Final Project Report

Geo-Location Clustering using the K-Means Algorithm

By

Afshar Mohammed

Under guidance of Vahid Behzadan

Motivation

The main objective of the project is to perform geo location clustering using K-means Algorithm. Clustering refers to grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups. I think that it can be used in market analysis of the organizations as well as it could be helpful for the broadcasting the business depending up on the location as well.

Approach

- Creating the S3 bucket in Aws
- Creating the spark context object and use it to load the data from the devicestatus.txt
- Applied some delimiters on my data by using .split() function
- Reading the text file using sc.TextFile() method
- Filtered entries where latitude or longitude are equal to 0
- Created the Dataframe using spark.createDataFrame

- I have saved the parsed data using save as TextFile () method and saved it in the bucket ffor the further reference
- Finally, I plotted between the Latitude and Longitude by using Matplotlib library
- Implementing the Kmeans Algorithm and determining the Kvalues as 2 which means the number of clusters
- Calculating the Eucledian distance and greater Circular distance
- Finally, I made a plot between the latitude and longitude

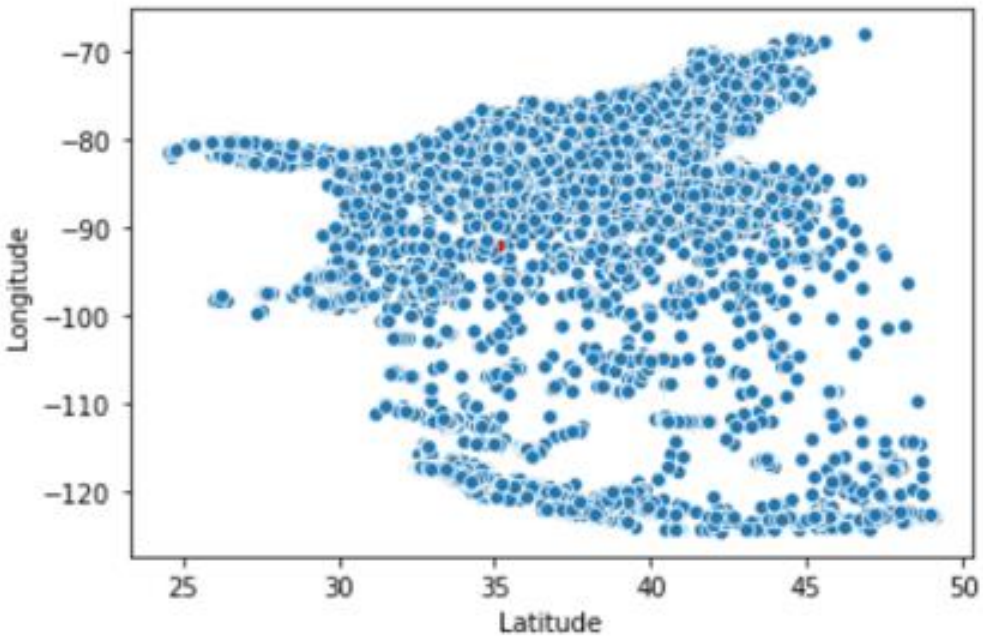
System Configuration:

Created a Key Value pair and downloaded a .pem file.

Spark EMR cluster m4.xlarge with 3 nodes 1 Master and 2 slave nodes.

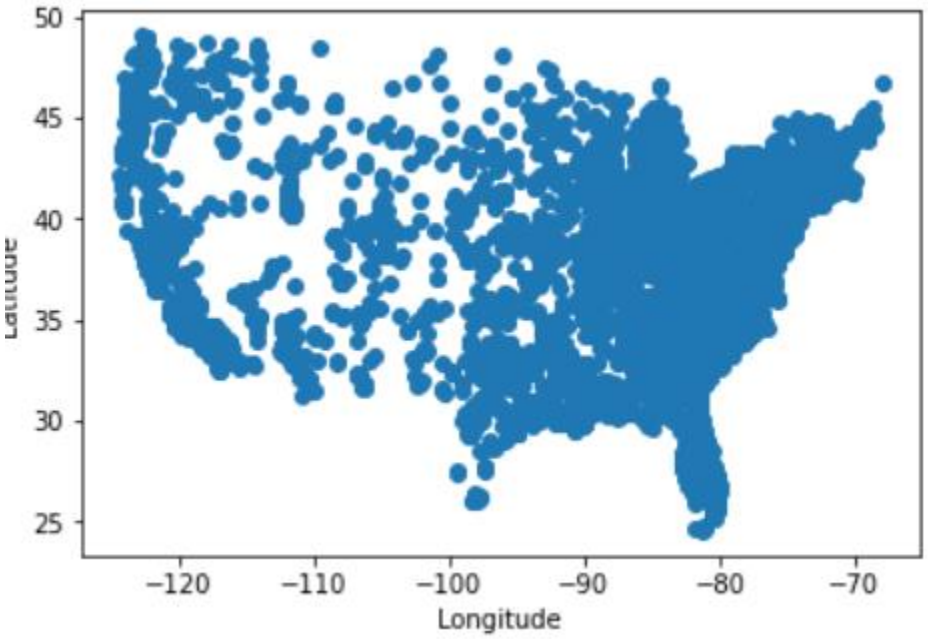
Visualizing the (latitude, longitude) pairs of the device location data

Uploaded the Device status.txt on the S3 Bucket and performing the



Analyzing and Visualizing synthetic location data

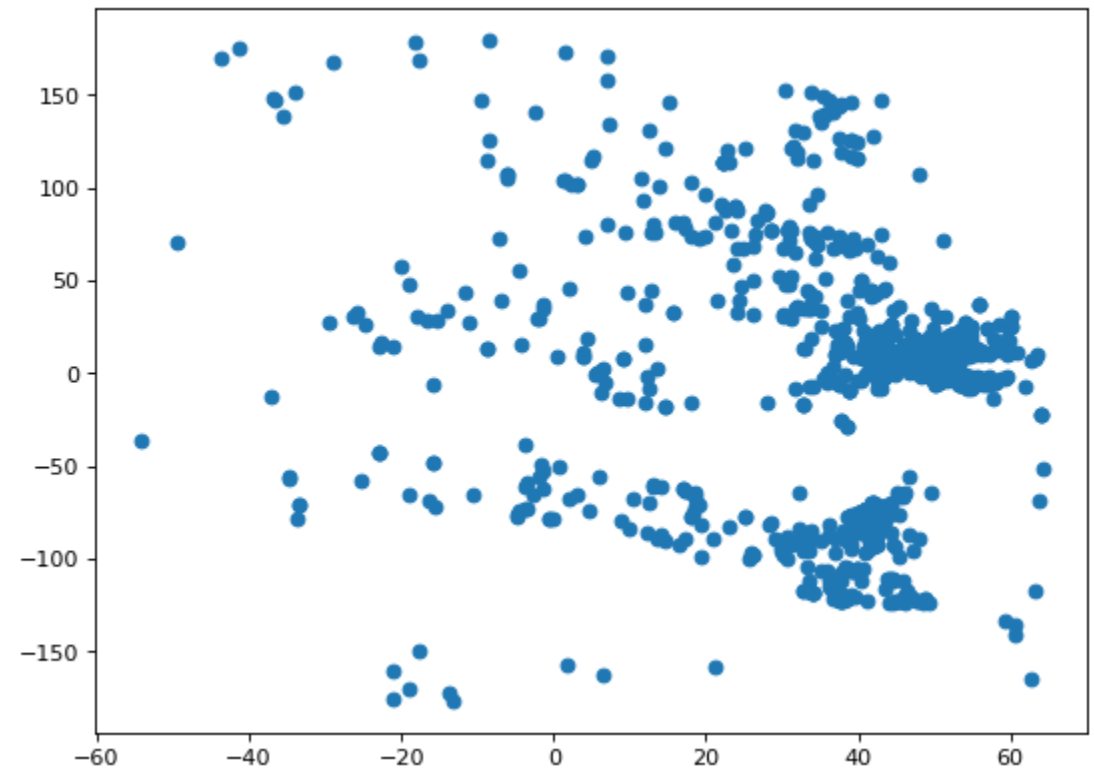
Uploaded samplegeo.txt in to the S3 bucket analyzing using the same methods as above



Preprocessing DBpedia location data:

Uploading Lat_long .txt and plotting between the latitude and longitude

```
<matplotlib.collections.PathCollection at 0x7f4fcf37fe48>
```



.

Building the Model

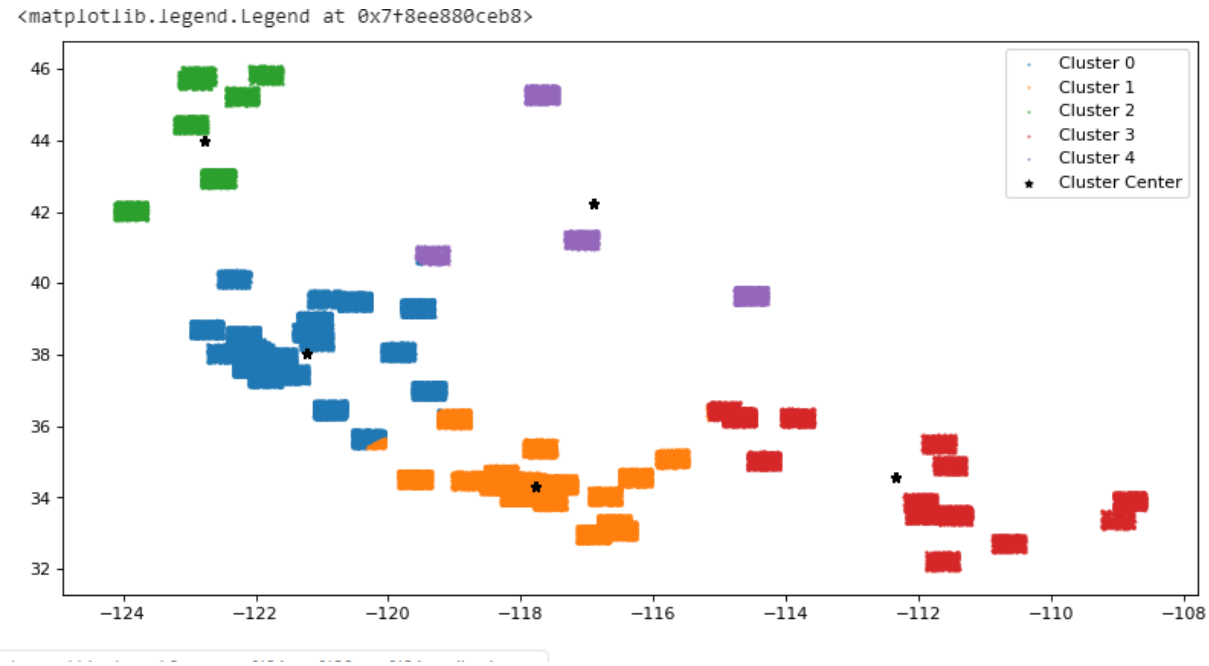
Knn model is build for clustering the geo locations and assigning the K values as 2,4,6 as the number of clusters on the locations of the data

```
centers = model.clusterCenters()
print("Cluster Centers: ")
for center in centers:
    print(center)
print("--- %s seconds ---" % (time.time() - start_time))
timetable.append((time.time() - start_time))
```

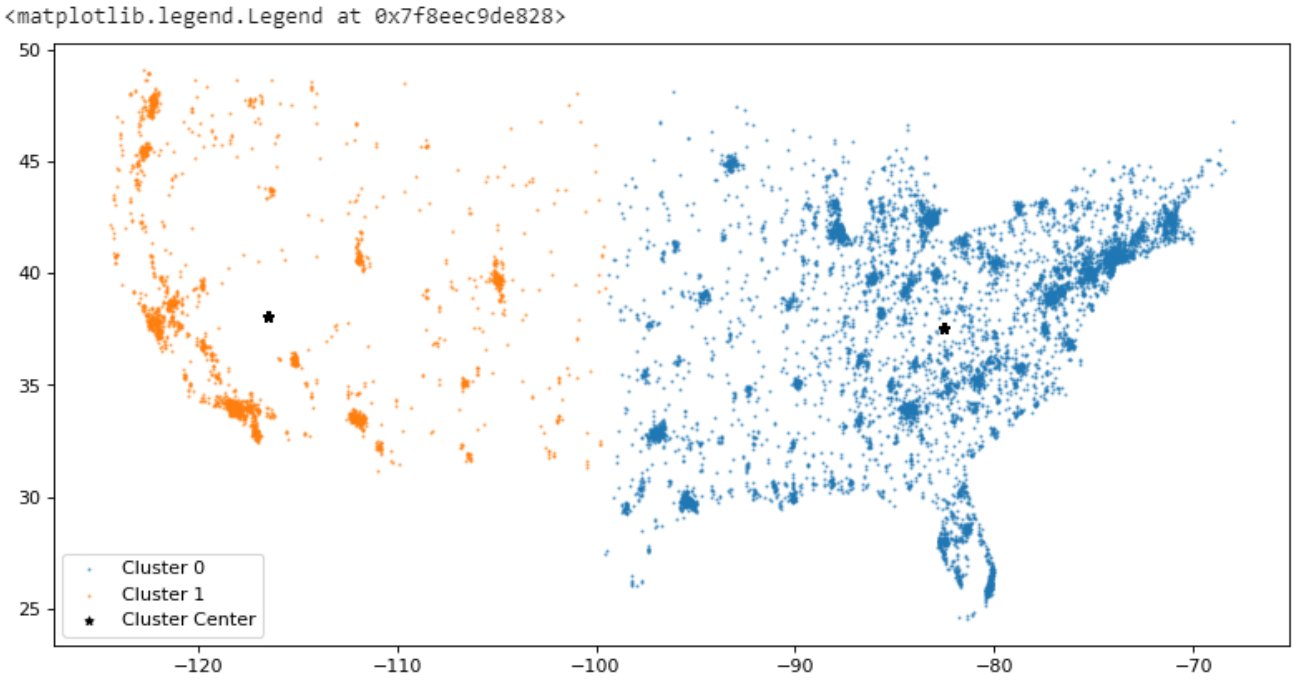
```
Silhouette with squared euclidean distance = 0.7779851895575357
Cluster Centers:
[  38.02864791 -121.23352192]
[  34.29718423 -117.78653245]
[  43.98989868 -122.77665336]
[  34.58818551 -112.35533553]
[  42.25924472 -116.90267328]
--- 13.201755285263062 seconds ---
```

Kmeans on Device location data

I build a knn model using the k value as 5 on the device location data.and plotted the clusters as shown below

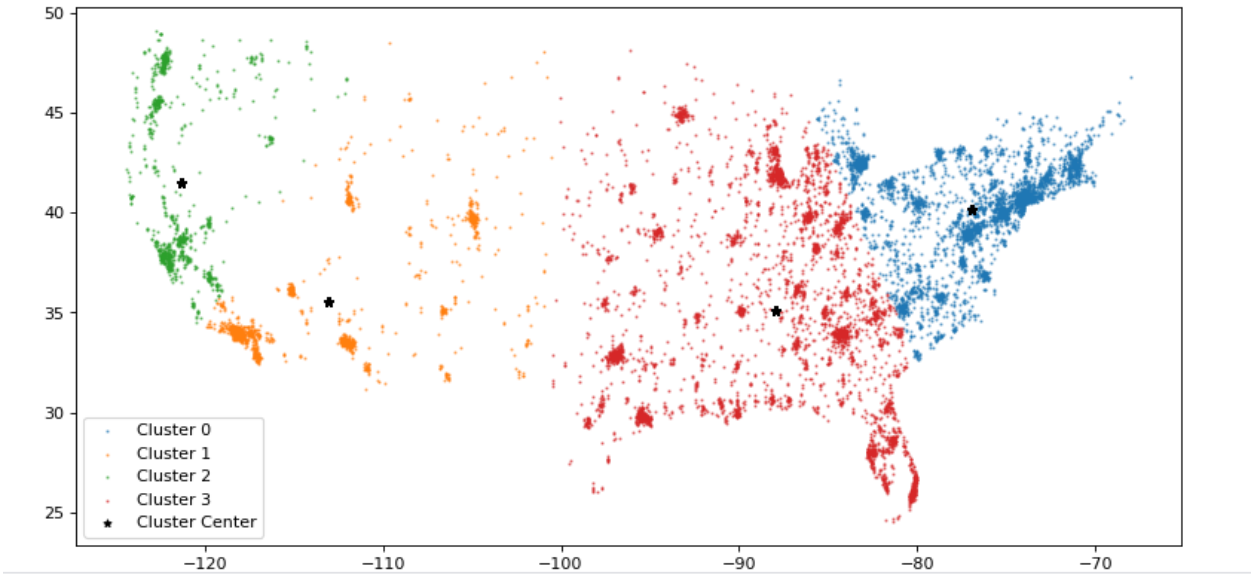


Analyzing and Visualizing k-means clusters for the synthetic location data using k = 2



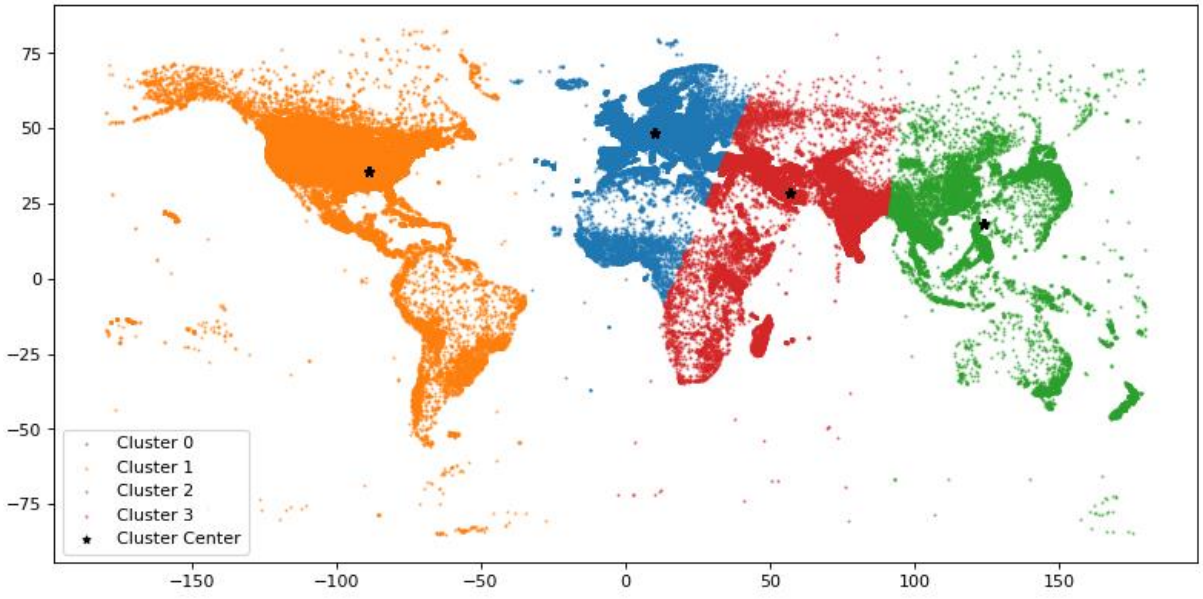
K=4

<matplotlib.legend.Legend at 0x7f8eea122e80>

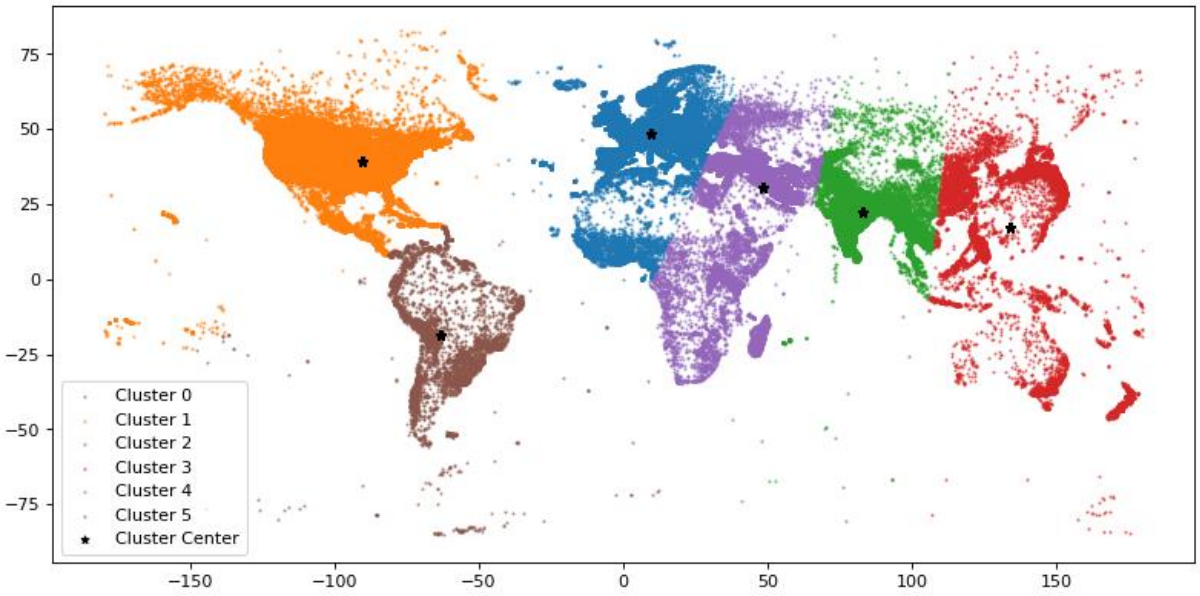


k-means clusters for the large-scale DBpedia location data.

K=4



K=6



Results:

Clusters

```
Cluster Centers:
[  38.02864791 -121.23352192]
[  34.29718423 -117.78653245]
[  43.98989868 -122.77665336]
[  34.58818551 -112.35533553]
[  42.25924472 -116.90267328]
```

Predictions by eucleadian distance and great circle distance

```
predictions_df_with_gcd.show(3)
```

```
+----------------+-----------------+----------+--------------+---------------+----------------+------------------+
|original_latitude|original_longitude|prediction|center_latitude|center_longitude|         gc_dist|           eu_dist|
+----------------+-----------------+----------+--------------+---------------+----------------+------------------+
|       33.689476|       -117.543304|         1|     34.297184|      -117.78653|35.59862841593989|0.42846743379777763|
|        37.43211|        -121.48503|         0|      38.02865|      -121.23352|34.96130983668151|0.41911582615284715|
|        39.43789|        -120.93898|         0|      38.02865|      -121.23352|79.38456955250766| 2.0727134647313505|
+----------------+-----------------+----------+--------------+---------------+----------------+------------------+
```

Runtime Analysis

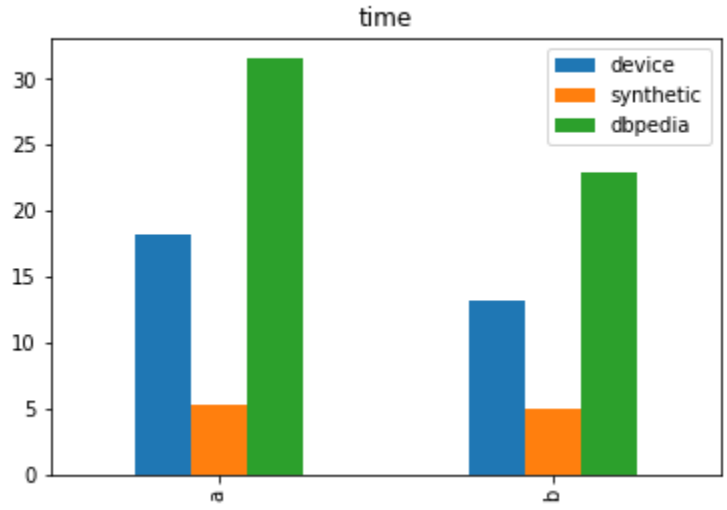
Compared the runtime of k-means implementation using the K value for all three datasets using the local mode with at least two threads

```
times=[[18.174409866333008, 5.210770845413208, 31.525365352630615],[13.201853036880493, 4.9684789180755615, 22.87245488166809]]
```

```
import pandas as pd
df=pd.DataFrame(times,columns=['device','synthetic','dbpedia'],index=["a","b"])
```

```
import matplotlib.pyplot as plt
df.plot(kind='bar',title='time')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f81115cdc18>
```



Conclusion:

Geolocation apps that run on mobile devices provide a richer experience than those that run on desktop PCs because the relevant data you send and receive changes as your location changes. . When a GPS signal is unavailable, geolocation apps can use information from cell towers to triangulate your approximate position, a method that isn't as accurate as GPS .So, in this project we introduced clusters for all continents in the world. I believe that geographical data is helpful for the people to know the routes easily.