**FOSS LAB**

**Project I Report: Uber Cluster Analysis**

**Mohit Motwani**

**205320012**

**Introduction:**

With the rising trend of private real-time cab services like Uber and Ola, commuters have many options, but drivers face much competition to get passengers, making it vital for them to be in the right place at the right time. Uber Technologies Inc. is a peer-to-peer ride-sharing platform. Uber's platform connects customers with drivers who can deliver them to their destination.

Uber uses machine learning to calculate prices and determine the best location for cars to maximize revenues. The public Uber trip dataset for New York is being evaluated as a real-time example for car GPS data processing and monitoring. We examined Uber pickups in New York City over six months in 2014, from April to September, and reported our data and analyses. We aggregated the activity across New York City to identify hotspots where a driver is most likely to locate a passenger at any given hour. We used different clustering algorithms for this unsupervised learning task and compared our predictions with general results from NYC. To validate our predictions, we plot the centroids of clusters over the NYC map and see locations manually.

Further exploring the dataset, we analysed the nightlife spots in NYC from Uber pickup activity from 9 PM to 3 AM. Again, we assumed that because Uber is the most ubiquitous and popular cab service, it could predict commute behaviour in general.

**Objective:**

For each ride, Uber has car GPS location and timing information for a pickup spot. Therefore, it is critical for Uber to strategically position their available Drivers/Cars so that assigning/picking up an incoming Uber trip from a customer for a specific location point takes the least amount of time. If the trip's waiting time is excessively long, customers are more likely to cancel the ride. They would look for a competition with a shorter travel wait time. Ride pickup clusters are identified as zones, and their centroids are identified as hubs using Uber pickup data. Around these hubs, as many cars as possible should be available. This would ensure that the distance and time required to reach the assigned customer trip request are minimal.

When a journey is completed, the driver should be informed of the cluster(zone) he is in as well as the centroid(hub) location so that he can reach out to the hub and maximize his chances of catching the next ride. Uber can use the centroids for the optimal placing of cars based on peak demand during the day. In addition, Uber may use the centroids to determine the best pricing depending on car demand and supply. Surge pricing occurs when demand is high, and supply is low.

**The Data:**

The data was obtained from fivethirtyeight’s GitHub page. The data were from the months of April 2014 through September 2014 and contained pickup locations in and around New York City.

The files contained four columns:

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

Table

Description automatically generated

There are no missing values in the data. If missing data makes up less than 10% of the total data, those records may be removed from the database. Because the majority of the columns are categorical, the appropriate strategy is to impute missing data using the MODE of the column data.

**Literature Review:**

For our analysis, we used Uber trip data which was obtained from fivethirtyeight’s GitHub page. This data has been analysed before and used for a few FiveThirtyEight stories: [Uber is Serving New York’s Outer Boroughs More Than Taxis Are](https://fivethirtyeight.com/features/uber-is-serving-new-yorks-outer-boroughs-more-than-taxis-are/), [Public Transit Should be Uber’s New Best Friend](https://fivethirtyeight.com/features/public-transit-should-be-ubers-new-best-friend/). There were a lot of insights gained by these articles, e.g., most of Uber rides start in Manhattan, Uber is busiest at the evening rush and so on.

A systematic study of these data is necessitated by the fact that we now have huge volume of such data. There were nearly 93 million trips taken by Uber and conventional taxis over a six-month period from April to September 2014. On one hand, these studies might help cab companies understand the customer demand better and gain more revenue. On the other hand, customers benefit from these studies as they might get faster and better services. A few of them and their findings:

1. Visualizing the paths of 10,000 taxi rides across Manhattan. Using data from 10,000 taxi trips and the Google Maps API, graduate students at Columbia University created an animation of the transit arteries of New York City. The visualization recreates a ‘breathing’ map of Manhattan based on the migration of vehicles across the city over a period of 24 hours, displaying the periods of intensity, density, and decreased activity.
2. Making a Bayesian Model to Infer Uber Rider Destinations. The Uber Data team analysed the riding patterns of over 3000 unique riders in San Francisco earlier in 2014. The analysis was aimed at determining which businesses Uber riders like to patronize, e.g. what kind of food or which hotels? Uber used Bayesian statistics and drop-off points for the trips to predict where a user would be going with an accuracy of 75%.
3. The Pulse of a City. How People Move Using Uber. Uber analysed their trip activity distributed hourly across each day of the ordinary week in various cities across the world. The data was then visualized as a heatmap and various inferences were made as well as cities were compared, e.g., When is a city most alive? Which cities are more nocturnal compared to others?

Our analysis conforms to a few findings of earlier studies, like most of Uber rides start at Manhattan and Uber is busiest at evening rush. At the same time our analysis focuses on a different aspect of finding locations where an Uber driver is most likely to find a ride at a given hour and inferring nightlife hotspots of NYC.

**Technology Stack Selection:**

Entire project is done on Python, in Jupyter Notebook environment. We use PySpark to analyse the data, MLlib library for machine learning algorithms and PySQL to use Relational database functionality and provide users with much more information from the analysis.

**Algorithms:**

1. **K-means Algorithm**: K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

It is an iterative algorithm that divides the unlabelled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabelled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabelled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

* Determines the best value for K centre points or centroids by an iterative process.
* Assigns each data point to its closest k-centre. Those data points which are near to the particular k-centre, create a cluster.

Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

The below diagram explains the working of the K-means Clustering Algorithm:



1. **Bisecting K-Means:** Bisecting k-means is a hybrid approach between Divisive Hierarchical Clustering (top-down clustering) and K-means Clustering. Instead of partitioning the data set into K clusters in each iteration, bisecting k-means algorithm splits one cluster into two sub clusters at each bisecting step (by using k-means) until k clusters are obtained.

Diagram

Description automatically generated

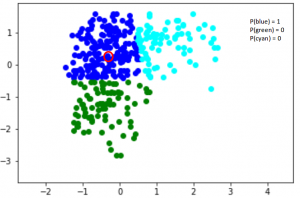
Divisive Clustering starts at the top level with a single cluster and divides it towards the bottom level. In order to decide which clusters should be split, a measure of dissimilarity between sets of observations is required. This is achieved by using a measure of distance between pairs of observations.

1. **Gaussian Mixture Models (GMMs**): Gaussian Mixture Models (GMMs) assume that there are a certain number of Gaussian distributions, and each of these distributions represent a cluster. Hence, a Gaussian Mixture Model tends to group the data points belonging to a single distribution together.

Let’s say we have three Gaussian distributions (more on that in the next section) – GD1, GD2, and GD3. These have a certain mean (μ1, μ2, μ3) and variance (σ1, σ2, σ3) value respectively. For a given set of data points, our GMM would identify the probability of each data point belonging to each of these distributions.

**Gaussian Mixture Models are probabilistic models and use the soft clustering approach for distributing the points in different clusters.**

Here, we have three clusters that are denoted by three colors – Blue, Green, and Cyan. Let’s take the data point highlighted in red. The probability of this point being a part of the blue cluster is 1, while the probability of it being a part of the green or cyan clusters is 0.



Now, consider another point – somewhere in between the blue and cyan (highlighted in the below figure). The probability that this point is a part of cluster green is 0, right? And the probability that this belongs to blue and cyan is 0.2 and 0.8 respectively.

Chart, scatter chart

Description automatically generated

Gaussian Mixture Models use the soft clustering technique for assigning data points to Gaussian distributions. I’m sure you’re wondering what these distributions are so let me explain that in the next section.

**Novelty proposed:**

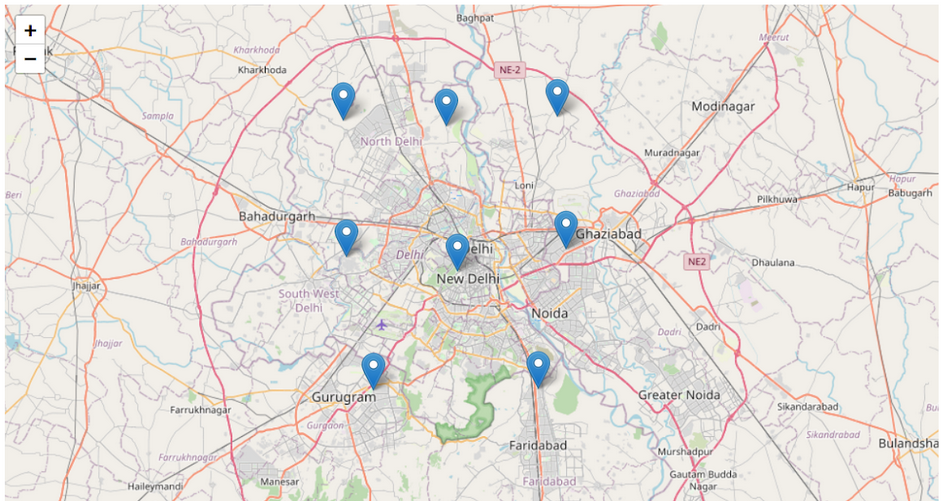
1. In most of the previous work K-Means Clustering algorithm was used to make clusters and those clusters centroids were used to create those locations as Uber hubs. In this project we would use more sophisticated Clustering algorithms such as **Bisecting K-means**, and **Gaussian mixture model** to improve results and minimized the time taken to perform the operations.
2. We would also try to create a much better UI using maps to visualize centroids, which will help drivers find the centroids faster and easier.

Map

Description automatically generated

1. We would also use Google's reverse geocoding API to get the street addresses corresponding to our cluster centroids at different times to get human-readable locations that can be used to inform drivers of current hot spots in the neighbourhood.
2. We can also retrieve this data. We will use Spark SQL to explore the dataset. We can create a query system which can retrieve data and visualize for queries like:
   1. Which hours of the day and which cluster had the highest number of pickups?
   2. How many pickups occurred in each cluster?
   3. How many pickups occurred in each hour?
3. We can also analyze the data for Holidays and find people's behaviour on holidays like Easter.

1. We also try to apply this algorithm to Data generated from New Delhi.

****

**Applications:**

* Uber can use these centroids as their hubs. Whenever Uber receives a new ride request, they can check the closeness with each of these centroids. Whichever particular centroid is closer than the Uber can direct the vehicle from that particular location to the customer location. Identified cluster-centroids should be designated as hubs. Maximum cars should be available around these hubs at any time for drivers to move minimum distance to reach the incoming trip demand.
* Uber has many drivers and provides services to many locations. If Uber knows the hub (particular centroid), and if they are getting many rides requests, they can strategically place their drivers in a good location wherein the probability of getting a ride request is huge. This will help Uber serve the customer faster as vehicles are placed closer to the location, and it helps to grow their business. For example, cars may drop passengers in a different cluster. The driver should reach the nearest hub while waiting for the next trip.
* Uber can make use of these centroids for the optimal placing of their vehicles. They can find which centroid at which part of the day more ride request come in. For example, if Uber get more request from centroid 0 (cluster 1) at 11 AM, but very less request from centroid 3 (cluster 4), then they can redirect the vehicles to cluster 1 from cluster 4 (if more vehicle presence in cluster 4).
* Uber can use these centroids for optimal pricing by analysing which cluster deals with maximum requests, peak times etc. Suppose if they don’t have too many vehicles to be sent to a particular location (more demand), then they can do optimal pricing as demand is high and supply is less.

Effective taxi dispatching will facilitate each driver and passenger to reduce the wait time to seek out one another. The model is employed to predict the demand on points of the city.

**Conclusion:**

The conclusion of the project is to project a basic outline of trips travelled concerning latitude and longitude of locations and pinpoint the locations travelled concerning the frequency of trips travelled by a uber cab during the day and also based on the cross analysing of the dataset based on the latitude and longitude of the point travelled by the cab which is then analysed by deploying k-means clustering which classifies the locations based on centroids and then orders the frequency of trips based on labels or clusters. By the location given by the user, the algorithm predicts the cluster nearest to the location so that cab can be assigned to the user for pickup.

The merit of the project is that it explains the functioning of how cabs are assigned to passengers based on an unsupervised algorithm and explains the key concepts of machine learning.