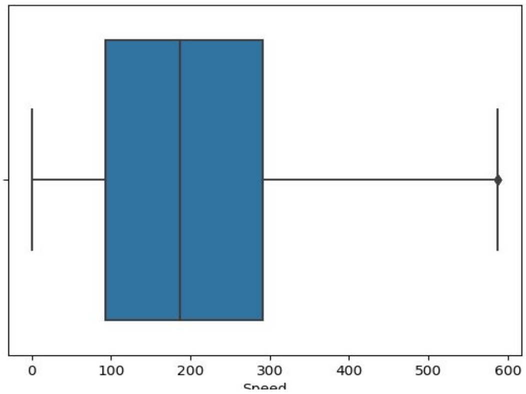
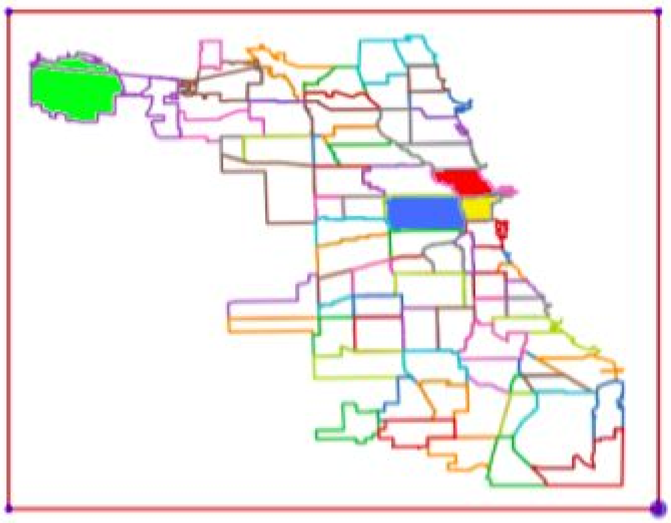


Fig 1: Data clean up for average speed

In the exploration phase, we garnered some insights about the part of the dataset we were dealing with and used data visualization as a technique in laying out relevant attributes.

a) The following plot shows the top 10 busiest routes during 2019.

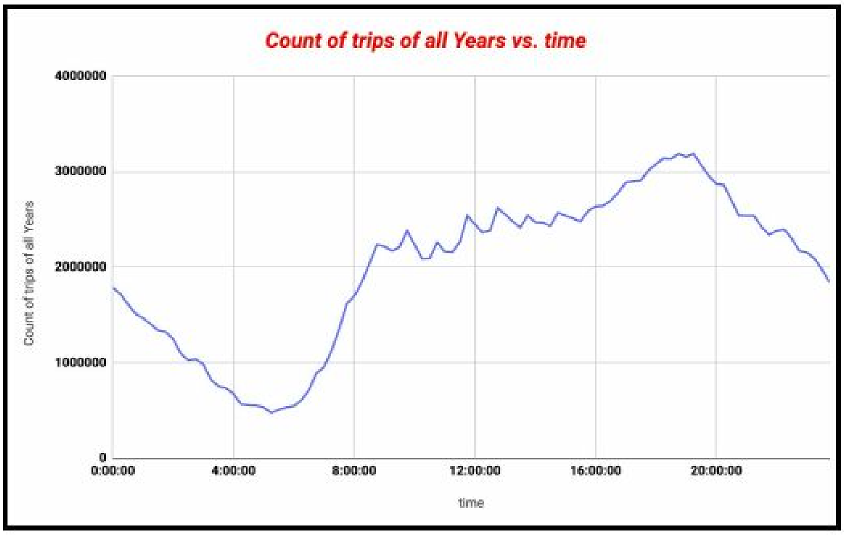


City of Chicago: Neighbourhood Map

As seen from the map, the color-filled neighborhoods had the most combinations of pickups and drop locations

i) Red and Yellow - Central
ii) Blue - West Side
iii) Green - Far North Side

b) On analyzing the correlation between the number of trips and the pickup time, we found the peak time for booking a taxi is between 5 PM - 8 PM and the off-peak hours were 3 AM - 6 AM (2013-Present)



Key Ideas

As mentioned earlier - we wanted to answer questions majorly about speed. We used regression as the underlying technique and varied our dataset according to the logical and practical barriers. We wanted to find out the answer to one particular question - “If someone were to take a trip, what would be their estimated average speed?” To answer this, we tried a few variations and performed regression on all of them. Next section talks of what we gathered from those experiments.

- **Based on the part of day:** Another criteria to classify the rides was tagging them as a ‘MORNING’, ‘NOON’, ‘EVENING’ or ‘NIGHT’ trip. We calculated the time slot for each ride using the start time of the ride. Since we wanted to calculate the average speed which would ultimately determine the end time. We created the following buckets:

- **Based on the hour of the day:** The last criterion for segregating the rides and thus creating prediction models was the hour of the day. Unlike the earlier approach where we created buckets with a few hours in each, this approach involved taking one-hour periods for time slots. For example, rides with a start time between 1 PM - 2 PM would be part of the train set for one regression model.

- **Based on months of the year:** This approach involved grouping the rides based on the month they were taken in. The idea was that holiday seasons might have a different rate of traffic as compared to non-holiday seasonal months. We were curious to find out if this would affect the average speed in a significant way. The test data for this experiment consisted of each calendar month’s rides for 2014, 2015 and

2016 in contrast with our other approaches where we only took 2016's data into consideration.

- **Based on pickup and dropoff locations:** For this experiment, we used 4D kmeans++ clustering and segregated the rides based on the cluster they belonged to. This was followed by using regression on each of the clusters separately. The motivation behind this approach was the difference in traffic situations in different areas of the city. Speed of rides in densely crowded areas would be different than speed of rides in sparsely crowded ones.

Experiments

Clustering

In the intermediate phase of our project, we used K-means++ clustering to make clusters of areas to aid in demand prediction. We used DBScan as a way to compare the clustering results. However, after deciding to move ahead with speed-related analysis we did a 4D clustering of data instead.

a) K-means++ [2D - Pickup latitudes and longitudes]

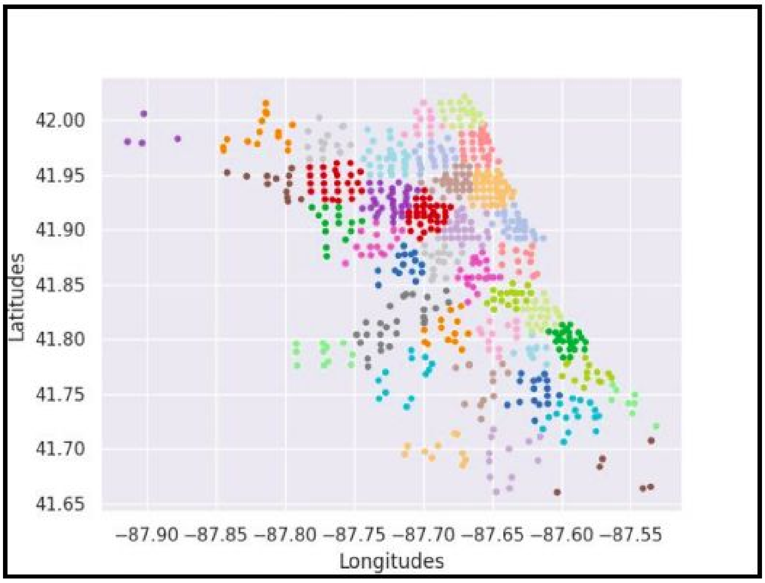
For the experiments below, we only consider data for the year 2016. The dataset provided pickup longitudes and latitudes. Our motivation behind clustering was to form pickup communities where the drivers could reach within a reasonable time. This would allow the drivers and users both to utilize the taxi-service efficiently.

The first step was to find the average speed of the taxi driver in Chicago. From our cleaned data which eradicated about 0.3% of it, we found out the average speed to be 15.07 miles/hr. This means a driver can cover around 2.5 miles in 10 minutes (reasonable wait time).

The next step in the process was to determine the number of clusters for the K-means++ clustering. From our calculation, the desired distance within a cluster was 2.5 miles. At the same time, we didn't want to have a majority of clusters less than 0.5 miles since that would render clustering useless.

The best ratio of clusters with the above specifications was found with 40 clusters.
The number of clusters less than 0.5 miles of distance increased substantially beyond this threshold; the number of clusters with a distance more than 2.5 miles increased substantially below this threshold. With 40 clusters we have 16 clusters which have 2.5 miles as the distance between their farthest points.

We used the haversine distance to calculate the distance between GPS coordinates.

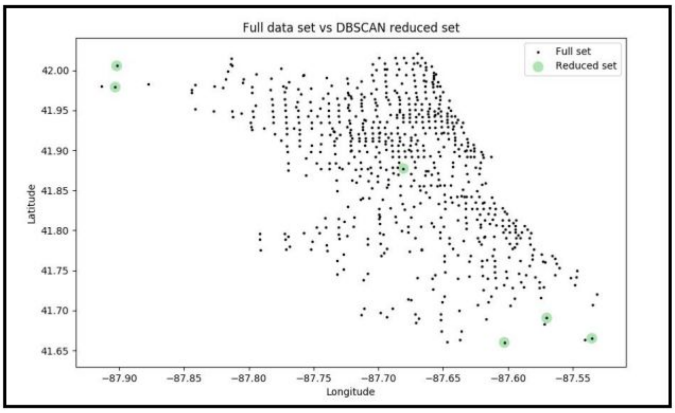


K-mean++ clustering of pickup locations.

b) DBSCAN [2D - Pickup latitudes and longitudes]

While we were exploring other algorithms for clustering, we decided to try DBSCAN as theory suggested it was superior to K-means clustering for geospatial points. We understood that k-means is not an ideal algorithm for latitude-longitude spatial data because it minimizes variance, not geodetic distance. DBSCAN clusters a spatial data set based on two parameters: a physical distance from each point, and a minimum cluster size. We use the haversine metric and ball tree algorithm to calculate great-circle distances between points. Unlike k-means++,
DBSCAN doesn't require us to specify the number of clusters in advance – it determines them automatically based on the epsilon(2.5 miles) and min_samples (1) parameters.

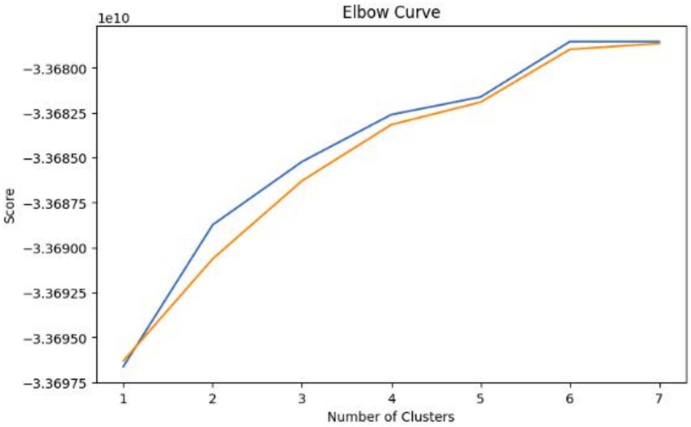
We see an interesting observation that DBSCAN has clustered 687 points down to 6 clusters, for 99.1% compression. Given the frequency of trips expected, it would be beneficial if the taxi aggregators had a bay somewhere near the "Near West Side", "South Deering" or "O'Hare" neighborhoods of Chicago to optimize both the driver and user time, which is in contrast with K-means++ result which clustered into 16 possible locations.



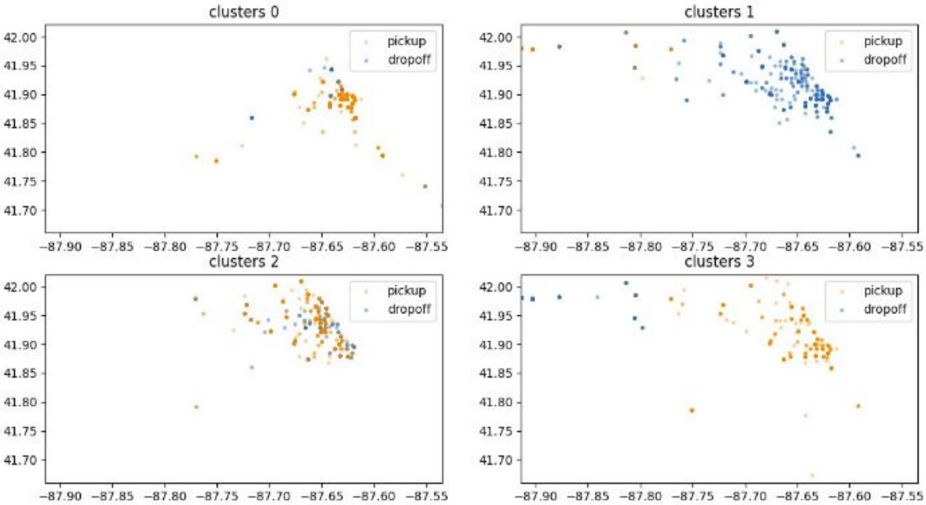
DBScan clustering

c) K-means [4D - Pickup, Dropoff Latitudes and Longitudes]

As suggested during our intermediate report meeting, we wanted to improve our metrics by clustering on four dimensions. We choose January 2016 month's data as a test model.
The elbow curve for the clustering was,



Therefore, we choose to perform k-means using 4 clusters.



We used these clusters to improve predictions, which are discussed below.

Some interesting facts we could conclude by just these clusters were:

i) Short trips are clustered together.
ii) Pickups were more concentrated than drop offs which might be because the taxi’s wouldn’t want to go too far for a pickup.

Simple Linear Regression

Linear regression attempts to model the correlation between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory or independent variable, and the other is considered to be a dependent variable.
The simple linear regression model is represented by $Y = \beta_0 + \beta_1 X + \varepsilon$
The error term is represented by ε, $\boldsymbol{\beta_0}$ is the y-intercept of the regression line, $\boldsymbol{\beta_1}$ is the slope, and ***E***(***y***) is the mean or expected value of ***y*** for a given value of ***x***. A regression line can depict a positive linear relationship, a negative linear relationship or no relationship.

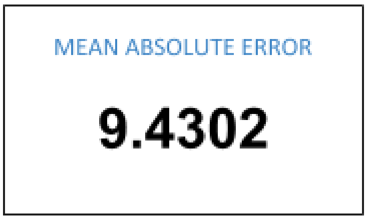
Implementation & Results

Key Idea 1: Based on the part of day

We classified the trips into being one of the morning, noon, evening or night trips based on the following time slots. For this experiment we then took the time slot as well as the day of the trip as explanatory variables. The dependent variable was speed.

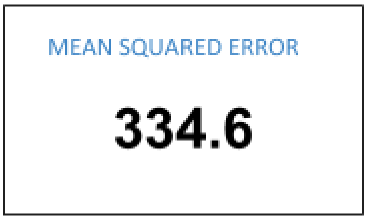
TIME SLOT	START TIME BETWEEN
Morning	4 AM - 12 PM
Noon	12 PM - 5 PM
Evening	5 PM - 9 PM
Night	9 PM - 4 AM

An analysis like this one was to answer questions of the kind - “How long would it take to take a ride on Monday evening?”



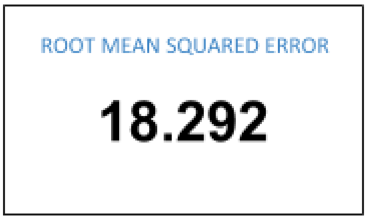
MEAN ABSOLUTE ERROR

9.4302



MEAN SQUARED ERROR

334.6



ROOT MEAN SQUARED ERROR

18.292

Key Idea 2: Based on hour of the day

We further fine tuned the time slot for the trips depending on the hour of day. Now each trip was classified based on its starting time between, say, 1PM - 2PM or 5PM - 6PM. The explanatory variables for this experiment still consisted of time slots (now, hourly) and day.

MEAN ABSOLUTE ERROR	MEAN SQUARED ERROR	ROOT MEAN SQUARED ERROR
5.123	46.77	6.839

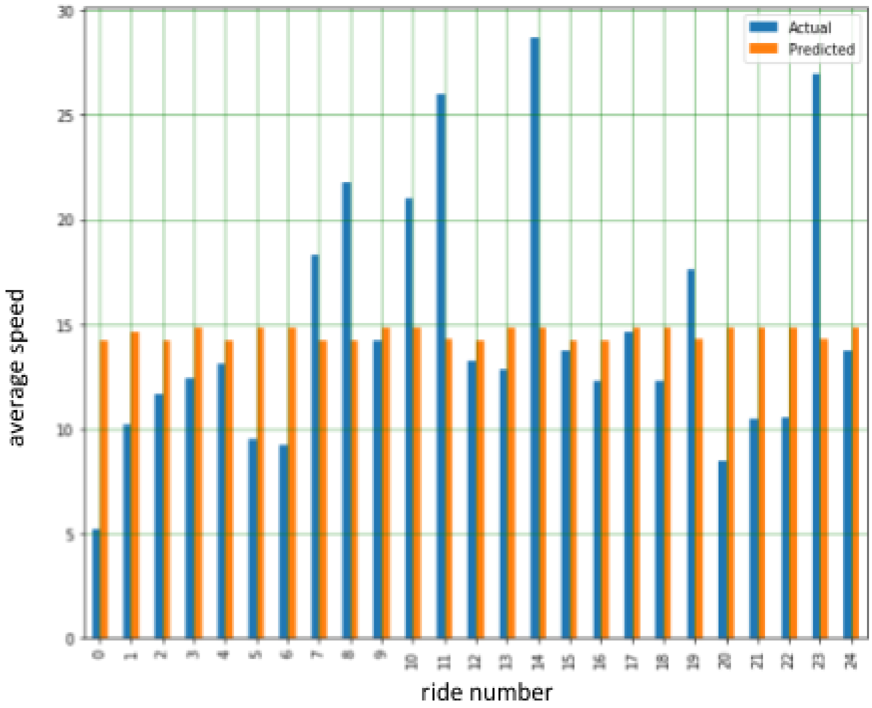


Fig. Comparison of actual & predicted speed of 2017’s first 25 rides

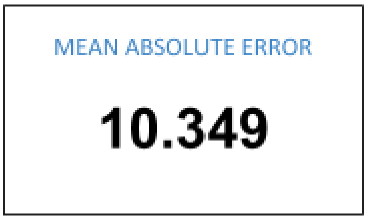
Key Idea 3: Based on peak and non-peak hours

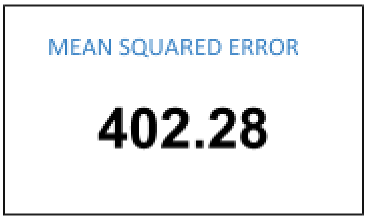
For 2016, we then calculated the peak and non-peak hours depending on the demand in 2016. We used the visualization below to classify rides taken between 4 PM - 8 PM as peak hour rides, rides taken between 4 AM - 8 AM as non-peak hour rides, and all the others in a separate bucket.

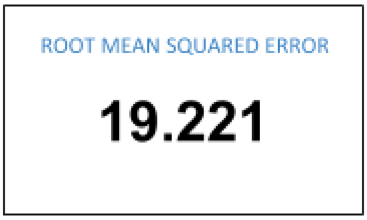
MEAN ABSOLUTE ERROR	MEAN SQUARED ERROR	ROOT MEAN SQUARED ERROR
9.361	333.3	18.256

Key Idea 4: Classification by month

We selected the best of the above three methods for time slotting and then further fine tuned our classification based on individual months of the year. This time we have 12 regression models based on each month of the year. Surprisingly, the results were no better than experiments done to implement Key Idea 2.


MEAN ABSOLUTE ERROR
10.349


MEAN SQUARED ERROR
402.28

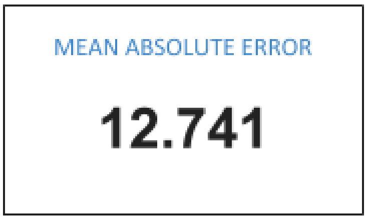

ROOT MEAN SQUARED ERROR
19.221

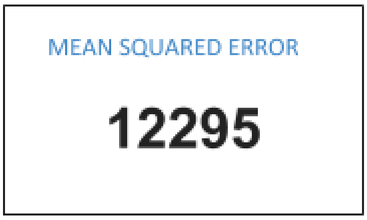
Key Idea 5: Based on area-wise clustering

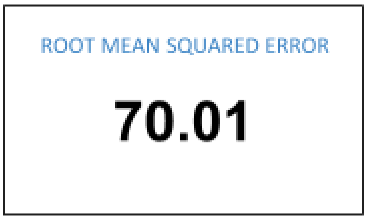
We relied on 4D clustering, as explained above to create 4 clusters. We then chose the best of our time slotting approaches from the above experiments and classified the data according to hours of the day. This gave us 4 models and the following are the average errors of those. For these models, we did not consider 2017's data. We instead went with 25% test data and 75% train data from the generated clusters.

Surprisingly this model performed the worst out of all the models based on the mean squared error metric to measure error. One possible cause might be lack of training data which resulted in a poor model.

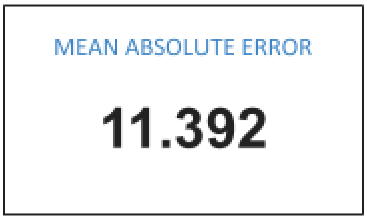
Cluster 1

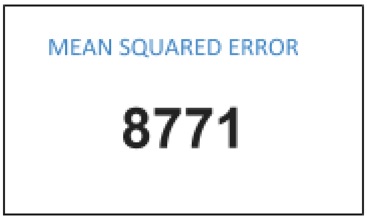

MEAN ABSOLUTE ERROR
12.741

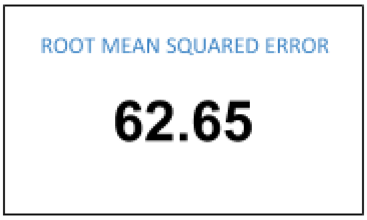

MEAN SQUARED ERROR
12295


ROOT MEAN SQUARED ERROR
70.01

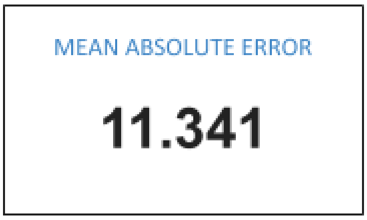
Cluster 2


MEAN ABSOLUTE ERROR
11.392

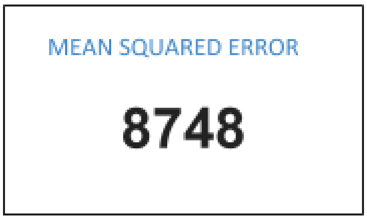

MEAN SQUARED ERROR
8771


ROOT MEAN SQUARED ERROR
62.65

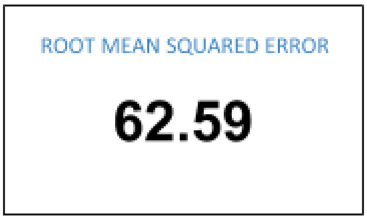
Cluster 3


MEAN ABSOLUTE ERROR

11.341

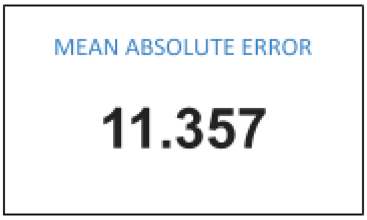

MEAN SQUARED ERROR

8748

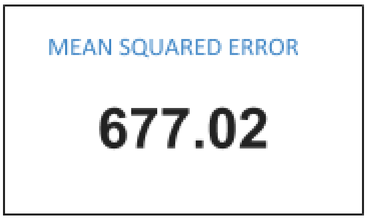

ROOT MEAN SQUARED ERROR

62.59

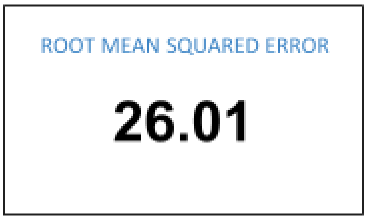
Cluster 4


MEAN ABSOLUTE ERROR

11.357


MEAN SQUARED ERROR

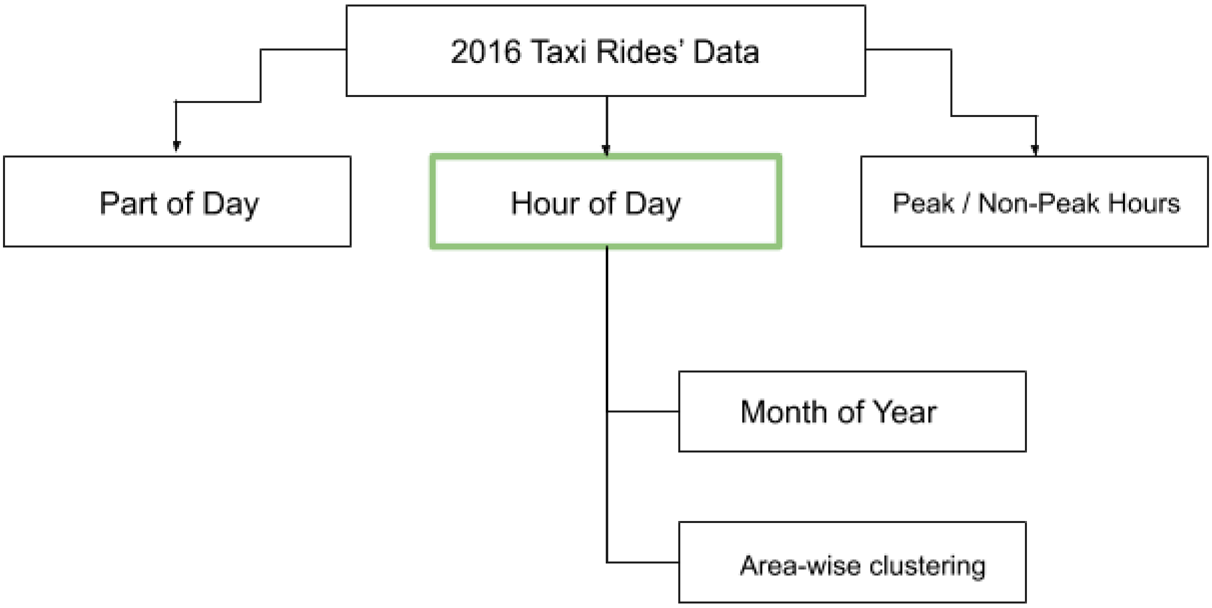
677.02


ROOT MEAN SQUARED ERROR

26.01

Pictorial Representation

Below is pictorial representation of our decision-making process at each step which involves classifications based on varied criteria.



Conclusion

From our experiments, we learned that the unclustered data classified by hour of the day works best in predicting the average speed of the ride. Although the error from this model is also huge which suggests that there are other factors which need to be accounted for while creating a robust and accurate prediction model.

Contributions

We would like to say that this process has been a great learning tool for all the team members. We all unanimously chose a dataset which seemed interesting and exploratory. We brainstormed our approaches in a friendly manner and each of us took an equivalent amount of work to complete this project. The specific details of each team member's contributions are as follows:

Abishek Krishnan

He primarily dealt with clustering the data in both the intermediate and final phase. He implemented the DBScan clustering of rides to garner comparative results with the k-means++ clustering. In the final phase, he worked on clustering the rides based on pickup and dropoff locations and extracted data for each of the clusters to perform regression on.

Shaurya Sahai

In the intermediate phase, she dealt with the k-means clustering and found out the average speed for a car ride in Chicago. In the final phase, she performed regression on the first three key ideas - which included writing code to clean data, check if there existed a linear relationship and finally generating the error values for those experiments.

Sushmitha Sunkurdi Nataraj

She extensively explored data in the intermediate phase and brought out a few interesting insights including the peak hours for trips, the correlation between trip fares and miles, etc. Since we changed our overall objective after the project meeting not all of the insights are part of this report. In the final phase, she worked on extracting the data for all the regression models and ran the experiment to generate models for the month wise and area-wise clustered data (key idea 4 and 5).

We would like to thank Prof. Jeff Phillips and all the TAs for their constant support and guidance during the entire course of the project even in unprecedented circumstances.