# Predict How To Prioritize Taxi Service in Different Locations and Different Times

**Group member:** Zhuohao Deng (zd612), Rui Jiang (rj2397), Muxin Jiang (mj3068), Yang Liu (yl9908), Zihao Liu (zl4969)

**CUSP-GX 5003: Machine Learning for Citie**

**Final Report**

**December 16, 2022**

## Abstract

The data shows that New York City generates roughly thirty million cab journeys annually. Urban taxi networks can operate more effectively if supply and demand information is used. More people utilize taxis than in any other American metropolis in New York City. The capacity to forecast taxi ridership may offer city leaders and taxi dispatchers useful information to address issues like where to place taxis for maximum efficiency, how many taxis to send out, and how ridership changes over time between variations. Our study focuses on estimating the demand for taxis in a certain area of New York City given a location.

## Introduction

The main purpose of this project is to analyze the demand for taxis, especially the yellow taxis in various districts of New York City at various times of a day through some K-means clustering method and Gaussian mixture models. Yellow cabs were incorporated in New York on April 4, 1912 and became a symbol of New York City. The taxis painted yellow can pick up customers anywhere in Manhattan, Brooklyn, the Bronx, Queens, and Staten Island boroughs. With the development of urban modernization, yellow taxis are less popular travel tools than before because of the rise of online car hailing like Uber and Lyft. However, sometimes the online car hailing is difficult to pick. For example, at the airport, it may take a long wait time to get an uber and the price is very high because of the distance and time. When people feel inconvenience and do not want to wait a long time, some are back in the classic yellow taxis. People can just wave hands to pick the yellow taxi at reasonable prices within a short time.

This project starts from a plot and normalizes the general trips per zip code to see which zip code has more trips and higher density. We focus on rush hours of the day. We will first discuss the data we used then go on to the pre-processing and processing part that we prepared. We also will discuss the method and model we used to analyze. Results and conclusions will be the end of the project with some related graphs and figures.

## Literature review

- *Predicting taxi passenger demand using artificial neural networks*, by Gustav Zander, 2017, Retrieved from https://www.diva-portal.org/smash/get/diva2:1082065/FULLTEXT01.pdf

This report proposes a machine learning method using artificial neural networks to estimate the demand for taxis in different geographical areas of the city of Stockholm. The difference between our project and his research is that he used the classification method while we focus on clustering. But he considered more input features than the factors of our project, which are the hour of the day, the day of the week, the month of the year, days after payment, the zone,

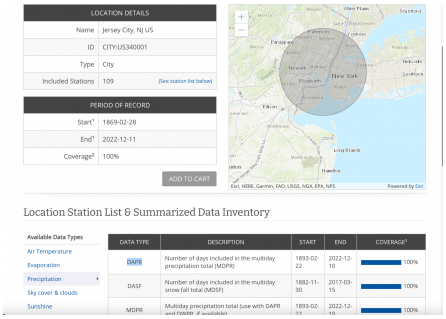
the rain and the temperature. In his report, he had more detailed considerations such as overfitting problems, loss function convergence, and optimization algorithms.

- *Real Time Prediction of Cab Fare Using Machine Learning*, By T. Prem Jacob; A. Pravin; K. Mohana Prasad; G. T. Judgi; R. Rajakumar, March 2022, Retrieved from https://ieeexplore.ieee.org/document/9752315/authors#authors

This report tries to know the patterns and predict the cab fare amount by using different methods within a certain city. The main focus of this research work, which is optimizing cab fare, is different from our project's aim. However, we used similar data features such as pickup and drop-off location. Their methodology, however, is different from ours, which are regression methods.

## Data

We downloaded and used TLC Trip Record data and NYC weather data on the National Center of Environmental information website. The dataset we choose is Yellow taxi trip records every month in 2021 including: VendorID, tpep_pickup_datetime, tpep_dropoff_datetime, passenger_count, trip_distance, RatecodeID, store_and_fwd_flag, PULocationID (Pick-up location), DOLocationID (Drop-off location) and the precipitation and non-precipitation daily weather data in 2021 for New York City.


| STATION | NAME | LATITUDE | LONGITUDE |
|---|---|---|---|
| US1NJBG0043 | MAYWOOD 0.2 SW, NJ US | 40.900010 | -74.066410 |
| US1NJBG0043 | MAYWOOD 0.2 SW, NJ US | 40.900010 | -74.066410 |
| US1NJBG0043 | MAYWOOD 0.2 SW, NJ US | 40.900010 | -74.066410 |
| US1NJES0031 | MONTCLAIR 0.7 N, NJ US | 40.835343 | -74.208967 |
| US1NJES0031 | MONTCLAIR 0.7 N, NJ US | 40.835343 | -74.208967 |

*Figure 1: Demonstration of Dataset Selection and Data Feature Selection.*

### Pre-processing

We merged the two datasets and performed an analysis based on this merged data.

### Process

We used both drop NaN and deduplication functions to filter data, and for special data, such as yellow taxis with time less than 5 min, and The travel time is less than 500 meters. And in the weather data, only New York City data is kept by limiting the latitude and longitude.

The final cleaned the dataset from 30 million to 25 million for yellow taxis and 110,000 to 7,000 for weather data. We convert the latitude and longitude to the zip code and merge both geo-datas together and try to compare precipitation and non-precipitation conditions. However, after we have processed the weather data by clarifying that only 147 valid data for 7000 total cells, as figure 2 shows, we realized that weather data does not satisfy our goal. The valid data only are 2.1% of datasets. Considering the authenticity of the overall results, we do not intend to

use WEATHER data as the subject of the Modeling.

fliter the ONlY NYC Data.

```
In [101]: index = (weather_data['LATITUDE'] > 40.495992) & (weather_data['LATITUDE'] < 40.915568) &\
              (weather_data['LONGITUDE'] > -74.257159) & (weather_data['LONGITUDE'] < -73.699215)
          weather_data = weather_data.loc[index]

In [102]: weather_data.count()

Out[102]: STATION             6992
          NAME                6992
          LATITUDE            6992
          LONGITUDE           6992
          ELEVATION           6992
          DATE                6992
          DAPR                 147
          DAPR_ATTRIBUTES      147
          MDPR                 146
          MDPR_ATTRIBUTES      146
          dtype: int64
```

Final Size is the 6992.

*Figure 2: The Total Available Dataset Cell Numbers (STATION) and Valid Data Count (DAPR) of Weather Data in New York City.*

**Method**

Since we indicated that the weather data is invalid and cannot produce useful results for machine learning, we decided to not consider the conduct of regression analysis. The clustering method is the most appropriate method to analyze the data of taxis in New York City.

The first step was to look at the demand for Taxis between the different zip code areas through geopandas. The figure  below demonstrates that central and lower Manhattan has the majority of the demand. We are also interested in the relationship between taxi demand and the hour of the day. We sorted the dataset by the hour and presented the demand for Yellow taxis by column chart. The trend of taxi trips starts from 4 am, which is the lowest of the day, and starts to increase. The trend peaks at 6 pm and declines after that.

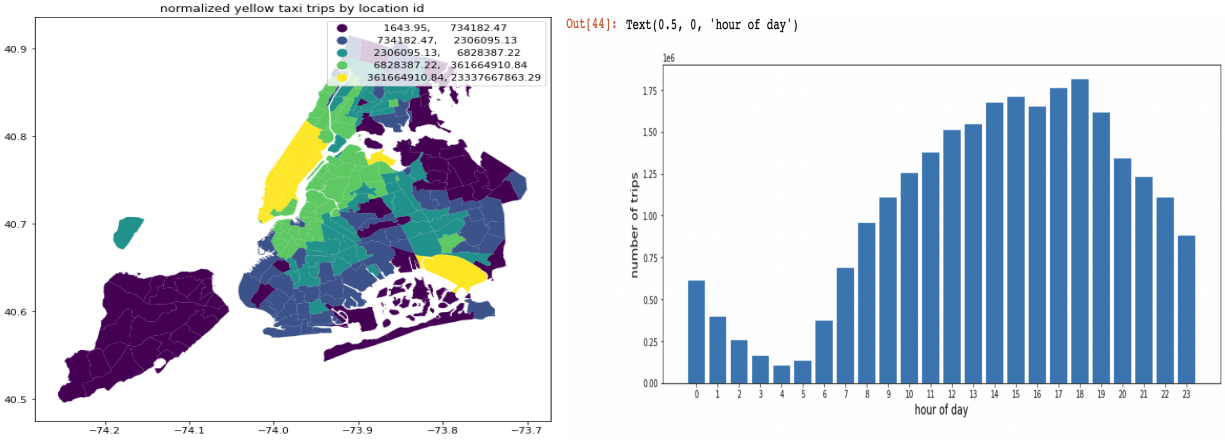



*Figure 3: Normalized Yellow Taxi Trips by Location ID and the Bar Chart of Total Numbers of Taxi Trips in Different Hours of Day (2021) in New York City.*

Moreover, we set a parameter for the Yellow Taxi activity. This parameter is used to determine the level of demand by each area.The parameter include the average distance traveled,

the average number of passengers, and the total number of trips by each hour in each region

Finally, we aggregate the Rush hour data. We split the data into two parts.Morning Rush Hour dataset is 7 to 10 a.m. Evening Rush Hour Dataset is 5 to 8 p.m. So, There are Two databases used to train models and make predictions by time.

**Model**

**K-Means Clustering**

K-Means Clustering is put observations into k clusters and each observation belongs to the nearest mean of the cluster. Our group uses this K-means clustering because the dataset is unlabeled, we do not know which area has the highest trip count. By the analysis we did in the last part, we have our initial prediction that Manhattan and two airports will have a higher value of trips compared to other areas.

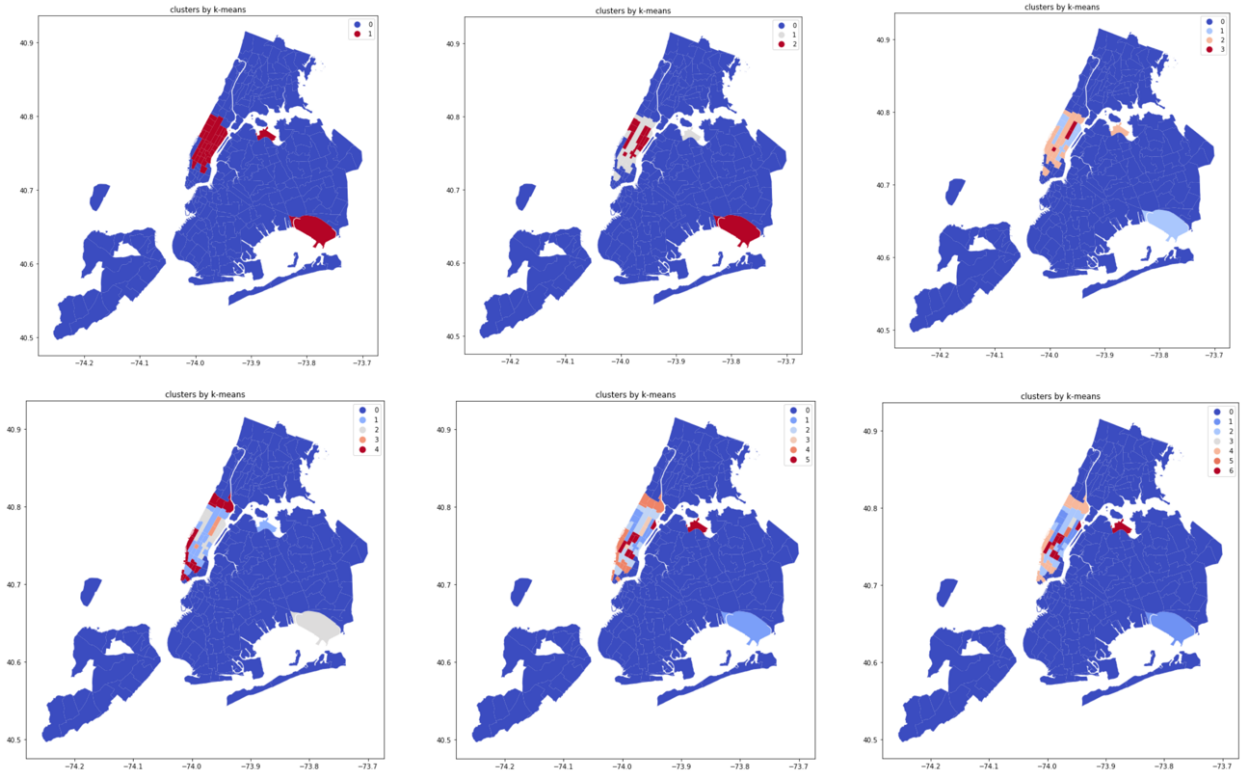



*Figure 4: 6 Clustering Results of the K-Means Clustering for Rush Morning (7:00 a.m. to 10:00 a.m.) in New York City.*

After we perform K-means clustering to the dataset from n=2 to 7, we can see that all the other parts of NYC (excluding JFK, LaGuardia and Manhattan) always share a cluster. Adding new clusters to the model highly probably assigns the cluster to some places in Manhattan. This is probably because there are different features shared by the zip code in Manhattan, indicating that the total trips per zip code in Manhattan has a large variety of different average trip distances and trips count total. From k-means, we can see the centers

of each cluster, and in this case, the center stands for (average trip distance, total trips) . And also the higher we get on total trips the lower we may get on the average trip distances. For n=7, cluster n=0 has the highest average trip distance but has the lowest trip count. cluster n=4 and n= 6 are the top two with highest trips count but have the lowest travel distance.

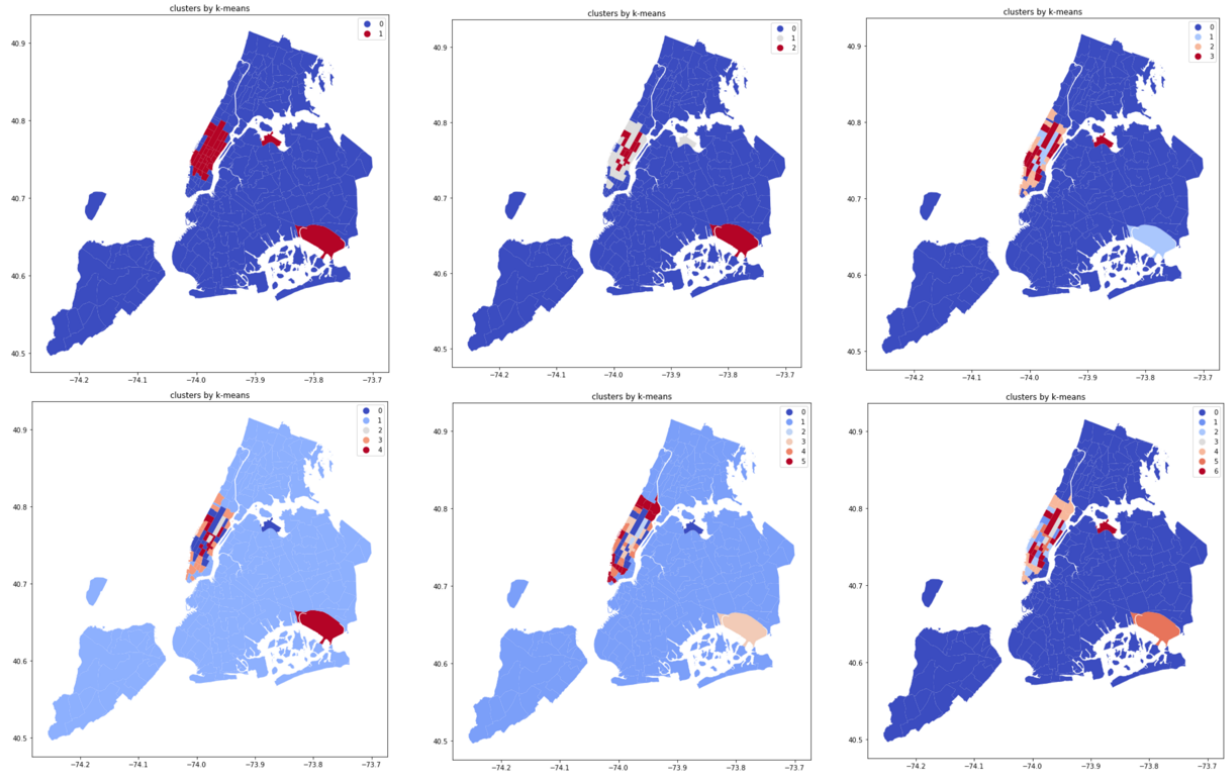



*Figure 5: 6 Clustering Results of the K-Means Clustering for Rush Evening (5:00 p.m. to 8: 00 p.m.) in New York City.*

Also, the thing is similar for 5 p.m. to 8p.m. Then, we want to find the best n for our model. We use Elbow plotting.

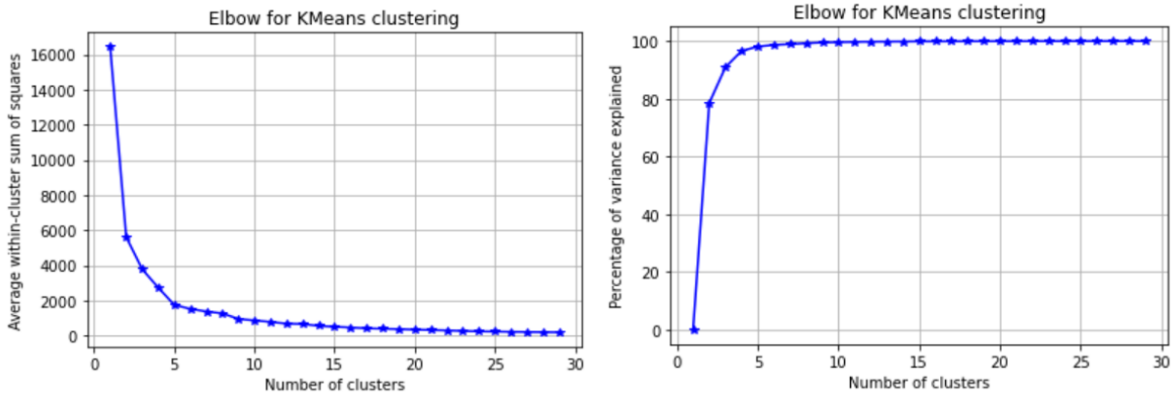



*Figure 6: Elbow Test Results of the K-Means Clustering in New York City.*

From the elbow theorem, we can say that the best performance of k-means in this situation is in range(3,6).

Also , we tried a silhouette score test for K-means. After getting the silhouette score for each scenario of a different choice of n, we get the conclusion that the best n for K-means clustering is n=4.

The silhouette score for n from 2 to 29

```
For n_clusters =2, the average silhouette_score is :0.8874625350628897
For n_clusters =3, the average silhouette_score is :0.8888600760631226
For n_clusters =4, the average silhouette_score is :0.9002936726448316
For n_clusters =5, the average silhouette_score is :0.8713634359864693
For n_clusters =6, the average silhouette_score is :0.8645870865411377
For n_clusters =7, the average silhouette_score is :0.8625469939414233
For n_clusters =8, the average silhouette_score is :0.8621573132782759
For n_clusters =9, the average silhouette_score is :0.8523073352577547
For n_clusters =10, the average silhouette_score is :0.8491442318138223
For n_clusters =11, the average silhouette_score is :0.8546305515042294
For n_clusters =12, the average silhouette_score is :0.8511019847689248
For n_clusters =13, the average silhouette_score is :0.8459933692931346
For n_clusters =14, the average silhouette_score is :0.8531502245976769
For n_clusters =15, the average silhouette_score is :0.8339134256084805
For n_clusters =16, the average silhouette_score is :0.810262402646688
For n_clusters =17, the average silhouette_score is :0.80834785323009
For n_clusters =18, the average silhouette_score is :0.8061805043692503
For n_clusters =19, the average silhouette_score is :0.8000211187781167
For n_clusters =20, the average silhouette_score is :0.8049078042748794
For n_clusters =21, the average silhouette_score is :0.7775586063070165
For n_clusters =22, the average silhouette_score is :0.8093328765027334
For n_clusters =23, the average silhouette_score is :0.7832436119330228
For n_clusters =24, the average silhouette_score is :0.7824077765725491
For n_clusters =25, the average silhouette_score is :0.7780764826627132
For n_clusters =26, the average silhouette_score is :0.7776917822898487
For n_clusters =27, the average silhouette_score is :0.7732614920473259
For n_clusters =28, the average silhouette_score is :0.7694431828369762
For n_clusters =29, the average silhouette_score is :0.7664288540737386
```

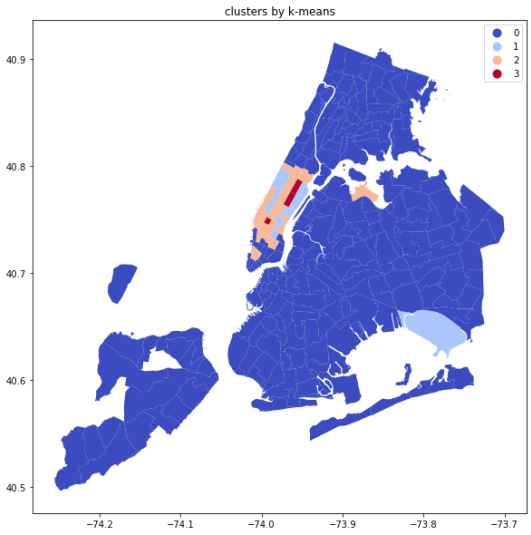



*Figure 7: The Analysis based on the Average Silhouette Score of the K-Means Clustering and the Graph of the Best K-Means Result (N=4) in New York City.*

Cluster centers:

[average distance, trip count]
[[5.76341110e+00 1.27349321e+03]
[3.36559677e+00 8.46266923e+04]
[2.84239230e+00 4.38908571e+04]
[2.01243342e+00 1.49917000e+05]]

When n=4, for morning rush hour, we can see that the whole other area in NYC is in cluster 0, which has an average of 1273 trips per each location id. Most of middle Manhattan and LaGuardia forms cluster 2 with an average of 43891 trips per id, while some part of Manhattan along with JFK is in cluster 1 with an average of 84627 trips per id. What's more, there is a small part of the area in Manhattan that has a surprisingly high average trips count(149917 trips per id), which is relatively higher than other clusters.

## Gaussian Mixture Models (EM)

Gaussian Mixture Models (EM) is a great comparison method with K-means clustering. The graphs below show the 6 different Gaussian Mixture Models Clustering for Rush Morning in New York City.

For Gaussian Mixture, some central areas in Queens and Brooklyn are assigned to groups in Manhattan. But also, significantly, we can see that, in the Gaussian Mixture model, neighboring zip codes tend to form in the same cluster, which is much more effective than K-Means Clustering due to better and more efficient taxi arrangement in a closer area. Through visualization, we can see a strong bond in Manhattan, JFK and LaGuardia, which is the same with the answer we got in K-means.


*Figure 8: 6 Clustering Results of the Gaussian Mixture Models (EM) for Rush Morning (7:00 a.m. to 10:00 a.m.) in New York City*

As figure 9 shows, GM clustering of Rush Evening shows similar results. The Gaussian Mixture Models still form clusters in a close and robust way. During the rush evening, Manhattan, JFK, LaGuardia, and certain parts of Queens and Brooklyn would be the main focus of Taxi Demand.

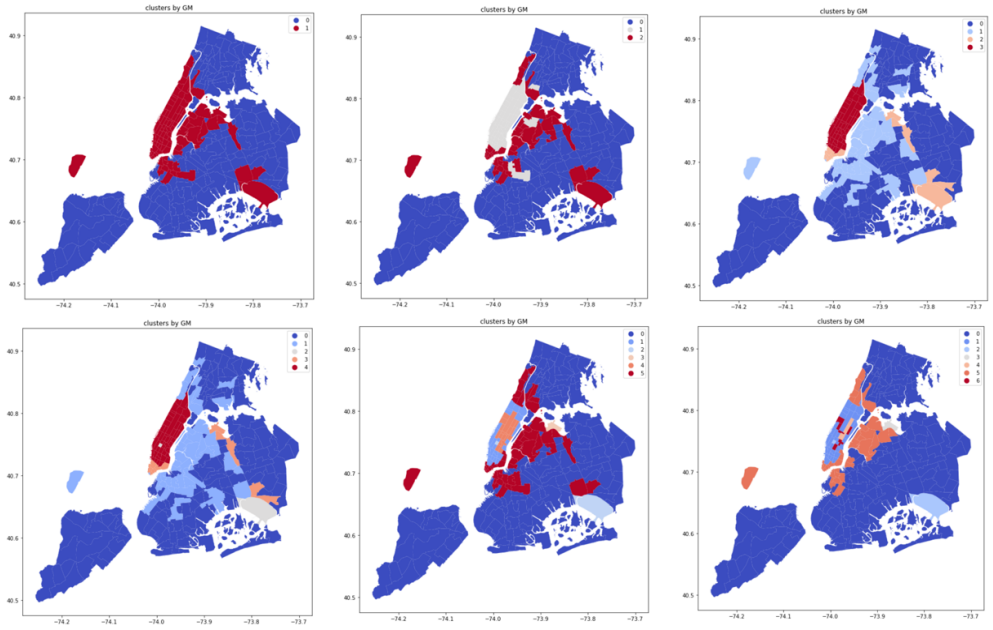



*Figure 9: 6 Clustering Results of the Gaussian Mixture Models (EM) for Rush Evening (5:00 p.m. to 8:00 p.m.) in New York City.*

As figure 10 below shows, from the analysis based on the average silhouette score of Gaussian Mixture Models (EM), when n_clusters is 13, it has the best performance. The graph of Gaussian Mixture Models (EM) with n_cluster = 13 demonstrates that every cluster should receive 1/13 of total taxi carrying capacity in New York City. It is still clear that most of Manhattan, the Airports are the main focus, but we can spare some taxis into the central area of Brooklyn and Queens.

```
For n_clusters = 2, the average silhouette_score is : 0.5640178768665115
For n_clusters = 3, the average silhouette_score is : 0.5866738014461085
For n_clusters = 4, the average silhouette_score is : 0.5322404572966052
For n_clusters = 5, the average silhouette_score is : 0.6360497254431448
For n_clusters = 6, the average silhouette_score is : 0.5373113653993753
For n_clusters = 7, the average silhouette_score is : 0.6059074329456122
For n_clusters = 8, the average silhouette_score is : 0.604828785115369
For n_clusters = 9, the average silhouette_score is : 0.6524045178087183
For n_clusters = 10, the average silhouette_score is : 0.6713221263982218
For n_clusters = 11, the average silhouette_score is : 0.6954104243319532
For n_clusters = 12, the average silhouette_score is : 0.6997606927581321
For n_clusters = 13, the average silhouette_score is : 0.715660433293852
For n_clusters = 14, the average silhouette_score is : 0.6578669207594211
For n_clusters = 15, the average silhouette_score is : 0.6698798322470867
For n_clusters = 16, the average silhouette_score is : 0.6644341369654667
For n_clusters = 17, the average silhouette_score is : 0.04438750719799862
For n_clusters = 18, the average silhouette_score is : 0.11938820429951844
For n_clusters = 19, the average silhouette_score is : 0.11860088317314757
For n_clusters = 20, the average silhouette_score is : 0.12086916576149905
For n_clusters = 21, the average silhouette_score is : 0.11490444366356595
For n_clusters = 22, the average silhouette_score is : 0.11560116769952895
For n_clusters = 23, the average silhouette_score is : 0.11126987378969315
For n_clusters = 24, the average silhouette_score is : 0.10427771422916089
For n_clusters = 25, the average silhouette_score is : 0.10353569310115723
For n_clusters = 26, the average silhouette_score is : 0.03534446442947956
For n_clusters = 27, the average silhouette_score is : 0.1015823577833874
For n_clusters = 28, the average silhouette_score is : 0.10746977208448945
For n_clusters = 29, the average silhouette_score is : 0.1038747102550386
the best cluster number is 13.
```

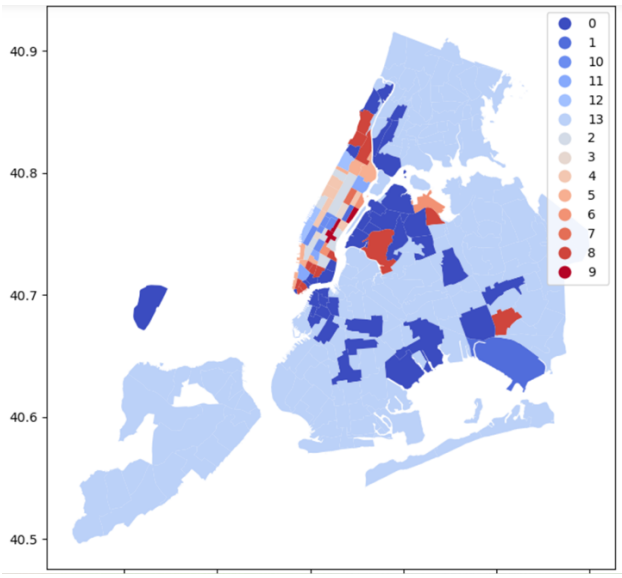


*Figure 10: The Analysis based on the Average Silhouette Score of the Gaussian Mixture Models (EM) and the Graph of the Best GM Result (N=13) in New York City.*

The first visible difference between K-Means and Gaussian Mixtures is the shape of the decision boundaries. Another difference is that GM assigned clusters in a closer way, Thus, GMs can predict better with better detailed clustering than K-Means, to arrange Taxi cars in a more effective way. Additionally, for the K-Means clustering results, we can only see the main cluster in the Manhattan area. But the Gaussian Mixtures model shows the cluster in Manhattan, Brooklyn and Queens. So if the government or taxi company want to know how to place the taxi in the area not only in Manhattan, the Gaussian mixtures model is better than the K-means clustering.

After analyzing the result of the clusters we got from K-means and Gaussian Mixture Models, we realized that the 'Number of trips' is overweight compared to the 'average trip distances' (very high variance versus low variance), which results to the high value of Sum Of Squared Errors we got for K-means method. Hence, we improved that in two ways. First one, we deleted the roll of average distances and only focused on the trips that are within the pick up location(PULocation), and got a similar result with our model, but more simpler since there is only one parameter trained for k-means. The second one, we use a logarithm function to transform the column 'N_trips' into 'logN_trips' and modify the other column. The result is much more complicated that more areas in Brooklyn and Queens are assigned to form new clusters with moderate travel distances and trip counts. However, the middle Manhattan, JFK and LaGuardia are still in control of the cluster with the highest trip counts which is consistent with our earlier result.

## Conclusion

JFK and LaGuardia airports had the most demand for cabs, per the images from k means and EM, which also include Manhattan and two other airports. The decision boundary's form is the first immediately noticeable distinction between K-Means and Gaussian mixtures. GM is a little more adaptable and comprehensive, and by employing the covariance matrix and K-means, we may create elliptical borders rather than circular ones. The fact that GMs are a probability algorithm is another issue. We may convey how strongly we think a given piece of evidence relates to a specific cluster by giving each data point a probability value. By contrasting the two methods with the number of yellow cab passengers in various parts of New York City and at various times of the day, Gaussian mixtures seem to be more reliable. Although GM takes more rounds of the EM method to attain convergence, it is often slower than K-Means. Additionally, they might quickly reach the global minimum which is not the best option.We all contribute equally to the project.

## Reference

*NCEI Data Service API User Documentation*. National Centers for Environmental Information (NCEI). (2022, September 12). Retrieved December 16, 2022, from https://www.ncei.noaa.gov/support/access-data-service-api-user-documentation

*Predicting taxi passenger demand using artificial neural networks*, by Gustav Zander, 2017, Retrieved from https://www.diva-portal.org/smash/get/diva2:1082065/FULLTEXT01.pdf

*Real Time Prediction of Cab Fare Using Machine Learning*, By T. Prem Jacob; A. Pravin; K. Mohana Prasad; G. T. Judgi; R. Rajakumar, March 2022, Retrieved from https://ieeexplore.ieee.org/document/9752315/authors#authors

*TLC Trip Record Data*. TLC Trip Record Data - TLC. (n.d.). Retrieved December 16, 2022, from https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page