EXPERIMENT – 6 (F)

import numpy as np

import pandas as pd

# Importing the dataset

dataset = pd.read\_csv("/content/IRIS.csv ")

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, -1].values

# Splitting the dataset into the Training set and Test set

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25, random\_state = 8)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

# Training the Decision Tree Classification model on the Training set

from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier(criterion = 'entropy', random\_state = 5)

classifier.fit(X\_train, y\_train)

# Display the Decision Tree

from sklearn.tree import plot\_tree

import matplotlib.pyplot as plt

plt.figure(figsize=(20,10))

plot\_tree(classifier, filled=True, rounded=True, feature\_names=dataset.columns[:-1])

plt.show()

# Predicting the Test set results

y\_pred = classifier.predict(X\_test)

# Display the results (confusion matrix and accuracy)

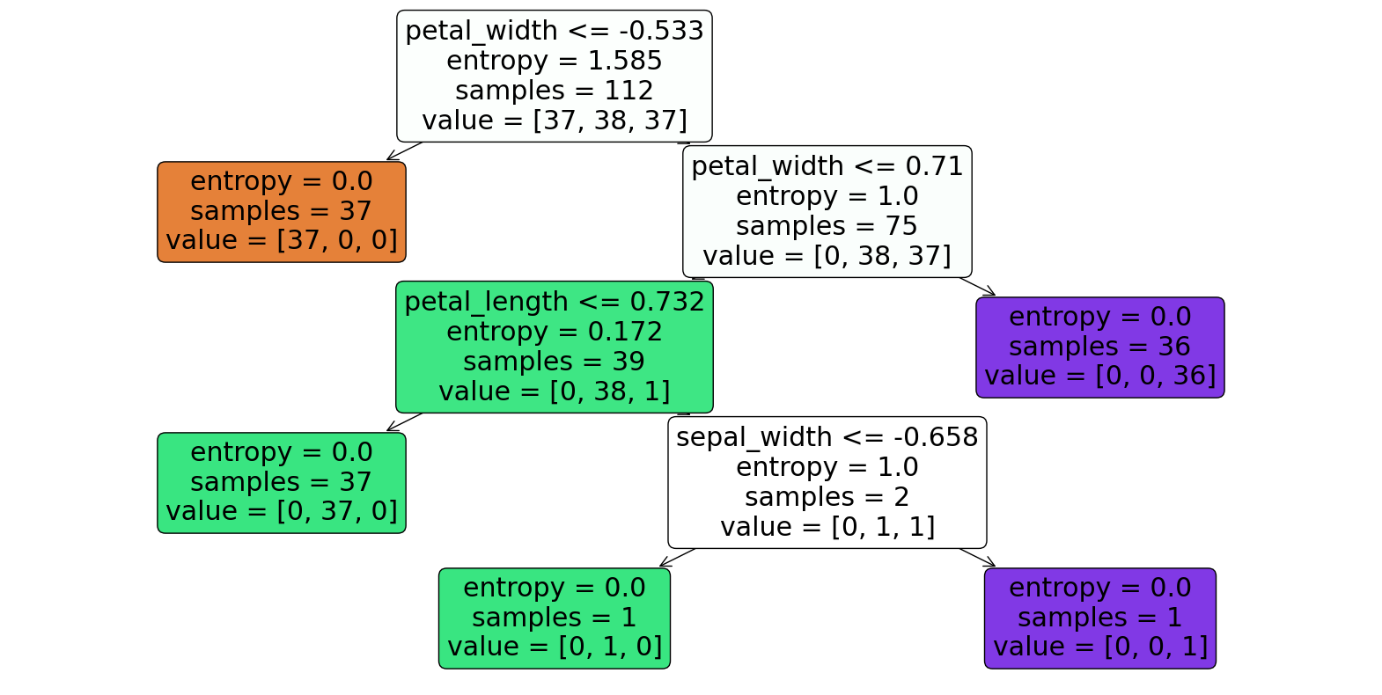
from sklearn.metrics import confusion\_matrix, accuracy\_score

cm = confusion\_matrix(y\_test, y\_pred)

print(cm)

accuracy\_score(y\_test, y\_pred)

**output:**

****

0.8947368421052632

[[13 0 0]

[ 0 11 1]

[ 0 3 10]]

**EXPERIMENT:6(E)SVM**

**AIM: :** finding accuracy value of iris data set using SVM algorithm

**PROGRAM:**

import numpy as np

import pandas as pd

# Importing the dataset

dataset = pd.read\_csv("/content/IRIS.csv ")

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, -1].values

# Splitting the dataset into the Training set and Test set

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=32)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

# Training the SVM model on the Training set

from sklearn.svm import SVC

classifier = SVC(kernel='linear', random\_state=0)

classifier.fit(X\_train, y\_train)

# Predicting the Test set results

y\_pred = classifier.predict(X\_test)

# Evaluating the performance of the model using confusion matrix and accuracy

from sklearn.metrics import confusion\_matrix, accuracy\_score

cm = confusion\_matrix(y\_test, y\_pred)

print(cm)

print('Accuracy: {:.2f}%'.format(accuracy\_score(y\_test, y\_pred) \* 100))

**OUTPUT:**

****

**EXPERIMENT:6(F)RANDOM**

**AIM: :** finding accuracy value of iris data set using RANDOM FOREST algorithm

**PROGRAM:**

import numpy as np

import pandas as pd

# Importing the dataset

dataset = pd.read\_csv("/content/IRIS.csv ")

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, -1].values

# Splitting the dataset into the Training set and Test set

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=39)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

# Training the Random Forest Classification model on the Training set

from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(n\_estimators=100, random\_state=42)

classifier.fit(X\_train, y\_train)

# Predicting the Test set results

y\_pred = classifier.predict(X\_test)

# Making the Confusion Matrix

from sklearn.metrics import confusion\_matrix, accuracy\_score

cm = confusion\_matrix(y\_test, y\_pred)

print(cm)

print('Accuracy:', accuracy\_score(y\_test, y\_pred))

