CSE4020 - ML Cyclesheet

File “MLdata2.txt” contains the data for k-means clustering with some outlier points.

Upload this file in Google Colab and load it to the numpy array ‘x’ as follows.

x = np.loadtxt("/MLdata2.txt")

Implement k-means clustering algorithm for this dataset. Handle the outliers (using IQR method preferably or any other method) and then apply k-means clustering for the processed data.

Use silhouette score to compare the performance of the algorithm before and after handling outliers.

Sample code:

from sklearn import metrics.silhouette\_score

kmeansmodel = KMeans(n\_clusters = 3, random\_state = 1).fit(X)

labels = kmeansmodel.labels\_

silhouette\_score(X, labels, metric = 'euclidean')

Note:

make\_blobs() method can be used to create synthetic data for clustering.

from sklearn.datasets import make\_blobs

x, clusterlabel = make\_blobs(n\_features=2, centers=4,random\_state=42)

# Here, x is the 2D array of data with size 100 X 2 and clusterlabel represents the id of the cluster to which each sample gets added to.

We can also pass two other useful parameters [n\_samples (default=100) and return\_centers (a Boolean parameter whose default value is false)] If it is True, it returns the centers of each cluster

MLdata2.txt has been created using make\_blobs() method.

For ‘weather.csv’, perform encoding of the categorical values using LabelEncoder() or OneHotEncoder() (whichever is appropriate) , perform missing value imputation using SimpleImputer() or KNNImputer(), balance the dataset using SMOTE() and use PCA to perform dimensionality reduction, then apply SVM classifier. Analyze its performance before and after balancing the dataset.

Sample code for using PCA:

from sklearn.decomposition import PCA

pca=PCA(n\_components=10) # out of 21 features (components), 10 principal components

#are selected

pca.fit(X)

Sample code for using SMOTE:

from imblearn.over\_sampling import SMOTE

sm=SMOTE(random\_state=42)

newx,newy=sm.fit\_resample(X,y)