Implementation of K-Means Algorithm in determining pickup hotspots for Cabs

PROJECT REPORT

Deepu John (20BDS0373)

Naru Venkata Pavan Saish (20BDS0167)

Tulika Garg (20BDS0348)

Vellore Institute of Technology, Vellore

Professor: Dr. Arup Ghosh

Course: Data Mining (CSE3054)

December 2021

Abstract

Drivers for Transport Network Companies need to find the most profitable areas to drive. Cab drivers would like to know what spots in the city would increase their chances of getting a ride request to maximize their earnings.

We can use the K-means clustering algorithm to pinpoint cluster centers within the traffic data. Companies would be able to improve their annual revenue and can expand their business by choosing the right location.

First, we need to collect data from a large cab rides dataset (from Kaggle) and clean it, so that we can perform analysis on it. Then, we would analyze the pickup points that occurred within a given area. From the dataset, we now implement the K-means clustering technique.

First, we need to assign each pickup point to a cluster. “K” in K-means represents the number of clusters. We can determine the optimal number of clusters into which the data may be clustered using the elbow method. Once we assign each pick-up data point to a cluster, we need to rank the clusters based on pickup volume. Then, we have to calculate the distance between each data point and the cluster using the Euclidean distance function. We need to group each pickup point based on how close it is to the cluster. Now we calculate the mean of each cluster. We keep measuring the distance between the centroid and data point, then cluster them using the mean values until there is no change in clustering between two consecutive iterations. Finally, we represent these groups of clusters in graphical format by importing libraries. These graphs help companies identify pickup areas with high traffic.

1. INTRODUCTION

"Passengers" generate the demand, "Drivers" supply the demand and "Cab Aggregator" acts as the marketplace/facilitator to make this all happen seamlessly on a mobile platform.

The cab aggregator business model operates around two major sections of customers – the users and the drivers. Users are people who don't own a car, don't like to drive, or want a cost-effective ride at their doorstep. The driver section includes people who own a car and want to earn some money via it. The success of every service is dependent upon the value it adds to its stakeholders including users and drivers. For all users it is important to have a feeling of safety while booking a ride. Users require the most economical ride in comparison to taxi dispatching services. Also added flexibility like ability to choose in the size of the cab, ability to share a ride and most importantly to be able to use various payment methods like Credit Cards, Google Pay, PayPal etc.

Drivers are the backbone of cab aggregators, and they create value for its drivers to earn their trust and motivate them to drive for the company. Drivers can work at their convenience and have an assured and fast payout system.

In the aggregator business, the key to success is to identify the clusters of Users(Passengers), peak demand time and the availability of Drivers to cater to the demand.

In this project, we intend to find such clusters on a dataset and present it.

1.1 Keywords

K-Means clustering, The Elbow Method

2. LITERATURE REVIEW

A paper on K-Mean algorithm using elbow method was submitted at the 2018 International Seminar on Application for Technology of Information and Communication and added to IEEE Xplore vide INSPEC Accession Number: 18290084.

K-means was introduced by James MacQueen in 1967 [26]. It is observed that a lot of work has been done in this field. In the time frame of 1967 to 1998, all the research work was related to the introduction of K-means in clustering area. After this all the modifications and improvements were started on K-means clustering.

K-means clustering is most widely used clustering algorithm which is used in many areas such as information retrieval, computer vision and pattern recognition. K-means clustering assigns n data points into K clusters so that similar data points can be grouped together. It is an iterative method which assigns each point to the cluster whose centroid is the nearest. Then it again calculates the centroid of these groups by taking its average.

2.1 Data Cleaning

Data that is to be analyzed by data mining techniques can be incomplete (lacking attribute values or certain attributes of interest, or containing only aggregate data), noisy (containing errors, or outlier values which deviate from the expected), and inconsistent. Therefore, once we have obtained our dataset, it needs to be cleaned to remove anomalies. The anomalies usually faced are – Irrelevant data, typos, filter out outliers, drop missing values and scale our data so that it fits within a specific scale.

2.2 K-Means Algorithm

K-Means was used as clustering method, and elbow method used to optimize number of clusters. SSE (Sum Square Error) of each cluster is calculated and compared to optimize number of clusters in the elbow method. The conclusion is the elbow method can be used to optimize number of clusters on K-Mean clustering method.

2.3 Elbow Method

The ‘‘elbow’’ method plots the sum of intra-cluster variances versus number of clusters and pick the optimal number of clusters as the ‘‘elbow’’ point which uses smallest number of clusters to explain most of the variances in the data. In the Elbow method, we are varying the number of clusters from 3 - 11. For each value of K, we calculate WCSS (Within-Cluster Sum of Square).

WCSS is the sum of squared distance between each point and the centroid in a cluster. When we plot the WCSS with the K value, the plot looks like an Elbow. As the number of clusters increases, the WCSS value will start to decrease. WCSS value is large when K = 1. When we analyze the graph, we can see that the graph will rapidly change at a point and thus creating an elbow shape. From this point, the graph starts to move almost parallel to the X-axis. The K value corresponding to that point will be the optimal K value.is the elbow method can be used to optimize number of clusters on K-Mean clustering method.

2.4 K-Means Advantages

- Relatively simple to implement
- Less in complexity
- Scales to large data sets

- K means may be computationally faster than hierarchical clustering for larger no of variables (if K is small)

2.5 K-Means Disadvantages

- Difficult to predict the number of clusters (K-Value)
- Different initial partitions can result in different final clusters
- Clustering data of varying sizes and density.
- Clustering outliers

3. METHODS

To apply the K-Means algorithm the following steps are followed:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third step, if any reassignment occurs, then go to step-4 else stop the algorithm.

3.1 Screenshots from the Code:

```
from scipy.spatial import ConvexHull
Import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
import numpy as np
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
import numpy as np
df = pd.read_csv('Rides.csv')
### BUILD A TWO DIMENSIONS CLUSTER AGAIN ###
# k means
```

```
kmeans = KMeans(n_clusters=3, random_state=0)
df['cluster'] = kmeans.fit_predict(df[['Longitude', 'Latitude']])
# get centroids
centroids = kmeans.cluster_centers_
cen_x = [i[0] for i in centroids]
cen_y = [i[1] for i in centroids]
## add to df
df['cen_x'] = df.cluster.map({0:cen_x[0], 1:cen_x[1], 2:cen_x[2]})
df['cen_y'] = df.cluster.map({0:cen_y[0], 1:cen_y[1], 2:cen_y[2]})
# define and map colors
colors = ['#DF2020', '#81DF20', '#2095DF']
df['c'] = df.cluster.map({0:colors[0], 1:colors[1], 2:colors[2]})

x = df.iloc[:, 4:6]  # 1t for rows and second for columns
kmeans = KMeans(3)
```

```
kmeans.fit(x)
wcss = []
for i in range(1, 7):
  kmeans = KMeans(i)
  kmeans.fit(x)
  wcss_iter = kmeans.inertia_
  wcss.append(wcss_iter)
number_clusters = range(1, 7)
plt.plot(number_clusters, wcss)
plt.title('The Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
```

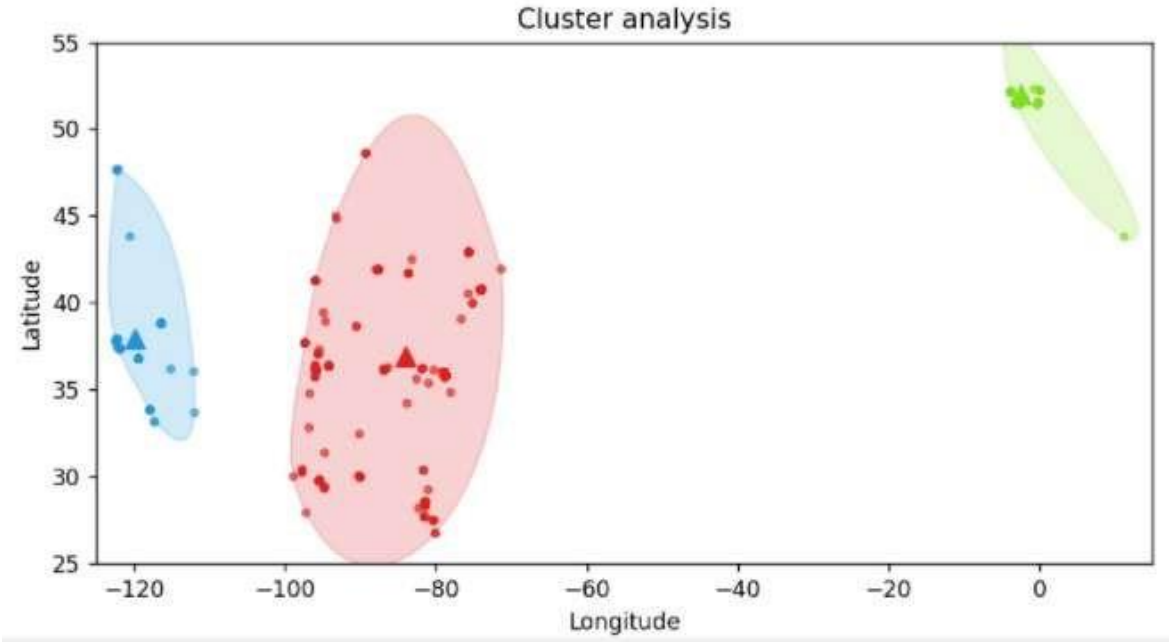
```
from scipy import interpolate
fig, ax = plt.subplots(1, figsize=(8,8))
plt.scatter(df.Longitude, df.Latitude, c=df.c, alpha = 0.6, s=10)
plt.scatter(cen_x, cen_y, marker='^', c=colors, s=70)

for i in df.cluster.unique():
    # get the convex hull
    points = df[df.cluster == i][['Longitude', 'Latitude']].values
    hull = ConvexHull(points)
    x_hull = np.append(points[hull.vertices,0],
                       points[hull.vertices,0][0])
    y_hull = np.append(points[hull.vertices,1],
                       points[hull.vertices,1][0])
    # interpolate
```

```
    dist = np.sqrt((x_hull[:-1]-x_hull[1:])**2+(y_hull[:-1]-y_hull[1:])**2)
    dist_along = np.concatenate(([0], dist.cumsum()))
    spline, u = interpolate.splprep([x_hull, y_hull],
                                    u=dist_along, s=0)
    interp_d = np.linspace(dist_along[0], dist_along[-1], 50)
    interp_x, interp_y = interpolate.splev(interp_d, spline)
    # plot shape
    plt.fill(interp_x, interp_y, '--', c=colors[i], alpha=0.2)
# plt.xlim(0,200)
# plt.ylim(0,200)
z=1
p=1
```

```
plt.ylim(25*z, 55*z)
plt.xlim(-125*p, 15*p)
plt.title('Cluster analysis')
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.show()
```

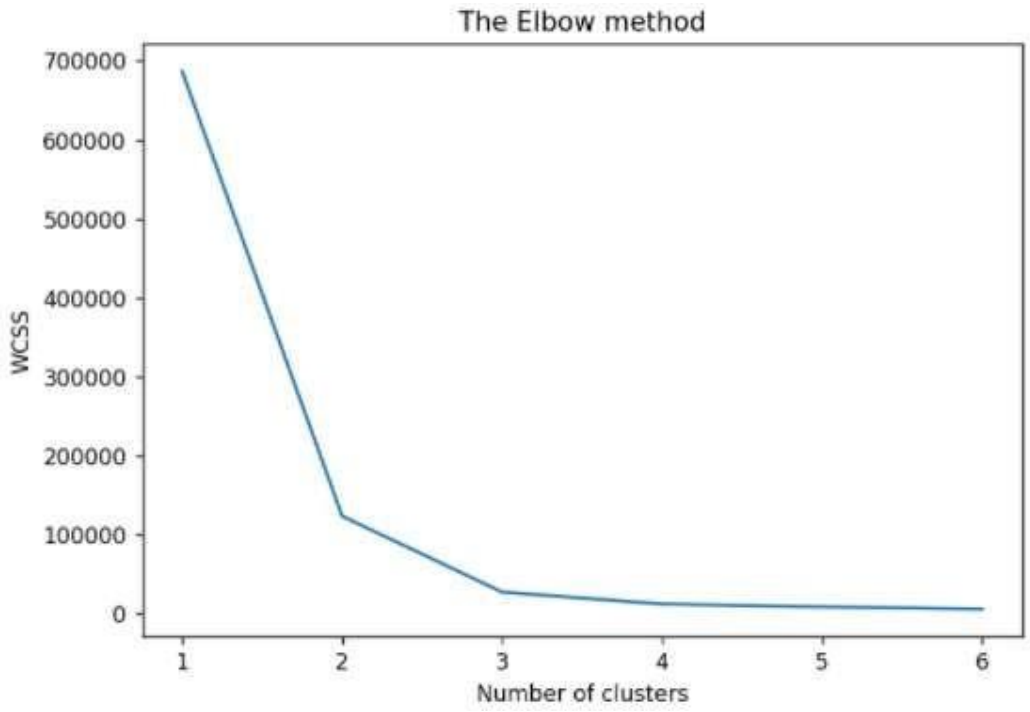
3.2 Output obtained when the Code is Run on the Data



3.3 Choosing K using Elbow method

In the Elbow method, we are actually varying the number of clusters (K) from 1 – 7. For each value of K, we are calculating WCSS (Within-Cluster Sum of Square). WCSS is the sum of squared distance between each point and the centroid in a cluster. When we plot the WCSS with the K value, the plot looks like an Elbow. The K value corresponding to the point where the bend occurs, is the optimal K value.

In the graph we obtained, the bend occurs as K = 3. Hence, that is the optimum number of clusters.



4. CONCLUSION

The above code helps us to find the hotspots by implementing K Means algorithm and elbow method. From the geographic data points (latitudes and longitudes), we clustered them with the help of elbow method and then implemented the K means algorithm to find the hotspots.

5. REFERENCES

Han, J., Pei, J., and Kamber, M. (2011). Data mining: concepts and techniques. Elsevier

Data Mining Practical Machine Learning Tools and Techniques Third Edition, Ian H. Witten Eibe Frank Mark A. Hall

2018 International Seminar on Application for Technology of Information and Communication and added to IEEE Xplore.

Introduction to Data Mining with Case Studies, G.K. Gupta.