# Lab 11. Hierarchical and DBSCAN clustering

You are given with Iris flower dataset file (`Iris.scv`). The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimetres.

1. Apply PCA and select first two directions to convert the data in to 2D. (Exclude the attribute “Species” for PCA)

2. Apply Agglomerative clustering with 3 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 1,5,10 and 20 and observe the resuts.*
(Use **`sklearn.cluster.DBSCAN`** )

4. Obtain and compare the purity score for all the clustering methods. Sample code snippet is given below.

```
#######################################################
#Purity score
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)
```