**Practical – 7**

Aim – Implementation of Decision Trees **Code**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn import metrics

import seaborn as sns

from sklearn.datasets import load\_iris

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

iris = load\_iris()

data = pd.DataFrame(data = iris.data, columns = iris.feature\_names) data['Species'] = iris.target

target = np.unique(iris.target)

target\_n = np.unique(iris.target\_names)

target\_dict = dict(zip(target, target\_n))

data['Species'] = data['Species'].replace(target\_dict)

x = data.drop(columns = "Species") y = data["Species"] names\_features = x.columns target\_labels = y.unique()

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.3, random\_state = 93)

from sklearn.tree import DecisionTreeClassifier

dtc = DecisionTreeClassifier(max\_depth = 3, random\_state = 93) dtc.fit(x\_train, y\_train)

plt.figure(figsize = (30, 10), facecolor = 'b')

Tree = tree.plot\_tree(dtc, feature\_names = names\_features, class\_names = target\_labels, rounded = True, filled = True, fontsize = 14)

plt.show()

y\_pred = dtc.predict(x\_test)

confusion\_matrix = metrics.confusion\_matrix(y\_test, y\_pred) matrix = pd.DataFrame(confusion\_matrix) axis = plt.axes() sns.set(font\_scale = 1.3)

plt.figure(figsize = (10,7))

sns.heatmap(matrix, annot = True, fmt = "g", ax = axis, cmap = "magma") axis.set\_title('Confusion Matrix')

axis.set\_xlabel("Predicted Values", fontsize = 10) axis.set\_xticklabels([''] + target\_labels)

axis.set\_ylabel( "True Labels", fontsize = 10) axis.set\_yticklabels(list(target\_labels), rotation = 0)

plt.show()

