**Computer Engineering Department National University of Technology Islamabad, Pakistan**

**Introduction to Data Mining**

**Practice Exercise 10**

****

Name: Muhammad Sami Uddin Rafay

Roll Number: F18604013

Submitted To: Dr. Kamran Javed

Date: 02 February 2020

**Practice Exercise 10**

**Supervised Machine Learning | Decision Tree**

**Objective:**

* Implement Decision Tree using Fisher iris Data.

**Equipment/Software Required:**

* Python (Spyder 4.0 Anaconda Distribution)

**Background:**

**Decision Tree:**

A decision tree is a structure that contains nodes (rectangular boxes) and edges(arrows) and is built from a dataset (table of columns representing features/attributes and rows corresponds to records). Each node is either used to make a decision (known as decision node) or represent an outcome (known as leaf node).

**Main Steps of ID3 Algorithm:**

1. Compute the entropy of Dataset ENTROPY(S)
2. For every attribute/feature: -
3. Calculate entropy for all other values ENTROPY(A)
4. Take AVERAGE INFORMATION ENTROPY for current attribute
5. Calculate Gain for current attribute
6. Pick the Highest Gain attribute
7. Repeat until we get the tree we desire

**Code:**

**# importing Libraries**

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.datasets import load\_iris

from sklearn import tree

**# Loading iris dataset**

fisher\_iris=pd.read\_csv(r"C:\Users\User\Downloads\iris.data")

**# Before implementing Decision Tree on fisher\_iris Data frame lets do some data quality assessment**

**# Converting fisher\_iris as a Data Frame**

fisher\_iris=pd.DataFrame(fisher\_iris)

**# Printing the head of fisher\_iris Data Frame**

print("Head of fisher\_iris: \n", fisher\_iris.head())

print("\n")

**# Spcifying the columns name**

attributes = ["sepal\_length", "sepal\_width", "petal\_length", "petal\_width", "class"]

fisher\_iris.columns = attributes

**# Printing shape/ Dimentions fisher\_iris of Data Frame**

print("Head of fisher\_iris: \n",fisher\_iris.shape)

print("\n")

**# Printing statistics of fisher\_iris Data Frame**

print("Head of fisher\_iris: \n", fisher\_iris.describe())

print("\n")

**#plotting the decision tree for iris dataset**

plt.figure(1)

**# Slicing Train and Target Data**

X, y = load\_iris(return\_X\_y=True)

**# Using Decision Tree Classifier**

clf = tree.DecisionTreeClassifier()

**# Fitting target and Train Data**

clf = clf.fit(X, y)

**# plotting Decision Tree**

tree.plot\_tree(clf)

anypoint=[[3., 2., 3, 4]]

**# predicting**

prediction=clf.predict(anypoint)

print("\n")

print("Prediction of any point: \n",prediction)

**Output:**

**Head of fisher\_iris:**

5.1 3.5 1.4 0.2 Iris-setosa

0 4.9 3.0 1.4 0.2 Iris-setosa

1 4.7 3.2 1.3 0.2 Iris-setosa

2 4.6 3.1 1.5 0.2 Iris-setosa

3 5.0 3.6 1.4 0.2 Iris-setosa

4 5.4 3.9 1.7 0.4 Iris-setosa

**Head of fisher\_iris:**

(149, 5)

**Head of fisher\_iris:**

**sepal\_length sepal\_width petal\_length petal\_width**

**count** 149.000000 149.000000 149.000000 149.000000

**mean** 5.848322 3.051007 3.774497 1.205369

**std** 0.828594 0.433499 1.759651 0.761292

**min** 4.300000 2.000000 1.000000 0.100000

**25%** 5.100000 2.800000 1.600000 0.300000

**50%** 5.800000 3.000000 4.400000 1.300000

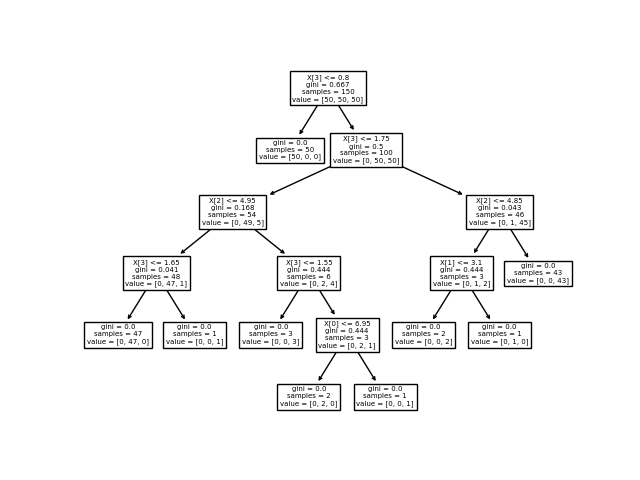
**75%** 6.400000 3.300000 5.100000 1.800000

**max** 7.900000 4.400000 6.900000 2.500000

**Prediction of any point:**

[2]

**Graphs:**

****

**Results and Discussions:**

**Decision Trees (DTs)** are a non-parametric supervised learning method used for [classification](https://scikit-learn.org/stable/modules/tree.html#tree-classification) and [regression](https://scikit-learn.org/stable/modules/tree.html#tree-regression). The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

**Conclusion:**

[Decision Tree](https://www.geeksforgeeks.org/decision-tree/) is one of the most powerful and popular algorithms. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.