Machine Learning (CS60050)

Project- Coronavirus Data Clustering using Complete Linkage Hierarchical Clustering

Vishal Gourav (20CS60R21)

March 18, 2021

Introduction

The project uses concepts of Unsupervised Learning, in particular **k-means Clustering** and **Complete Linkage Hierarchical Clustering** to create clusters from a given data set. Sticking to the modular approach of project development, the tasks have been divided into 3 programs which are as follows:-

- **kmean.py** The program performs **Tasks 1,2** and **3** i.e., k-means clustering for k in between 3 and 6.
- **heira.py** The program performs **Task 4a** i.e., Complete Linkage Hierarchical Clustering for best value of k in obtained in *kmean.py*.
- **jaccard.py** The program performs **Task 4b** i.e., find jaccard similarity scores of the outputs of the previous two programs.

The functionalities have been divided into different functions in each program which are discussed further in the report.

Functions

The program **kmean.py** takes in input the given data set *COVID_3_unlabelled.csv* and creates clusters for values of k between 3 and 6 and displays them. It also calculates the Silhouette coefficient for each value of k and returns the best value k. For doing this the program uses the following functions:-

- **preprocess()** In this function the given data is normalized by z-score normalization in the range -2 to 2.
- **init_centroids()** In this function the k centroids are randomly assigned from the given data set.
- **cosine_distance(a,b)** This function finds and returns the cosine distance between 2 lists a and b.

- **min_dist(dis_list)** This function returns the cluster ID and value of minimum distance of a point in the data set to the mean of each cluster.
- **update_centroids(clus)** In this function the centroids are updated to the new mean value of the points in each cluster.
- **plot_cluster()** This functions returns the cluster in the form of a 3D plot using matplotlib library in python.
- **clustering()** The clustering is done by this function which calls all above functions according to requirement.
- **find_closest_cluster()** This function creates a dictionary that contains every cluster ID and corresponding nearest cluster's cluster ID.
- **mean_distance(one_row,clus)** The function returns the mean of distances between a point in the data set and every point that belongs to the given cluster in input.
- **index_cluster(clus)** The function returns the index of cluster closest to a given cluster.
- **silhoute_coeff()** This function calculates the Silhouette coefficient for each point for a given value of k.
- **cluster_info(file)** This function creates the file *kmeans_<file>.txt* and prints the clusters' information in the file.

The program **heira.py** takes in input the given data set *COVID_3_unlabelled.csv* and creates clusters for values of k obtained in *kmean.py*. It then creates a distance matrix and then performs hierarchical clustering using the following functions:-

- **preprocess()** In this function the given data is normalized by z-score normalization in the range −2 to 2.
- **cosine_distance(a,b)** This function finds and returns the cosine distance between 2 lists a and b.
- **plot_cluster()** This functions returns the cluster in the form of a 3D plot using matplotlib library in python.
- **find_max_list(list1,list2,ind1,ind2)** This function takes as input 2 lists list1 and list2, and for every element in the 2 lists first finds maximum and stores it in *final_list*. The *final_list* and its copy are returned after deleting the values at given indices ind1 and ind2, and adding a max value of 100 to one of the lists.
- **create_distance_matrix()** This function creates the distance matrix for each point with respect to every other point in the given data set. The distance metric used is cosine distance.
- **hier_clusters()** This function creates the clusters in a bottom up fashion using complete linkage hierarchical clustering technique.
- **cluster_info()** This function creates the file *agglomerative.txt* and prints the clusters' information in the file.

The final program **jaccard.py** maps each cluster in the output of *kmean.py* to the output clusters of *heira.py* and lists the jaccard similarity score between each cluster in the 2 outputs.

For this it uses the following functions:-

- **jaccard_similarity(list1, list2)** The function takes in as input 2 sets and returns the jaccard similarity score between them.
- **find_jaccard_mapping()** This function does the bijective mapping between the 2 cluster sets.

Results

All programs have been executed using python interpreter on command prompt on a Windows 7 system. It must be kept in mind that the output **may vary on every execution** as the initial centroids in *kmean.py* are chosen **randomly** each time. The results obtained for the various programs are described further in the report.

Results for kmean.py(Task 1,2 and 3)

The program takes about 8 seconds to plot and display each cluster and about 30 seconds to calculate Silhouette coefficient for each value of k.
The cluster plots for different values of k are as follows:-

- For **k=3**, the clusters are diagrammatically represented in Figure 1.

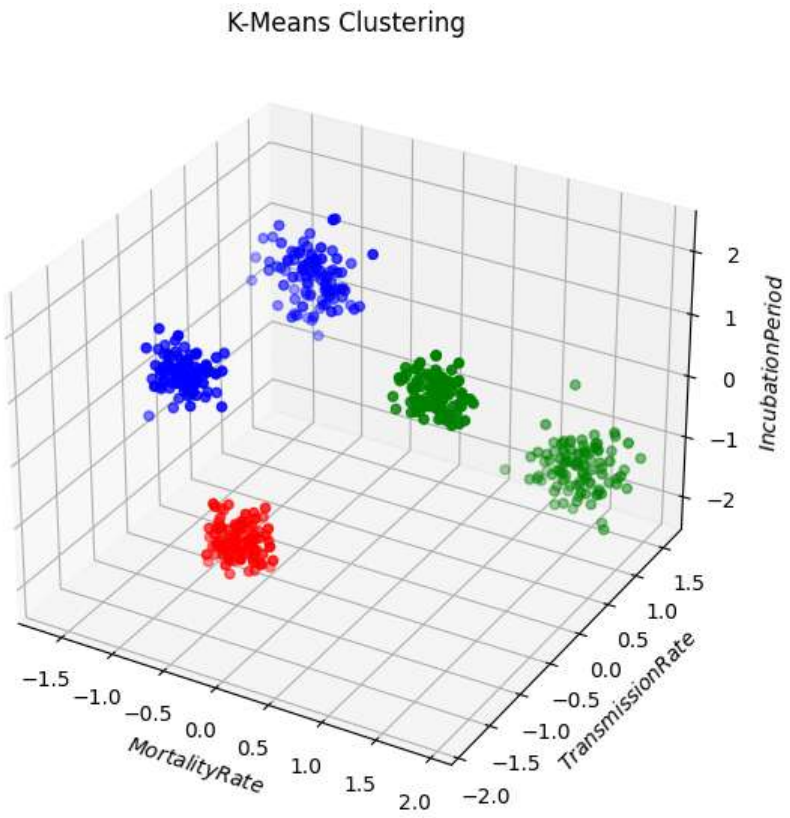

Figure 1: K-mean Clusters at k=3

- For **k=4**, the clusters are diagrammatically represented in Figure 2.

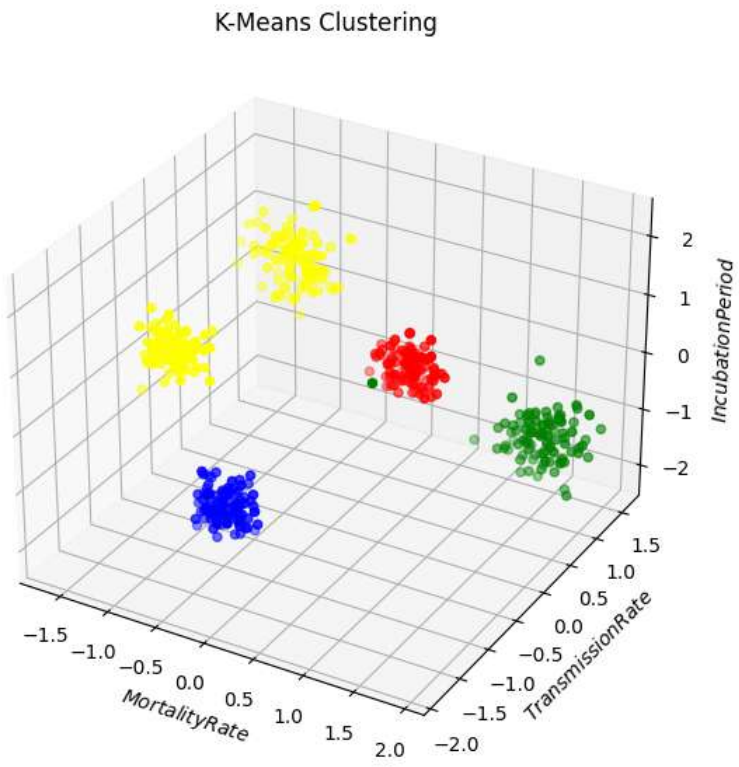

Figure 2: K-mean Clusters at k=4

- For **k=5**, the clusters are diagrammatically represented in Figure 3.

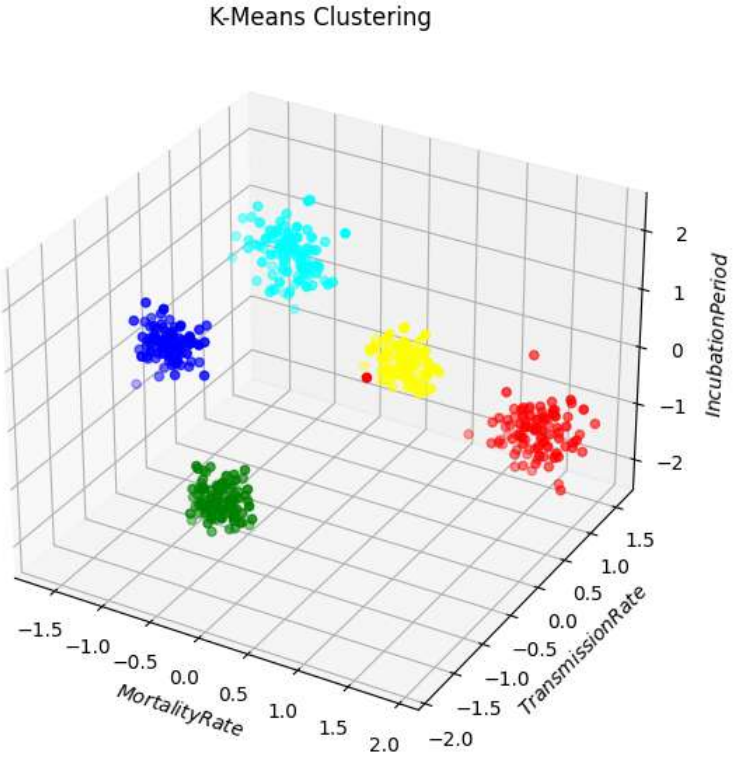

Figure 3: K-mean Clusters at k=5

- For **k=6**, the clusters are diagrammatically represented in Figure 4.

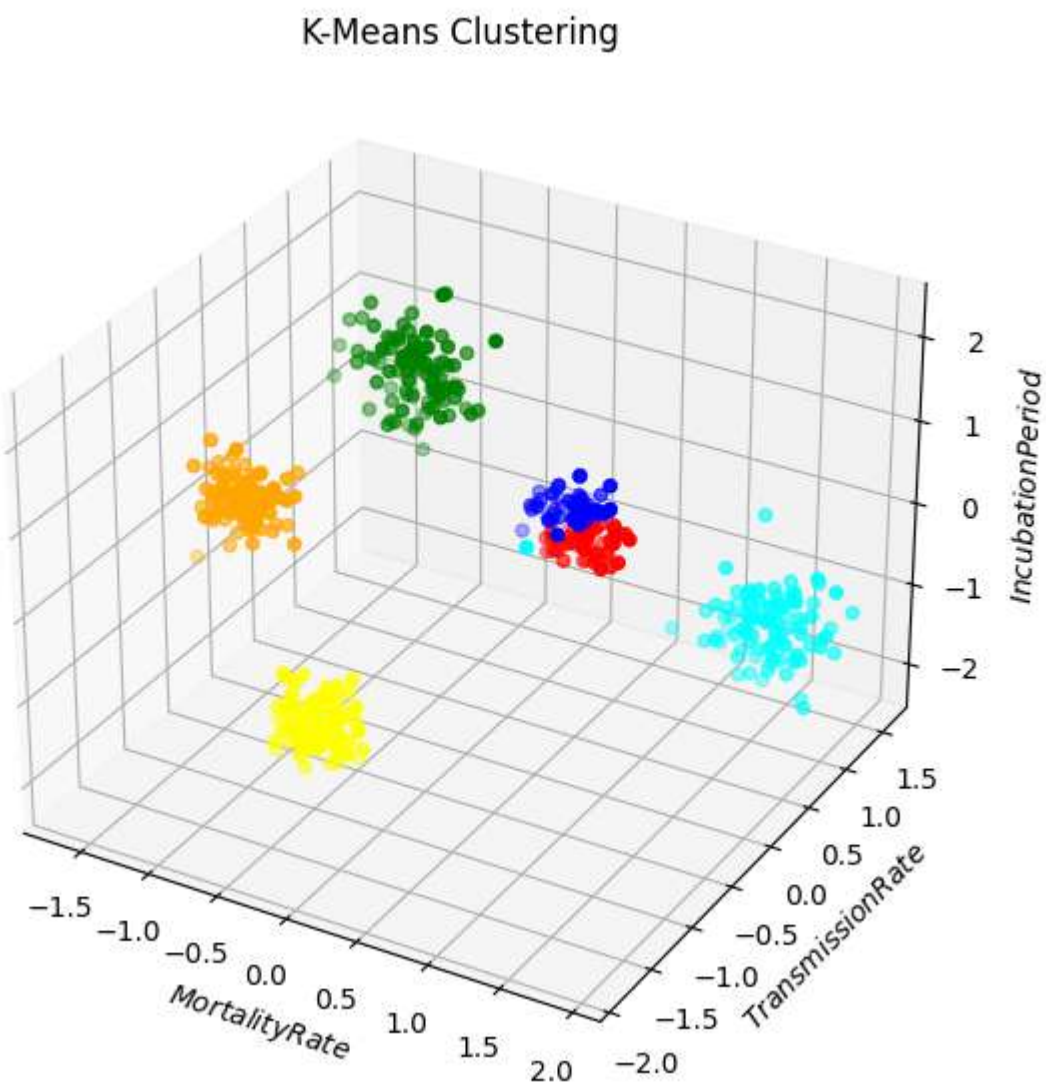

Figure 4: K-mean Clusters at k=6

- The Silhouette coefficients obtained is shown in Figure 5

```
C:\Users\VISHAL\Desktop\ML Assignments\Ass3>python kmean.py
At k= 3 , Silhouette Coefficient:  0.7829629793207037
At k= 4 , Silhouette Coefficient:  0.850233815828951
At k= 5 , Silhouette Coefficient:  0.9335492338563922
At k= 6 , Silhouette Coefficient:  0.9011168006216054

The best clustering is reached at k= 5 at a value of Silhouette Coefficient: 0.9
335492338563922
```

Figure 5: Silhouette coefficients at different values of k

Result for hiera.py(Task 4a)

The output cluster obtained by using Complete Linkage Hierarchical Clustering Technique at **k=5** which is the best value of k obtained according to Silhouette coefficients of different k values as seen in Figure 5 is shown diagrammatically in Figure 6. It takes about 45 seconds to execute.

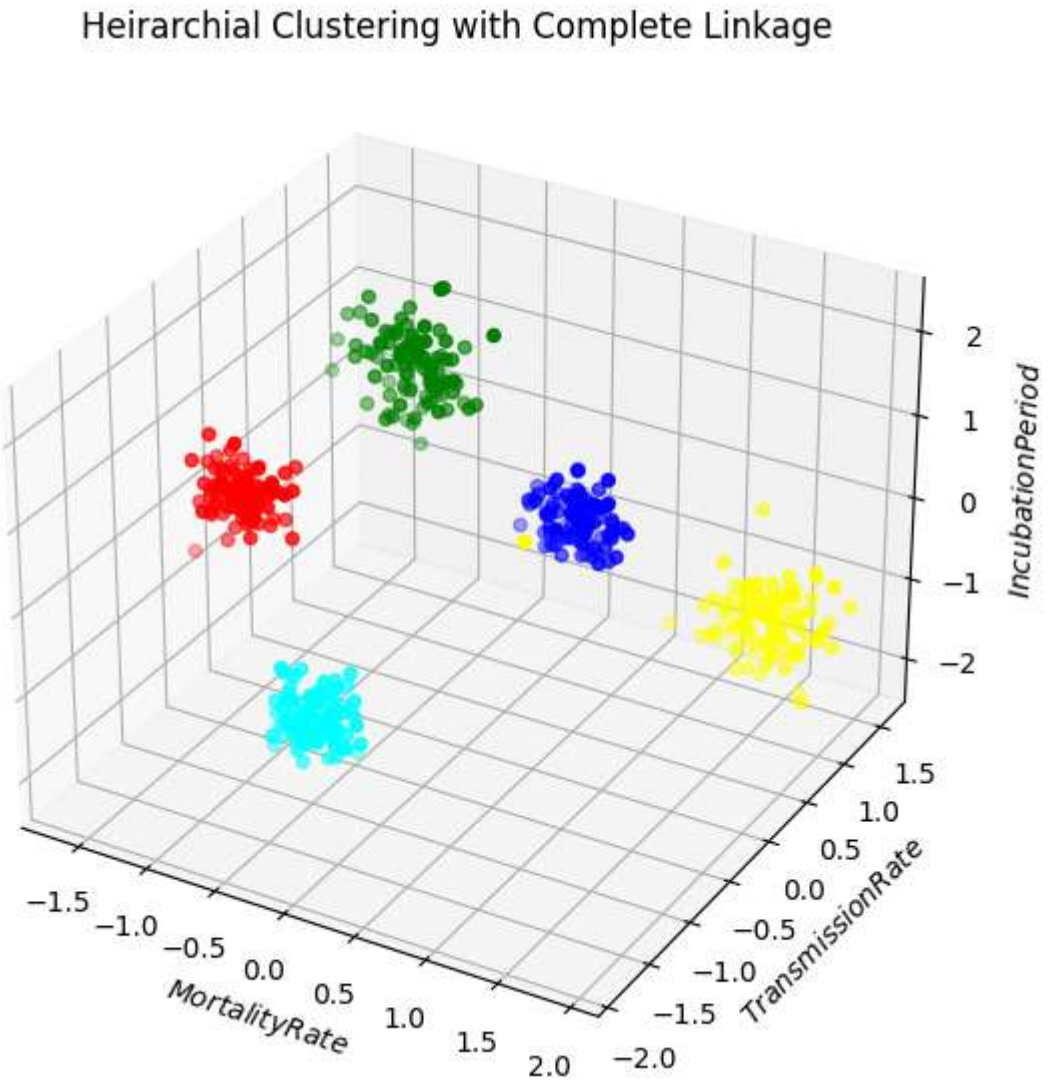

Figure 6: Complete Linkage Hierarchical Clustering at k=5

Result of jaccard.py(Task 4b)

The bijective mapping obtained from the output clusters of *kmean.py* and *heira.py* can be seen in Figure 7. The figure also contains overall output of all programs together.

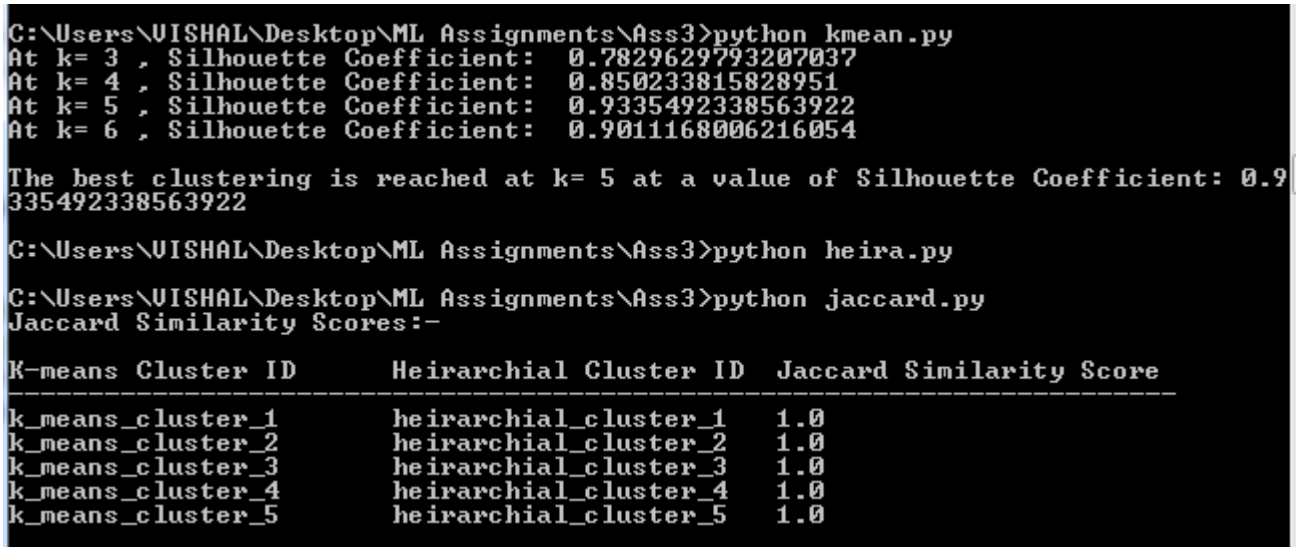

Figure 7: Jaccard Similarity Scores and overall output

Conclusion

The results help us visualize Coronavirus Data using k-means clustering technique to find best value of k and use it to cluster efficiently using Complete Linkage Hierarchical Clustering Technique.