# FARE OPTIMIZER

***Dissertation submitted to***
***Shri Ramdeobaba College of Engineering & Management, Nagpur***
***in partial fulfillment of requirement for the award of***
***degree of***

## Bachelor of Engineering

In

## Computer Science and Engineering

*By*


**Krunal Pande**

**Kunal Khadkeshwar**

**Nachiket Satpute**

**Navin Rathi**

**Pranshu Singh**


*Guide*

**Prof Vrushali K Bongirwar**


**Computer Science and Engineering**
**Shri Ramdeobaba College of Engineering & Management, Nagpur 440 013**
(An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur University Nagpur)


**November 2019**

**SHRI RAMDEOBABA COLLEGE OF ENGINEERING & MANAGEMENT, NAGPUR**

(An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur University Nagpur)

Department of Computer Science and Engineering

# CERTIFICATE

This is to certify that the Thesis on **"FARE OPTIMIZER"** is a bonafide work of Krunal Pande, Nachiket Satpute, Kunal Khadkeshwar, Navin Rathi, Pranshu Singh submitted to the Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur. It has been carried out at the Department of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur during the academic year 2019-20.

Date:

Place:Nagpur

Prof. Vrushali Bongirwar
Project guide
Department of Computer Science and Engineering

Dr. Manoj Chandak
H.O.D
Department of Computer Science and Engineering

.

# DECLARATION

I, hereby declare that the thesis titled **"FARE OPTIMIZER"** submitted herein, has been carried out in the Department of Computer Science Of Engineering of Shri Ramdeobaba College of Engineering & Management, Nagpur. The work is original and has not been submitted earlier as a whole or part for the award of any degree at this or any other institution.

Date :

Place : Nagpur

Krunal Pande

Nachiket Satpute

Kunal Khadkeshwar

Navin Rathi

Pranshu Singh

# APPROVAL SHEET

This thesis/dissertation/report entitled **FARE OPTIMIZER** by Krunal Pande, Nachiket Satpute, Kunal Khadkeshwar, Navin Rathi, Pranshu Singh is approved for the degree of Computer Science and Engineering.

Name & signature of Supervisor(s)

---

---

Name & signature of External.Examiner(s)

---

---

Name & signature of HOD

---

---

Date:
Place:

# ACKNOWLEDGMENT

We would like to take the opportunity to acknowledge our profound indebtedness and extend our deep sense of gratitude to our respected guide Prof. Vrushali Bongirwar for her valuable guidance, profound advice and encouragement that has led to the successful completion of this project .

We are grateful to our respected Head of the Department Dr. Manoj Chandak for his full cooperation and help. We also thank the Department's faculty members.

We would like to thank to all the people who have directly or indirectly help us in the completion of the project.

Finally, we would like to express our deepest gratitude to our parents for encouraging through the progress in the work.

.

# ABSTRACT

With the ever-expanding cities and urbanization, the big issue that is prevalent today is the wait involved with booking taxis. It is becoming increasingly difficult for Taxi drivers and commuters alike to find cabs in the vicinity. Taxi drivers are hit with a financial crisis and the supply demand chain isn't met. Users are unable to find cabs leading to long waiting times and drives higher fares.

The application we developed addresses this problem by analyzing the previous taxi Booking data to predict areas with dense bookings, within the user specified range to reduce user efforts and aim to maximize the driver profit by also predicting the expected cab charges. In this report, we provide a detailed description of building the application from the previous years' dataset, predicting the booking dense areas, and connecting the analytical and regression models to the application.

**TABLE OF CONTENTS**

## LIST OF FIGURES

**Figures**

# INTRODUCTION

# CHAPTER 1

# INTRODUCTION

**Introduction:**

India's taxi market stood at around $ 6.4 billion in 2016, and is forecast to grow at a CAGR of 13.7% during 2017 – 2022, to reach $ 14.3 billion. Surging demand for taxi services in India can be attributed to changing lifestyles of travelers and increasing disposable income of consumers, especially in Tier-I and Tier-II cities.

With the massive increase in the number of drivers and with the companies increasingly focusing on profitability, and the expanding cities, drivers' finances have worsened. Uber and Ola drivers are only paid once they pick up a passenger. Every minute they spend waiting for a pickup or even driving to meet a rider they are simply losing money. This roaming situation not only wastes energy but also increases pollution and unnecessary traffic. One of the reasons cab drivers are idle for the day is lack of passengers in a particular area, new Cab Drivers to a city are unaware of the traffic hotspots and popular locations of the city.

The problem for the crisis hasn't been addressed yet in the interest of the Taxi Drivers. The application aims to minimize the supply-demand gap that exists in the system to improve drivers' finances and boost overall productivity. The project aims to address this situation by predicting cab booking hotspots throughout the area using previous dataset and also predict the estimated fare the driver should expect. Our approach involves analyzing the previous cab booking dataset to understand the model for cab booking. We performed Exploratory Data analysis and visualization techniques on the dataset. Clustering methods were then applied on the dataset to predict hotspots. The pickup data of all clusters was analyzed using regression algorithms to predict fare data for the cluster pickup points. Cluster hotspots are then color coded according to fare, so the cab drivers can choose the cluster they should navigate to.

The remainder of this paper is organized as follows. Chapter 1.1 describes the related work. The problem formulation is presented in 1.3. Following the definition, 1.4 details our approach. The next chapter 2, details the technology stack we used. Next, Chapter 3 describe the implementation of the clustering used and the implementation of regression and the experiment results. Conclusion, summary and future scope are stated in Chapter 4.

# LITERATURE REVIEW

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Context-aware taxi demand hotspots prediction in Taiwan

In the project developed by Han-wen Chang, Yu-chin Tai and Jane Yung-jen Hsu **[3]** which was made for the people of Taiwan (officially the Republic of China, State in East Asia) where the problem was very critical. In urban areas the demand of Taxis was not always matched up with the supply. This paper proposes mining historical data to predict demand distributions with respect to contexts of time, weather, and taxi location.

The four-step process consists of data filtering, clustering, semantic annotation, and hotness calculation. The results of three clustering algorithms are compared and demonstrated in a web mash-up application to show that context-aware demand prediction can help improve the management of taxi fleets. Keywords used in their project are hotspot mining, data mining, clustering.

According to the Institute of Transportation (IOT) Survey of Taxi Operation Conditions in Taiwan Area 2006, a particular driver operated taxi business for 9.9 h a day, driving approximately 147.3 km. However, about one-third of the time drivers were on the roads driving without passengers. The time and energy wastage is more severe in Taipei urban area. This roaming situation not only wastes energy but pollutes the environment.

One of the reasons for the drivers driving without passengers is that they do not know where potential customers are there. It leaves them with no choice but to move around the city.
They used past history to predict the areas with potential demand.They applied Clustering methods on primitive data to find locations with high densities.

They mapped these clusters to known road segments which helped in understanding the semantic meanings of the geometries. Once the clusters are identified, the hotness scores can be calculated. Combining the cluster geometries, the semantic road names, and the hotness scores, hotspots are defined. Accordingly, cab drivers can adjust their strategies according to the demand distribution prediction

Consider the case when a taxi driver is taking a passenger to the destination. When the taxi driver approaches the location and drops off the passenger, the system detects the need of the driver to know the potential taxi demand. As a result, the prediction service begins and the results will be displayed to the driver for reference.

Clustering the GPS coordinates into locations is an important step before doing any further analysis. On spatial dimension, millimeter-level scale is too detailed to make comprehensive conclusions for real applications. In the scenario of analyzing taxi demand, city-block-level or road-level scale with semantic meaning is much easier to describe the distribution of request records.

Passengers coming from a business building, which may be a hotspot for taxi, may actually get on the taxis at slightly different GPS coordinates on the roads around the building. These nearby GPS coordinates should be viewed as one location instead of several independent locations.Clustering is a process for grouping the similar items together.

# TECHNOLOGY

# CHAPTER 3

# TECHNOLOGY

## 3.1 Technology Stack

### 3.1.1 Pandas

In computer programming, pandas **[9]** is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

Pandas is well suited for many different kinds of data:

- Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet
- Ordered and unordered (not necessarily fixed-frequency) time series data.
- Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels
- Any other form of observational / statistical data sets. The data actually need not be labelled at all to be placed into a pandas data structure


Figure 3.1 :Pandas

We used the Pandas API for Data Cleaning and Data Analysis, The dataset obtained from Kaggle had several outliers as mentioned earlier and were removed.

Code Snippet from Project:

```
#Drop Coordinates which are outliers to NYC
data[(data.pickup_latitude<lat_min) | (data.pickup_latitude>lat_max) ]

data.drop(data[(data.pickup_latitude<lat_min) | (data.pickup_latitude>lat_max)].index,axis=0,inplace=True)
```

```
data[(data.dropoff_latitude<lat_min) | (data.dropoff_latitude>lat_max)]


data.drop(data[(data.dropoff_latitude<lat_min) |
(data.dropoff_latitude>lat_max)].index,axis=0,inplace=True)
#Drop Null Fields
data.isnull().sum()
data[data.dropoff_longitude.isnull()==True].head(1)
data.drop(data[data.dropoff_longitude.isnull()==True].index,axis=0,inplace=True)
```

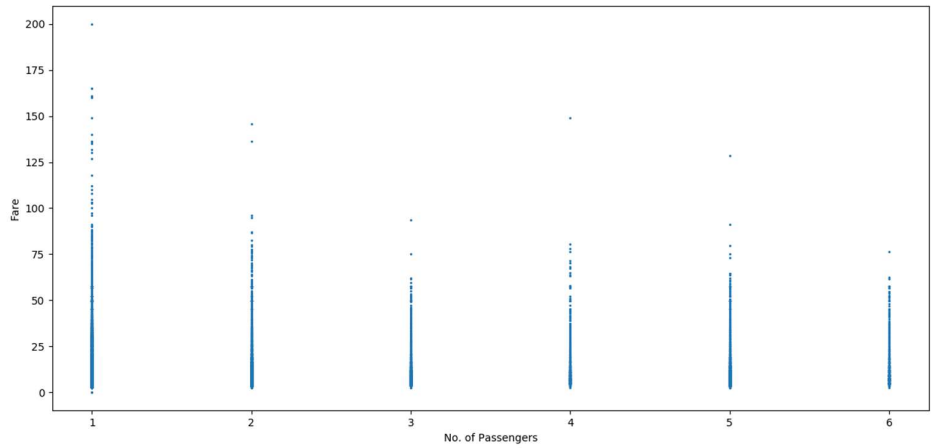


Figure 3.2 Exploratory data analysis for no of passengers vs fare

As implied from the figure, the passenger count of the dataset does not affect the fare, therefore it is removed from the dataset.

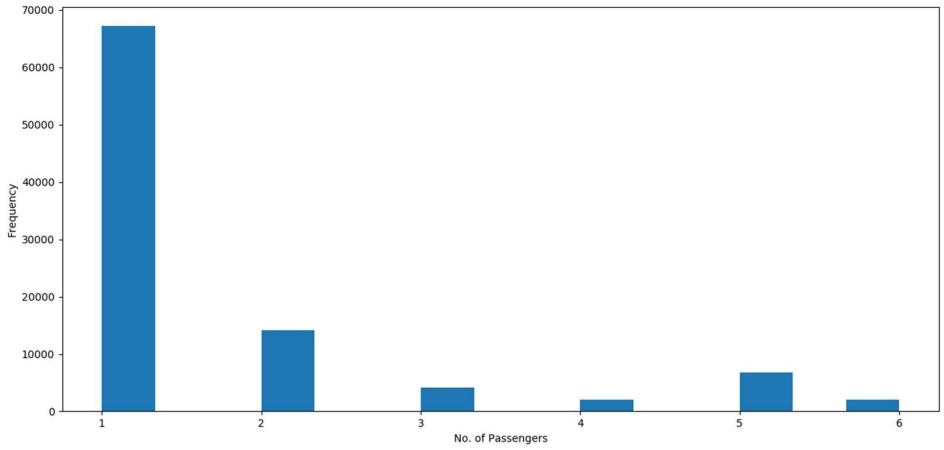


Figure 3.3 Exploratory data analysis for no of passengers vs frequency

The Following figure represents the frequency of number of passengers in cabs. Does not affect analysis.

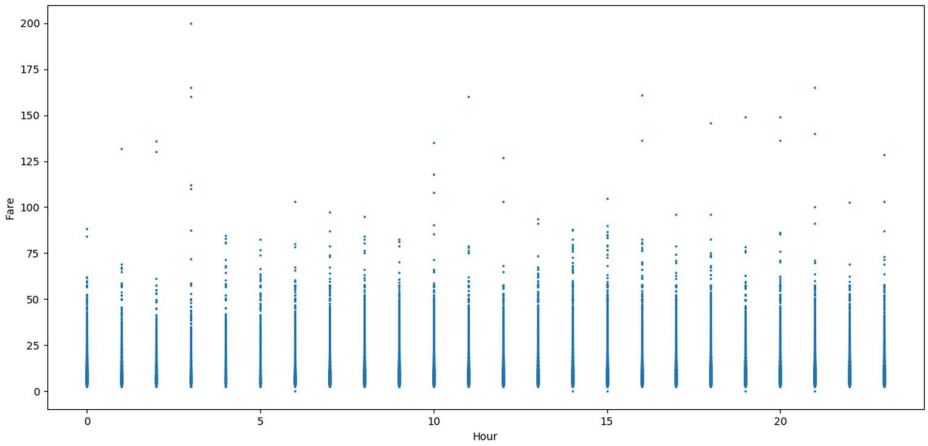


Figure 3.4 Exploratory data analysis for no of hour vs fare

The above figure shows the relation between the Hour and Fare. as implied, the hour affects the fare. Fares are more during peak hours, and then reduced consistently.

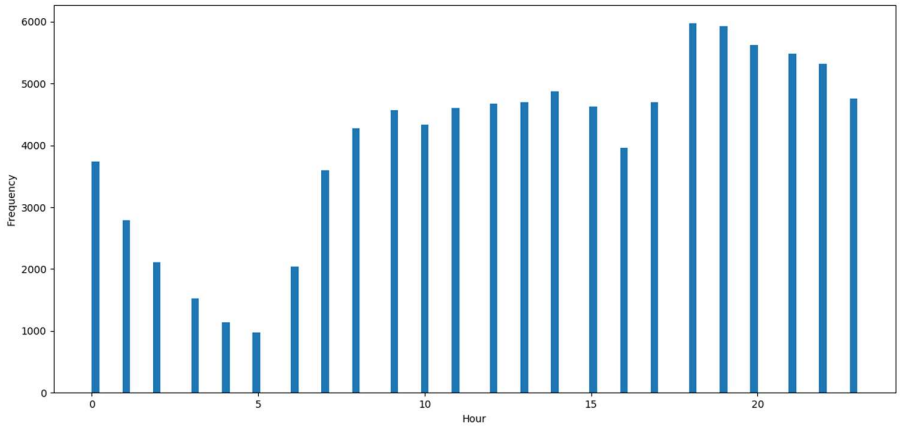


Figure 3.5 Exploratory data analysis for hour vs frequency

Most of the cabs are booked during late hours, preferably after office hours.

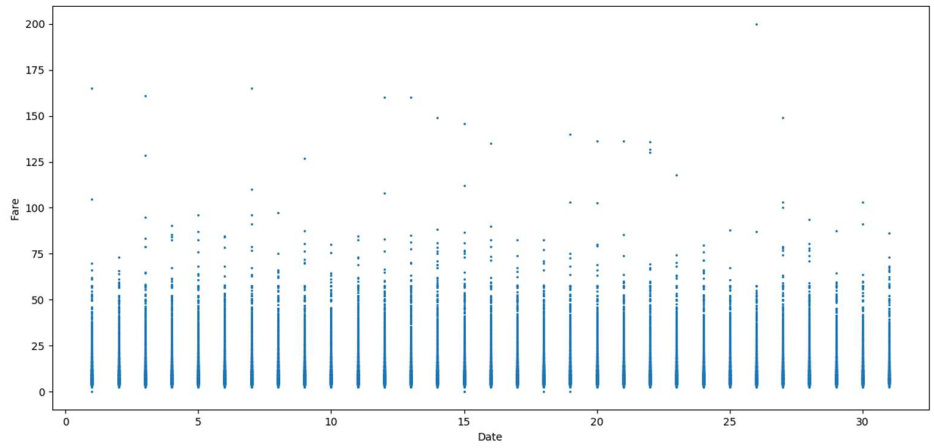


Figure 3.6 Exploratory data analysis for date vs fare

Fares are almost the same during all days, and are a bit higher on Fridays owing to higher office hours and more rush.

### 3.1.2 Firebase

Firebase **[5]** is Google's mobile application development platform that helps one build, improve and grow one's app. Firebase has various features like Hosting, Storage, Cloud Messaging, Realtime Database etc. In our project we will be using Realtime Database to store data.

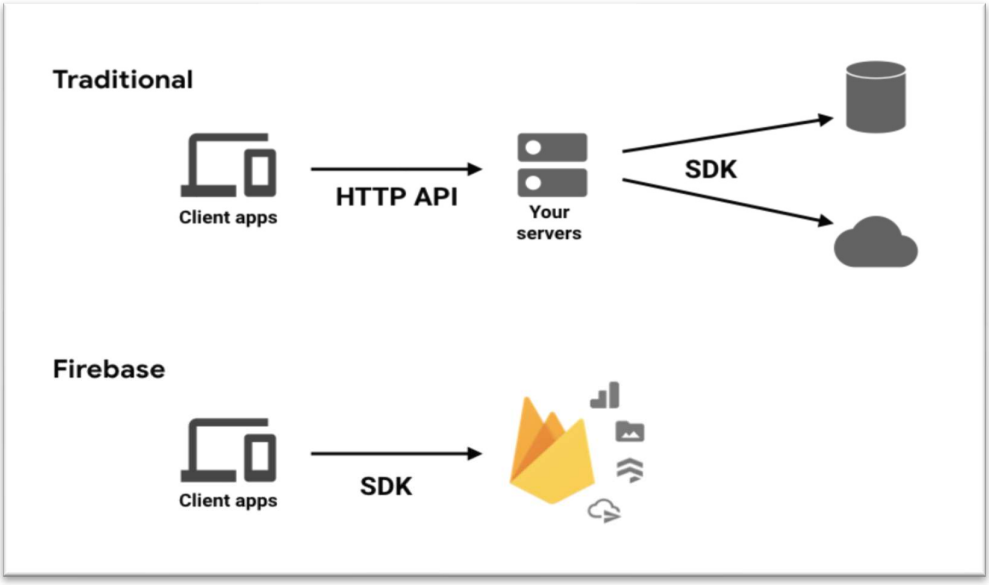


Figure 3.7 Firebase vs Traditional Storage

### 3.1.3 RealTime Database

Firebase Realtime Database is a cloud hosted database, it runs on a cloud and access to

The user. It stores the data in JSON Format for connections.

Properties:

- Real Time
- No need of Application Server
- Support by various language and platforms
- Cross-Platform

### 3.1.4 Creating Project

It is very easy to create a project in firebase. Login to firebase and add a project to firebase. Android app can be linked with firebase from Android Studio or firebase also.After the app is linked the instance of firebase can be obtained in android and used many times.

The below line of code gives the reference of database in android.

```
DatabaseReferencemref=FirebaseDatabase.getInstance().getReference();
```

To link a python script with we need to get crediantials of firebase projest. We need to add a web model in firebase project.After adding we get all the crediantials we need for linking the script. Below is the configuration used to link firebase and script.

```
firebaseConfig = {
apiKey: YOUR_API_KEY,
authDomain: "myproject.firebaseapp.com",
databaseURL: "https://myproject.firebaseio.com",
projectId: "myproject",
storageBucket: "myproject.appspot.com",
messagingSenderId: "1234556789",
appId: APP_ID
};
```

The below line of code gives the reference of database in Python.

```
firebase=pb.initialize_app(firebaseConfig)
x=firebase.database().child("College")
```

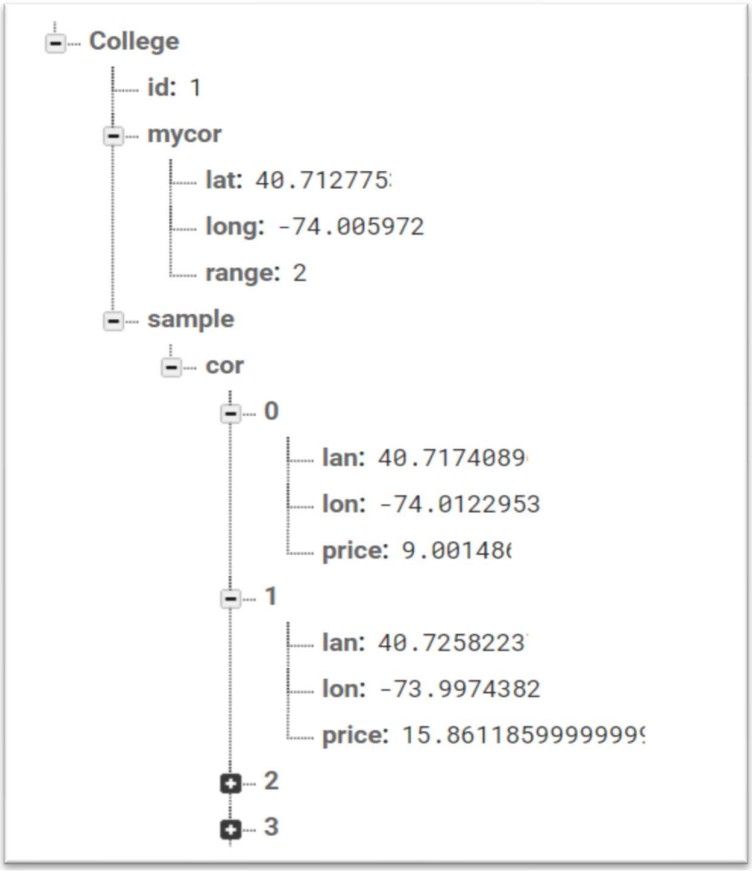


Figure 3.8 Snapshot of Firebase Database

### 3.1.5 Python

**Python** is a general-purpose programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use **Python** for developing complex scientific and numeric applications. **Python** is designed with features to facilitate data analysis and visualization.

Python is very concise and the code is easily readable and understood. It's simplicity helps developers to code reliable program. Python is open source language and is supported by large community. Python's extensive selection of machine learning libraries and frameworks simplify the development process and cut development time.

In our project pythons was use to clean data. As data contains lot of impurities is becomes necessary to clean and the use for better utilization of data. Also Python was used for machine learning algorithms like k-means clustering and Random Forest Algorithm. Jupyter Notebook was use for implementing python code.

Pyrebase Library **[6]** is used in python to connect with firebase.

*import pyrebase as pb*

*firebase=pb.initialize_app(config)*

The above code snippet shows how to import pyrebase and initialize firebase in python. Here config is a variable which consists of all credentials of firebase.

### 3.1.5.1 Python Script

In our project python script is used to perform as a server. The script will be continuously be running and listening to the request for hotspots from the taxi drivers. As soon as there is a request it will find related hotspots and return the information to respective taxi driver. Spyder IDE was use for running the server. The server can be stopped by the admin by changing the value of id to 99.

## 3.1.6 K-Means Clustering

K-means **[10]** is a centroid-based algorithm, or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, each cluster is associated with a centroid. The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms.

Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes.

Andrey Bu, who has more than 5 years of machine learning experience and currently teaches people his skills, says that "the objective of K-means is simple: group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset."

The approach K-means follows to solve the problem is called Expectation-Maximization. The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster. Below is a breakdown of how we can solve it mathematically (feel free to skip it).

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \|x^i - \mu_k\|^2 \qquad (1)$$

Figure 3.9 Formula for K-Means Clustering

We used K-Means to divide the cab booking data into clusters. The clusters were then plotted on the Android Map. The number of Clusters was set at 150, since it covers most of the hotspots

in New York City without overwhelming the map or the user.

Code Snippet:

*plt.scatter(X[:,0],X[:,1], label='True Position')*

*kmeans = KMeans(n_clusters=150)*

*kmeans.fit(X)*

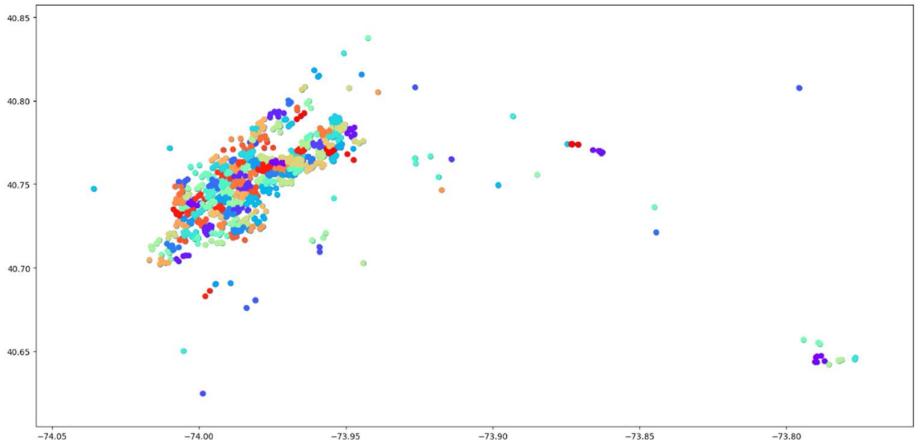


Figure 3.10 150 Clusters across NYC

The above figure represents the clusters that were obtained from the dataset. 150 clusters were formed across the city.

### 3.1.7 Random Forest Regression

Random forest is a Supervised Learning algorithm which uses ensemble learning method for classification and regression. Random forest is a bagging technique and not a boosting technique. The trees in random forests are run in parallel. There is no interaction between these trees while building the trees.It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.The number of features that can be split on at each node is limited to some percentage of the total (which is known as the hyper parameter). This ensures that the ensemble model does not rely too heavily on any individual feature, and makes fair use of all potentially predictive features.

Each tree draws a random sample from the original data set when generating its splits, adding a further element of randomness that prevents over fitting.

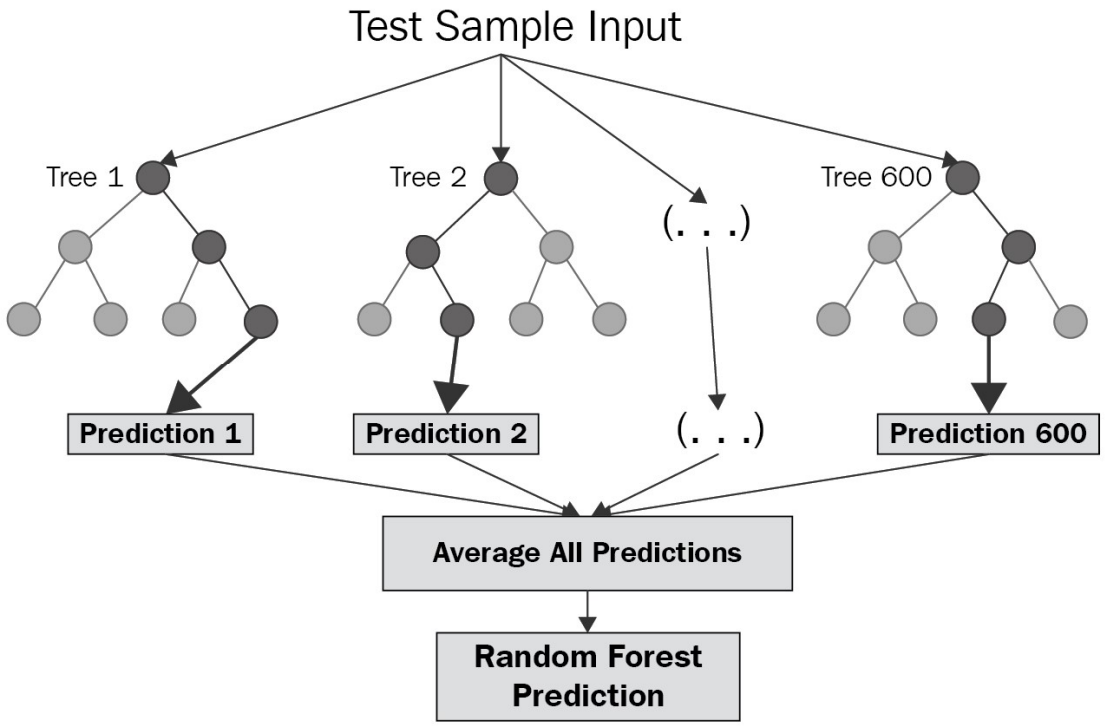


Figure 3.11 Representation of random Forest Regression

**Approach:**
Pick at random K data points from the training set.
Build the decision tree associated with those K data points.
Choose the number Ntree of trees you want to build and repeat step 1 & 2.
For a new data point, make each one of your Ntree trees predict the value of Y for the data point, and assign the new data point the average across all of the predicted Y values.

### 3.1.7.1 Why Random Forest Regression?

Random forest regression is a accurate learning algorithm available. It gives what variables important in classification.It can handle thousands of input variables without variable deletion.It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
Linear regression is a linear model, which means it works really nicely when the data has a linear shape. But, when the data has a non-linear shape, then a linear model cannot capture the non-linear features. So in this case, we can use the decision trees, which do a better job at capturing the non-linearity in the data by dividing the space into smaller sub-spaces depending on the questions asked.

### 3.1.8 Android Studio

Android has became a day to day necessity in today's world. Large Population use mobile so this project is suitable on portable device i.e. Mobile.

Android Studio is official IDE (Integrated Development Environment) for Android application development. It is based on Java. It is very is use and make application.

Android Project is created and Google maps API is use to show hotspot.For google mas to be used we first have to generate a API key which enables us to use and manipulate google maps on out application.

Various other elements of android are used in the app such as to a progress app to show the loading Apps homepage. When a request is generate little delay is observer so a progress dialog box is used.

### 3.1.9 The Dataset

The data used in this study are all subsets of New York City Taxi and Limousine Commission's trip data, which contains observations on around 1 billion taxi rides in New York City between 2009 and 2016. The total data is split between yellow taxis, which operate mostly in Manhattan, and green taxis, which operate mostly in the outer areas of the city. For the main analyses of this study, the data for yellow taxi rides during the month of May 2016 were used, although the models were validated on additional data. Since each month consists of about 12 million observations, and there were computational limitations, subsets of the monthly data were used for model building, and other subsets were used for validation.

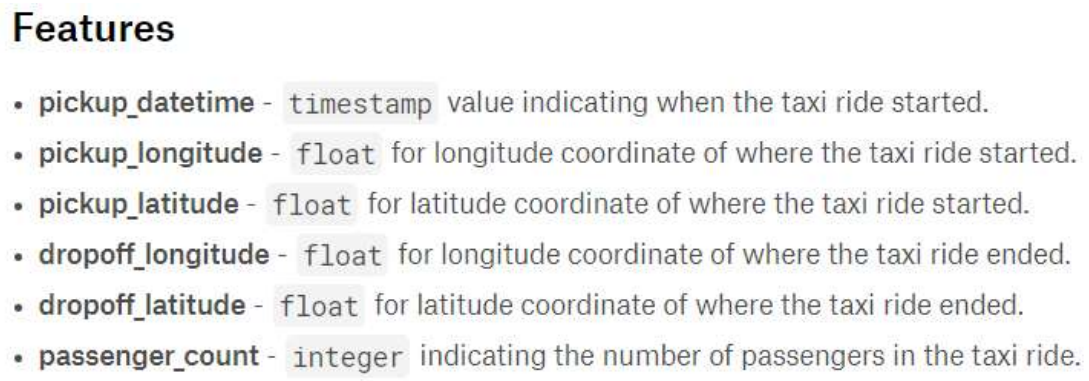


Figure 3.12 Features of Data Set

A single row consists of the passenger pickup time and pickup coordinates.

The dataset is collected for New York City for a duration of three years and contains outliers as well as reversed coordinates and blank fields. Dataset Source Kaggle.

## 3.2 Methodology

Cab Drivers new to a city have difficulty in finding new customers, since they are unaware of the popular areas where they could get potential customers. The same problem is also faced by the consumers, who sometimes are unable to get cabs in their area.

The factors to consider for the problem are pickup latitude, pickup longitude, passenger count and the drop-off coordinates.

### 3.2.1 Proposed Solution

Consider the case when a cab driver comes to a halt after a ride and is unable to find more customers. In this case, the app would record User's current location and predict clusters up to the diameter specified by the user. Same goes for a passenger searching for cabs.The clusters hotspots now predicted will be color coded according to mean expected fare.

According to the location, records from dataset are obtained, and then grouped into clusters. The clusters are then marked on the map using Google API. Clicking on the hotspot displays the predicted price by using Random Forest Regression. The clusters are given a fixed diameter which can be changed by the user apart from the cluster finding range.

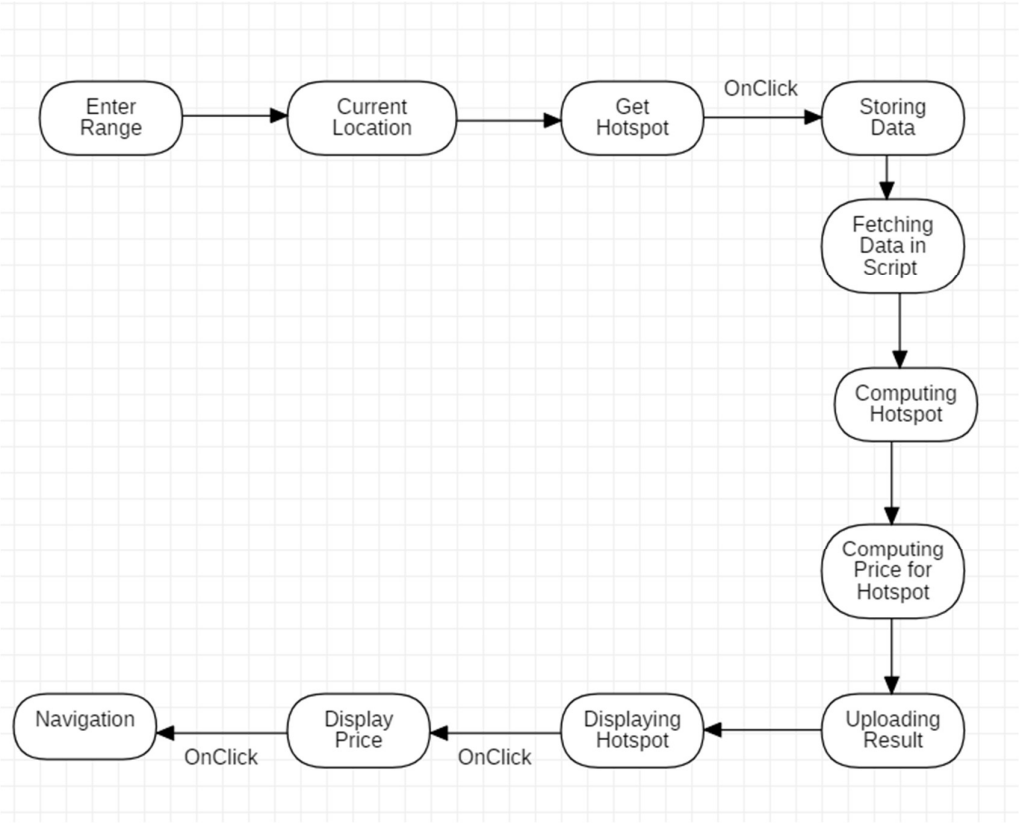


Figure 3.13 FlowChart of Project

The working of this project is based on three components Android , Firebase and Python Script.These components makes the working of these project possible. Each are linked and dependent on each other. The main link is firebase all communication is through firebase. Firebase acts as mediator between Android App and Python Script.

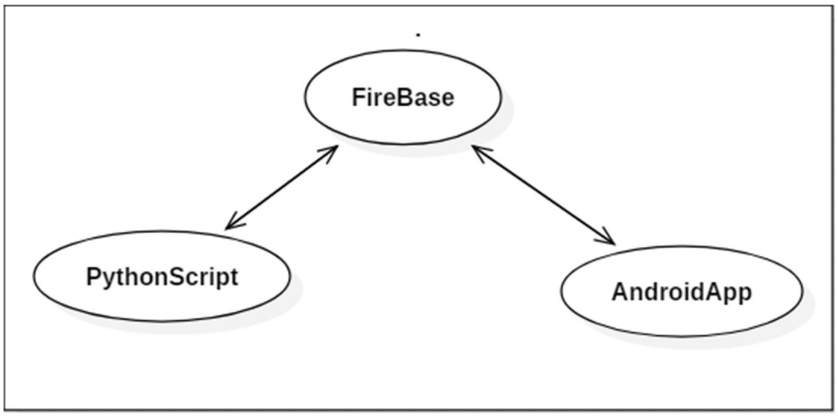


Figure 3.14 Main three Components

From figure 2 we can see that AndroidApp needs a medium to communicate with Script i.e. Firebase.The server can be started by the admin and stopped by admin.To stop the server the value of id in the database must be change to 99 by the admin.

### 3.2.2 Data Cleaning and Exploratory Data Analysis

The dataset obtained from Kaggle is unclean and contains several blank fields also. The data is the cab booking data for New York City from 2010 to 2016. The data needed to be cleaned for:

1. Outliers: Several Coordinated were outside the city, so were useless and removed.
2. Blank Rows: Several Rows were left blank or had negative values and needed to be removed.
3. Negative Fares: Fares were zero or negative.
4. Reversed Coordinates: Coordinates were reversed, the coordinates in dataset led to Antarctica, thus were removed.

All in all, about 61% data was removed from the dataset. This was a onetime process and improved the results a lot and saved huge computational power.
Data cleaning was done in python using the Pandas framework.

| key | fare_amou | pickup_da | pickup_lor | pickup_lat | dropoff_lc | dropoff_la | passenger_count |
|---|---|---|---|---|---|---|---|
| 40:00.0 | 17.3 | 2012-03-2 | 2522.271 | 2621.628 | -1718.12 | -2864.47 | 1 |
| 36:00.0 | 9.7 | 2011-11-2 | 2140.601 | 1703.093 | -1251.2 | -1189.62 | 1 |
| 34:00.0 | 7.7 | 2011-05-1 | 351.0525 | 1669.582 | 1717.003 | 1989.728 | 1 |
| 31:00.0 | 5.7 | 2012-04-1 | -73.9571 | 424.0833 | -73.9683 | 40.76512 | 1 |
| 55:00.0 | 9.3 | 2011-04-2 | -74.0025 | 405.35 | -73.9786 | 40.73996 | 1 |
| 13:00.0 | 8.5 | 2012-03-1 | -73.9943 | 405.1333 | -73.9568 | 40.78374 | 1 |
| 58:00.0 | 6.9 | 2012-03-0 | -73.9943 | 404.9667 | -73.9739 | 40.75546 | 1 |
| 08:00.0 | 9.3 | 2012-03-0 | -73.9943 | 404.9667 | -73.9864 | 40.7769 | 1 |
| 15:00.0 | 3.3 | 2011-07-3 | -73.9472 | 401.0833 | -73.9514 | 40.77893 | 1 |
| 44:00.0 | 5.3 | 2012-05-2 | -73.9892 | 91.26667 | -73.9824 | 40.76311 | 1 |
| 44:00.0 | 8.9 | 2012-06-2 | -73.9785 | 69.4 | -73.9551 | 40.77999 | 1 |
| 21:00.0 | 2.9 | 2012-07-1 | -73.9875 | 65.16667 | -73.9873 | 40.74314 | 1 |
| 18:48.0 | 4.5 | 2013-10-1 | -66.7258 | 51.08437 | -65.2388 | 51.34652 | 1 |
| 19:00.0 | 7.7 | 2012-01-2 | -90.2391 | 47.89009 | -90.2391 | 47.89009 | 1 |
| 05:40.0 | 12 | 2014-05-0 | -69.9549 | 47.2674 | -73.9974 | 40.76368 | 2 |
| 52:26.0 | 7.5 | 2013-04-0 | -75.4653 | 47.02395 | -73.9947 | 40.72843 | 1 |
| 52:44.0 | 11 | 2015-03-2 | -65.7015 | 46.20996 | -68.9743 | 45.19198 | 1 |

Figure 3.15 Table showing data before cleaning

| key | fare_amou | pickup_da | pickup_lor | pickup_lat | dropoff_lc | dropoff_la | passenger_count |
|---|---|---|---|---|---|---|---|
| 00:46.0 | 15 | 2009-01-0 | -73.9537 | 40.80676 | -73.9894 | 40.76954 | 1 |
| 01:04.0 | 5.8 | 2009-01-0 | -73.9951 | 40.73411 | -73.9982 | 40.72287 | 2 |
| 11:42.0 | 29.4 | 2009-01-0 | -73.96 | 40.77361 | -73.8215 | 40.86134 | 2 |
| 12:02.0 | 7.4 | 2009-01-0 | -73.9595 | 40.77126 | -73.9675 | 40.78774 | 1 |
| 20:31.0 | 7.4 | 2009-01-0 | -73.9885 | 40.74667 | -73.9898 | 40.72651 | 1 |
| 25:57.0 | 5 | 2009-01-0 | -73.9852 | 40.77896 | -73.9797 | 40.77372 | 3 |
| 31:32.0 | 11.8 | 2009-01-0 | -73.9551 | 40.76899 | -73.9732 | 40.74798 | 1 |
| 35:03.0 | 16.5 | 2009-01-0 | -73.9919 | 40.73841 | -74.0035 | 40.68333 | 1 |
| 40:21.0 | 9.4 | 2009-01-0 | -73.9907 | 40.73074 | -73.9921 | 40.75003 | 1 |
| 41:00.0 | 6.2 | 2009-01-0 | -73.9942 | 40.75101 | -73.9858 | 40.75058 | 2 |
| 46:56.0 | 11.8 | 2009-01-0 | -74.0009 | 40.72934 | -74.0039 | 40.743 | 1 |
| 47:26.0 | 5.4 | 2009-01-0 | -73.9873 | 40.74957 | -73.976 | 40.74195 | 1 |
| 53:38.0 | 11 | 2009-01-0 | -73.9646 | 40.76733 | -73.996 | 40.7436 | 2 |
| 58:31.0 | 5.4 | 2009-01-0 | -73.9837 | 40.74383 | -73.9734 | 40.75468 | 1 |
| 11:28.0 | 13 | 2009-01-0 | -74.0056 | 40.74111 | -73.9725 | 40.75559 | 2 |
| 11:28.0 | 8.2 | 2009-01-0 | -73.9758 | 40.78152 | -73.9519 | 40.77325 | 2 |
| 12:36.0 | 11.3 | 2009-01-0 | -73.9165 | 40.80825 | -73.8867 | 40.84237 | 1 |
| 16:02.0 | 6.1 | 2009-01-0 | -73.9589 | 40.76912 | -73.9599 | 40.77056 | 1 |
| 19:32.0 | 9 | 2009-01-0 | -73.9735 | 40.7603 | -73.9463 | 40.78228 | 2 |
| 21:54.0 | 4.9 | 2009-01-0 | -73.9518 | 40.7149 | -73.9622 | 40.70935 | 4 |

Figure 3.16 Dataset after Cleaning

### 3.2.3 Clustering Coordinates

Clustering the coordinates before analyzing the data is an important approach. But firstly, the distance between two points on the map given by their respective coordinates is calculated by the formula given below.

$$\mathrm{dist}(a,b) = 6372.795 \times \Delta\widehat{\sigma}(a,b)$$

$$\Delta\hat{\sigma}(a,b) = \arctan\left(\frac{\sqrt{(\cos\phi_b \sin\Delta\lambda)^2 + (\cos\phi_a \sin\phi_b - \sin\phi_a \cos\phi_b \cos\Delta\lambda)^2}}{\sin\phi_a \sin\phi_b + \cos\phi_a \cos\phi_b \cos\Delta\lambda}\right).$$

Figure 3.17 Formula for Euclidian Distance

Clustering is the way to group similarly plotted items together in a common cluster, which needs to be calculated by finding the distance between the points and then grouping them.

This distance is calculated by the Euclidean Formula**[8].** The latitude and longitude of the pickup points are already present in the dataset and are used in the formula. The radius of the Earth is 6372.795 and is multiplied with the radius angular distance Δ σ which is given in the Figure 1Now, this data needs to be divided into clusters based on their distances. There are several algorithms out there to train data, each one better than the previous one. However, we went with K-Means Clustering for training our dataset, described in the next section.

### 3.2.4 Sending Request

The app is easy to use. Enter the basic details in the page such as name and range of the area to predict hotspots. The an activity will open which consists of a map, where one can enter his/her current location and find hotspots.On Clicking the 'Get Hotspot' button a request is generated via firebase. Also, a marker is added to the Drivers current Location which will be shown on the map. Taxi Drivers current location will converted into latitude and longitude by the help of Geocoder function. The location points so generated will be stored in the firebase under child 'mycors' and simultaneously the value of id in firebase will be overwritten by 1. Once it in overwritten in means that a request has been generated.

### 3.2.5 Processing Request

As soon as the id value in firebase is changed to 1 the role of server comes to play. Then the python script fetches the location coordinates along with the range from firebase to process. The data is been processed all the hotspot in the specified range of Taxi Driver are calculated. Then the result is stored in the firebase.Also, the estimated price in the particular hotspot is calculated. After that all information is calculated in stored in firebase. And then the value of id variable is changed to 0 so another request can be handled.

### 3.2.6 Displaying Hotspot

Google Maps API is very useful to display the hotspots. When computed hotspots are added to the firebase, the android app takes into notice and fetches all the coordinated and prices related to each hotspot.These values are first stored in an array. Then this valueis use to display hotspot on maps. To display hotspot, google maps in built function 'addCircles' is used. Color of each hotspot may vary accordingly to the estimated price, for example if the estimated price is below 10 then the color of hotspot is red and if it is greater the 10 the color in blue.

```
mMap.addCircle(new CircleOptions()
        .center(xy[i])
        .radius(300)
        .strokeColor(Color.GREEN)
        .fillColor(Color.argb(170,34,139,34))
        .clickable(true));
```

The above code adds a circle on the map on the location with center stored in the array xy and color green.

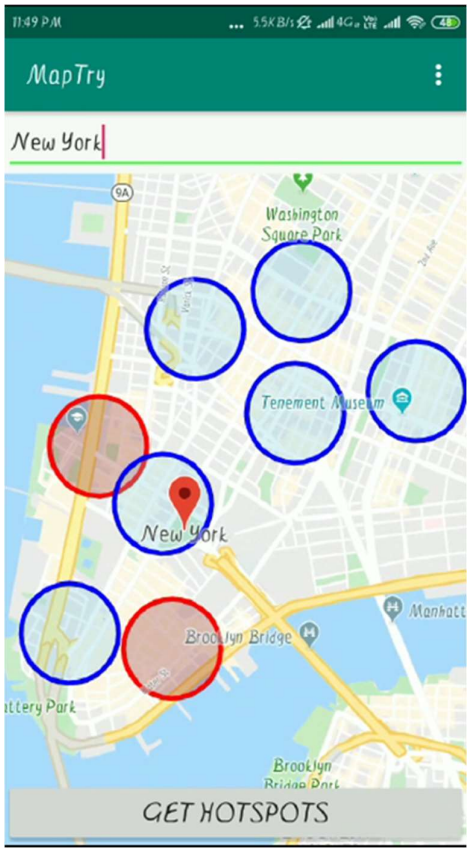


Figure 3.18 Screenshot of actual app interface displaying hotspots

Each hotspot is having a click listener in it. The event is add by using clickable method in map object. If one click a hotspot a dialog box will pop up with estimated price and if driver is interested in the hotspot he can click 'yes' button to see the route to the particular hotspot in google map. Else he can select another hotspot ,and repeat until he get a satisfying hotspot

### 3.2.7 Predicting Prices

The clusters marked on the map will also present the user with predicted fares. The fares will be predicted using Random Forest Regression, described more in the next section. The fares will be calculated assuming the pickup coordinates as the coordinates of the cluster. The cluster will be allotted a random drop-off point for calculation purposes. The predicted fare will then be calculated and displayed in an Android Dialog to the user.

1. Much of the data preprocessing was performed in conjunction with the data exploration and but that was only done on a small subset (the first 2 million rows) of the total dataset due to memory constraints (the full 55M row CSV is over 5.5GB, but it'd be much larger as a Pandas DataFrame held in memory).

2. To ease the memory burden a numeric downcast function was created to reduce the variable sizes of the values stored in the DataFrame, which was rather successful resulting in about a 40% size reduction. Unfortunately, it wasn't feasible to load the entire train dataset into memory toprocess all at once and the preprocessing needed to be systematized into something that is nearly ready for deploying into a production model.

3. We dropped some features "Pickup_datetime" and "key" which were not required in the data set

*X_test.drop(["key","pickup_datetime"],axis=1,inplace=True)*

4. XGBoost is an advanced gradient boosting tree library. It is integrated DSS visual machine learning, meaning that you can train XGBoost models without writing any code.
*n_estimators : int -Number of boosted trees to fit.*
*max_depth : int-Maximum tree depth for base learners.*

```
import xgboost
Regression_xgbooost =xgboost.XGBRegressor(n_estimators=50,max_depth=15,max_leaves=9,random_state=0)
Regression_xgbooost.fit(X,y)
y_test_pred=Regression_xgbooost.predict(X_test)
```

Figure 3.19 Code snippet 1 for Random forest

5.We split this into two different datasets, one for the independent features — x, and one for the dependent variable — y (which is the last column). We'll now split the dataset x into two

separate sets — xTrain and xTest. Similarly, we'll split the dataset y into two sets as well — yTrain and yTest. Doing this using the sklearn library is very simple. Then We get results on test data that we need to predict.

```
from sklearn.model_selection import train_test_split
train_X, val_X, train_y, val_y =train_test_split(X,y,test_size=0.2,random_state=0)

import xgboost
Regression_xgbooost =xgboost.XGBRegressor(n_estimators=300,max_depth=15,max_leaves=9,random_state=0)
Regression_xgbooost.fit(train_X,train_y)
y_pred=Regression_xgbooost.predict(val_X)
```

Figure 3.20 Code snippet 2 for Random forest

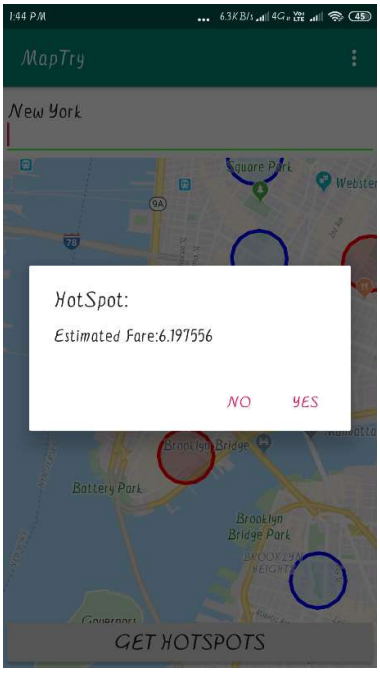


Figure 3.21:Screenshot of actual app interface displaying fares for hotspots

### 3.2.8 Navigation

The user will also have the option to navigate to the cluster. This is made possible by using the Google Maps API. More in the next section.

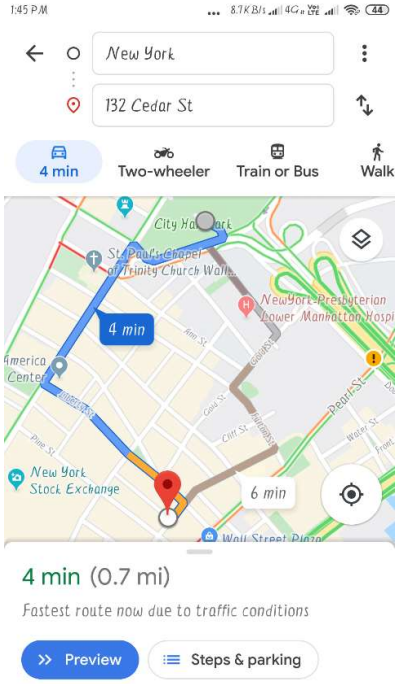


Figure 3.22 Screenshot of actual app interface displaying navigation using Google API

# RESULT AND FUTURE SCOPE

# CHAPTER 4

# RESULT AND FUTURE SCOPE

## 4.1 Performance Results

The initial dataset encompassed to 1.6 GB and had the following entries:

The dataset had about 55M rows before the Data cleaning procedure. The cleaning reduced the dataset to about 11M rows. The number of columns were 11 and the model was tested against test.csv file included with the dataset to calculate the RSME.

Different clustering methods had different performances on different kind of data distributions. The multi driver 3-year dataset cannot represent all taxi drivers for the whole New York. Moreover, the dataset only records where drivers picked up customers, and there is no information about where drivers met no demands. To assess the effectiveness of the proposed solution, deployments of the proposed solution to a taxi fleet and a long-term evaluation on the average roaming time of taxis are necessary.

## 4.2 Conclusion

The dataset we actually used to implement the model was about 10M rows due to the computational difficulties. The best model for Regression had a RSME of 0.84% which was significantly better than the Linear Regression Model.

Fuzzy C Clustering model wasn't used due to manual centroid entries. The Firebase model performed efficiently on the tested 150 tries simultaneously. The application handled multiple users in parallel efficiently.

## 4.3 Future Scope

There lies an abundant future scope to the project for improvements. For example, the regression algorithms can be changed to suit the requirements according to the city. The city can be changed and an improved and accurate dataset can be used.
The next steps for this project on its path towards final deployment, would also include cleaning up the data a bit further, optimizing the features, tuning the parameters with a grid search, reducing the model complexity, and maybe even trying a different model. Moreover, the dataset can be changed entirely for a better one, with useful features leading to a better model.

The project can be implemented by cab aggregators in India to improve driver productivity and efficiency. The individual app can be deployed to the App store for mobiles.

The clustering model can be changed to another one, including but not only fuzzy c means clustering.

# REFERENCES


1. Vanajakshi, L., S. C. Subramanian, and R. Sivanandan. "Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses." IET intelligent transport systems 3.1 (2009): 1-9.
2. Wu, Chun-Hsin, Jan-Ming Ho, and Der-Tsai Lee. "Travel-time prediction with support vector regression." IEEE transactions on intelligent transportation systems 5.4 (2004): 276-281
3. https://www.researchgate.net/publication/220579543_Context-aware_taxi_demand_hotspots_prediction
4. Van Lint, J. W. C., S. P. Hoogendoorn, and Henk J. van Zuylen. "Accurate freeway travel time prediction with state-space neural networks under missing data." Transportation Research Part C: Emerging Technologies 13.5 (2005): 347-369.
5. www.firebase.com
6. https://github.com/thisbejim/Pyrebase
7. https://medium.com/firebase-developers
8. https://www.movable-type.co.uk/scripts/latlong.html
9. https://pypi.org/project/pandas/
10. https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-.