DISTRIBUTED AND SCALABLE DATA ENGINEERING

DSCI - 6007

Final Project Report

Geo-location Clustering using the k-means Algorithm

Motivation:

This project main objective is to find the common preference and similarities among the users and group them to improve their experiences based on their content or finding spatial clusters using their location. The main motivation here is that when we group people then we can develop different way to gain the market in that locations by studying them and this will help in growing the business and understanding the customer needs based on grouping them together.

System Configuration :

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

Approach:

Step 1 :

Creating S3 buckets:

First create a bucket and upload all the data files

Parsing the data:

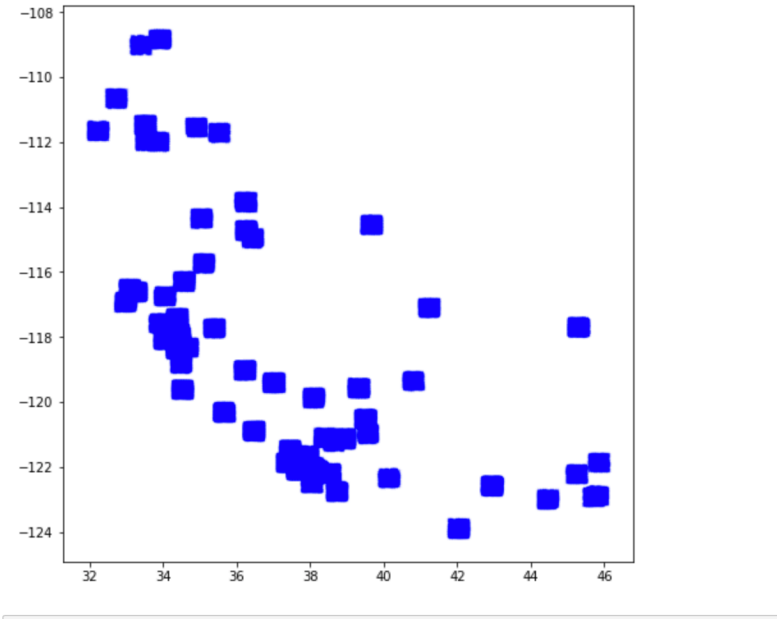
- Here we have three different data files and we have use multiple delimiters to import the data in a clean format.
-
- I have defined a function to read the text data and split them using the delimiter and I have defined the datatypes for all the columns with the help of .map() function.
- Then I have casted my input data Into their exact datatype
- We need to take care of latitude and longitude data types to be in float
- Then I have filtered the data where the latitude and longitude are “0”
- Then I have saved the parsed data using SaveAsTextFile() and saved it in the bucket for further purpose using comma as delimiter.
- Here we need to be very careful while parsing the data because each data file has different formats and different delimiters.
- Finally I have stored the parsed data in new folder in s3 bucket which will be used to build the K means model with k values of 2,4,5,6 as specified in the instructions.

- Then we have converted the spark data frame into pandas data frame from visualization purpose using topandas() method.

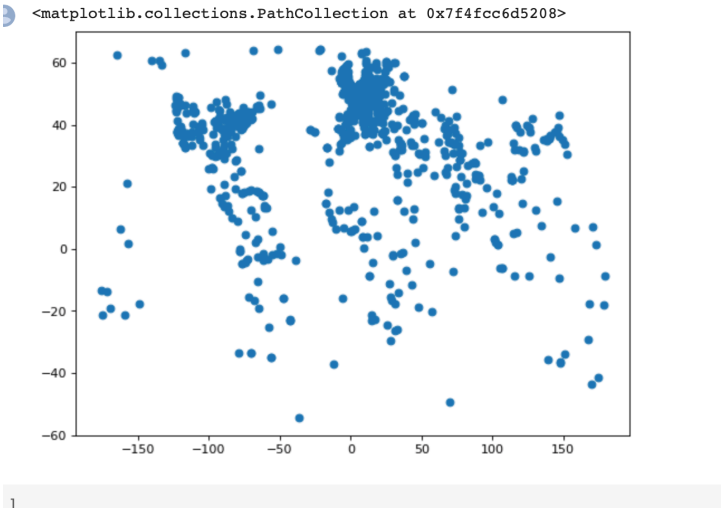
Step 2 :

Analyzing the Data By using the third party libraries like matplot and seaborn:

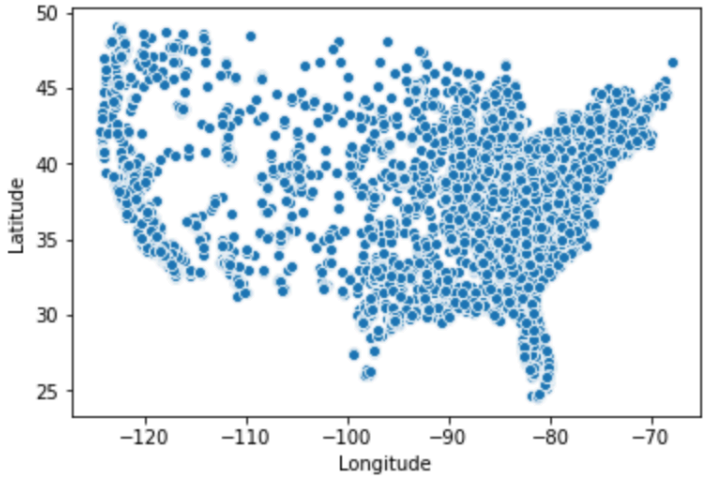
Analyzing device status text file and plotting latitude and longitude.



Analyzing the lat long text file



Analyzing sample_geo text file



Step 3 :

Building K-Means Model:

Reference : https://spark.apache.org/docs/latest/ml-clustering.html
In every iteration of the algorithm, each data point is assigned to its nearest cluster based on some distance metric, which is usually it is great circle distance as we are dealing with geo locationIn a nutshell, *k*- means groups the data by minimizing the sum of squared distances between the data points and their respective closest centroid.
With help from the above resources I have built the k means model for data here we have used the number of clusters = “2,4,5,6” the sample shown below is here for k=5 and built three models using different clusters

```
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]
--- 12.97334909439087 seconds ---
```

Step 4 :

The next step is calculating the Eucledian and Great Circle Distance for the points:

Reference : https://gist.github.com/pavlov99/bd265be244f8a84e291e96c5656ceb5c

https://medium.com/@nikolasbielski/using-a-custom-udf-in-pyspark-to-compute-haversine-distances-d877b77b4b18

https://stackoverflow.com/questions/4913349/haversine-formula-in-python-bearing-and-distance-between-two-gps-points

You can see that we have calculated the Eucledian and Great circle Distance between the points.This is the most challenging part of the project.

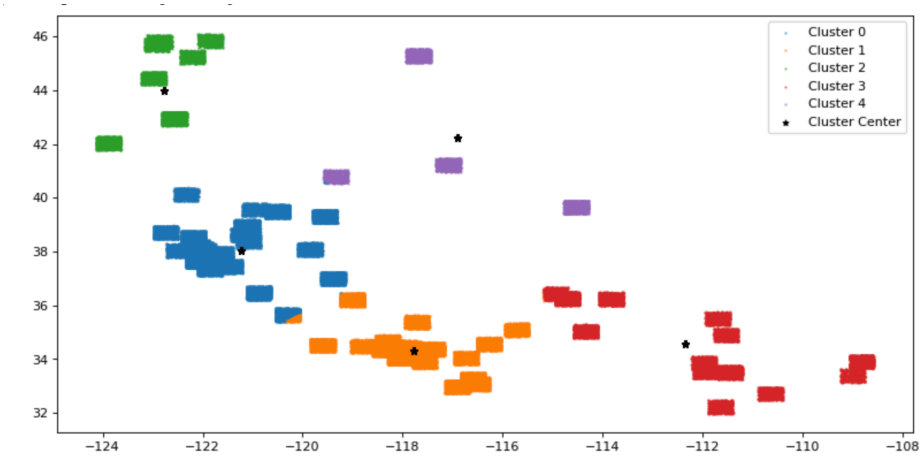
```
+-----------------+------------------+----------+---------------+----------------+-----------------+-------------------+
|original_latitude|original_longitude|prediction|center_latitude|center_longitude|          gc_dist|            eu_dist|
+-----------------+------------------+----------+---------------+----------------+-----------------+-------------------+
|        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|
+-----------------+------------------+----------+---------------+----------------+-----------------+-------------------+
only showing top 3 rows
```

With the help from above link I have figure it how to calculate the distance between two points

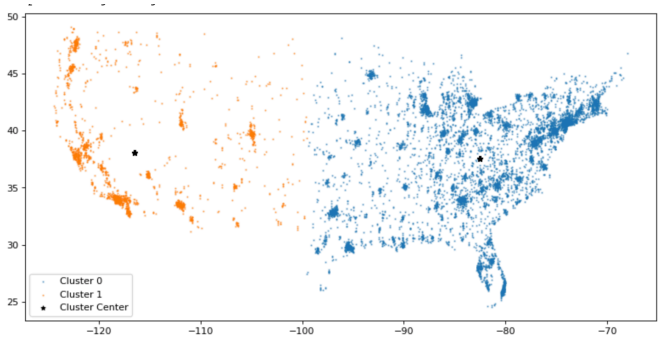
Step 5:

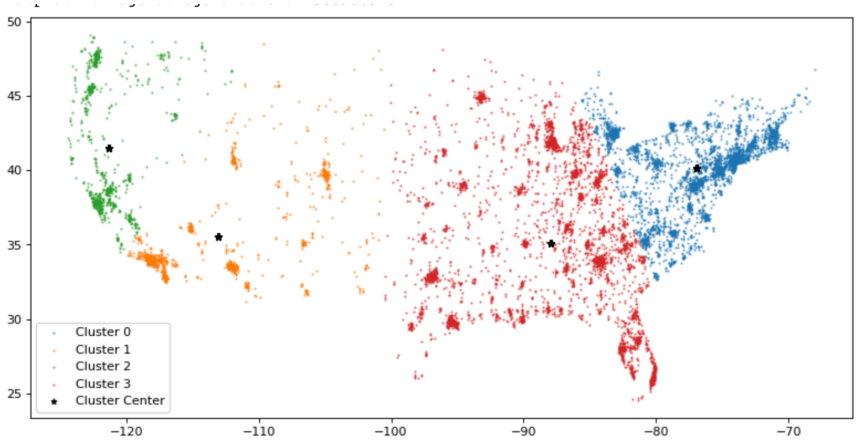
Cluster visualization:

Device status when K = 5

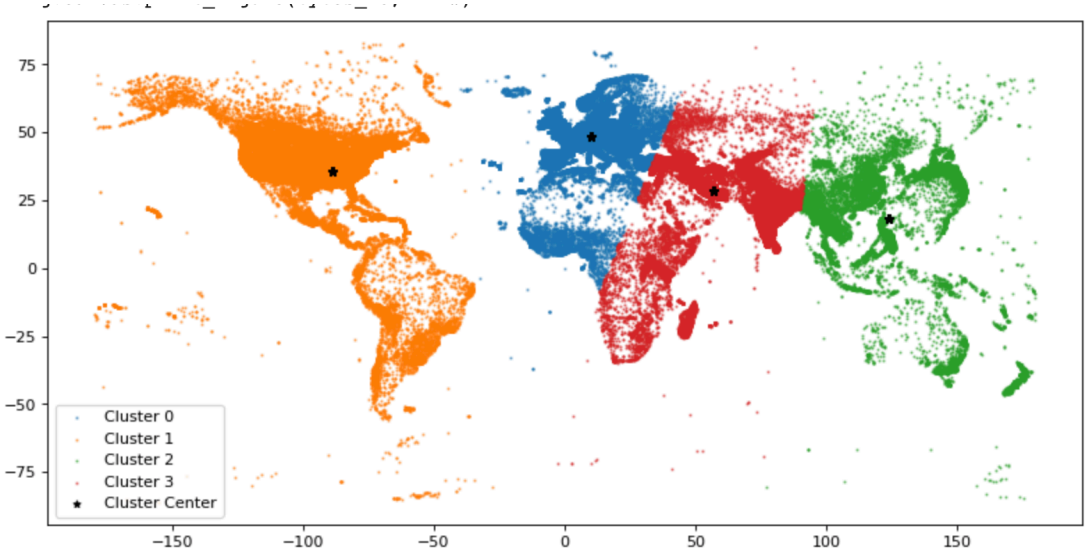


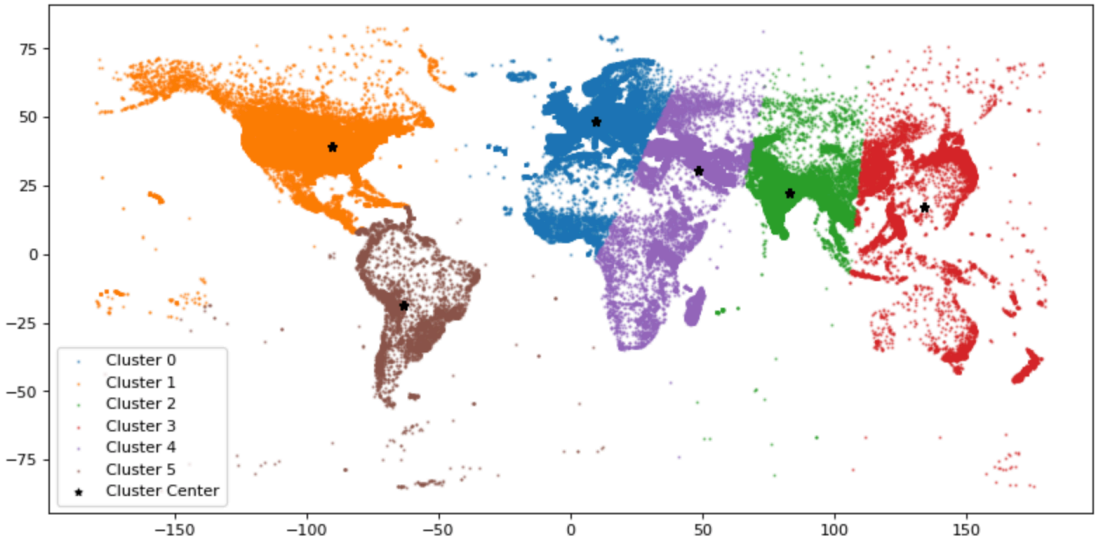
Synthetic location data where K = 2 , 4





Dbpedia location data where K = 4 , 6



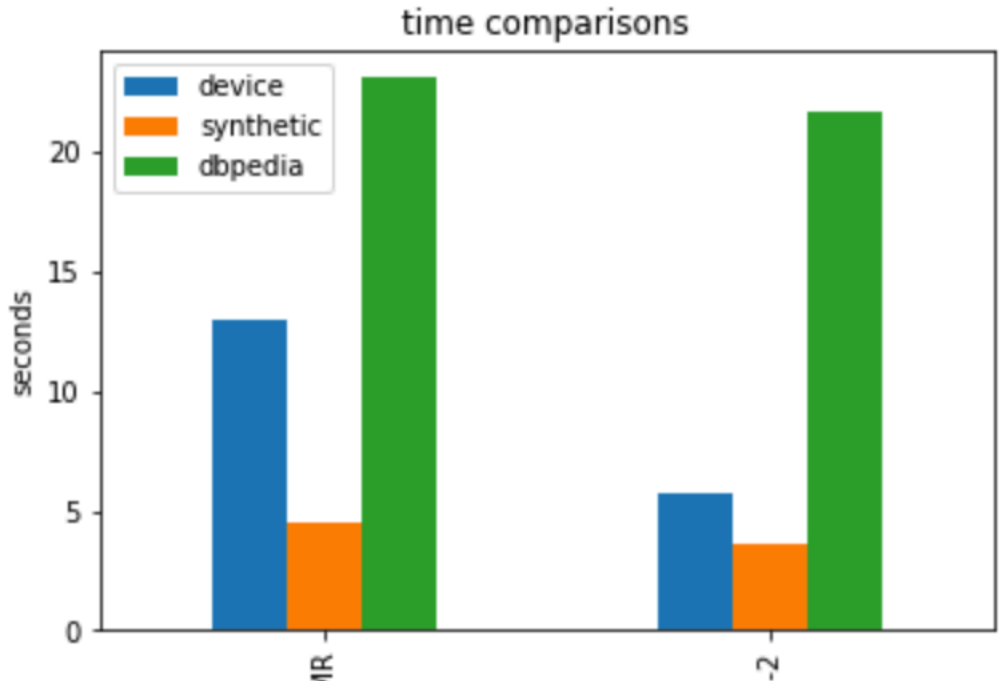


Step6:

Runtime Analysis:

Here we have create two models one in EMR and other one in local mode with two threads the above plot shows the compile time for all the three models where k=5 k=2,6 k= 2,4.

With different method one without using persistent RDD and local mode with two threads



Final Conclusion:

From the above analysis of all the three files when we see the data like sites they have listed the sites and from this we can understand what site the user are interested and we can group them according to it and we can also analyze them based on this we can use what they are interested and helps the industries to focus on the needs to improve the business. With help of the clustering algorithm we can group the users based on pattern and in future if we have more amount of data we can use this in different domain to solve the problems for example we also have the details like service provider in other data file from this we can see the area where the service providers are good and areas where they need to improve.