**Assignment- 4: IRIS Dataset**

**Created by:**

**Srinidhi Devan**

***Step-1: Importing the necessary libraries***

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn import tree

from sklearn.tree import DecisionTreeClassifier, plot\_tree

from sklearn.model\_selection import train\_test\_split

* Here, apart from NumPy and Pandas, the additional libraries imported are: ‘tree’, ‘DecisionTreeClassifier’, ‘plot\_tree’ and ‘train\_test\_split’ from ‘sklearn’ module of Python for this particular problem.
* ‘DecisionTreeClassifier’: Facilitates in building a decision tree model in Python
* ‘plot\_tree’: For tree visualization purpose
* ‘train\_test\_split’: For splitting the data into train (70%) & test (30%) set respectively

***Step-2: Reading the Iris Dataset***

data = pd.read\_csv('Iris.csv')

data = data.drop(['Id'],axis=1)

* Using ‘pd.read\_csv’ command
* Dropping the ‘Id’ variable from the dataset; axis=1 implies we are dropping from column

***Step-3: Separating the Independent and Dependent variables in the data***

x = data[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]

y = data['Species']

* Here, the Independent Variables are: 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm'.

This is called ‘x’

* The Dependent Variable is: 'Species'

This is called ‘y’

***Step-4: Splitting the Dataset into Train & Test set***

(x\_train, x\_test, y\_train, y\_test) = train\_test\_split(x, y, train\_size=0.7, random\_state=1)

* The train set consists of 70% of the dataset and test set consists of 30%
* The splitting is done using ‘train\_test\_split’ command
* 'random\_state'--> to ensure that there is always uniformity in splitting

***Step-5: Building the Decision Tree model***

dt\_model = DecisionTreeClassifier(criterion = 'gini' )

* This is done using ‘DecisionTreeClassifier’ package in Python

dt\_model.fit(x\_train, y\_train)

* And we fit the model for further analysis—using ‘.fit()’ for training set (including both Independent (x) and dependent variables(y))

***Step-6: Finding the Feature Importance***

pd.DataFrame(dt\_model.feature\_importances\_, columns = ["Imp"], index = x\_train.columns)

* This will show us which of the Independent variables have more weightage relative to the other.



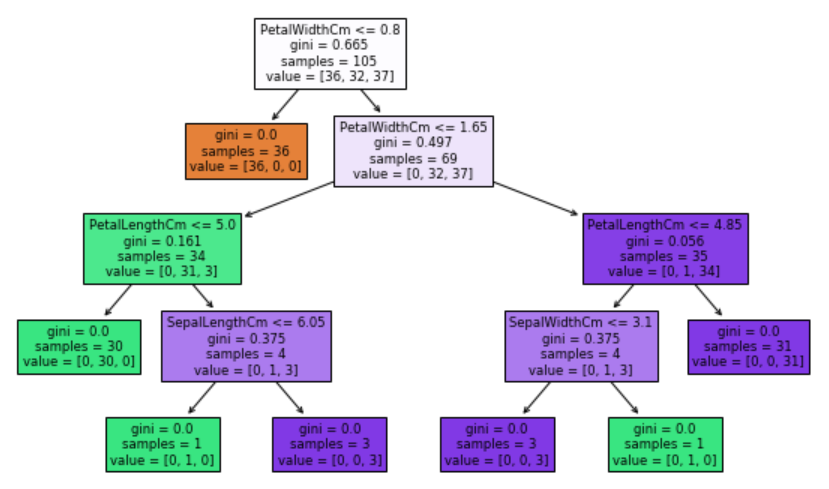
* From the above table, we can see that ‘PetalWidthCm’ has the highest importance followed by ‘PetalLengthCm’, ‘SepalWidthCm’ and ‘SepalLengthCm’.
* This is done on independent variables of the training dataset.

***Step-7: Plotting the Decision Tree***

fig = tree.plot\_tree(dt\_model,feature\_names=data.columns,filled='True')

* This is done with the package ‘plot\_tree’ on ‘dt\_model’

Where dt\_model = DecisionTreeClassifier(criterion = 'gini' )



***Step-8: Creating Classification Report***

* This is done on both Train and Test dataset
* Classification Report shows us the ‘Precision’, ‘Recall’, ‘F1 Score’ and ‘Accuracy’

classification\_report(y\_train, ytrain\_predict)

* Where

ytrain\_predict = dt\_model.predict(x\_train)

Classification Report for Train Set



1. PRECISION: Percentage of Iris-setosa correctly predicted is 100%

Percentage of Iris-versicolor correctly predicted is 100%

Percentage of Iris-virginica correctly predicted is 100%

1. RECALL: Percentage of positive cases in Iris-setosa is 100%

Percentage of positive cases in Iris-versicolor is 100%

Percentage of positive cases in Iris-virginica is 100%

1. F1-SCORE: Percentage of positive predictions in Iris-setosa which were correct is 100%

Percentage of positive predictions in Iris-versicolor which were correct is 100%

Percentage of positive predictions in Iris- virginica which were correct is 100%

1. ACCURACY: 100%

classification\_report(y\_test, ytest\_predict)

* Where

ytest\_predict = dt\_model.predict(x\_test)

Classification Report for Test Set



1. PRECISION: Percentage of Iris-setosa correctly predicted is 100%

Percentage of Iris-versicolor correctly predicted is 94%

Percentage of Iris-virginica correctly predicted is 92%

1. RECALL: Percentage of positive cases in Iris-setosa is 100%

Percentage of positive cases in Iris-versicolor is 94%

Percentage of positive cases in Iris-virginica is 92%

1. F1-SCORE: Percentage of positive predictions in Iris-setosa which were correct is 100%

Percentage of positive predictions in Iris-versicolor which were correct is 94%

Percentage of positive predictions in Iris- virginica which were correct is 92%

1. ACCURACY: 96%

***Step-9: Accuracy***

* This is found using Confusion Matrix as shown above.
* Alternatively,

dt\_model.score(x\_train,y\_train)

1. **For Train Dataset: 100%**

dt\_model.score(x\_test,y\_test)

1. **For Test Dataset: 95.55%**