IDM REPORT

Members:

Samia Sohail Azim 18553

Tooba Nadeem Khan 18558

Vishal 18562

# Business Understanding

Uber Technologies Inc. is a peer-to-peer ride sharing platform. Uber's platform connects the drivers who can drive to a customer's location. Uber uses machine learning for calculating pricing to find the optimal positioning of cars to maximize profits. We have used a public Uber trip dataset to answer the following business questions:

* Days when Pickup happen regularly
* Time when Uber pickup happens regularly
* Pickup distribution in the Zones
* Pickup distribution on the map, and finding out the highly clustered/hotspot areas

# Data Understanding

**Dataset:**<https://www.kaggle.com/fivethirtyeight/uber-pickups-in-new-york-city?select=uber-raw-data-aug14.csv>

<https://www.kaggle.com/fivethirtyeight/uber-pickups-in-new-york-city?select=uber-raw-data-sep14.csv>

**Data Description:**

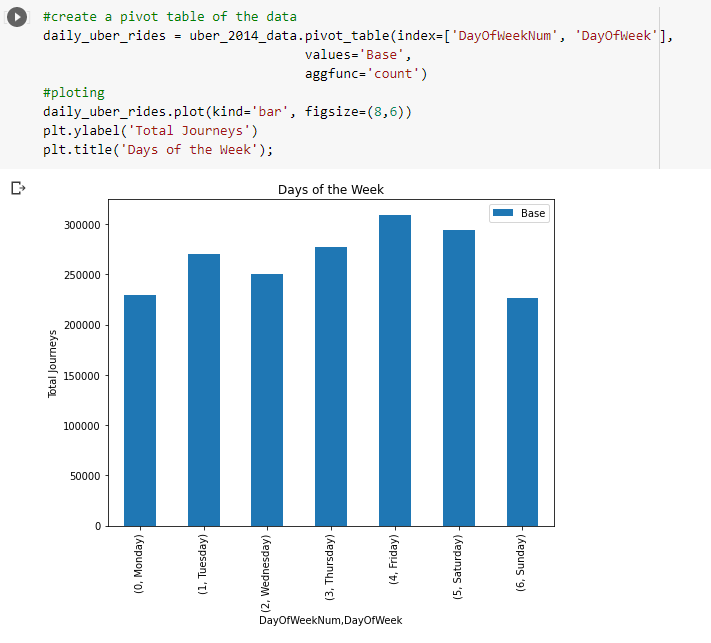
There are two csv files of data on Uber pickups in New York City from August to September 2014. The total number of samples is 1857411. And each sample contains 4 features/columns. Each file contains data for the particular month and each has the following columns:

* Date/Time : The date and time of the Uber pickup (Numerical)
* Lat : The latitude of the Uber pickup (Numerical)
* Lon : The longitude of the Uber pickup (Numerical)
* Base : The TLC base company code affiliated with the Uber pickup (Categorical)

**To carry out data exploration we used feature engineering on our dataset which is described in Section III Data Preparation.**

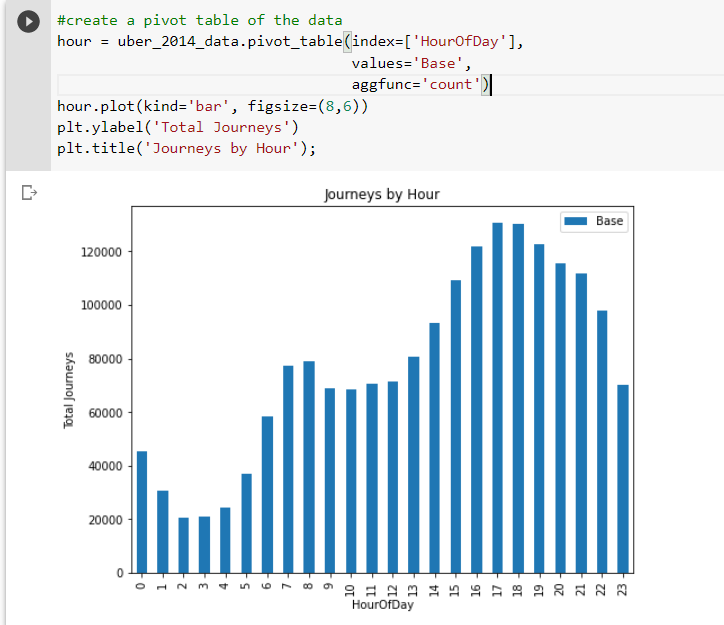
**Data Exploration:**

* Distribution of pickups by Days of The Week:



From this data we can see that the most pickups occurred on Friday followed closely by Saturday over the course of two months.

* Distribution of pickups by the hours of day:



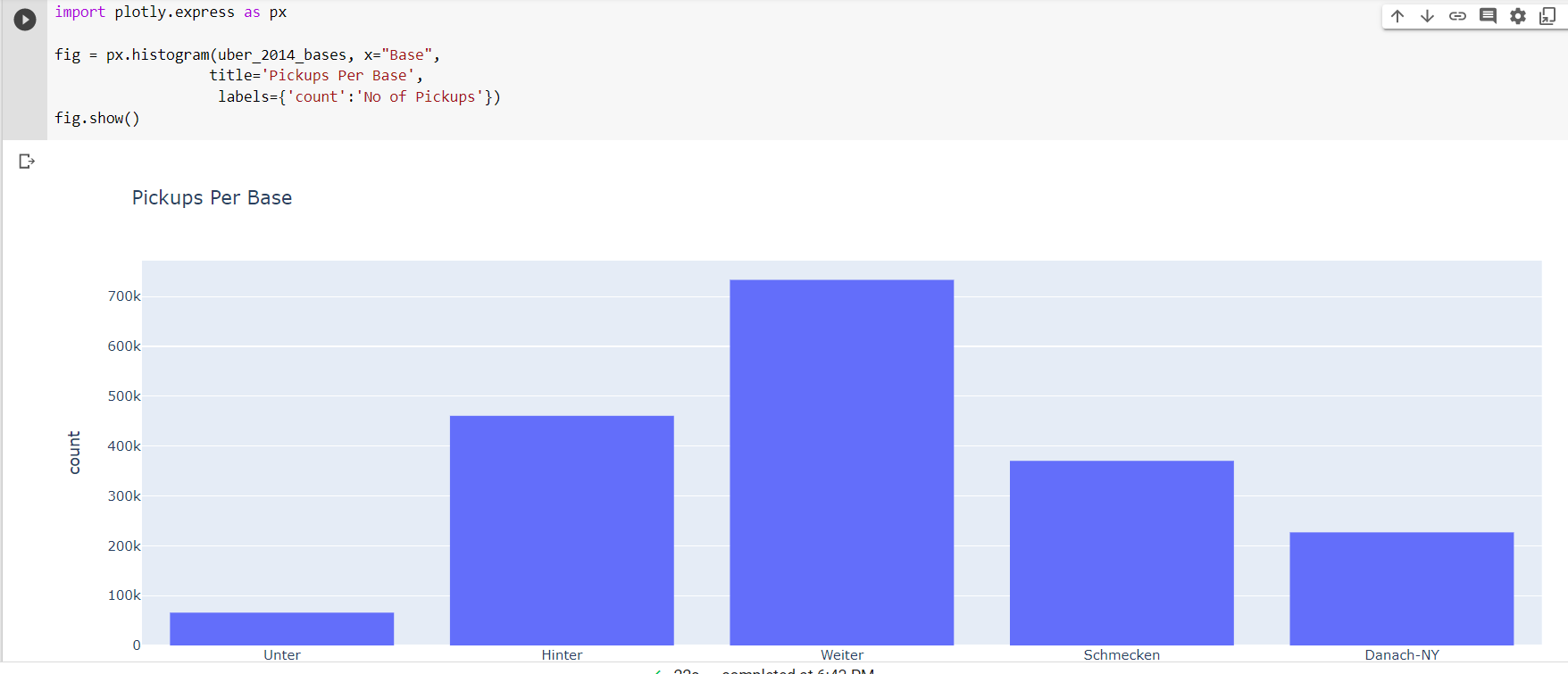
From a general overview of the data, the journeys peak between 7-8am and then peak again even more from 3pm and rise steadily until it's on its peak at around 5pm before the pickups start to slowly drop.

* Distribution of uber pickups among the Bases:

From our data, there are 5 bases and each one of them has a unique area code that is used to identify it. To make it easier for us to identify the bases, we’ll replace the base codes on the Base column with the actual names of the bases.



Then we created a plot to find out the distribution of the pickups among the bases over those months.

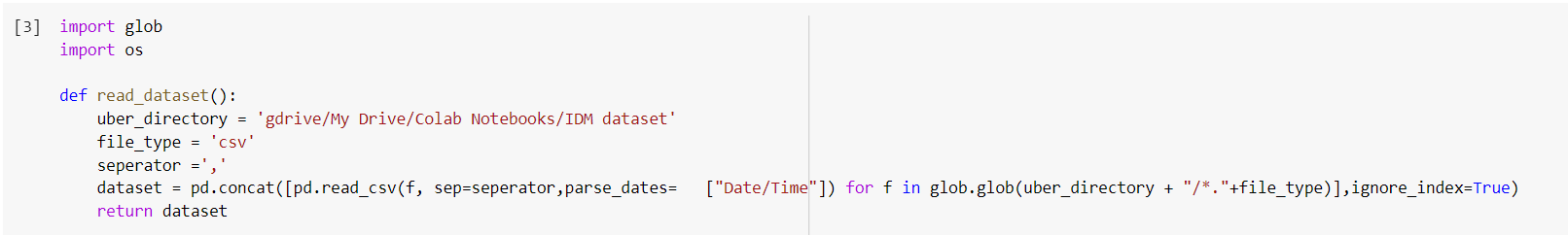


From the plot above, we can easily see that the most pickups were in Weiter, closely followed by Hinter and Schmecken. That’s probably because these areas are in highly populated areas. Unter had the least amount of pickups at less than 100k.

# Data Preparation:

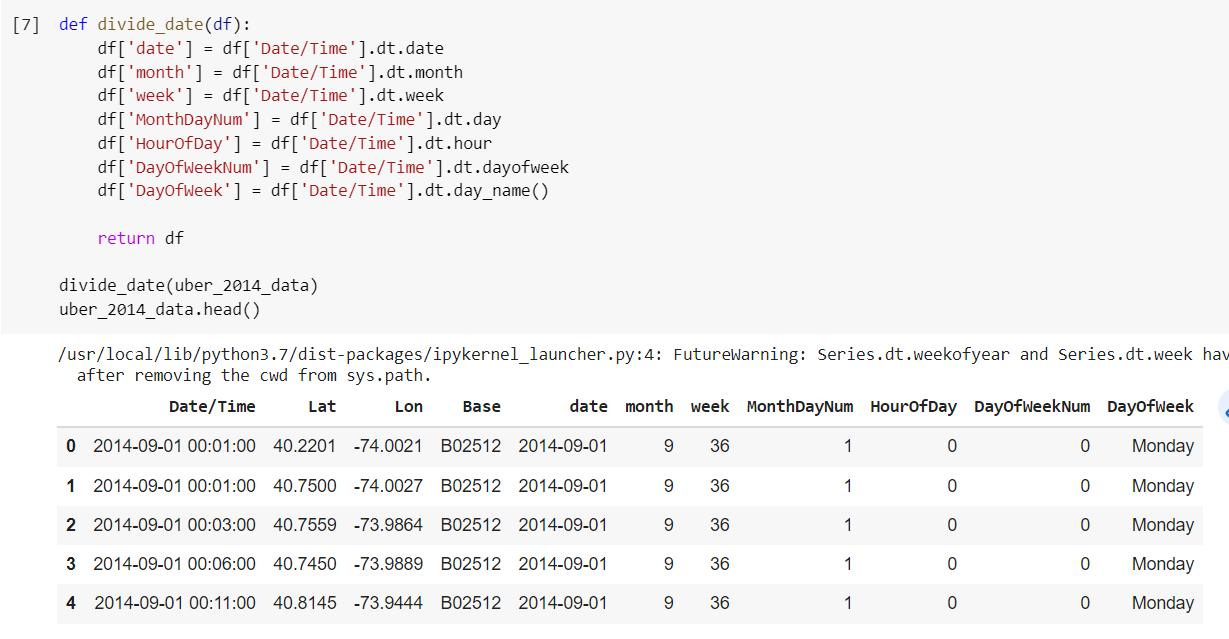
We first concatenated the two csv files into one DataFrame using a simple function that reads multiple csv files in the directory and concatenates them into one DataFrame.

The parse\_date function parses the Date/Time column as a datetime datatype.



**Feature engineering:**

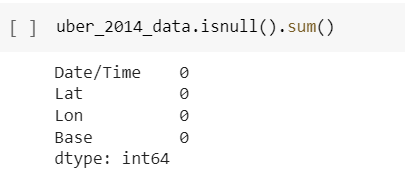
We realized we needed to add more features to answer our business questions so we broke down the datetime feature into a date, month, week, day and hour features.



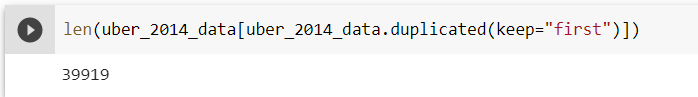
This helped us answer our first three business questions as we used them in our visualizations while we were exploring our data.

**Pre-Processing:**

We carried out basic cleaning and pre-processing techniques such as checking for null values in the data.

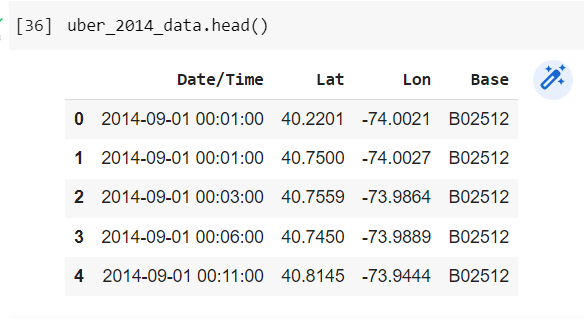


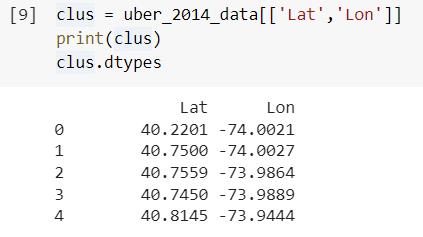
While checking for duplicates we came to the conclusion that different pickups can happen at the same time and location so removing duplicates might lead to loss in data.



**Feature Selection:**

Since our dataset only has 4 features and taking into account the fact that k means uses continuous data we picked “Lat” and “Lon” as features for our clustering model.





# Modeling + Evaluation

**Software**:

Software used is Google Collab and the language used was Python.

**Machine(s) used:**

Google Cloud Platform, Google Collab was connected to Python 3 Google Compute Engine Backend with a total of 12.69 GB RAM and 107.72 GB Disk Space.

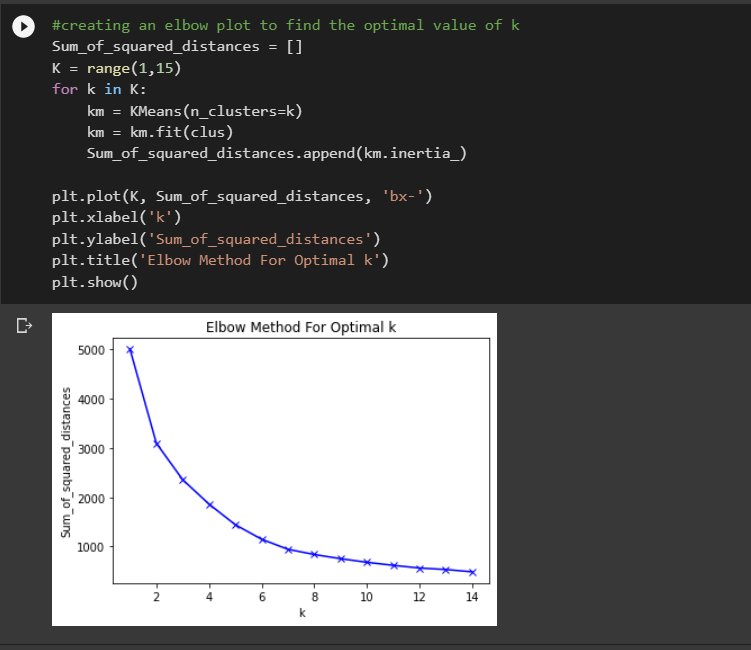
**Libraries used:**

* Pandas
* NumPy
* Matplotlib
* Sklearn
* Glob
* Folium
* Fuzzy-c-means

**Evaluation Metric Used:**

**Elbow Curve Method:**

The Elbow approach is a heuristic used in cluster analysis to determine the number of clusters in a set of data.



In this method, we used a varying number of **k** values ranging from 1 to 14. For each value of **k,** we calculated the **WCSS** (Within-Cluster Sum of Square) which is the sum of squared distance between each point and the centroid in a cluster.

At first, we set the range for k from 1 to 14. We then started a loop and calculated the sum of squared distances (**WCSS**) with each value of k till the 15th value. After obtaining the array of WCSS, we then used matplotlib library’s line-plot to plot the values of **WCSS** (y-axis) against **k** values (x-axis) and the graph turned out to be in the shape of an elbow. While analyzing the graph, the point at which the elbow shape is created is 5 or 6, that is, our **k** value or the optimal number of clusters can be either 5 or 6.

## Model(s) Implemented:

Since we are working with spatial data, to answer our business objective through prediction using modeling, we are using an unsupervised learning method which is clustering to answer our business objective.

We implemented two models:

1. Fuzzy c means clustering
2. K means clustering

### 

### **Fuzzy c means clustering:**

Fuzzy C-Means clustering is a soft clustering method in which each data point is allocated a probability score to belong to that cluster. A data point can belong to multiple clusters.

We tried fuzzy c means taking initial clusters **c** = 6.

We used the fuzzy-c-means library from PyPI to implement our c means algorithm.

In the code below, we first converted our dataset to a numpy array and then trained our model on it. We then obtained our centroids and their cluster labels. We then plotted our results on a scatterplot showing the distribution of our clusters.



### **K means clustering:**

K-means is an iterative algorithm that tries to partition the dataset into **k** pre-defined distinct non-overlapping clusters where each data point belongs to only one group. Our model includes the following:

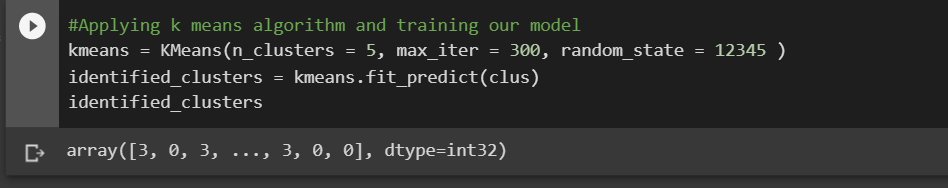
We used the elbow curve method to find the optimal value of **k** for our model which according to our curve seems to be 5 and 6. We run our model on both values of k.

We applied the k-means clustering using the sklearn library which we imported.

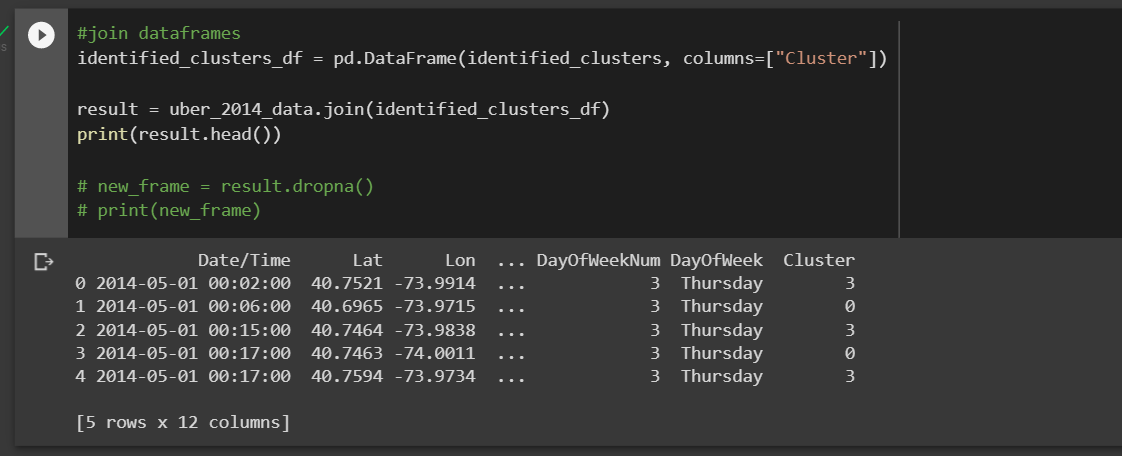
**Using k = 5:**

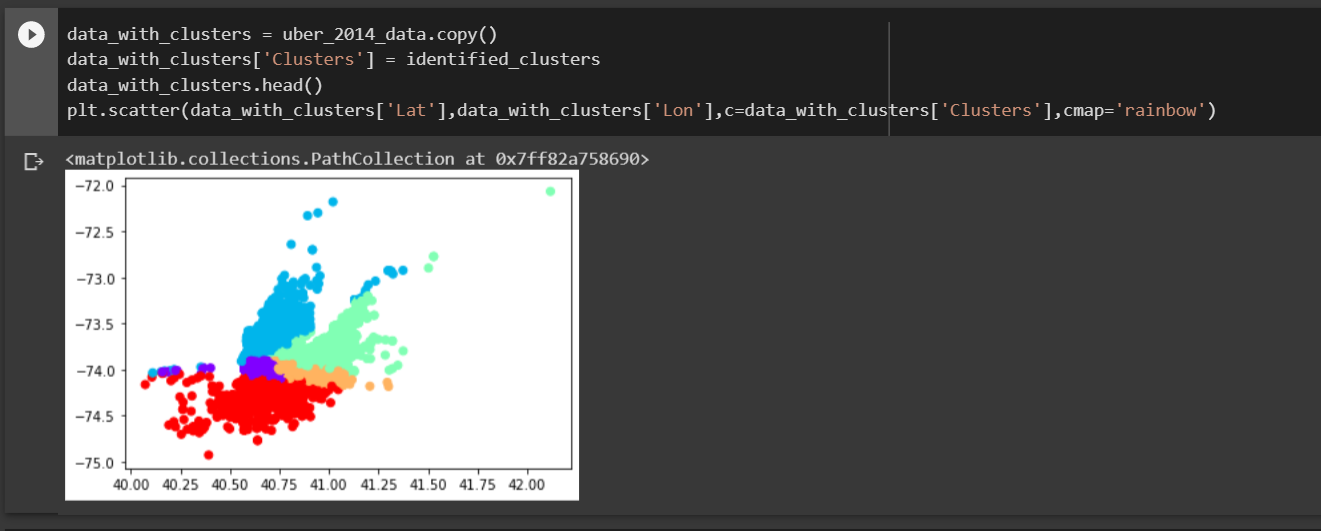
We set our max iterations to 300 and random state to 12345.

We get an array of clusters after applying the kmeans.fit\_predict() method on our prepared data which is ‘clus’ below.



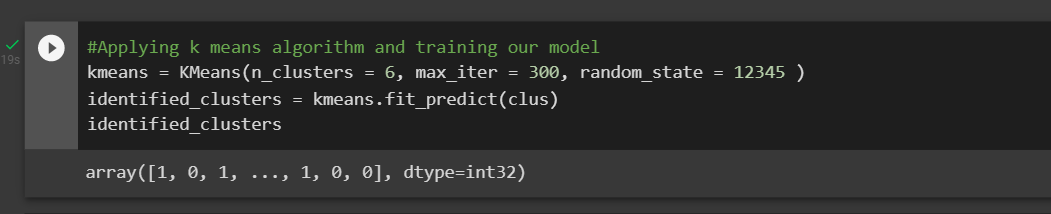
We assigned the identified clusters to our dataset:



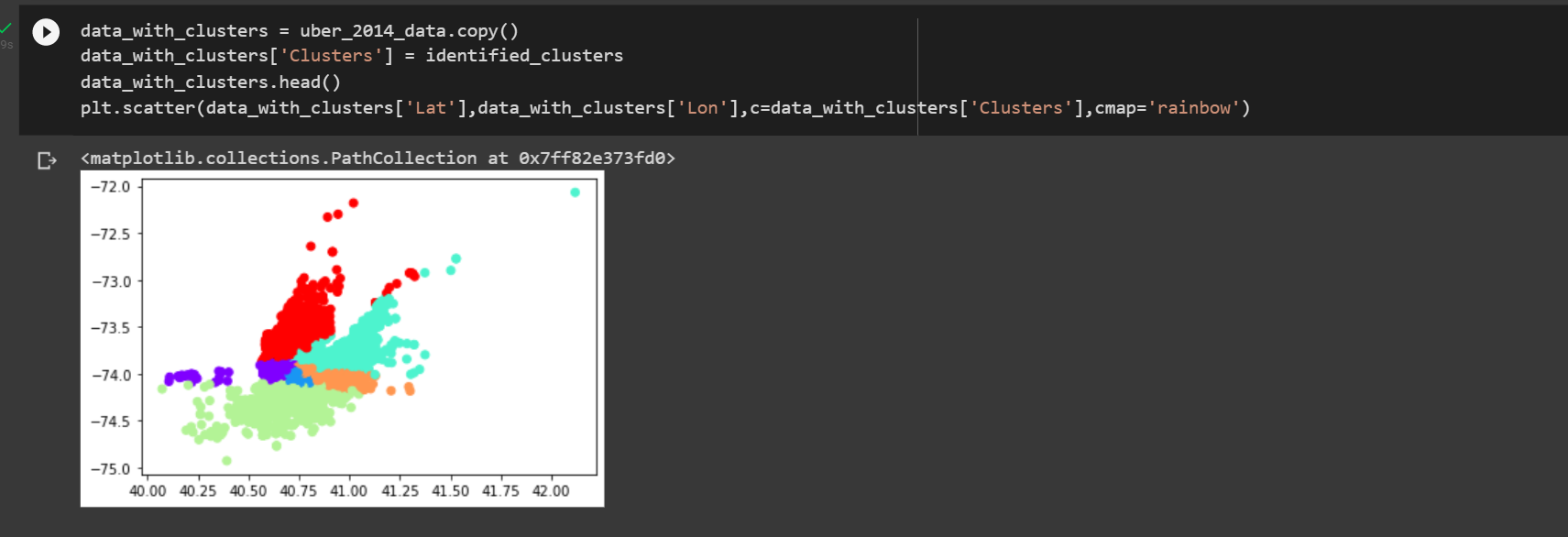
We then make a scatter plot of our original dataset and clusters obtained and use a colored effect to see the distribution of clusters on our scatterplot. As shown in the plot, we can identify 5 different clusters.

We now apply k=6 to check how our model trains on initially 6 clusters.

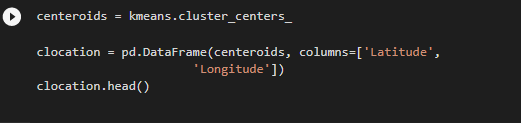
**Using k = 6:**

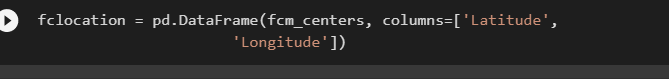


The visualization for the model on k = 6 is as follows. As you can see, we are now forming 6 clusters distributed over this scatter plot.

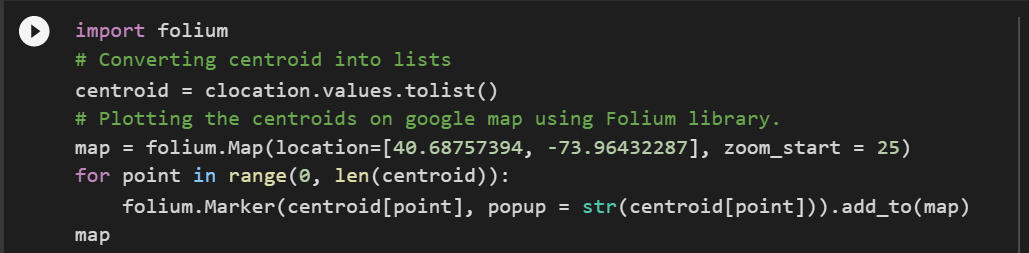


Now to visualize our centroids obtained from both our models, we plotted these clusters on a map to visualize and extract meaningful information out of these clusters since they’re in the form of latitudes and longitudes. We saved our centroids and converted them into a dataframe clocation and fclocation from both algorithms.

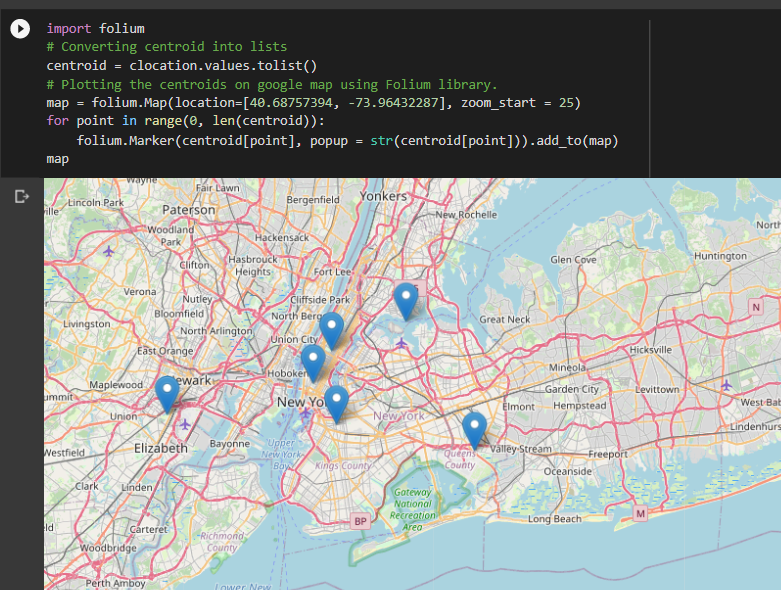




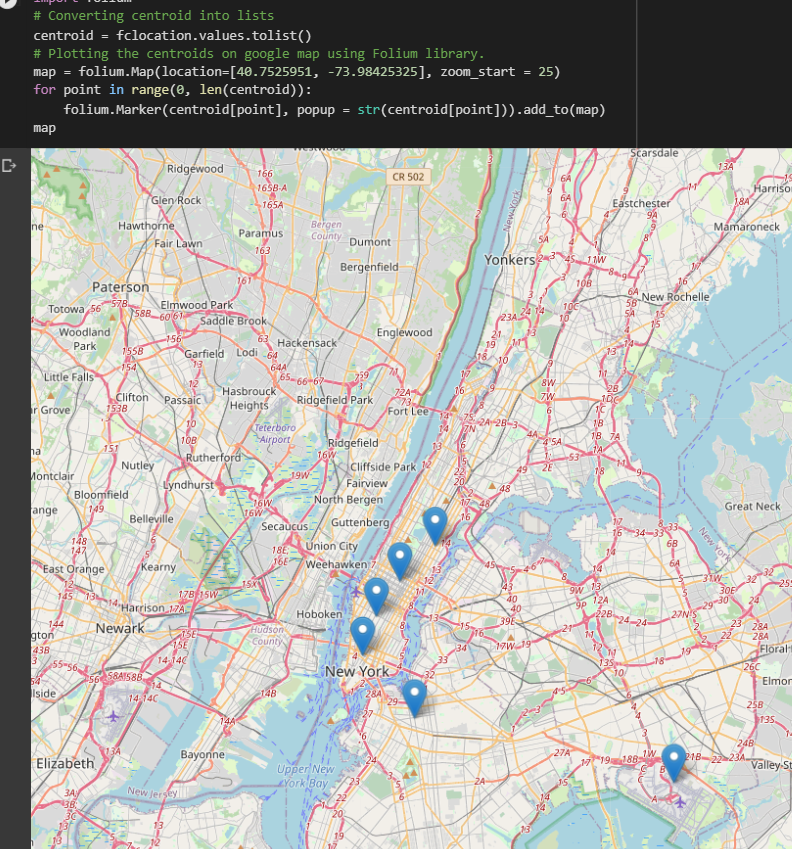
We then used the folium library to map our centroids. This is the code snippet from k means algorithm however the same method was used in c means.



Visualization of centroids obtained from k means, k = 6:



Visualization of centroids obtained from c means, c = 6:



### **Evaluation:**

Our best model is the K-Means clustering algorithm at k = 6 because we obtained our optimal value of k from the elbow method.

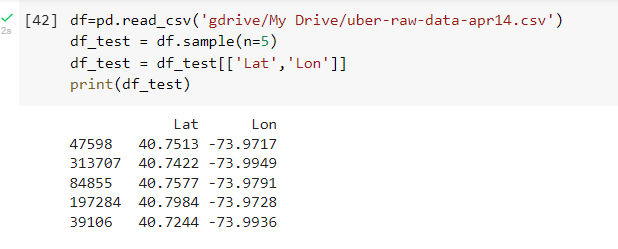
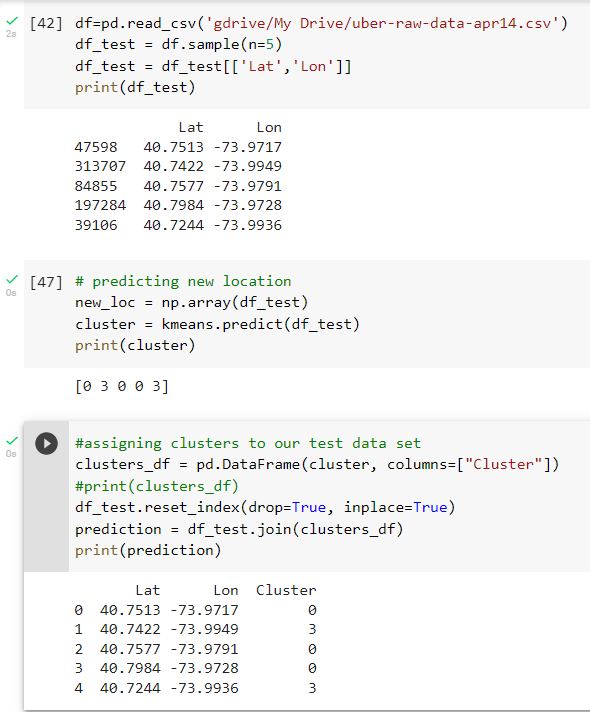
In hard clustering i.e. K-Means each data point is clustered or grouped to any one cluster. However, in soft clustering i.e Fuzzy C-Means each data point is assigned a likelihood or probability score to belong to that cluster. Therefore, the data point can belong to more than one cluster. Since we have spatial data where we want each coordinate to belong to one cluster exclusively, K-Means performed much better than Fuzzy C-Means.

# Findings

### **Insights:**

Using our model we plotted the centroids for each cluster on the map. With these centroids and the visualization data we were able to draw some insights:

* Uber can use these centroids as their hubs and place their vehicles at the optimal location. They can also see which time of the day most pickups are noticed in a specific cluster and send more vehicles towards that hub. This will help Uber to maximize the number of rides they receive in a day and therefore maximize profits.
* Whenever Uber receives a new ride request, they can predict which cluster the new request belongs to and then Uber can direct the vehicle from that hub to the customer location. This will help Uber to serve the customer faster.



* Uber can use these centroids for optimal pricing by analyzing which cluster deals with maximum requests and peak times.

### **Challenges and Limitations:**

For this model we have only taken data of two months, August and September 2014 which relate to the five boroughs of NYC. In real-time, there is far more data (and clusters) to consider, as Uber has a presence in many countries around the world. Our model was done on a small scale and for Uber to actually implement this they would have to do it on a very large scale.

### **When would the model expire?**

The model would require to be retrained if we want to run it on a larger dataset to get more clusters. And the model would need to be retrained every year as we could see that the number of rides increased each month.