RUTGERS

THE STATE UNIVERSITY
OF NEW JERSEY

CAPSTONE PROJECT

COMPARING MINI BATCH K-MEANS AND K-MEANS USING NYC CRIME AND POLICE PRECINCT SHAPEFILE

PROJECT GUIDE:

Michael Katehakis

PROJECT MEMBERS:

Akhil Patil

Palash Nandecha

Arjun Prakash

2019

INTRODUCTION:-

The City of New York, usually referred to as either New York City (NYC) or simply New York (NY), is the most populous city in the United States. With an estimated 2018 population of 8,398,748 distributed over a land area of about 302.6 square miles (784 km2), New York is also the most densely populated major city in the United States.The New York metropolitan area, the largest metropolitan area in the world by urban landmass and one of the world's most populous megacities,with an estimated 19,979,477 people in its 2018 Metropolitan Statistical Area and 22,679,948 residents in its Combined Statistical Area. A global power city,New York City has been described as the cultural,financial,and media capital of the world,and exerts a significant impact upon commerce,entertainment, research, technology, education, politics, tourism, art, fashion, and sports.

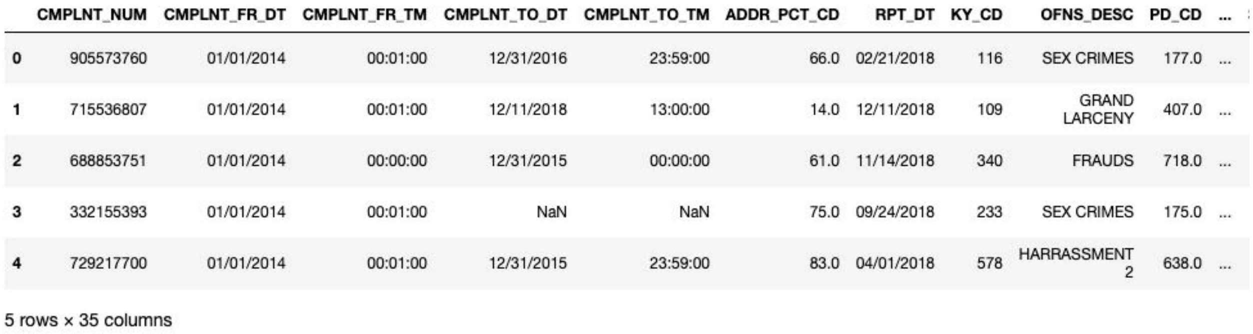
PROBLEM:-

New York being one of the most important metropolitan on the globe, hence safety is one of the most fundamental needs for the residents of the city,and it has been one of the major problems over the decades for the city.Hence our main focus point for this project was to use the crime data of the city of the period of 5 years from 2014 start to the end of 2018, and then we have implemented two K-means algorithm , one normal K-means and other mini batch K-means algorithm and through conclusions based on implementation in these two cases we will decide which one is better.

DATA ACQUISITION AND CLEANING:-

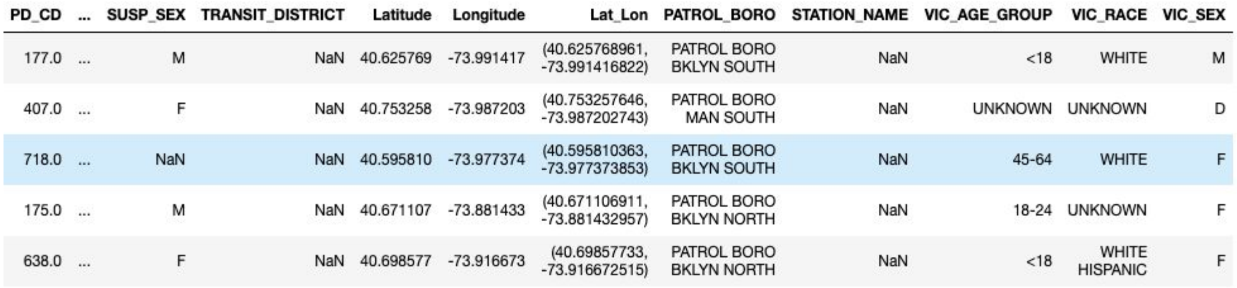
DATA SOURCES:-

For our analysis we required crime data and precinct data which we obtained from **NYC Open Data**.Then after pulling the data foremost thing is to clean the data and point out the necessary attributes which are required for the utmost accuracy.Hence after pulling the crime data it looked like this:-



	CMPLNT_NUM	CMPLNT_FR_DT	CMPLNT_FR_TM	CMPLNT_TO_DT	CMPLNT_TO_TM	ADDR_PCT_CD	RPT_DT	KY_CD	OFNS_DESC	PD_CD	...
0	905573760	01/01/2014	00:01:00	12/31/2016	23:59:00	66.0	02/21/2018	116	SEX CRIMES	177.0	...
1	715536807	01/01/2014	00:01:00	12/11/2018	13:00:00	14.0	12/11/2018	109	GRAND LARCENY	407.0	...
2	688853751	01/01/2014	00:00:00	12/31/2015	00:00:00	61.0	11/14/2018	340	FRAUDS	718.0	...
3	332155393	01/01/2014	00:01:00	NaN	NaN	75.0	09/24/2018	233	SEX CRIMES	175.0	...
4	729217700	01/01/2014	00:01:00	12/31/2015	23:59:00	83.0	04/01/2018	578	HARRASSMENT 2	638.0	...

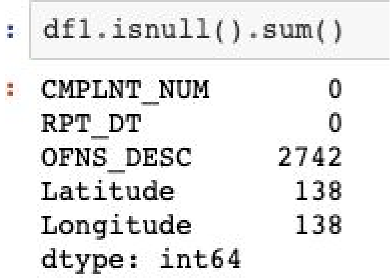
5 rows × 35 columns



PD_CD	...	SUSP_SEX	TRANSIT_DISTRICT	Latitude	Longitude	Lat_Lon	PATROL_BORO	STATION_NAME	VIC_AGE_GROUP	VIC_RACE	VIC_SEX
177.0	...	M	NaN	40.625769	-73.991417	(40.625768961, -73.991416822)	PATROL BORO BKLYN SOUTH	NaN	<18	WHITE	M
407.0	...	F	NaN	40.753258	-73.987203	(40.753257646, -73.987202743)	PATROL BORO MAN SOUTH	NaN	UNKNOWN	UNKNOWN	D
718.0	...	NaN	NaN	40.595810	-73.977374	(40.595810363, -73.977373853)	PATROL BORO BKLYN SOUTH	NaN	45-64	WHITE	F
175.0	...	M	NaN	40.671107	-73.881433	(40.671106911, -73.881432957)	PATROL BORO BKLYN NORTH	NaN	18-24	UNKNOWN	F
638.0	...	F	NaN	40.698577	-73.916673	(40.69857733, -73.916672515)	PATROL BORO BKLYN NORTH	NaN	<18	WHITE HISPANIC	F

DATA CLEANING:-

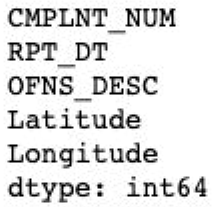
Since our data is raw and we have directly used from nyc open data online portal hence there will be a lot of anomalies in the dataset like null values, then there will be some unassigned values in the dataset. Our primal work was to get those attributes through which we will solve our problem statement.Hence we extracted the necessary attributes and then we checked for any discrepancies in these attributes and whatever discrepancies we found like null values or redundancy of values using the necessary functions like **is.null.sum()** and other functions and then using functions like **dropna()** we checked whether any anomalies are still prevalent in our dataset or not.



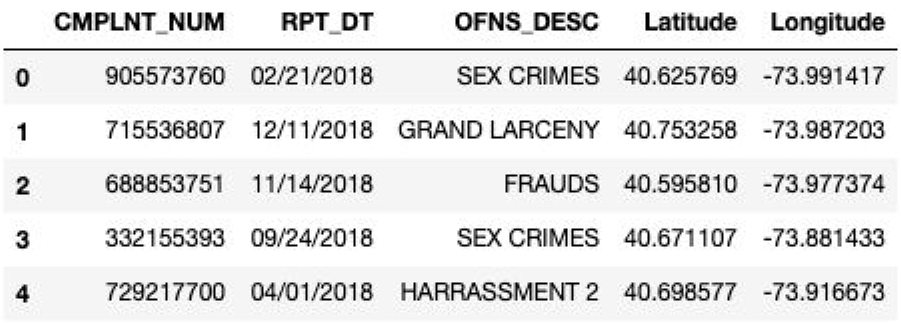
```
: df1.isnull().sum()

: CMPLNT_NUM        0
  RPT_DT            0
  OFNS_DESC      2742
  Latitude        138
  Longitude       138
  dtype: int64
```

Hence after all the necessary actions we used five attributes for our analysis which were complaint number,report date,offense description,latitude and longitude and after the data cleaning step dataset looked like this:-



```
CMPLNT_NUM    0
RPT_DT        0
OFNS_DESC     0
Latitude      0
Longitude     0
dtype: int64
```

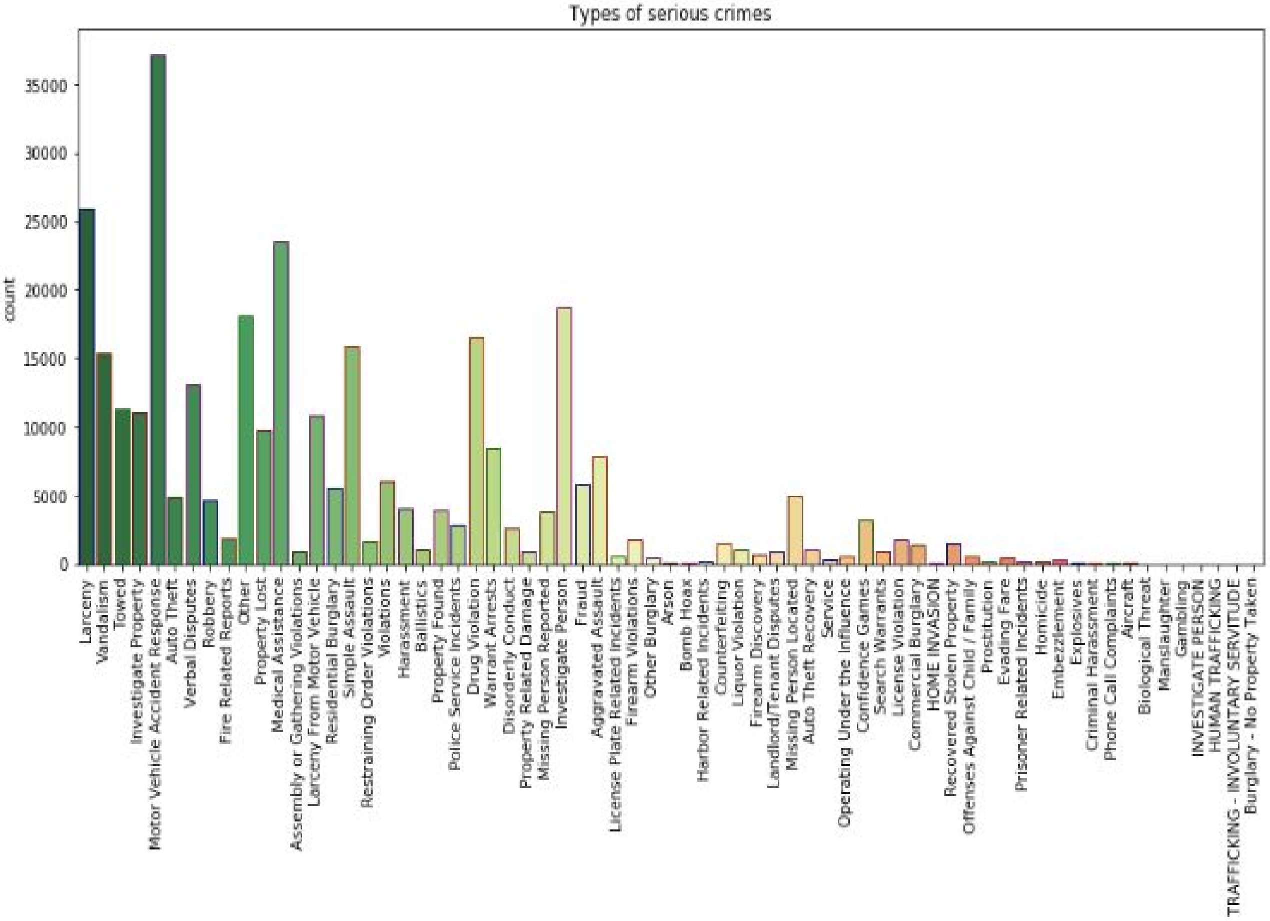


	CMPLNT_NUM	RPT_DT	OFNS_DESC	Latitude	Longitude
0	905573760	02/21/2018	SEX CRIMES	40.625769	-73.991417
1	715536807	12/11/2018	GRAND LARCENY	40.753258	-73.987203
2	688853751	11/14/2018	FRAUDS	40.595810	-73.977374
3	332155393	09/24/2018	SEX CRIMES	40.671107	-73.881433
4	729217700	04/01/2018	HARRASSMENT 2	40.698577	-73.916673

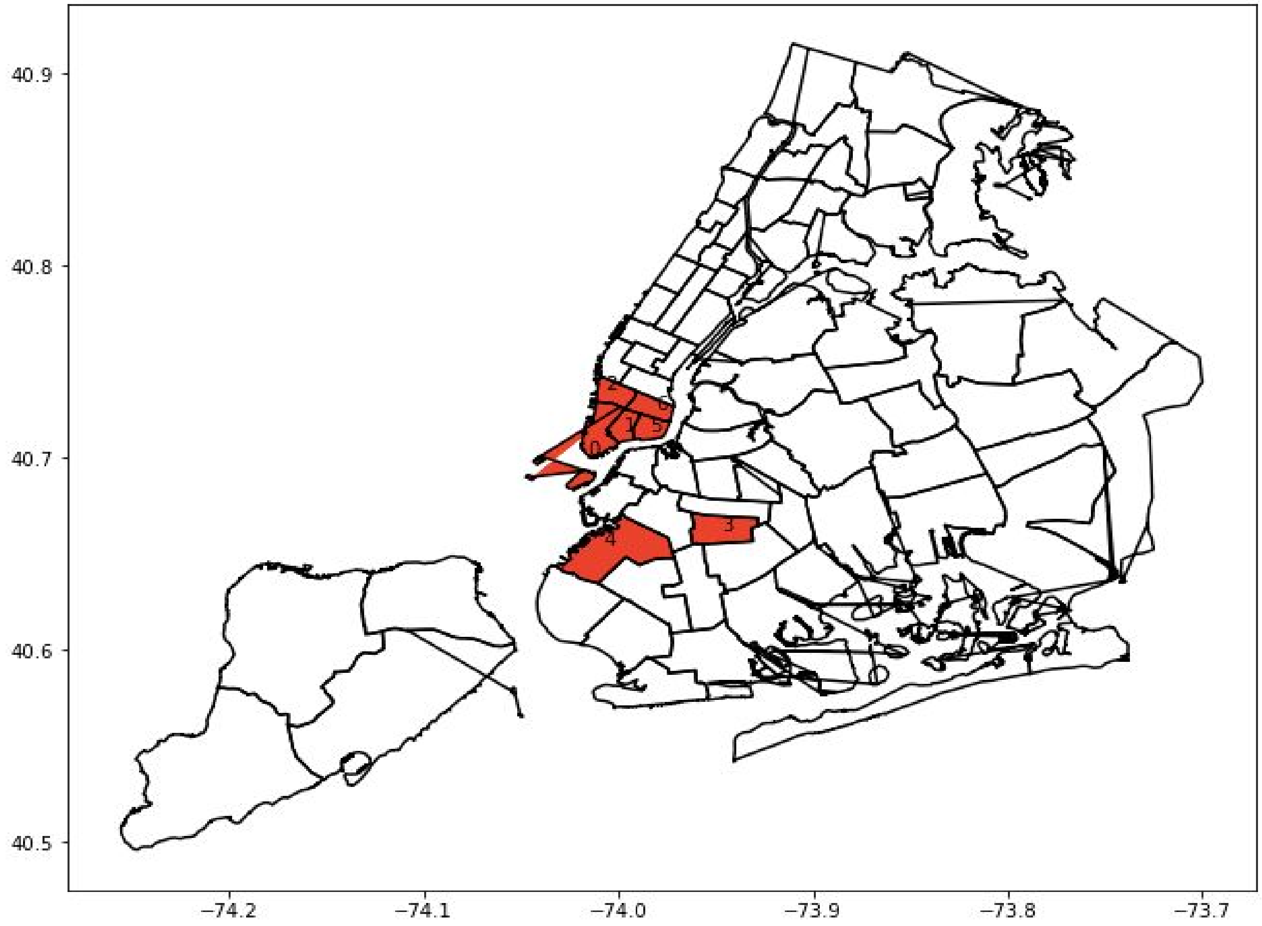
EXPLORATORY DATA ANALYSIS:-

Getting Majority Crime count:-

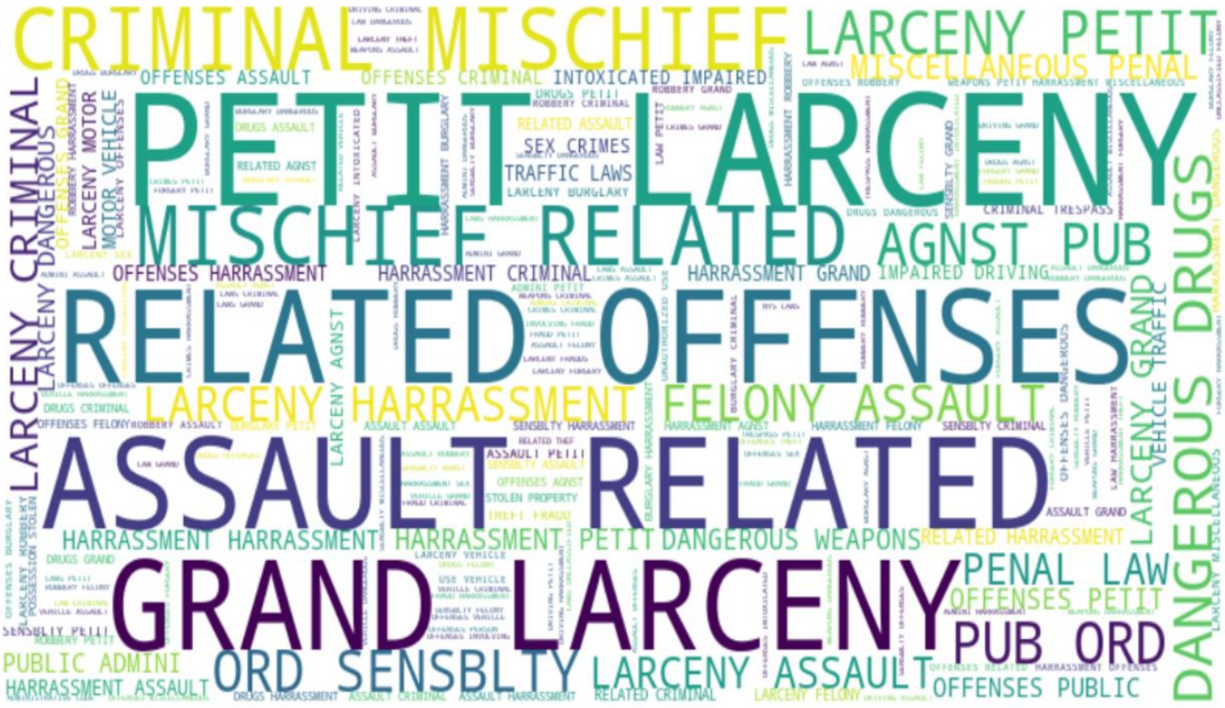
In this we wanted to have an idea about how data was behaving hence we checked crime statistics in the city



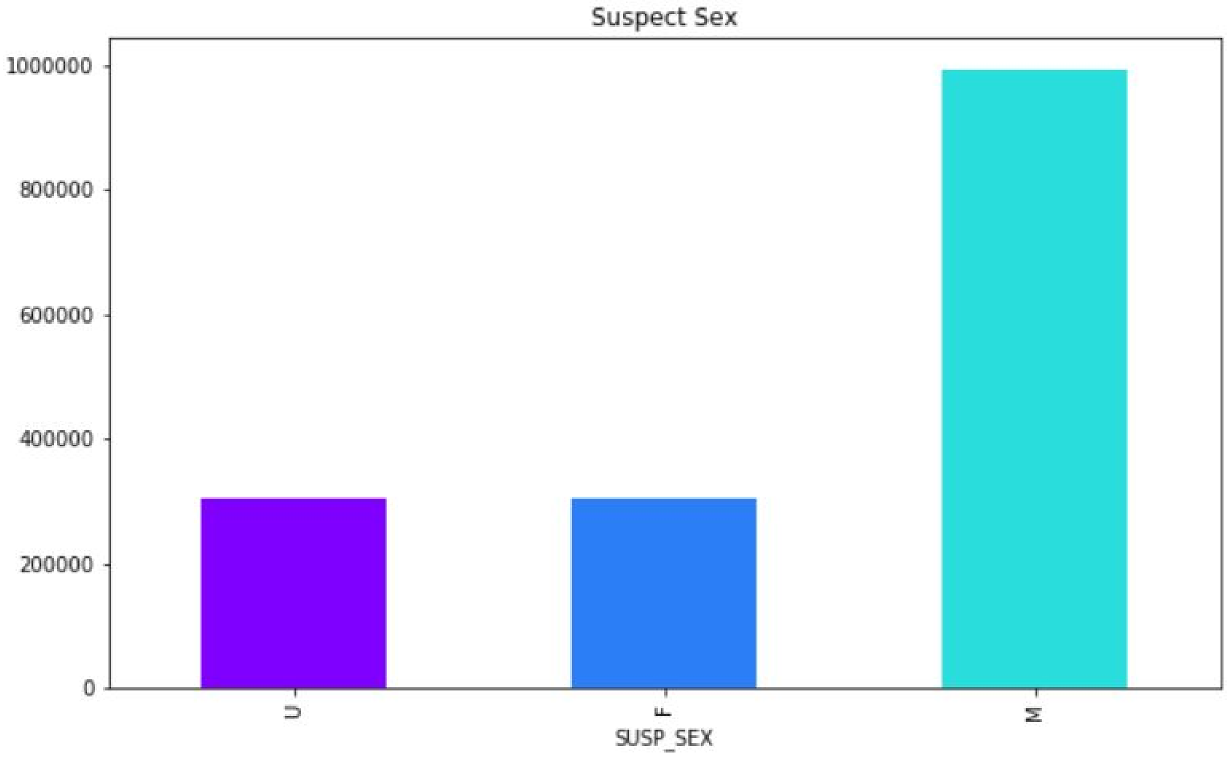
We also used shape file of Police Precinct to get accuracy on the map for fitting latitude and longitude from the nypd crime data hence we implemented it like wise:-



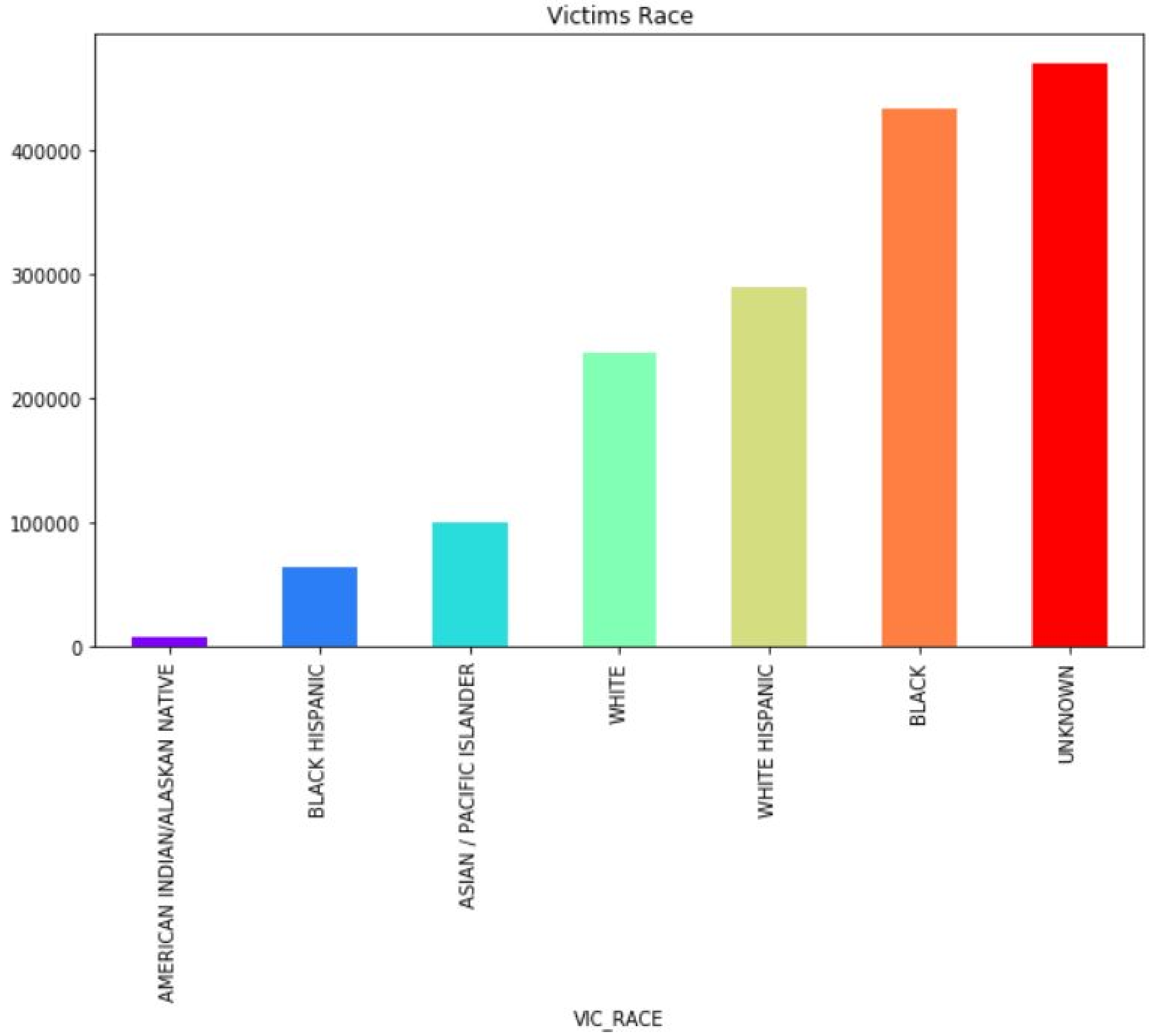
During our exploration process we ran through offense column where we tried to picture the most redundant crime which we have tried to depict through word cloud below:-



Also through the course of project we explored other attributes , like the gender which is at the forefront of execution of these crimes which is explained below in graph:-



Also we implemented the most targeted races during these crimes which is depicted in the graph below:-



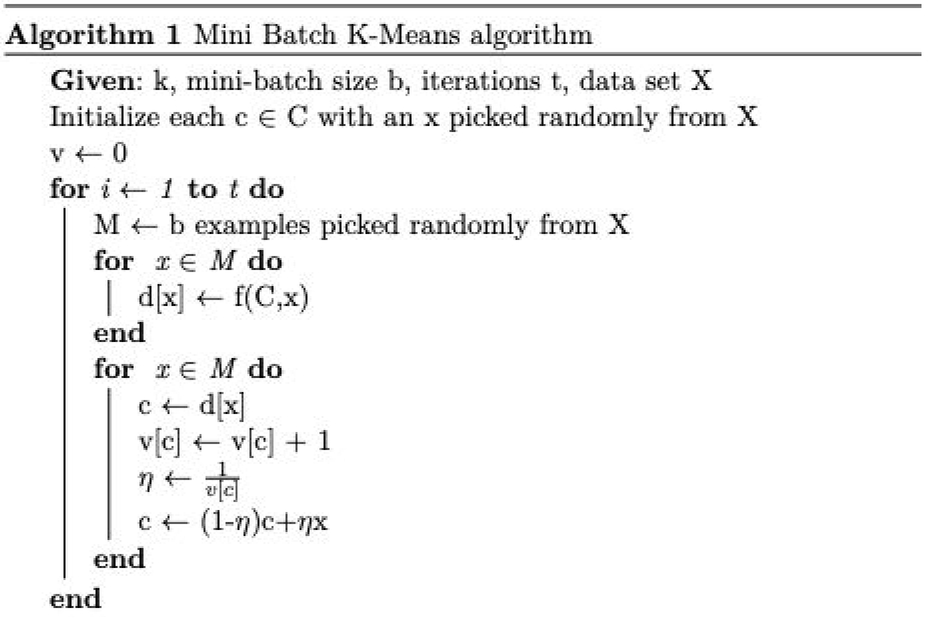
METHODOLOGY:-

In this we have applied 2 sets of K-Means as the modelling algorithm, first is the normal K-Means and the other is mini-batch K-Means and based on time difference i.e. which algorithm implemented the problem faster will be better.

Mini Batch K-Means:-

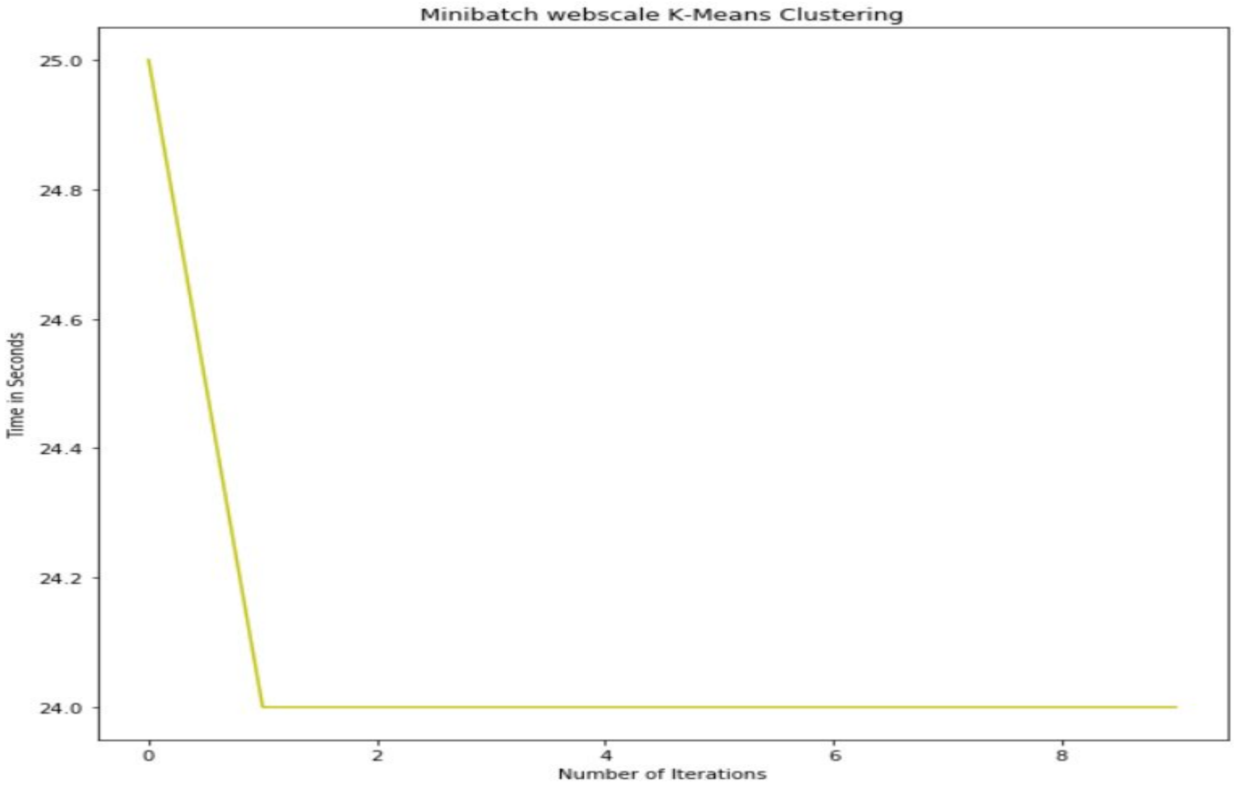
K-means is one of the most used clustering algorithms, mainly because of its good time performance.K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.k-Means minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. Better Euclidean solutions can for example be found using k-medians and k-medoids.

Main idea is to use small random batches of examples of a fixed size so they can be stored in memory. Each iteration a new random sample from the dataset is obtained and used to update the clusters and this is repeated until convergence. Each mini batch updates the clusters using a convex combination of the values of the prototypes and the examples, applying a learning rate that decreases with the number of iterations. This learning rate is the inverse of number of examples assigned to a cluster during the process.



Algorithm 1 Mini Batch K-Means algorithm

```
Given: k, mini-batch size b, iterations t, data set X
Initialize each c ∈ C with an x picked randomly from X
v ← 0
for $i$ ← $1$ to $t$ do
    M ← b examples picked randomly from X
    for $x \in M$ do
        d[x] ← f(C,x)
    end
    for $x \in M$ do
        c ← d[x]
        v[c] ← v[c] + 1
        $\eta \leftarrow \frac{1}{v[c]}$
        c ← (1-$\eta$)c+$\eta$x
    end
end
```

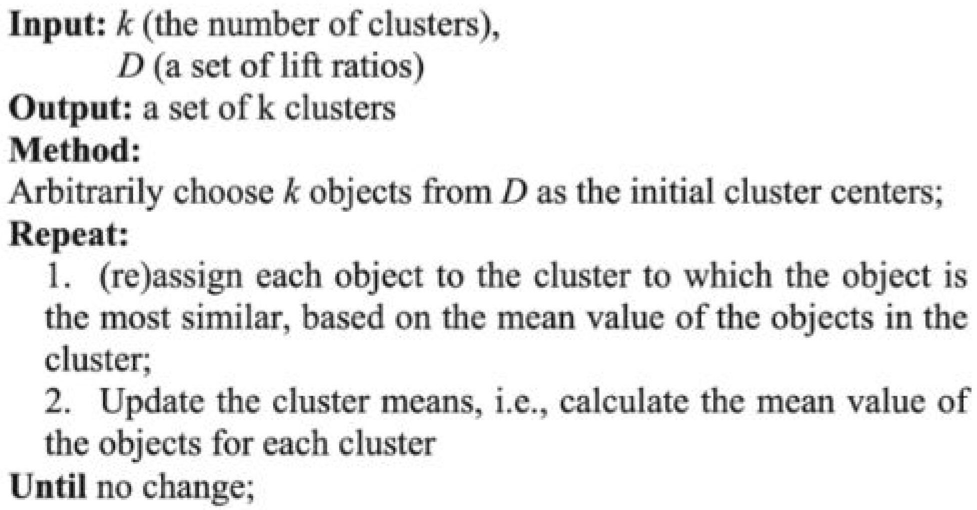


K-Means:-

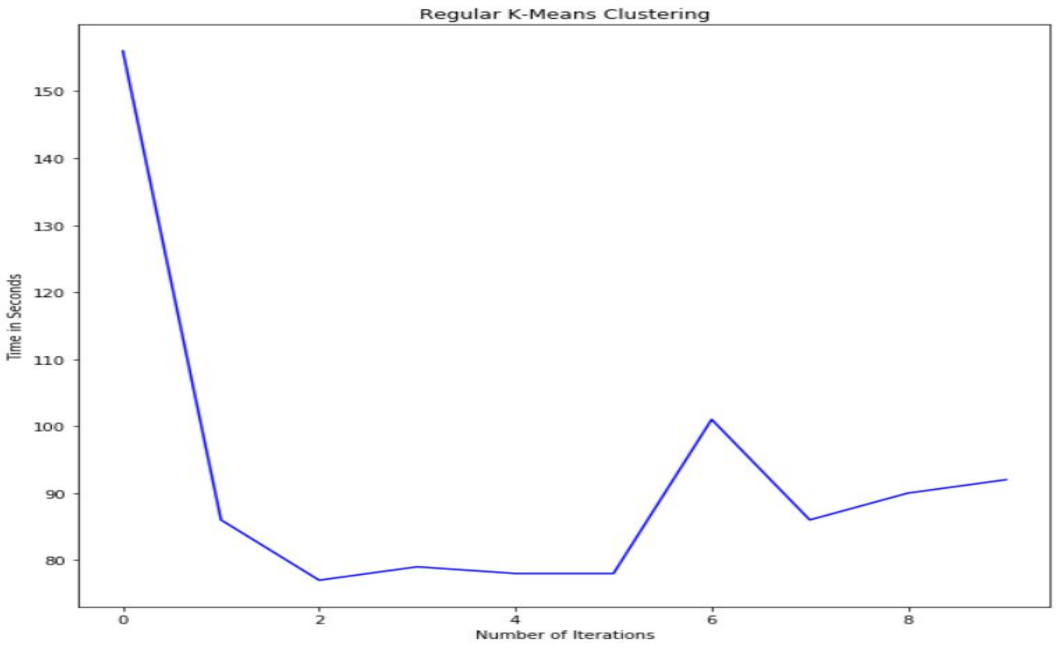
K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.k-Means minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. Better Euclidean solutions can for example be found using k-medians and k-medoids.

The most common algorithm uses an iterative refinement technique. Due to its ubiquity, it is often called "the k-means algorithm"; it is also referred to as Lloyd's algorithm, particularly in the computer science community. It is sometimes also referred to as "naive k-means", because there exist much faster alternatives.
The two common methods are Forgy and Random Partition. The Forgy method which we have used randomly chooses k observations from the crime dataset and uses these as the initial means. The Random Partition method first randomly assigns a cluster to each observation and then proceeds to the update step, thus computing the initial mean of the crime data to be the centroid of the cluster's randomly

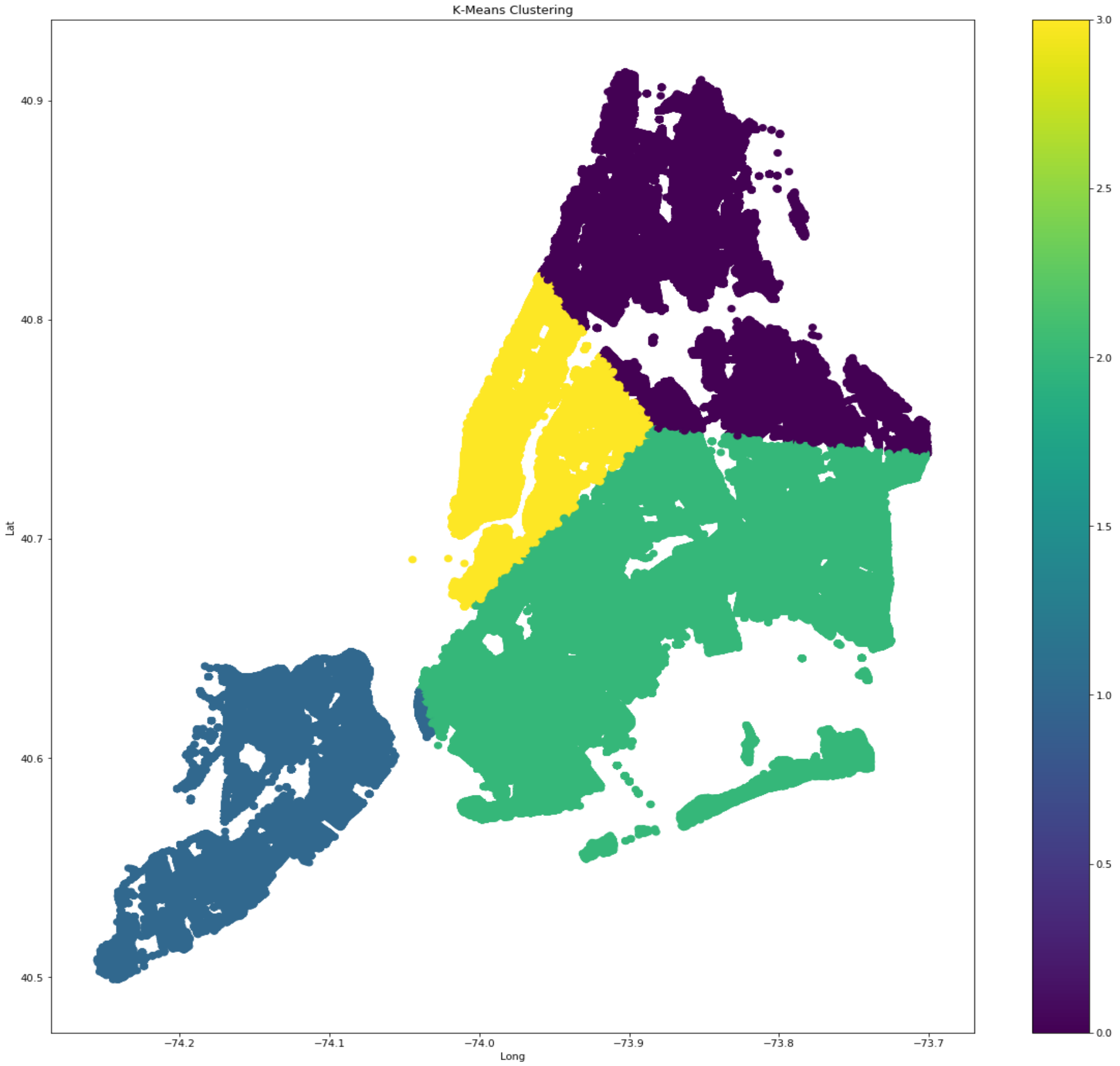
assigned points. The Forgy method tends to spread the initial means out, while Random Partition places all of them close to the center of the data set.

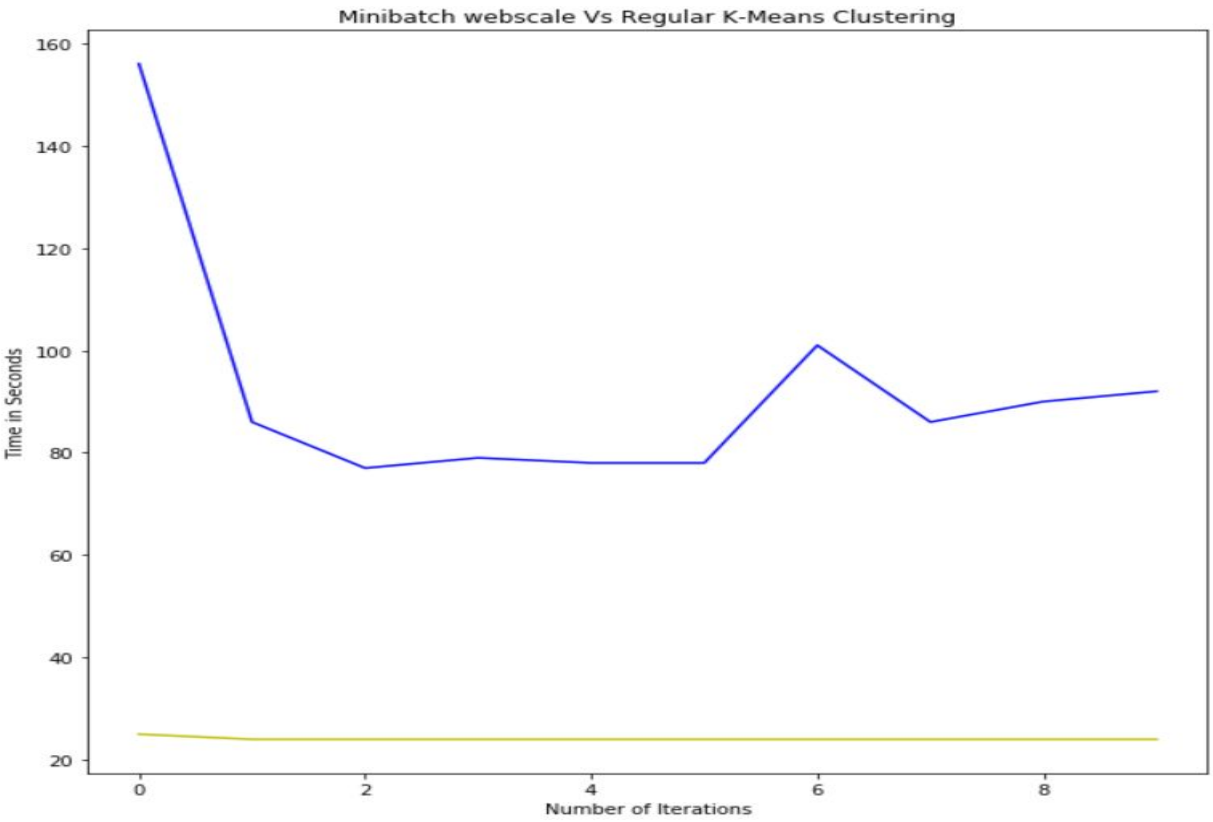


```
Input: k (the number of clusters),
          D (a set of lift ratios)
Output: a set of k clusters
Method:
Arbitrarily choose k objects from D as the initial cluster centers;
Repeat:
   1. (re)assign each object to the cluster to which the object is
   the most similar, based on the mean value of the objects in the
   cluster;
   2. Update the cluster means, i.e., calculate the mean value of
   the objects for each cluster
Until no change;
```



CONCLUSION:-


K-Means Clustering
Lat
Long
40.9
40.8
40.7
40.6
40.5
−74.2
−74.1
−74.0
−73.9
−73.8
−73.7
3.0
2.5
2.0
1.5
1.0
0.5
0.0



1. Mini batch K-means is taking less time as compared to Normal K-means algorithm.We can deduce from the graph above that mini batch k-means is faster and is quite consistent in its execution as compared to regular k-means which is following a haphazard trajectory hence showing its inability to be consistent and quick.
2. Each run of the K-means and mini batch K-means algorithms for a dataset was repeated 10 times and then the result was plotted on the graph.
3. Mini batch k-means has the main advantage of reducing computational cost of finding a partition.
4. Through implementation of these two algorithms we concluded that mini batch through its reduced computational time and the process of implementing itself in mini batches has helped Mini batch K- Means to implement faster.
5. The number of clusters has an important impact in the difference between the partition obtained by k-means and mini batch k-means.
6. The larger the dataset, the better it is for mini batch k-means algorithm to implement than the regular k-means algorithm, because in bigger datasets it will reduce the process time and also clusters formed will be no different than k-means.

REFERENCES:-

1. https://en.wikipedia.org/wiki/K-means_clustering#
2. https://upcommons.upc.edu/bitstream/handle/2117/23414/R13-8.pdf
3. Anil K. Jain. Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8):651–666, 2010.
4. Weiliang Qiu and Harry Joe. Generation of random clusters with specified degree of separation. Journal of Classification, 23(2):315–334, 2006.
5. Hamerly, Greg; Elkan, Charles (2002). "Alternatives to the k-means algorithm that find better clusterings" (PDF). Proceedings of the eleventh international conference on Information and knowledge management (CIKM).
6. https://en.wikipedia.org/wiki/New_York_City_Police_Department
7. https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i
8. https://data.cityofnewyork.us/Public-Safety/Police-Precincts/78dh-3ptz