University of New Haven
TAGLIATELA COLLEGE OF ENGINEERING
Electrical & Computer Engineering and Computer Science

GEO LOCATION CLUSTERING

TECHNICAL REPORT

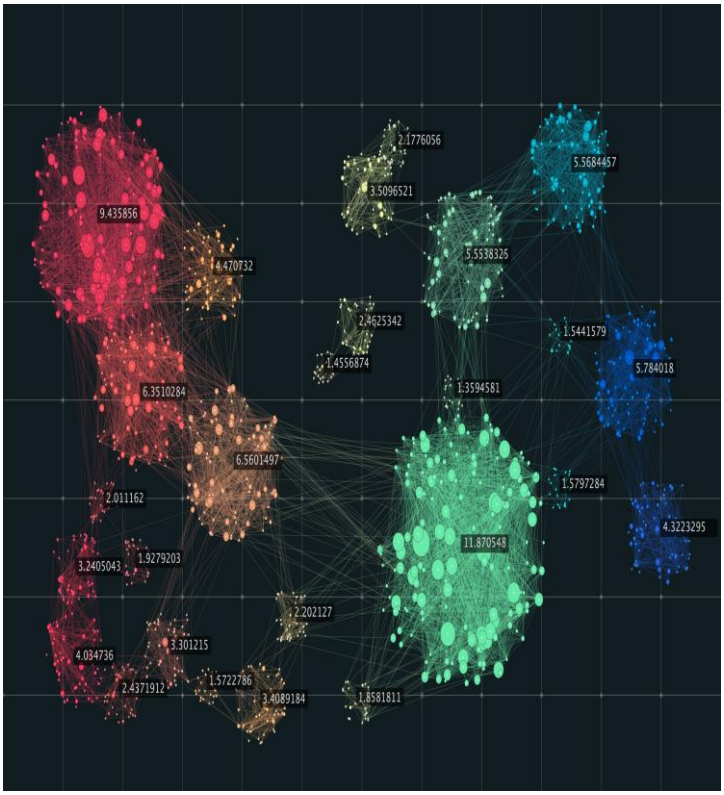
SPRING 22

CONTENTS

TEAM MEMBERS

1.Riz Amatya
2.ThrivenuBabu Manukonda
3.Upendra Boddu
4.Syeda Fatiha Buttul

GEO LOCATION CLUSTERING



Highlights of Project

The project aims to develop kmeans clustering of geolocation data.

Submitted on:

05/03/2022

Abstract

The main aim of the project is building a kmeans clustering algorithm for clustering of geolocation data, so that we can cluster large dataset on the basis of the Euclidean distance or the greater circle dataset. The centroids is generated and the dataset is used for visualization purpose

Introductory Section

Clustering:

Clustering is the process of splitting a set of data points into many groups so that data points in the same group are more similar than data points in other groups. To put it another way, the goal is to separate groups with similar characteristics and assign them to clusters.

Let's look at an example to help you understand. Assume you are the owner of a rental store and want to learn more about your customer’s preferences in order to expand your business. Is it possible for you to examine each customer's details and design a unique business plan for each? Certainly not. However, you can group all of your customers into, say, ten groups depending on their purchase behaviors, and employ a different method for each of these ten groups. This is referred to as clustering.

Now that we know what clustering is, let's look at some examples. Let's look at the many sorts of clustering.

Types of clustering:

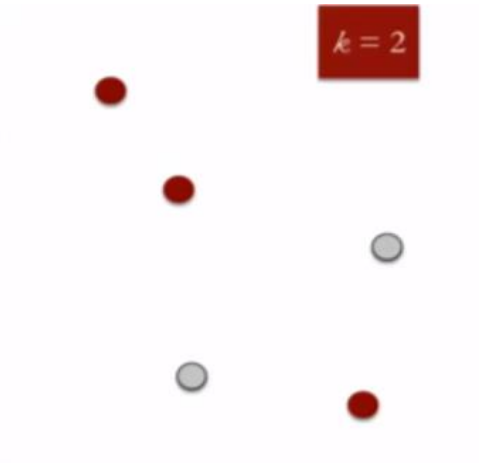
1.Hard clustering: In hard clustering, each data point is either totally or partially associated with a cluster. For example, in the above example each customer is put into one group out of the 10 groups.

2.soft clustering: Instead of assigning each data point to a separate cluster, soft clustering assigns a chance or likelihood of that data point being in those clusters. For example, in the aforementioned scenario, each customer is allocated a likelihood of being in one of ten retail store clusters.

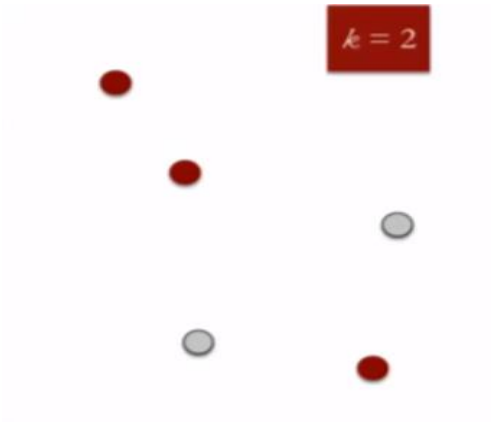
K Means Clustering:

The K means algorithm is an iterative clustering algorithm that seeks out local maxima in each iteration. This algorithm is made up of five steps:

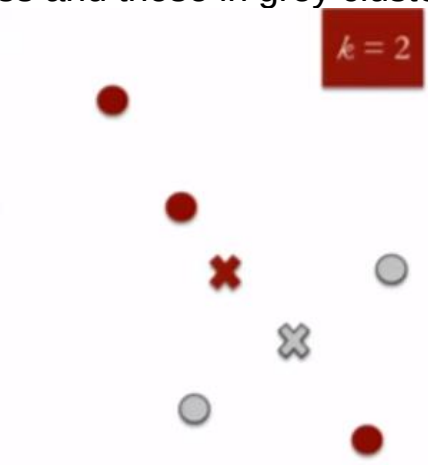
1. K: Indicate the desired number of clusters. For these 5 data points in 2-D space, let's use k=2.



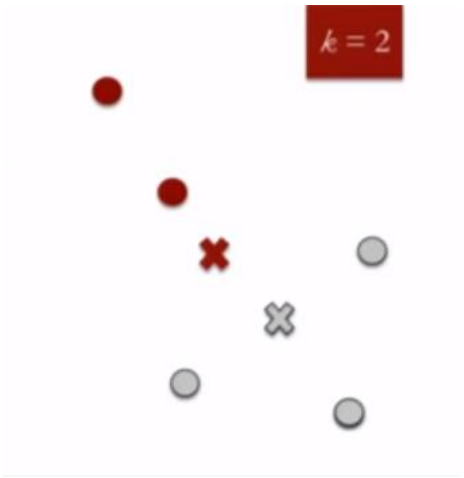
2. Assign each data point to a cluster at random: Assign three points to cluster 1 (shown by the red hue) and two points to cluster 2 (represented by the grey color).



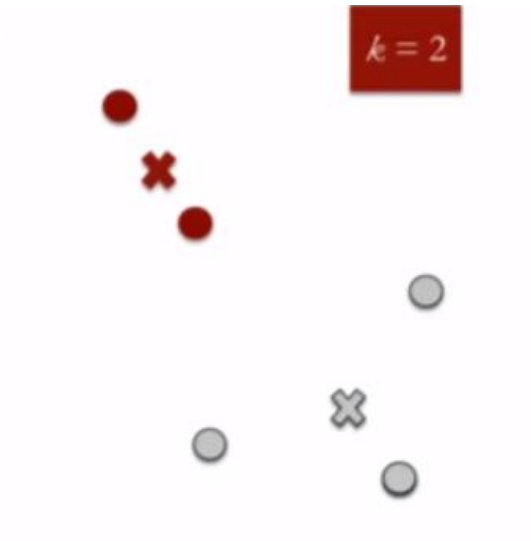
3. Compute cluster centroids: The centroid of data points in the red cluster is depicted using red cross and those in grey cluster using grey cross.



4. Reassign each point to the cluster centroid that is closest to it: Even though it is closer to the grey cluster's centroid, only the data point at the bottom is allocated to the red cluster. As a result, we place that data point in the grey cluster.



5. Recalculate cluster centroids: Recalculate the centroids for both clusters now.



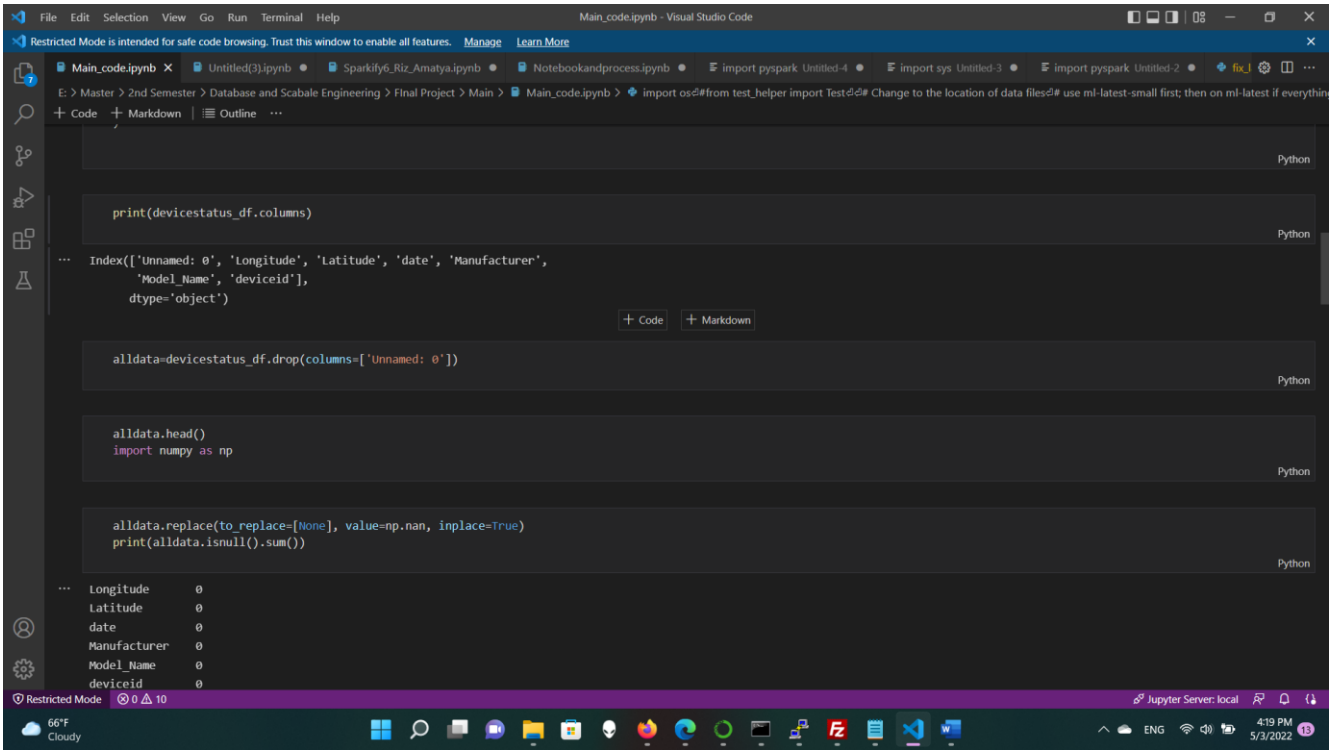
Review of available research

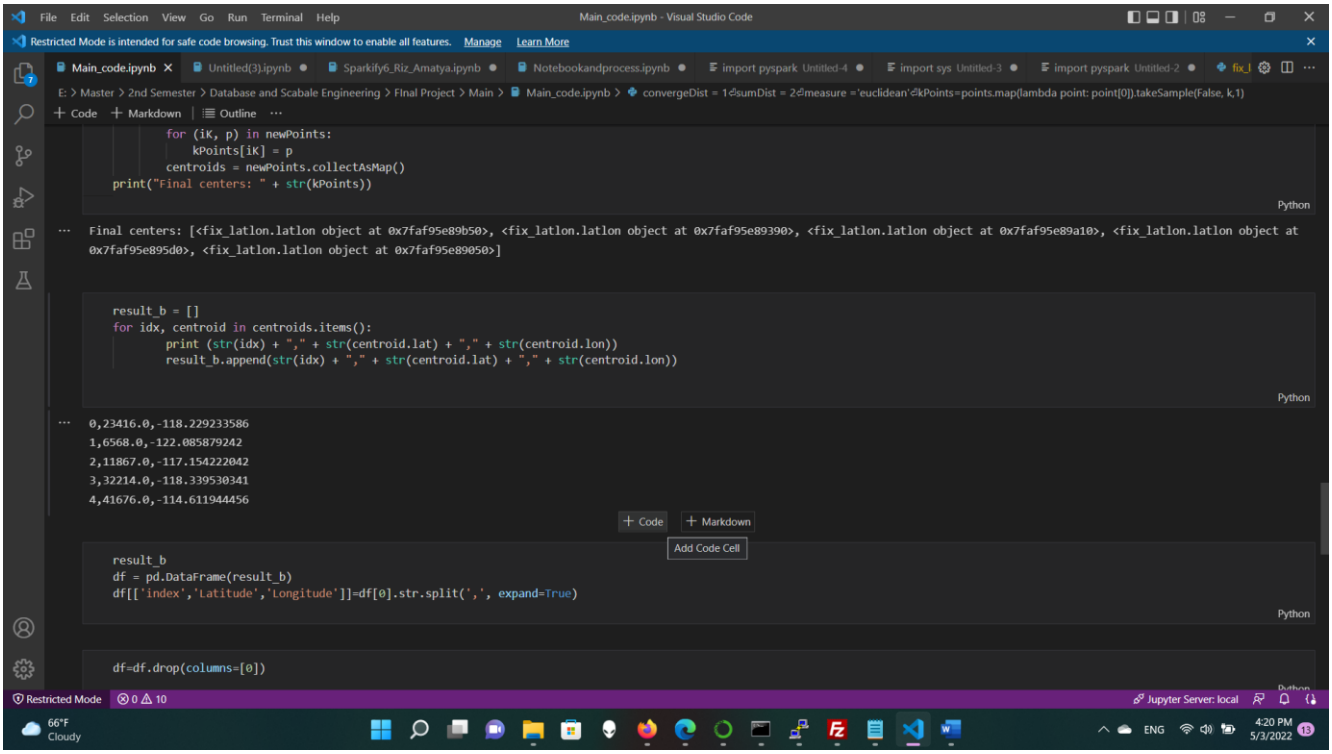
There is a lot of research being done in the field of k-means clustering. The geolocation data can be clustered and can be brought into a dynamic form by various process and implementation among which Kmeans is one of the measures, we can take.

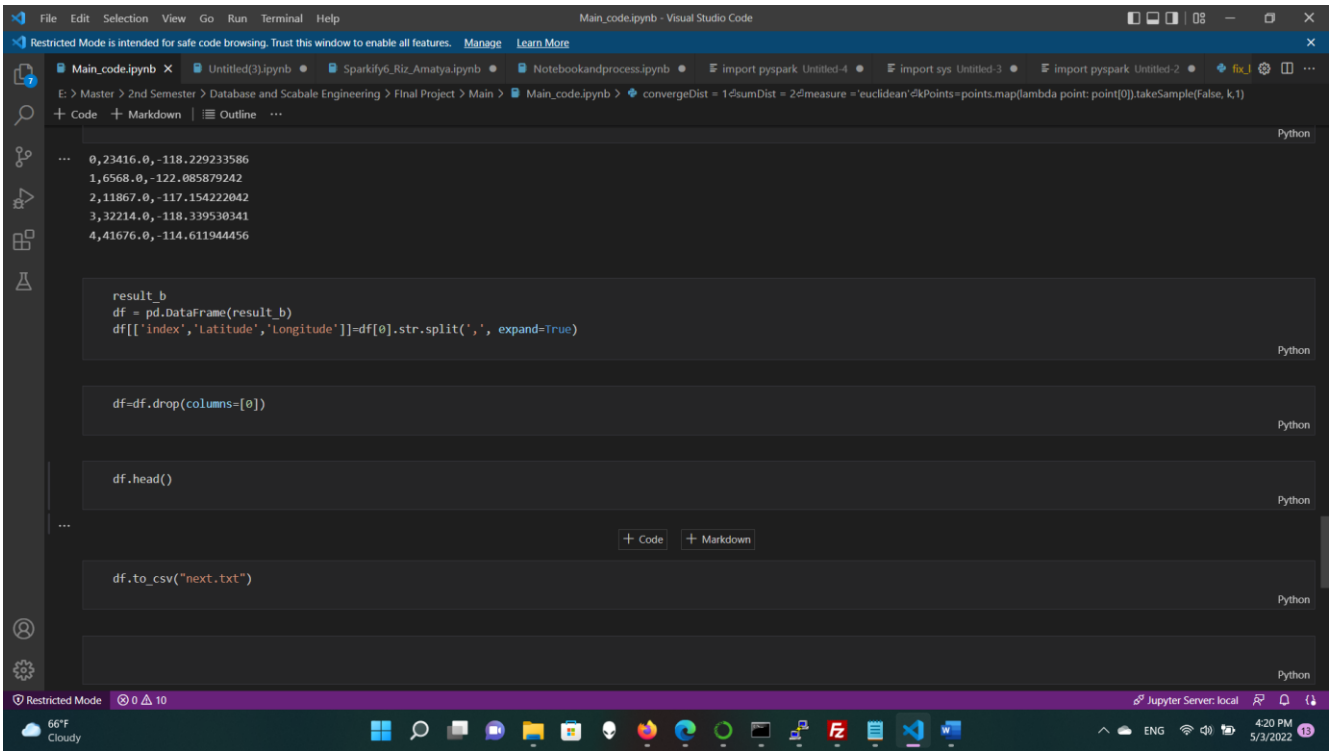
Methodology

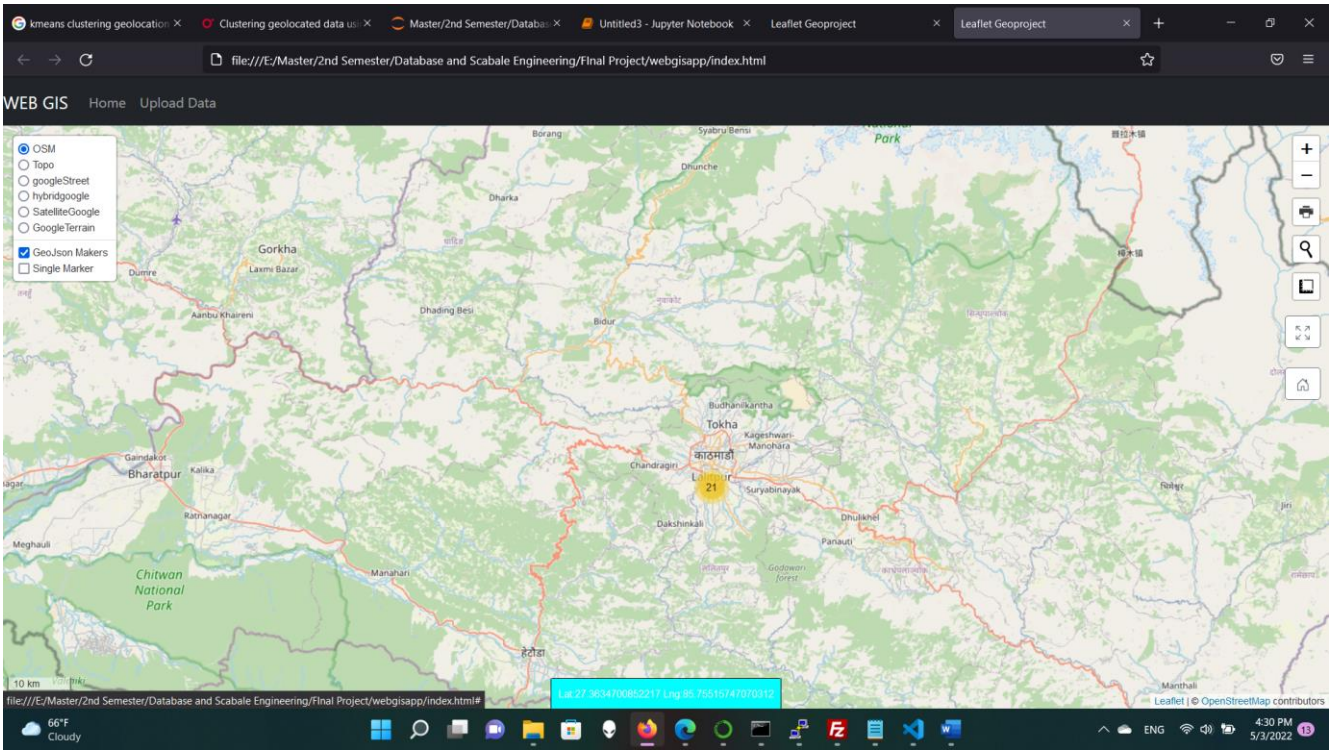
Firstly, we create virtual environment in amazon, and installed some libraries which we used for processing steps. We uploaded the file from our local computer to the s3 bucket in the amazon services. We create a Python file to store all the procedure and steps needed for the calculation process. Then the main mapper and reducer code were implemented and the centroids we found out and the centroids was saved into a text file using saves as text and the centroids were further utilized for the classification of the cluster. Initially the distance measure was given as Euclidean and the k was given as 5. The data was then processed using these variables and to find the best centroids for the latitude and longitude present in the data. For the processing step the data was firstly clean, checked for null value and then processed, otherwise we will get error in the later step of finding centroids and mapping and reducing the value. The latitude and longitude must me the first and second value for code to take other dataset as for the processing through the file.

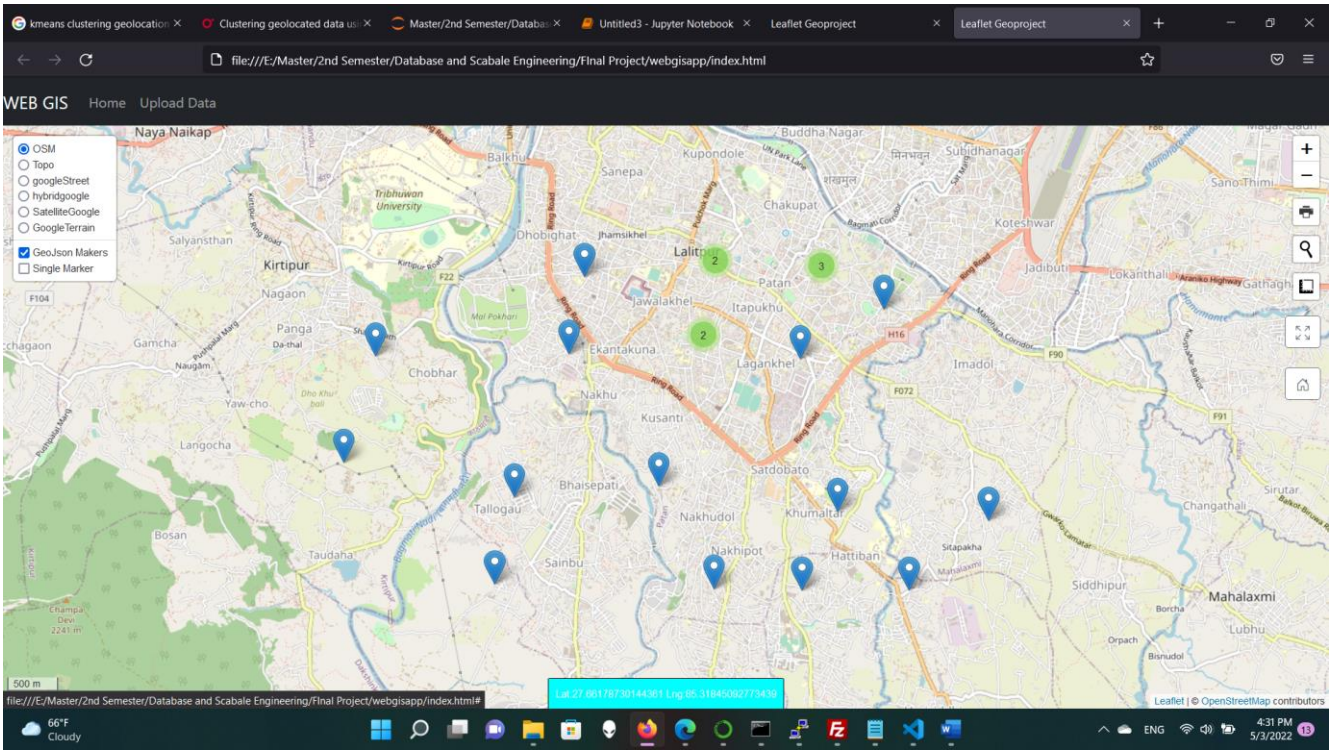
Results Section




File Edit Selection View Go Run Terminal Help
Main_code.ipynb - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
Main_code.ipynb
Untitled(3).ipynb
Sparkify6_Riz_Amatya.ipynb
Notebookandprocess.ipynb
import pyspark Untitled-4
import sys Untitled-3
import pyspark Untitled-2
fix_l
E: > Master > 2nd Semester > Database and Scabale Engineering > FInal Project > Main > Main_code.ipynb > convergeDist = 1.0sumDist = 2.0measure ='euclidean'.0kPoints=points.map(lambda point: point[0]).takeSample(False, k,1)
+ Code + Markdown Outline
for (iK, p) in newPoints:
kPoints[iK] = p
centroids = newPoints.collectAsMap()
print("Final centers: " + str(kPoints))
Python
Final centers: [<fix_latlon.latlon object at 0x7faf95e89b50>, <fix_latlon.latlon object at 0x7faf95e89390>, <fix_latlon.latlon object at 0x7faf95e89a10>, <fix_latlon.latlon object at 0x7faf95e895d0>, <fix_latlon.latlon object at 0x7faf95e89050>]
result_b = []
for idx, centroid in centroids.items():
print (str(idx) + "," + str(centroid.lat) + "," + str(centroid.lon))
result_b.append(str(idx) + "," + str(centroid.lat) + "," + str(centroid.lon))
Python
0,23416.0,-118.229233586
1,6568.0,-122.085879242
2,11867.0,-117.154222042
3,32214.0,-118.339530341
4,41676.0,-114.611944456
+ Code
+ Markdown
Add Code Cell
result_b
df = pd.DataFrame(result_b)
df[['index','Latitude','Longitude']]=df[0].str.split(',', expand=True)
Python
df=df.drop(columns=[0])
Restricted Mode
0 10
Jupyter Server: local
66°F
Cloudy
ENG
4:20 PM
5/3/2022


File Edit Selection View Go Run Terminal Help
Main_code.ipynb - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
Main_code.ipynb
Untitled(3).ipynb
Sparkify6_Riz_Amatya.ipynb
Notebookandprocess.ipynb
import pyspark Untitled-4
import sys Untitled-3
import pyspark Untitled-2
fix_l
E: > Master > 2nd Semester > Database and Scabale Engineering > FInal Project > Main > Main_code.ipynb > convergeDist = 1.0sumDist = 2.0measure ='euclidean'.0kPoints=points.map(lambda point: point[0]).takeSample(False, k,1)
+ Code + Markdown | Outline ...
Python
0,23416.0,-118.229233586
1,6568.0,-122.085879242
2,11867.0,-117.154222042
3,32214.0,-118.339530341
4,41676.0,-114.611944456
result_b
df = pd.DataFrame(result_b)
df[['index','Latitude','Longitude']]=df[0].str.split(',', expand=True)
Python
df=df.drop(columns=[0])
Python
df.head()
Python
+ Code + Markdown
df.to_csv("next.txt")
Python
Python
Restricted Mode 0 10
Jupyter Server: local
66°F Cloudy
ENG
4:20 PM 5/3/2022


WEB GIS
Home
Upload Data
OSM
Topo
googleStreet
hybridgoogle
SatelliteGoogle
GoogleTerrain
GeoJson Makers
Single Marker
Gorkha
Bharatpur
Tokha
काठमाडौं
Chitwan National Park
Leaflet | © OpenStreetMap contributors


WEB GIS
Home
Upload Data
OSM
Topo
googleStreet
hybridgoogle
SatelliteGoogle
GoogleTerrain
GeoJson Makers
Single Marker
Naya Naikap
Kirtipur
Chobhar
Balkhu
Sanepa
Kupondole
Chakupat
Patan
Jawalakhel
Ekantakuna
Lagankhel
Nakhu
Kusanti
Satdobato
Imadol
Koteshwar
Jadibuti
Sano Thimi
Gathaghar
Changathali
Sirutar
Mahalaxmi
Siddhipur
Lubhu
Bisnudol
Orpach
Sitapakha
Hattiban
Khumaltar
Nakhipot
Nakhudol
Bhaisepati
Tallogau
Sainbu
Taudaha
Bosan
Langocha
Yaw-cho
Chobhar
Gamcha
Panga
Nagaon
Salyansthan
Dhobighat
Jhamsikhel
Lalitpur
Buddha Nagar
Subidhanagar
Tribhuwan University
Ring Road
Kirtipur Ring Road
Araniko Highway
Leaflet | © OpenStreetMap contributors
500 m

Discussion

The overall result was just calculated using the kmeans clustering, but we can furthermore use it for further calculation and evaluation of the clusters on the maps. The location based clustering with various measure to take into consideration can be used for better clustering and modification of data according to the needs.

Conclusion

The project is about how we can utilize the function and properties of spark for RDD for running mapper and reducer task for big data processing and created clustering based on the distance from centroid. The projects can be further enhanced and used for analysis various analysis purposes. The overall task of the algorithm is to create a cluster using kmean clustering with greater circle or Euclidean distance.

Contributions/References

- K-Means Cluster Analysis | Columbia Public Health
- book7.dvi (stanford.edu)
- B. Babcock, M. Datar, R. Motwani, and L. O'Callaghan, "Maintaining variance and k-medians over data stream windows," Proc. ACM Symp. on Principles of Database Systems, pp. 234–243, 2003.
- V. Ganti, R. Ramakrishnan, J. Gehrke, A.L. Powell, and J.C. French:, "Clustering large datasets in arbitrary metric spaces," Proc. Intl. Conf. on Data Engineering, pp. 502–511, 1999.
- Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algo rithm for large databases," Proc. ACM SIGMOD Intl. Conf. on Manage ment of Data, pp. 73–84, 1998.
- T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," Proc. ACM SIGMOD Intl. Conf. on Management of Data, pp. 103–114, 1996