BIG DATA PROJECT (19B12CS419)

EVEN SEMESTER 2020

CLUSTERING INDIAN STATES INTO DIFFERENT COVID-19 ZONES

Submitted by:

Pulkit Khurana (9917103237)

Vrinda Goyal (9917103238)

Subhradip Mukherjee (9917103234)

Shivam Singh (9917103257)

Submitted To:

Dr. Shikha Mehta

Department of CSE/IT

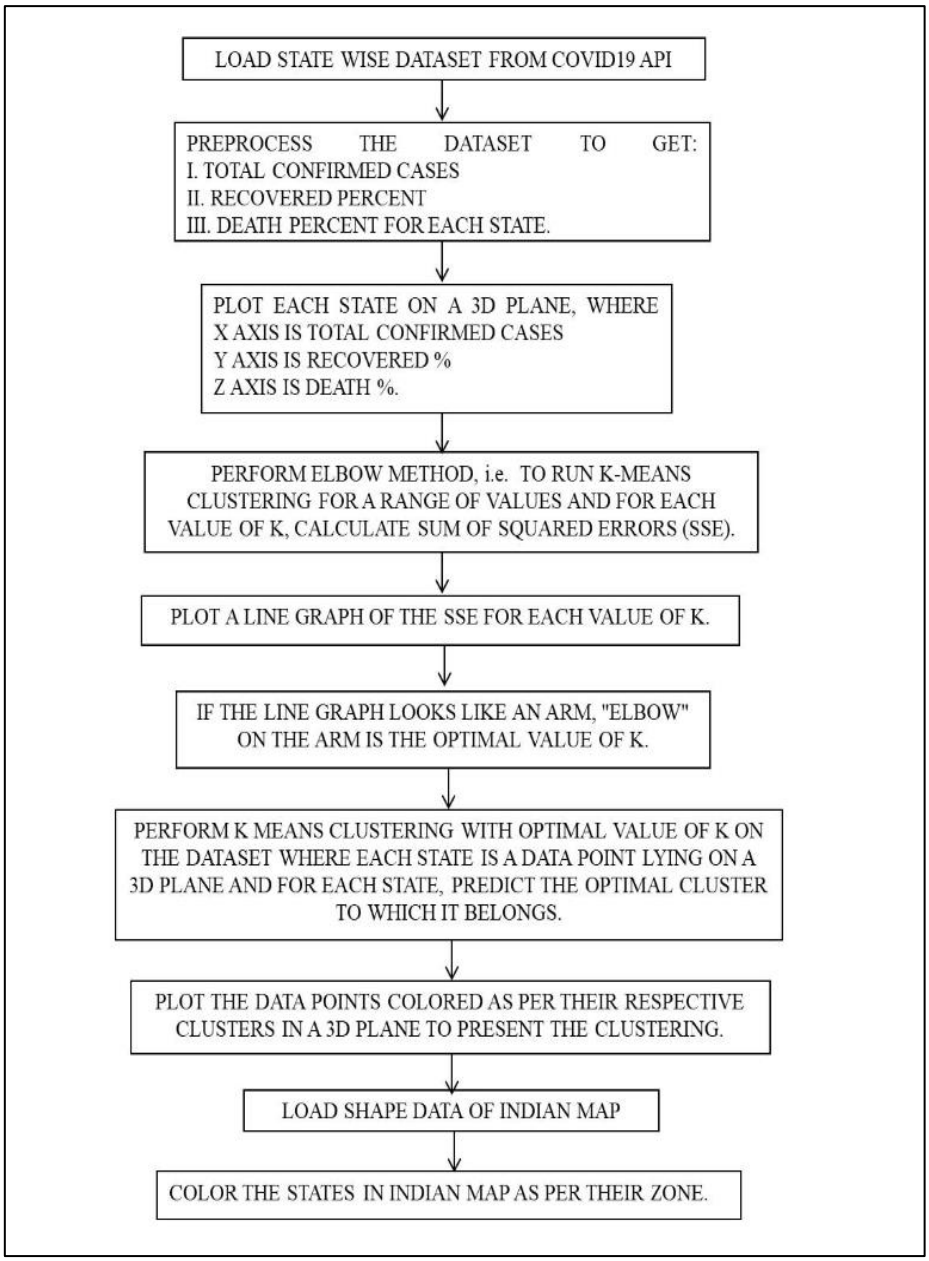
Jaypee Institute of Information Technology University, Noida

May 2020

PROBLEM STATEMENT

With rapid exponential increase of COVID-19 patients, the Govt. of India is going shorthanded in finding ways to approach this epidemic. Our project uses machine learning algorithm in order to divide all the states/districts across the country into different clusters ranging from most affected area to least affected area so as to give more focus to the more affected areas.

CONTROL FLOW GRAPH



ALGORITHM AND TECHNIQUES USED

We are using k-means algorithm, which is an unsupervised clustering algorithm. K-means algorithm is an iterative algorithm that tries to partition the dataset into *K* distinct non-overlapping clusters (subgroups) where each data point belongs to only one cluster**.** It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as far as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid is at the minimum.

The way k-means algorithm works is as follows:

1. Specify number of clusters K.
2. Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement. These k centroids represents k cluster.
3. Compute the sum of the squared distance between data points and all centroids.
4. Assign each data point a cluster id to which it will belong based on the distance between that data point and cluster centroid being minimum.
5. Compute the new centroids from the clusters formed by taking the average of the all data points that belong to each cluster.
6. Keep repeating steps 3 to 5 until there is no change in the centroids found in 2 consecutive iterations I.e. assignment of data points to clusters isn't changing.

Here we aren't taking a random value of k, rather we are using a method called elbow method in order to find k which will give us the best clusters possible.

The idea behind elbow method is to run k-means clustering on a given dataset for a range of values of k (num_clusters, e.g. k=1 to 10), and for each value of k, calculate sum of squared errors (SSE).

After that, plot a line graph of the SSE for each value of k. If the line graph looks like an arm - a red circle in below line graph (like angle), the "elbow" on the arm is the value of optimal k (number of cluster). Here, we want to minimize SSE. SSE tends to

decrease toward 0 as we increase k (and SSE is 0 when k is equal to the number of data points in the dataset, because then each data point is its own cluster, and there is no error between it and the centre of its cluster).
So the goal is to choose a small value of k that still has a low SSE, and the elbow usually represents where we start to have diminishing returns by increasing k.

Our project is divided into 2 parts –

1st dividing the country into k clusters at the state level and in the 2nd part diving it into k clusters at district level. For the 1st part we have taken states as our data points and for the 2nd part districts as data points.
The attributes that define these data point are x – Total covid-19 cases, y – No. of recovered cases, z- No. of death cases. The distance between 2 data points is calculated using Euler's distance formula-

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}.$$

WORK DONE

- Pre-processed the dataset (handling Null/inconsistent/missing data) to get Total confirmed cases, Recovered percent (No of persons recovered per 100 positive cases), Death % (No of persons deceased per 100 positive cases) for each state.
- Perform elbow method, i e to run k-means clustering for a range of values of and for each value of k, calculate sum of squared errors (SSE) to find the optimum value of k.
- We got optimal value of k as 4 and therefore performed K Means clustering with K=4 on the dataset and for each state, predicted the optimal cluster to which it belongs.
- Plotted the 3D scatter plot (Figure 2.1, 2.1) coloured as per their respective clusters in a 3D plane to present the clustering and Coloured the states in Indian map as per their zone calculated using K means algorithm.
- Plotting the 2D elbow curve (Figure 1.1 , 1.2) for a value of k
- Plot of the India map divided into 4 clusters (Figure 3)

RESULT AND ANALYSIS :

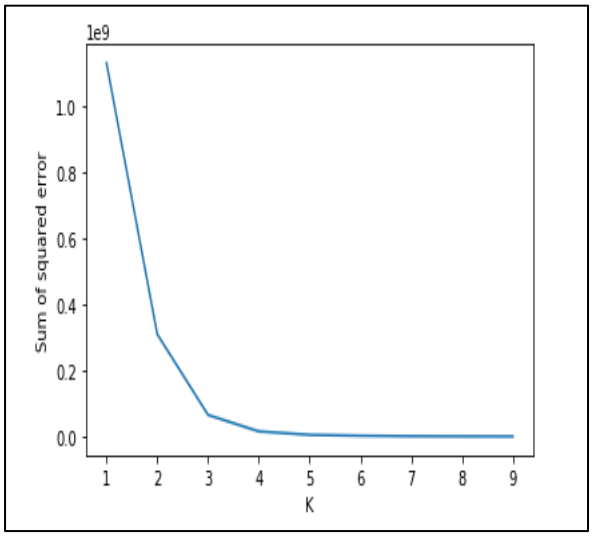


Figure 1.1 State Elbow curve

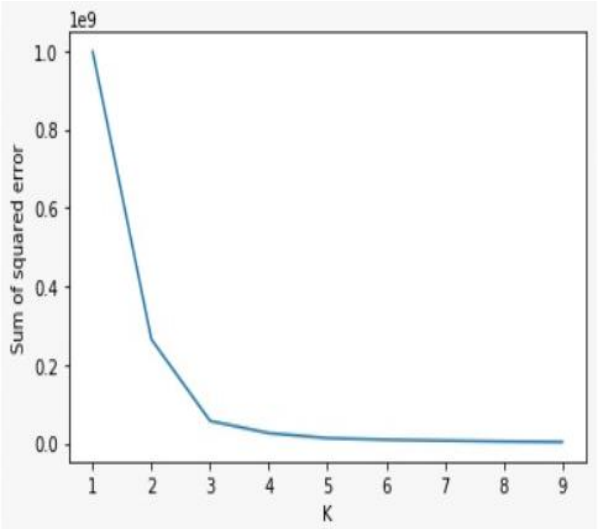


Figure 1.2 District Elbow curve

Using Elbow Method, We got optimal value of k as 4, therefore we clustered all the states into 4 clusters using K means algorithm:

1. **Green**: Arunachal Pradesh, Assam, Bihar, Chhattisgarh, Goa, Haryana, Himachal Pradesh, Jharkhand, Karnataka, Kerala, Manipur, Meghalaya, Mizoram, Nagaland, Odisha, Punjab, Sikkim, Telangana, Tripura, Uttarakhand, Andaman and Nicobar Chandigarh, Daman and Diu, Dadar and Nagar Haveli, Jammu and Kashmir, Ladakh, Lakshadweep, Puducherry

2. **Orange** : Madhya Pradesh, Rajasthan, Uttar Pradesh, Andhra Pradesh, West Bengal

3. **Red** : Tamil Nadu, Gujarat, Delhi

4. **Black**: Maharashtra

Black Zone has the highest number of COVID-19 related cases and deaths in the country. Red Zones also have a high number of cases while Orange Zones have

comparatively fewer cases and Green Zones have the least no of cases and deaths related to COVID-19.

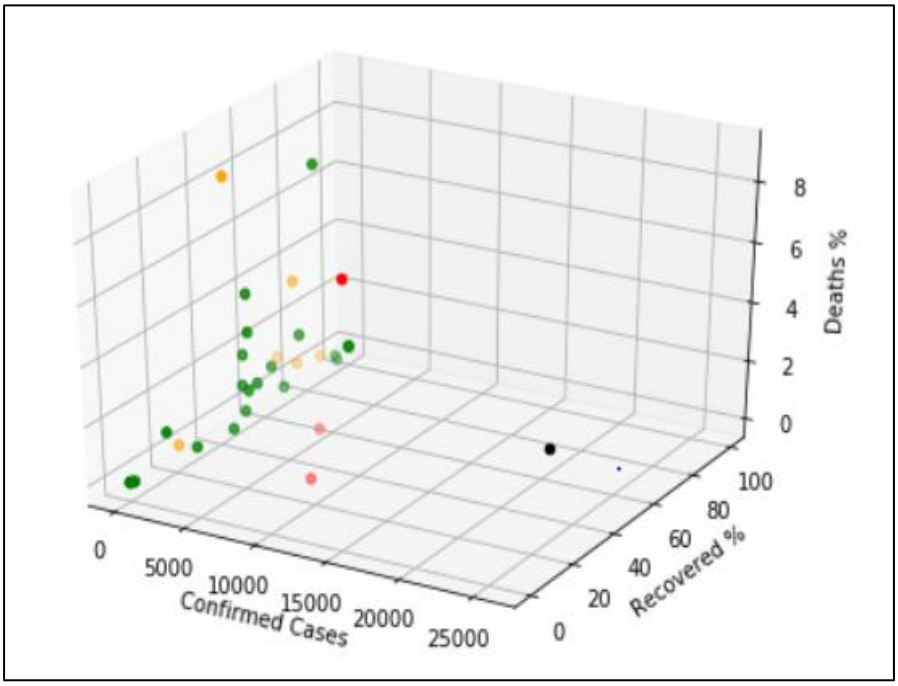


Figure 2.1 State Plotting

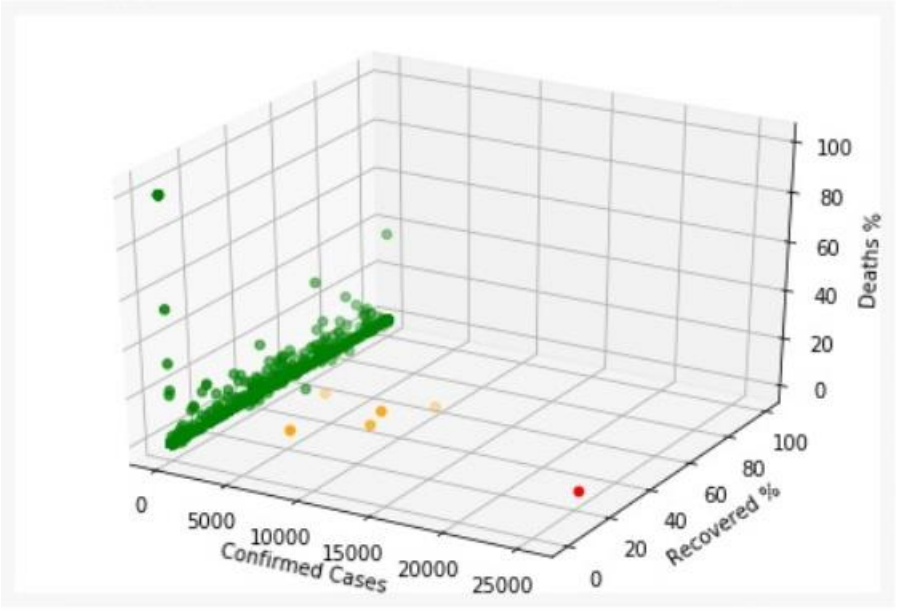


Figure 2.2 District Plotting

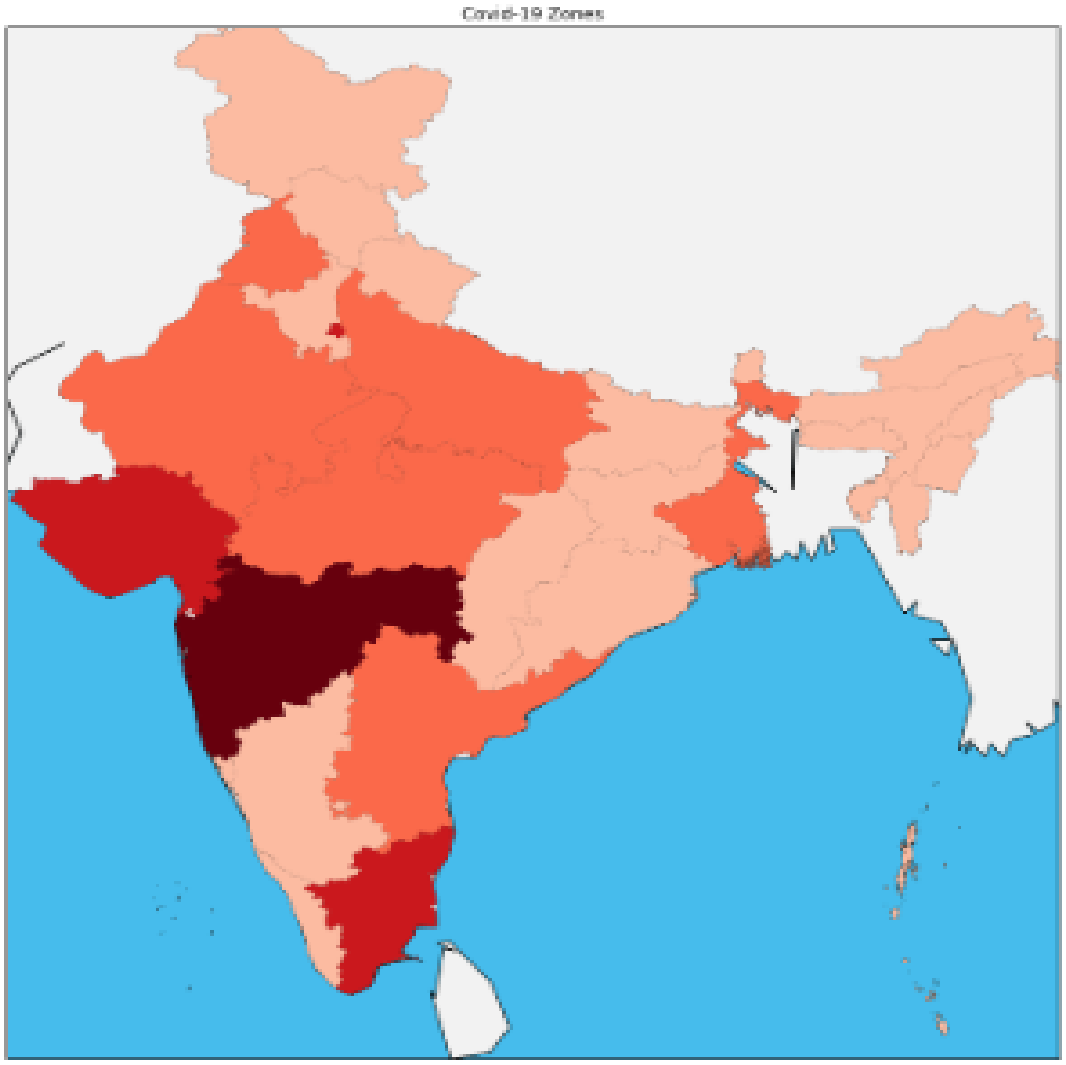


Figure 3 State wise division on india map

DATASET USED:

The data has been taken from covid19india.org. It contains state wise and district wise data of India i.e. number of confirmed, recovered, death, active cases of related to COVID 19 until 16 May, 2020.

State wise data - https://api.covid19india.org/csv/latest/state_wise.csv

District wise data - https://api.covid19india.org/csv/latest/district_wise.csv

Indian Map Boundaries -

https://www.arcgis.com/home/item.html?id=cf9b387de48248a687aafdd4cdff1127

CODE FILE:

https://github.com/Pulkit-100/Clustering_Indian_States_Covid19_KMeans

STEPS TO EXECUTE THE CODE:

Prerequisites to execute the code:

In addition to basic Python, Jupyter and Anaconda dependencies you need :

- Numpy
- Pandas
- Sklearn
- Matplotlib
- Basemap

Steps to install these requirements :

1. pip install numpy pandas sklearn matplotlib
2. conda install basemap

Code Execution :

1. Open the notebook file either on jupyter notebook or in Google colab (http://colab.research.google.com/)
2. Make sure you have all the pre requisites installed in the python environment.
3. Make sure you have the Indian Map Boundary Dataset downloaded in the notebook's directory.
4. Run all the cells of the notebook.

WORK DISTRIBUTION

- Appying k-means algorithm, elbow method for finding k and plotting Indian map data and performing State Level Clustering by Pulkit Khurana (9917103237) and Vrinda Goyal (9917103238)

- Data pre-processing and applying k-means for district level clustering by Subhradip Mukherjee (9917103234) and Shivam Singh (9917103257)