DSCI-6007

Student Name:Nikhitha Nagirimadugu

Geo-Location Clustering using the k-means algorithm

Introduction:

Geolocation is a geographical location i.e.,Latitudinal and Longitudinal location of any electronic device used to access internet. Geolocation is widely used to group customers based on preferences for commercial companies, police can use geolocation for safety procedures and tracking down cyber criminals, Owners of food company can use it make a decision on potential expansion based on how many users search their website from which location.

we can use clustering concept of machine learning to find and group the customer's with similar preferences. Through clustering we divide the data points into some n groups. Data points of same group should be similar and data points of different groups should be different. K-means is one of the popular algorighms that can help us accomplish Clustering.

Data Used:

Three datasets given to accomplish this project are:

- devicestatus.txt
- sample_geo.txt
- lat_longs.txt

Data Preparation:

Initially the textfiles are uploaded to a S3 bucket on Amazon web Services(AWS)

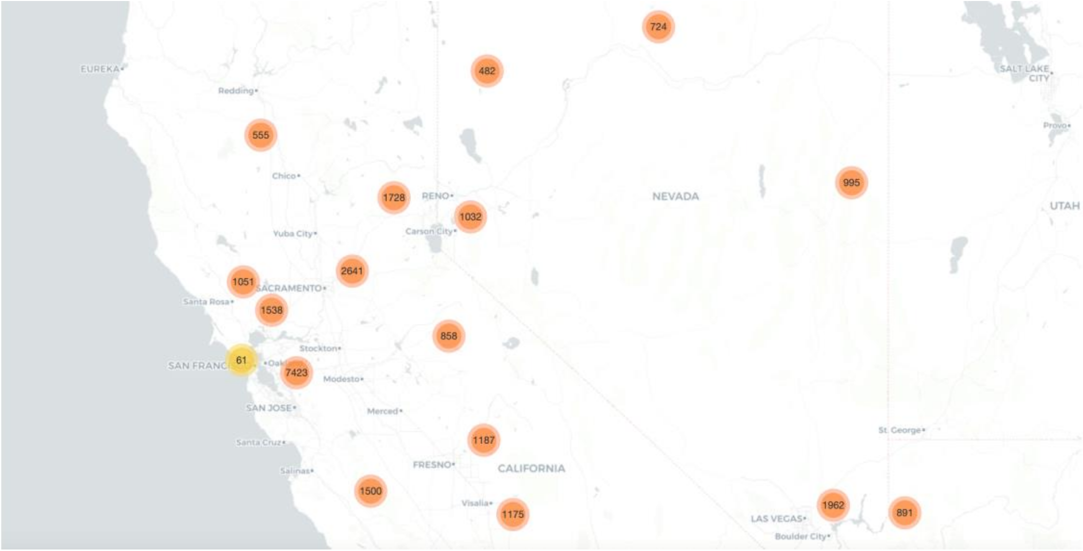
The Data preprocessing done on devicestatus.txt are as below:

- Loading the dataset from S3 bucket.

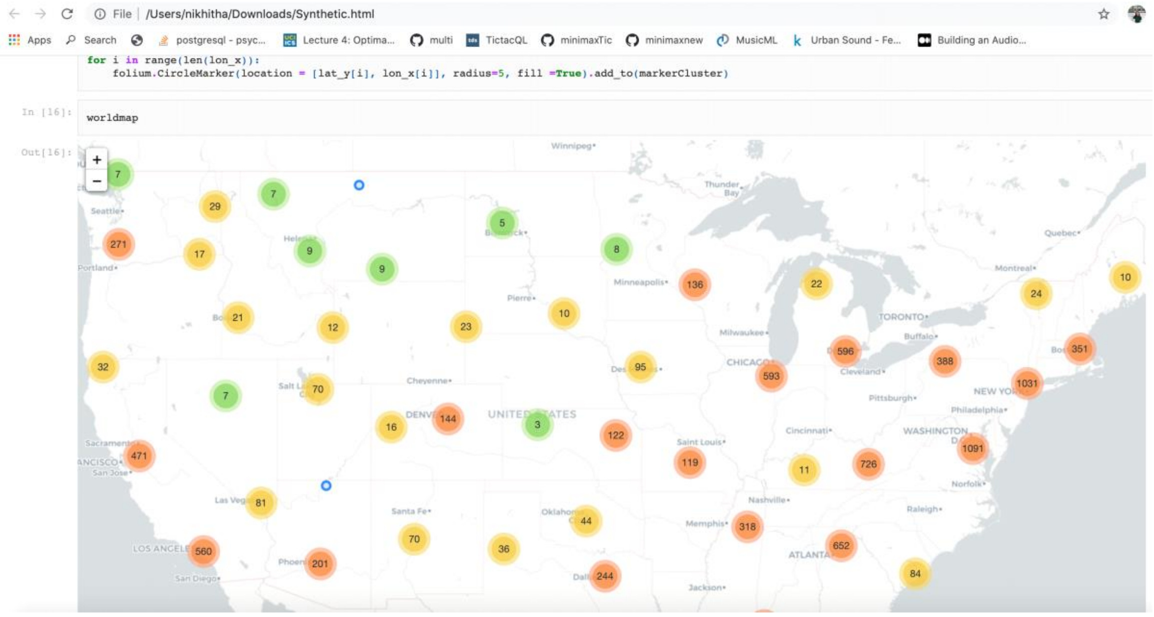
- devicestatus.txt has multiple delimiters. So we need to determine which delimiter to be used
- After processing data with the delimiter selected above the part of data used different delimiter wouldn't have parsed correctly. So Eliminate such rows
- Extract the date, model, device ID, latitude and longitude as 1 through 5 columns
- Rearrange the columns such that latitude and longitude as first two columns
- Eliminate rows that has latitude and longitude value as 0.
- Splitting the model field into device manufacturer and model name. Split is done at first space.
- Save the extracted data to a csv file of S3

Visualizing the data files:

Device Status Location:



Synthetic Data:

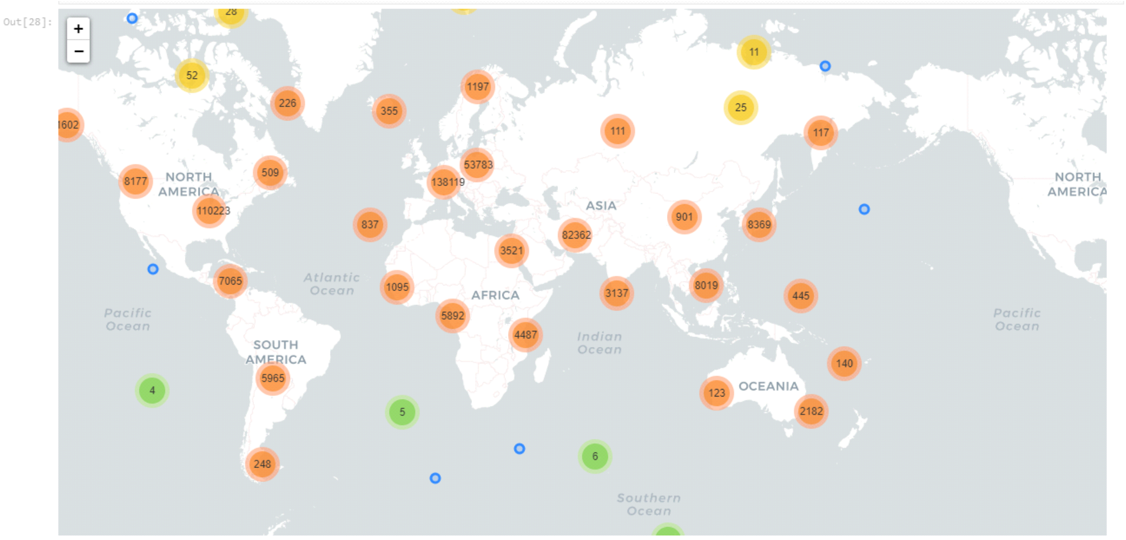


```
for i in range(len(lon_x)):
    folium.CircleMarker(location = [lat_y[i], lon_x[i]], radius=5, fill =True).add_to(markerCluster)
```

```
In [16]: worldmap
```

```
Out[16]:
```

DBPedia Data



```
Out[28]:
```

Clustering Approach:

After selecting K value in our k-means the initial centroids will be random sample of all points in the dataset. In each iteration it calculates the distance between the data point and the centroid of all the clusters. The distance is calculated using Euclidean distance or Great Circle distance. The data point is assigned a cluster whose centroid is near to the data point. Once the data point is added to the cluster, the centroid of the cluster is updated with the mean of the new values of the cluster