Machine Learning Lab Assignment 4

Sourav Patra

Roll No:- 001811001044

Iris plants dataset: [***https://archive.ics.uci.edu/ml/datasets/Iris/***](https://archive.ics.uci.edu/ml/datasets/Iris/)

Wine Dataset: [***https://archive.ics.uci.edu/ml/datasets/wine***](https://archive.ics.uci.edu/ml/datasets/wine)

The code used for the this assignment is as follows:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def preprocess_cluster(X,y,te_size=0.3,label=False,scale=False,pca=False):

    if label:
        from sklearn.preprocessing import LabelEncoder
        y = LabelEncoder().fit_transform(y)

    if scale:
        from sklearn.preprocessing import StandardScaler
        sc = StandardScaler()
        X = sc.fit_transform(X)
        X = sc.transform(X)
        # X = pd.DataFrame(X)

    if pca:
        from sklearn.decomposition import PCA
        pca = PCA(n_components='mle')
        X = pca.fit_transform(X)
        X = pca.transform(X)
```

```
    return X, y

def tester_cluster(X,y,model):

    from sklearn.metrics import
rand_score,adjusted_rand_score,mutual_info_score,adjusted_mutual_info_scor
e,normalized_mutual_info_score,homogeneity_score,completeness_score,v_meas
ure_score

    print("Performance evaluation:")
    print('-----------------------------------')
    print('-----------------------------------')

    print("Rand Score")
    print(rand_score(y,model.labels_))

    print('-----------------------------------')

    print("Adjusted Rand Score")
    print(adjusted_rand_score(y,model.labels_))

    print('-----------------------------------')
    print('-----------------------------------')

    print("Mutual Info Score")
    print(mutual_info_score(y,model.labels_))

    print('-----------------------------------')

    print("Adjusted Mutual Info Score")
```

```
    print(adjusted_mutual_info_score(y,model.labels_))


    print('-----------------------------------')


    print("Normalized Mutual Info Score")
    print(normalized_mutual_info_score(y,model.labels_))


    print('-----------------------------------')
    print('-----------------------------------')


    print("Homogenity Score")
    print(homogeneity_score(y,model.labels_))


    print('-----------------------------------')


    print("Completeness Score")
```

```
    print(completeness_score(y,model.labels_))


    print('-----------------------------------')


    print("V-measure Score")
    print(v_measure_score(y,model.labels_))


def tester_cluster_2(X,y,model):


    from sklearn.metrics import
silhouette_score,calinski_harabasz_score,davies_bouldin_score


    print("Performance Evaluation:")
    print('-----------------------------------')
    print('-----------------------------------')
```

```
    print("Silhouette Coefficient")
    print(silhouette_score(X,model.labels_, metric='euclidean'))

    print('-----------------------------------')

    print("Calinski Harabasz Score")
    print(calinski_harabasz_score(X,model.labels_))

    print('-----------------------------------')

    print("Davies Bouldin Score")
    print(davies_bouldin_score(X,model.labels_))

    print('-----------------------------------')

def tester_cluster_3(X,y,model,kmeans=False):

    from scipy.cluster.vq import vq

    if kmeans:
        codebook = model.cluster_centers_
    else:
        codebook = []
        for i in np.unique(model.labels_):
```

```
            cluster_data = X[model.labels_ == i]
            centroid = cluster_data.mean(0)
            codebook.append(centroid)

    partition, euc_distance_to_centroids = vq(X, codebook)
```

```python
    tss = np.sum((X-X.mean(0))**2)
    sse = np.sum(euc_distance_to_centroids**2)
    ssb = tss - sse
    print("Cohesion Score")
    print(sse)
    print('-----------------------------------')
    print("Seperation Score")
    print(ssb)
    print('-----------------------------------')

df1 = pd.read_csv('data/iris.data', header=None)
X = np.array(df1.iloc[:,:-1])
y = np.array(df1.iloc[:,-1])

# df1 = pd.read_csv('data/wine.data', header=None)
# x = df1.iloc[:,1:]
# y = df1.iloc[:,0]

X, y = preprocess_cluster(X,y,te_size=0,label=True,scale=True)

from sklearn.cluster import KMeans
model = KMeans(n_clusters=3,n_init=100,max_iter=1000)

# from sklearn_extra.cluster import KMedoids
# model =
KMedoids(n_clusters=3,method='pam',init='k-medoids++',max_iter=1000)

# from sklearn.cluster import AgglomerativeClustering
# model = AgglomerativeClustering(n_clusters=3,linkage='single')

# from sklearn.cluster import Birch
# model = Birch(n_clusters=3)
```

```
# from sklearn.cluster import DBSCAN
# model = DBSCAN(eps=2,min_samples=8)

# from sklearn.cluster import OPTICS
# model = OPTICS(max_eps=2,min_samples=8)

model.fit(X)

tester_cluster_2(X,y,model)
tester_cluster_3(X,y,model,False)
```

Partition Based Algorithms:

i) Kmeans:-

The scores for iris dataset are:-

```
Performance Evaluation:
-----------------------------------
-----------------------------------
Silhouette Coefficient
0.4317682861510166 -------------------
----------------
Calinski Harabasz Score
152.3870943273455
-----------------------------------
Davies Bouldin Score
0.8205624685391698 -------------------
----------------
Cohesion Score
433.15661144203204
-----------------------------------
Seperation Score
898.060917092929
-----------------------------------
```
The

scores for wine dataset are:-

```
Performance Evaluation:
-----------------------------------
```

```
-----------------------------------
Silhouette Coefficient
0.2844772464432905 -------------------
----------------
Calinski Harabasz Score
111.72360529635833
-----------------------------------
Davies Bouldin Score
1.3010268749872544 -------------------
----------------
Cohesion Score
8505.241813812467
-----------------------------------
Seperation Score
10859.843192645298
-----------------------------------
```

ii)Kmeans++:-

This was pre-implemented in kmeans library function of scikit learn by using the ‘k-means++’ option in the init argument during model initialisation

The scores for iris dataset are:-

```
Performance Evaluation:
-----------------------------------
-----------------------------------
Silhouette Coefficient
0.4317682861510166 -------------------
----------------
Calinski Harabasz Score
152.3870943273455
-----------------------------------
Davies Bouldin Score
0.8205624685391698 -------------------
----------------
Cohesion Score
433.15661144203204
-----------------------------------
Seperation Score
898.060917092929
-----------------------------------
```

The scores for wine dataset are:-

```
Performance Evaluation:
-----------------------------------
-----------------------------------
Silhouette Coefficient
0.2844772464432905 -------------------
----------------
Calinski Harabasz Score
111.72360529635833
-----------------------------------
Davies Bouldin Score
1.3010268749872544 -------------------
----------------
Cohesion Score
8505.241813812467
----------------------------------
```

iii)Bisecting Kmeans:-

The code for this algorithm is as follows:

```
import pandas as pd
import numpy as np

df1 = pd.read_csv('data/iris.data', header=None)
X = np.array(df1.iloc[:,:-1])
y = np.array(df1.iloc[:,-1])

from sklearn.preprocessing import LabelEncoder
y = LabelEncoder().fit_transform(y)

from sklearn.cluster import KMeans
K = 3
current_clusters = 1
centroids = []
clusters = []
X_ = X
labels = [0]*X.shape[0]
label = 1
```

```
while current_clusters != K:
    kmeans = KMeans(n_clusters=2)
```

```
    kmeans.fit(X_)
    current_clusters += 1
    cluster_centers = kmeans.cluster_centers_

    sse = [0]*2
    for point, label in zip(X_, kmeans.labels_):
        sse[label] += np.square(point-cluster_centers[label]).sum()
    chosen_cluster = np.argmax(sse,axis=0)

    centroids.append(cluster_centers[1-chosen_cluster])
    clusters.append(X_[kmeans.labels_ == 1-chosen_cluster])

    chosen_cluster_data = X_[kmeans.labels_ == chosen_cluster]
    X_ = chosen_cluster_data

centroids.append(X_.mean(0))
clusters.append(X_)

labels = np.zeros(X.shape[0])
labels[clusters[0].shape[0]:clusters[0].shape[0]+clusters[1].shape[0]] = 1
labels[clusters[0].shape[0]+clusters[1].shape[0]:] = 2

labels = labels.astype(int)

from sklearn.metrics import
silhouette_score,calinski_harabasz_score,davies_bouldin_score

print("Performance Evaluation:")
print('-----------------------------------')
print('-----------------------------------')
```

```
print("Silhouette Coefficient")
print(silhouette_score(X,labels, metric='euclidean'))

print('-----------------------------------')

print("Calinski Harabasz Score")
print(calinski_harabasz_score(X,labels))

print('-----------------------------------')

print("Davies Bouldin Score")
print(davies_bouldin_score(X,labels))

print('-----------------------------------')

from scipy.cluster.vq import vq
codebook = centroids
partition, euc_distance_to_centroids = vq(X, codebook)

tss = np.sum((X-X.mean(0))**2)
sse = np.sum(euc_distance_to_centroids**2)
ssb = tss - sse
print("Cohesion Score")
print(sse)
print('-----------------------------------')
print("Seperation Score")
print(ssb)
print('-----------------------------------')
```

The scores for iris dataset are:-

```
Performance Evaluation:
-----------------------------------
-----------------------------------
```

```
Silhouette Coefficient
0.37352230022461397 ------------------
-----------------
Calinski Harabasz Score
243.27002984971273
-----------------------------------
Davies Bouldin Score
1.025154504871039 --------------------
---------------
Cohesion Score
80.09330502556426 --------------------
---------------
Seperation Score
600.7310949744357
-----------------------------------
```

The scores for wine dataset are:-

```
Performance Evaluation:
-----------------------------------
-----------------------------------
Silhouette Coefficient
0.13354160766718481 ------------------
-----------------
Calinski Harabasz Score
133.95058126266278
-----------------------------------
Davies Bouldin Score
16.93020288935997 --------------------
---------------
Cohesion Score
2498290.9965025433
-----------------------------------
Seperation Score
15094005.38700593
-----------------------------------
```

iv)Kmediods:-

The scores for iris dataset are:-

```
Performance Evaluation:
-----------------------------------
-----------------------------------
```

```
Silhouette Coefficient
0.4390041518129282 -------------------
----------------
Calinski Harabasz Score
151.00104809628613
-----------------------------------
Davies Bouldin Score
0.7970734618618747 -------------------
----------------
Cohesion Score
435.22340023159717
-----------------------------------
Seperation Score
895.9941283033638 --------------------

--------------
```

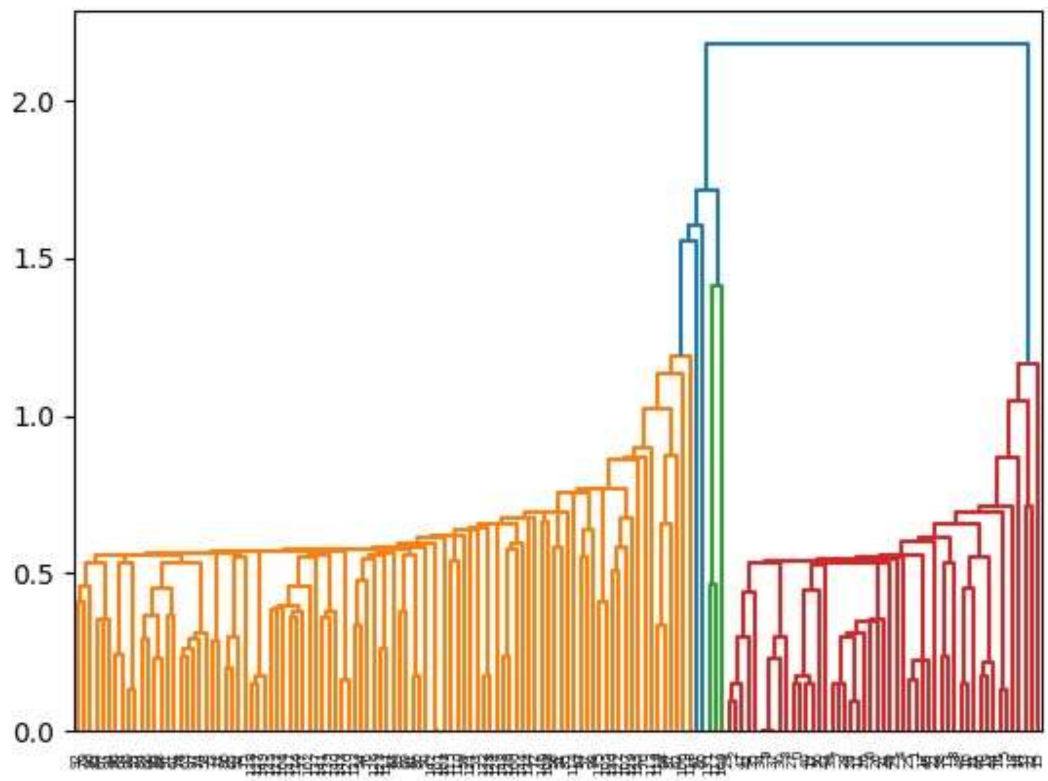
The scores for wine dataset are:-

```
Performance Evaluation:
-----------------------------------
-----------------------------------
Silhouette Coefficient
0.25788384772276307 ------------------
-----------------
Calinski Harabasz Score
107.4402890033373
-----------------------------------
Davies Bouldin Score
1.3582758881134855 -------------------
----------------
Cohesion Score
8652.61794565623
-----------------------------------
Seperation Score
10712.467060801535
----------------------------------
```

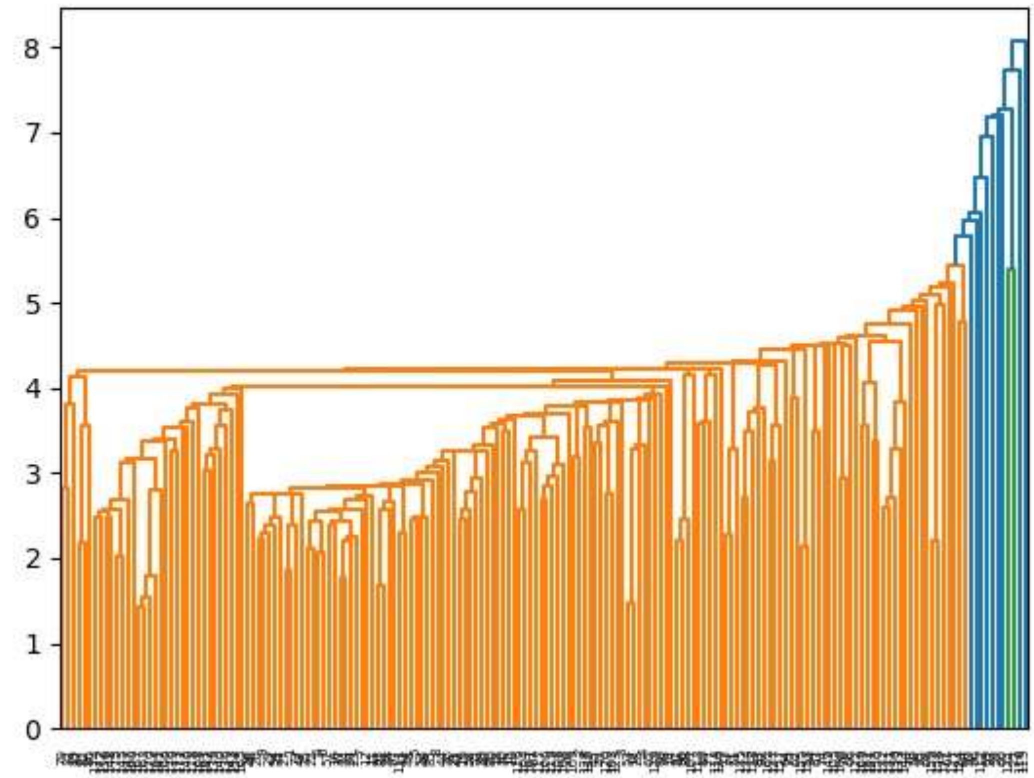
Hierarchical Algorithms:

i)Dendrogram:-

The graph for iris dataset:-



The graph for wine dataset:-



ii)AGNES:-

The scores for iris dataset are:-

```
Performance Evaluation:
-----------------------------------
-----------------------------------
Silhouette Coefficient
0.25788384772276307 ------------------
-----------------
Calinski Harabasz Score
107.4402890033373
-----------------------------------
Davies Bouldin Score
1.3582758881134855 -------------------
----------------
Cohesion Score
8652.61794565623
```

```
----------------------------------
Seperation Score
10712.467060801535
```

The scores for wine dataset are:-

```
Performance Evaluation:
-----------------------------------
-----------------------------------
Silhouette Coefficient
0.05506029279005502 ------------------
-----------------
Calinski Harabasz Score
2.393717637307116 --------------------
---------------
Davies Bouldin Score
0.6534805641366035 -------------------
----------------
Cohesion Score
17730.974867562945
-----------------------------------
Seperation Score
1634.110138894819
----------------------------------
```

iii)BIRCH:-

The scores for iris dataset are:-

```
Performance Evaluation:
-----------------------------------
-----------------------------------
Silhouette Coefficient
0.4207655357916486 -------------------
----------------
Calinski Harabasz Score
146.28041877203535
-----------------------------------
Davies Bouldin Score
0.8388514354531184
```

```
-----------------------------------
Cohesion Score
440.9596839852269
------------------------------------
Seperation Score
890.2578445497342
```

The scores for wine dataset are:-

```
Performance Evaluation:
------------------------------------
------------------------------------
Silhouette Coefficient
0.26428431893338583 ------------------
-----------------
Calinski Harabasz Score
107.19355782534467
------------------------------------
Davies Bouldin Score
1.3653874210280403 -------------------
----------------
Cohesion Score
8568.906004725632
------------------------------------
Seperation Score
10796.179001732133
------------------------------------
```

Density Based Algorithms:

i)DBSCAN:-

The scores for iris dataset are:-

```
Performance Evaluation:
------------------------------------
------------------------------------
Silhouette Coefficient
```

```
0.4672532531297798 --------------------
----------------
Calinski Harabasz Score
86.97354665917852 --------------------
---------------
Davies Bouldin Score
0.6285907442747066 -------------------
----------------
Cohesion Score
590.532907833632
-----------------------------------
Seperation Score
740.6846207013291
```

The scores for wine dataset are:-

```
Performance Evaluation:
-----------------------------------
-----------------------------------
Silhouette Coefficient
0.5587229611951657 -------------------
----------------
Calinski Harabasz Score
237.70687070972235
-----------------------------------
Davies Bouldin Score
0.6551076423025366 -------------------
----------------
Cohesion Score
5019958.209778264
-----------------------------------
Seperation Score
12572338.17373021
-----------------------------------
```

ii)OPTICS:-

The scores for iris dataset are:-

```
Performance Evaluation:
-----------------------------------
-----------------------------------
Silhouette Coefficient
```

```
-0.2538505885661452
----------------------------------
Calinski Harabasz Score
19.3913120185153 ---------------------
--------------
Davies Bouldin Score
2.83428709567964 ---------------------
--------------
Cohesion Score
210.68419432792155
-----------------------------------
Seperation Score
470.1402056720784
----------------------------------
```

The scores for wine dataset are:-

```
Performance Evaluation:
-----------------------------------
-----------------------------------
Silhouette Coefficient
0.013388601390969089 -----------------
------------------
Calinski Harabasz Score
25.417115659353996 -------------------
----------------
Davies Bouldin Score
0.9717411608615347 -------------------
----------------
Cohesion Score
10029749.502308398
-----------------------------------
Seperation Score
7562546.881200075
-----------------------------------
```

Finally summarising all scores for each model and dataset:

		Kmeans	Kmeans++	Bisecting Kmeans	Kmediods	AGNES	BIRCH	DBSCAN	OPTICS

Iris dataset	Silhouette Coeff.	0.4317682861510166	0.4317682861510166	0.37352230022461397	0.4390041518129282	0.25788384772276307	0.4207655357916486	0.4672532531297798	-0.2538505885661452
	Calsinki Harabsz Score	152.3870943273455	152.3870943273455	243.27002984971273	151.00104809628613	107.4402890033373	146.28041877203535	86.97354665917852	19.3913120185153
	Davies Bouldin Score	0.8205624685391698	0.8205624685391698	1.025154504871039	0.7970734618618747	1.3582758881134855	0.8388514354531184	0.6285907442747066	2.83428709567964
	Cohesion Score	433.15661144203204	433.15661144203204	80.09330502556426	435.22340023159717	8652.61794565623	440.9596839852269	590.532907833632	210.68419432792155
	Seperation Score	898.060917092929	898.060917092929	600.7310949744357	895.9941283033638	10712.467060801535	890.2578445497342	740.6846207013291	470.1402056720784
Wine dataset	Silhouette Coeff.	0.2844772464432905	0.2844772464432905	0.13354160766718481	0.25788384772276307	0.05506029279005502	0.26428431893338583	0.5587229611951657	0.013388601390969089
	Calsinki Harabsz Score	111.72360529635833	111.72360529635833	133.95058126266278	107.4402890033373	2.393717637307116	107.19355782534467	237.70687070972235	25.417115659353996
	Davies Bouldin Score	1.3010268749872544	1.3010268749872544	16.93020288935997	1.3582758881134855	0.6534805641366035	1.3653874210280403	0.6551076423025366	0.9717411608615347
	Cohesion Score	8505.241813812467	8505.241813812467	2498290.9965025433	8652.61794565623	17730.974867562945	8568.906004725632	5019958.209778264	10029749.502308398
	Seperation Score	10859.843192645298	10859.843192645298	15094005.38700593	10712.467060801535	1634.110138894819	10796.179001732133	12572338.17373021	7562546.881200075