Hierarchical Clustering for Seed Categorization:

Description: Implement Hierarchical Clustering on the UCI seed dataset to divide it in groups and then use nearest neighbour classification based on the cluster IDs to identify the species.

Implementation: The implementation involves a Hierarchical agglomerative clustering algorithm. The algorithm starts by making small cluster from the given data points and recursively merging them until the end condition is met.

The data set used in the UCI seed dataset. The dataset is downloaded from “<https://archive.ics.uci.edu/ml/machine-learning-databases/00236/>” . The downloaded file is converted into a csv file and renamed as “seeds.csv”. This csv file has the data without the labels present for the data points. We also have a duplicate csv file named “seedsDuplicate.csv”, this csv has an additional column to hold the labels. The labels are added to the data set once the clustering is complete and based on the cluster ID (say cluster1, cluster2, etc.).

Since the UCI seed data set has 3 main labels for the datapoints(1,2,3), we provide the number of cluster value as 3 and based on this the program will return 3 main clusters based on the respective Euclidean distances calculated.

The program uses priority queue to store the distances between each pair of clusters and find any 2 clusters with minimum distance efficiently. For this we make use of the “heapq.py” obtained from “<https://github.com/python/cpython/blob/3.7/Lib/heapq.py>”.

Three main clusters are formed based on the number of clusters initially provided. The dataset values are populated with respect to these 3 clusters. Once the clusters are obtained the labels are added to the dataset. For example : 1 is added as label to the datapoints belonging to cluster 1 and so on. The minimum distance between the datapoints are obtained and for each feature value the centroid is calculated and the datapoints nearest to these centroid are grouped to form the clusters.

Once all the labels are added to the datapoints they are trained in the KNN classifier function which is an implementation taken from Homework 1 assignment. Once the data is trained the test data present in “testData.csv” is given as input and the predictions are printed out in the console. The nearest neighbour classification is used to make the prediction for the test data.

Conclusion: The final output of the program gives us the list of the test data and their respective predictions based on the KNN classifier. The program can be modified to create more than 3 clusters, but we must make sure the number of clusters do not exceed the number of datapoints in the input dataset.

**Note: The code execution procedure and steps are provided in the README.txt file**