Lab 11. Hierarchical and DBSCAN clustering

You are given with two data sets. Both folders contain "2D_points.txt" file having points belonging to 2 classes. Write a program to do the following.

1. Apply K-means (K=2) clustering on the original data. Plot the points in these clusters using different colors.

2. Apply Agglomerative clustering with (2 clusters) on the data. Plot the points in these clusters using different colors. Compare with the clustering obtained by K-means approach.
(Use **`sklearn.cluster.AgglomerativeClustering`**)

3. i) Apply DBSCAN clustering with default parameters and compare the results.
ii) Vary the parameter ***eps*** *(maximum distance between two samples to be considered) to 0.05, 0.5 and 0.95 and observe the results. Vary* ***min_samples*** *(The number of samples in neighbourhood**)** to 1,10,30 and 50 and observe the resuts.*
(Use **`sklearn.cluster.DBSCAN`**)

4. Obtain the purity score for both clustering methods. Sample code snippet is given below.

(For **NLS** data, assume first 500 points belong to class 0 and rest belong to class 1. For **Ring** data, assume first 150 points belong to class 0 and rest all points in class 1.)

```
##############################################
# Purity score
import numpy as np
from sklearn import metrics
from scipy.optimize import linear_sum_assignment

def purity_score(y_true, y_pred):
    # compute contingency matrix (also called confusion matrix)
    contingency_matrix    =    metrics.cluster.contingency_matrix(y_true,
y_pred)
    print(contingency_matrix)

    # Find optimal one-to-one mapping between cluster labels and true
labels
    row_ind, col_ind = linear_sum_assignment(-contingency_matrix)

    # Return cluster accuracy
    return       contingency_matrix[row_ind,       col_ind].sum()       /
np.sum(contingency_matrix)
```