**Department of Computing**

**CS370: Artificial Intelligence**

**Class: BSCS-10AB**

**Lab 11: Random Forest Classifier**

**Date: 27-04-2023**

**Instructor: Dr. Hashir Kiani**

**Lab Engineer: Ms Shakeela Bibi**

# Lab 11: Random Forest Classifier

**Introduction:**

Bagging identification trees often ends up giving us K trees that are highly correlated. If there is one attribute with a very low disorder, it is likely to be the root of most of the trees. The random forest model is a form of bagging in which we take extra steps to make the ensemble of trees more diverse. The key idea is to randomly vary the attribute choices rather than the training examples. At each split point in constructing the tree we select a random sampling of attributes and then compute which of those have the lowest disorder. If there are 𝑑 attributes, a common default choice is that each split randomly picks attributes to consider for classification problems. Random forests reduce the problem of overfitting.

**Lab Task:**

In this lab, you are supposed to implement the Random Forest Classifier. Data in excel files (both the training and test sets) are uploaded on LMS (same dataset that was used in previous labs). In the said training and test data files, each row contains data about one instance of a plant category where four predictors/attributes are recorded for each plant (namely, leaf length, leaf width, flower length, and flower width), while “plant” is the target class which could be any one of the following at a time: “Arctica” or “Harlequin” or “Caroliniana”.

You are supposed to develop random forests with

a. 100 trees

b. 300 trees

c. 500 trees

Figure out how the results vary on the test data (How results vary by increasing the number of trees). Please submit your results for the complete test set using the three abovementioned random forests (a, b, and c).

Note: Scikit-learn might be a useful resource during implementation, feel free to use this-or any other suitable library.

**Training Set (70 % Training Set / 30 % Testing Set 120 Values)**

**Code:**

**# Data Processing**

**import pandas as pd**

**import numpy as np**

**# Modelling**

**from sklearn.ensemble import RandomForestClassifier**

**from sklearn.metrics import accuracy\_score, confusion\_matrix, precision\_score, recall\_score, ConfusionMatrixDisplay**

**from sklearn.model\_selection import RandomizedSearchCV, train\_test\_split**

**from scipy.stats import randint**

**# Tree Visualisation**

**from sklearn.tree import export\_graphviz**

**from IPython.display import Image**

**import graphviz**

**train\_data = pd.read\_excel('/content/TrainingSet.xlsx', engine='openpyxl')**

**print(train\_data)**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(train\_data.iloc[:, :-1], train\_data.iloc[:, -1], test\_size=0.3,random\_state=109) # 70% training and 30% test**

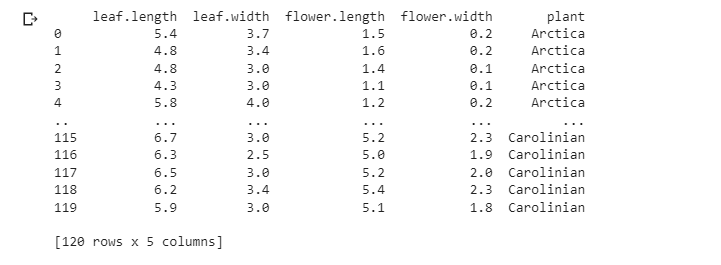
**X\_train, X\_test, y\_train, y\_test = train\_test\_split(train\_data.iloc[:, :-1], train\_data.iloc[:, -1], test\_size=0.3,random\_state=109) # 70% training and 30% test**

**accuracy = accuracy\_score(y\_test, Prediction)**

**print("Accuracy:", accuracy)**

**Screenshot:**

**Dataset**

****

**Testing Set (30 Values)**

**Code:**

**# Data Processing**

**import pandas as pd**

**import numpy as np**

**# Modelling**

**from sklearn.ensemble import RandomForestClassifier**

**from sklearn.metrics import accuracy\_score, confusion\_matrix, precision\_score, recall\_score, ConfusionMatrixDisplay**

**from sklearn.model\_selection import RandomizedSearchCV, train\_test\_split**

**from scipy.stats import randint**

**# Tree Visualisation**

**from sklearn.tree import export\_graphviz**

**from IPython.display import Image**

**import graphviz**

**test\_data = pd.read\_excel('/content/TestingSet.xlsx', engine='openpyxl')**

**print(test\_data)**

**X\_test=test\_data.iloc[:, :-1]**

**#print(X\_test)**

**print("")**

**print("")**

**Prediction = rf.predict(X\_test)**

**print("Predicted Values:")**

**print(Prediction)**

**print("")**

**print("Number of Predicted Values:")**

**print(len(Prediction))**

**test\_data = pd.read\_excel('/content/TestingSet.xlsx', engine='openpyxl')**

**test\_data.iloc[:, -1] = Prediction**

**test\_data.to\_excel('/content/TestingSet.xlsx', index=False)**

**#FOR Tree n=100**

**# Export the first 100 decision trees from the forest**

**n=3**

**for i in range(n):**

**tree = rf.estimators\_[i]**

**dot\_data = export\_graphviz(tree,**

**feature\_names=X\_test.columns,**

**filled=True,**

**max\_depth=2,**

**impurity=False,**

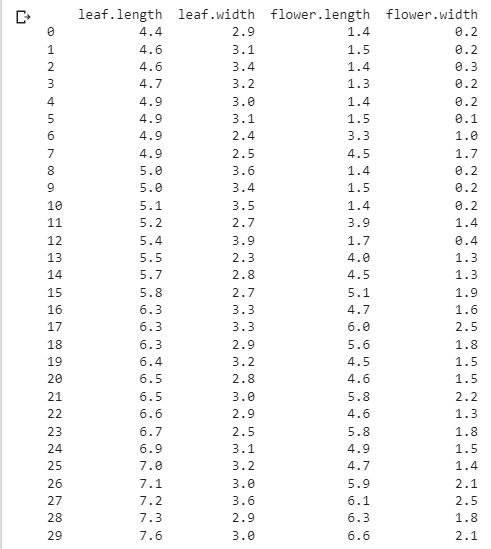
**proportion=True)**

**graph = graphviz.Source(dot\_data)**

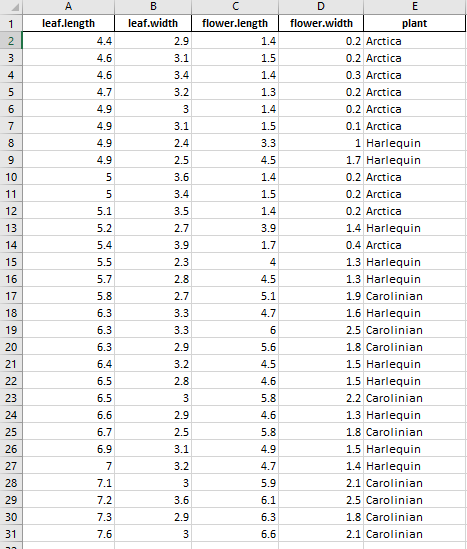
**display(graph)**

**Screenshot:**

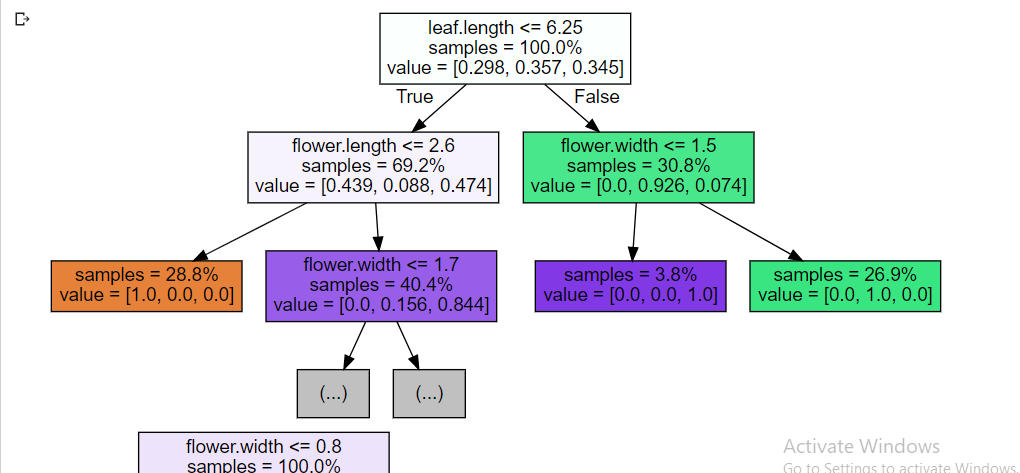
**Dataset**

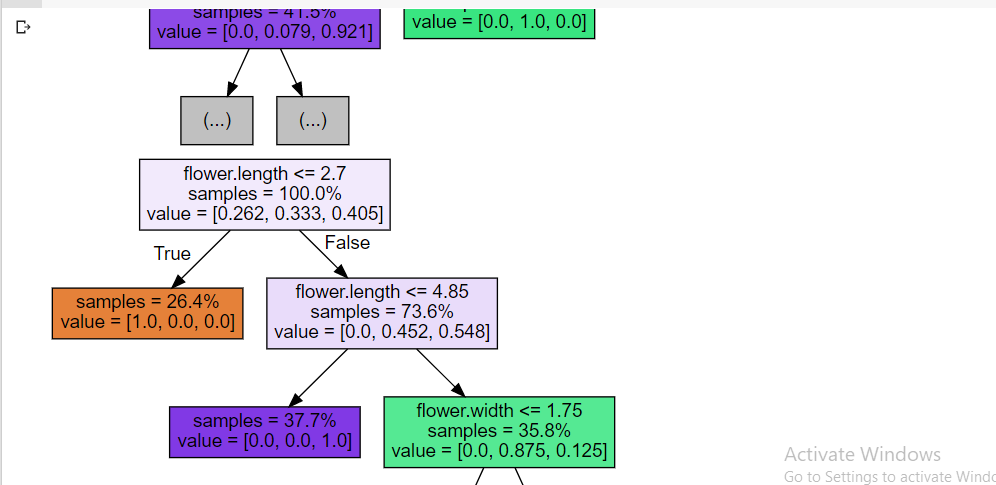
****

**Screenshot (Excel):**

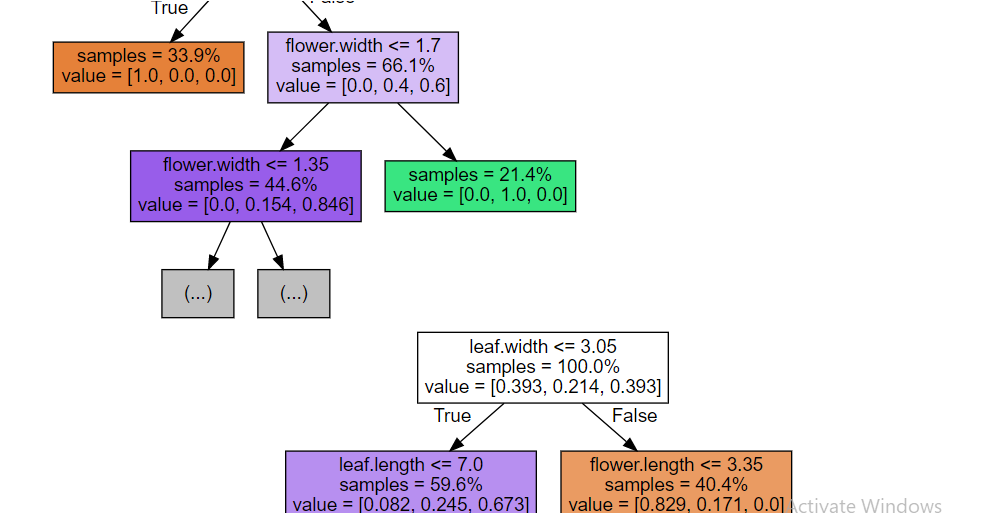
****

**Screenshot (100 Trees) :**

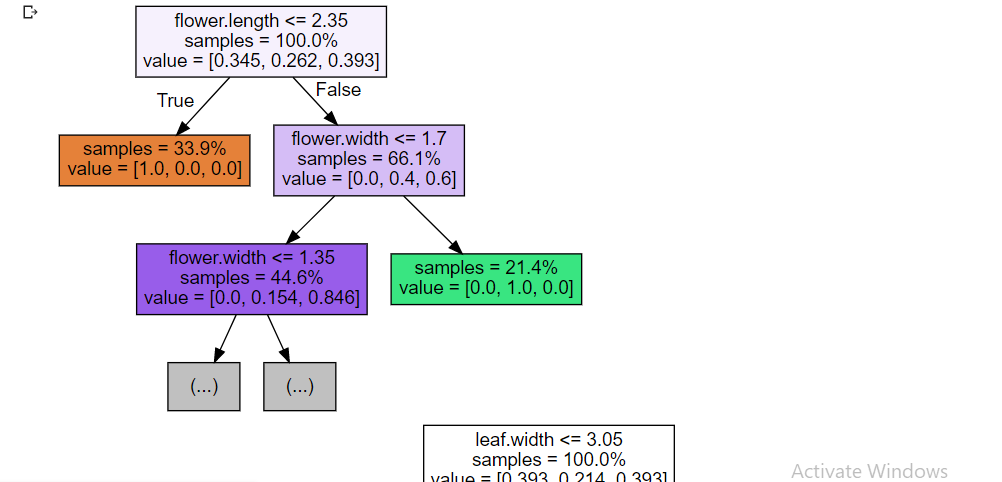
****

****

**Screenshot (300 Trees) :**

****

**Screenshot (500 Trees) :**

****