**Lab Assignment 2**

Arundhati Das

BT17GCS016

**Objective:** Building Univariate Decision Tree and understanding the effect of various choices.

Download the “Seeds” dataset from the following link: <https://archive.ics.uci.edu/ml/datasets/seeds>

It is a tab separated text file that contains numeric values for seeds of three different varieties of wheat with 7 features. Thus we have three classes and seven features in our dataset.

The purpose of this exercise is that you should be able to observe the effect of the choices that are available when using ID3 for building univariate trees.

1. Choice regarding training set: Choose 70% of samples randomly for training. Build the tree and write down the corresponding decision rules. Use Information Gain for finding the split point. Now choose some other 70% samples for training and again build the tree (using IG) and the corresponding rules. Did the rules change? Why? What was the accuracy for the two cases? You can check the accuracy by predicting the class of the remaining 30% of samples and then checking versus the actual class.
2. Choice regarding impurity measure: Repeat the task of (a) above but use Ginni Index for finding the split point instead of IG. Did the rules change from (a)? Why? Which impurity measure gave a higher accuracy?

**Code:**

!pip install --upgrade scikit-learn==0.20.3

import pandas as pd

import graphviz

from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier

from sklearn.model\_selection import train\_test\_split # Import train\_test\_split function

from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation

data = pd.read\_csv('seeds\_dataset.txt', sep=" ", header=None)

data.columns = ["area", "Perimeter", "Compactness", "Length of Kernel","Width of Kernel","Asymmetry Coefficient","Length of Kernel Grove","Class"]

feature\_cols = ["area", "Perimeter", "Compactness", "Length of Kernel","Width of Kernel","Asymmetry Coefficient","Length of Kernel Grove"]

X = data[feature\_cols]

y = data.Class

print(data)

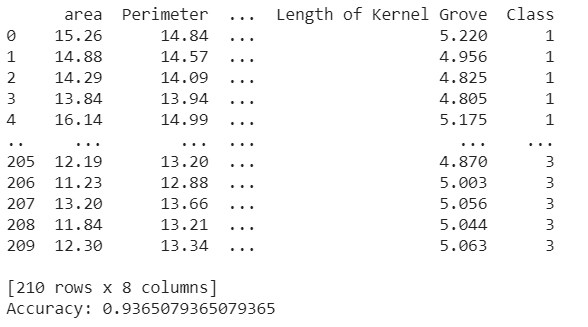
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=1)

clf = DecisionTreeClassifier()

clf = clf.fit(X\_train,y\_train)

y\_pred = clf.predict(X\_test)

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))



from sklearn.tree import export\_graphviz

from sklearn.externals.six import StringIO

from IPython.display import Image

import pydotplus

dot\_data = StringIO()

export\_graphviz(clf, out\_file=dot\_data,

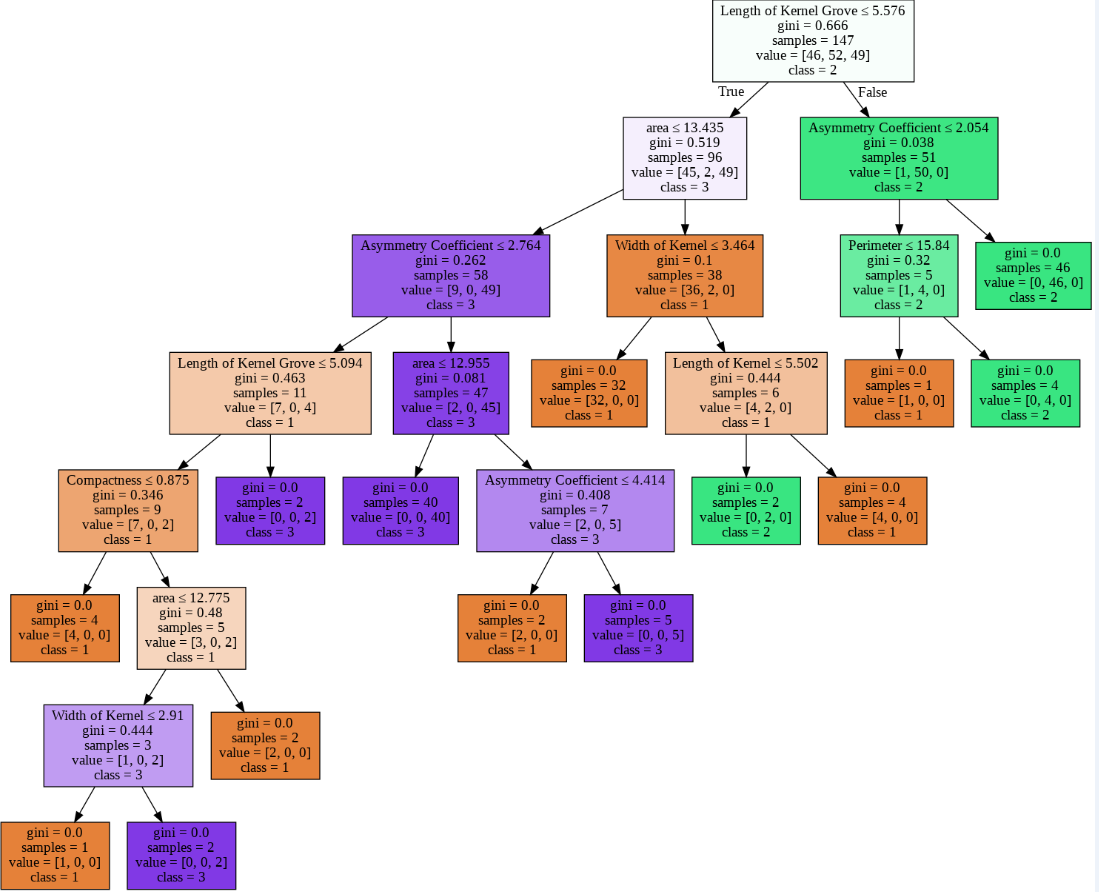
                filled=True, rounded=False,

                special\_characters=True,feature\_names = feature\_cols,class\_names=['1','2','3'])

graph = pydotplus.graph\_from\_dot\_data(dot\_data.getvalue())

graph.write\_png('gini.png')

Image(graph.create\_png())



clf\_entropy=DecisionTreeClassifier(criterion='entropy')

dt\_entropy=clf\_entropy.fit(X\_train,y\_train)

dt\_entropy

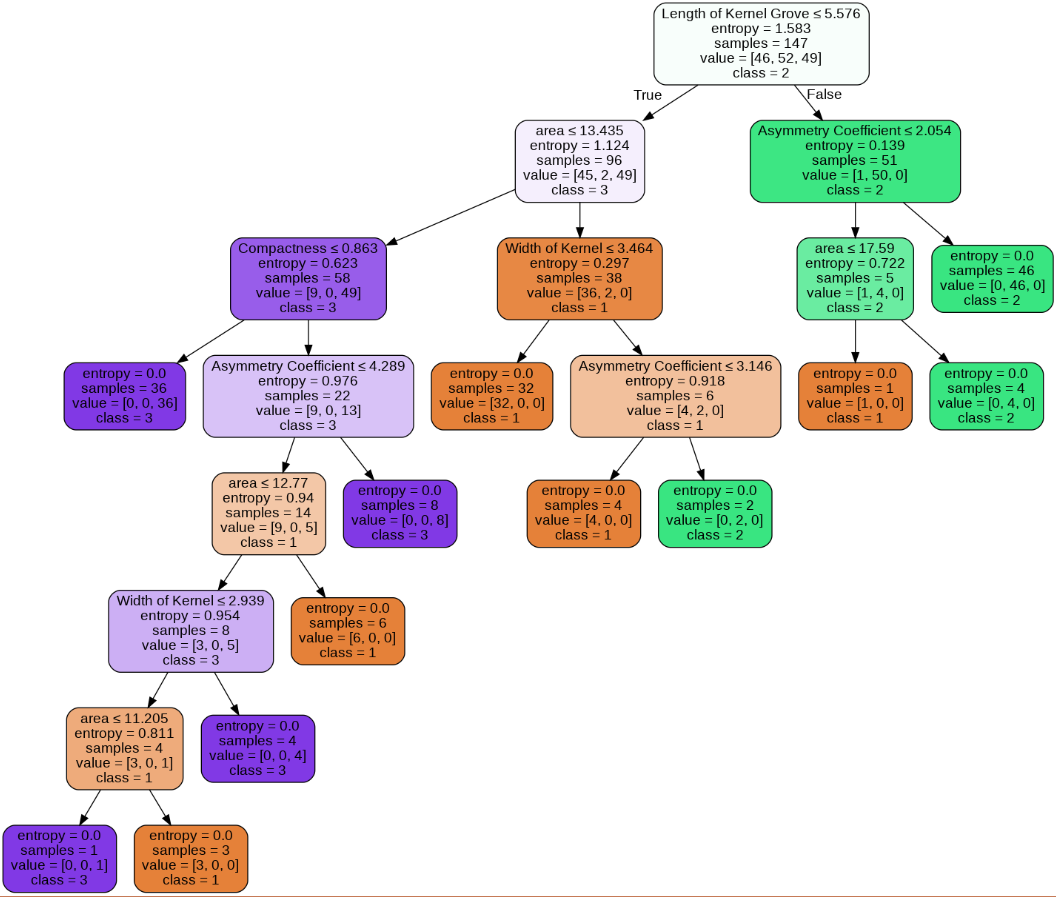
dot\_data\_entropy=StringIO()

export\_graphviz(dt\_entropy,out\_file=dot\_data\_entropy,filled=True,rounded=True,special\_characters=True,feature\_names=feature\_cols,class\_names=['1','2','3'])

graph\_entropy=pydotplus.graph\_from\_dot\_data(dot\_data\_entropy.getvalue())

graph\_entropy.write\_jpg('train\_entropy.jpg')

Image(graph\_entropy.create\_png())



y\_pred\_entropy=clf\_entropy.predict(X\_test)

print("Accuracy : ",(metrics.accuracy\_score(y\_test,y\_pred\_entropy)))



**Analysis:**

The accuracy obtained after IG was 92.06% whereas the accuracy obtained after using Ginni Index came out to be 93.65%