

Program No:10

Aim:Program to implement decision tree using any standard dataset available in the public domain and find the accuracy of the algorithm

Program:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import
classification_report, confusion_matrix
from sklearn.tree import plot_tree
df = sns.load_dataset('iris')
print(df.head())
print(df.info())
df.isnull().any()
print(df.shape)
sns.pairplot(data=df, hue = 'species')
plt.savefig("pne.png")
sns.heatmap(df.corr())
plt.savefig("one.png")
target = df['species']
df1 = df.copy()
df1 = df1.drop('species', axis =1)
print(df1.shape)
print(df1.head())
X = df1
print(target)
le = LabelEncoder()
target = le.fit_transform(target)
print(target)
y = target
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size = 0.2, random_state = 42)
print("training split input", X_train.shape)
print("testing split input" , X_test.shape)
```

```

dtree = DecisionTreeClassifier()
dtree.fit(X_train,y_train)
print("decision tree classifier created")
y_pred = dtree.predict(X_test)
print("classification report \n",
classification_report(y_test,y_pred))
cm = confusion_matrix(y_test,y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5,annot=True,square=True,
e,cmap='Blues')
plt.ylabel('Actual label')
plt.xlabel('predicted label')
all_sample_title = 'accuracy score
:{0}'.format(dtree.score(X_test , y_test))
plt.title(all_sample_title,size =15)
plt.savefig("two.png")
plt.figure(figsize = (20,20))
dec_tree = plot_tree(decision_tree=dtree,feature_names=
df1.columns,

class_names=["setosa","vercicolor","verginica"],filled
= True, precision =4 ,rounded =True)
plt.savefig("three.png")

```

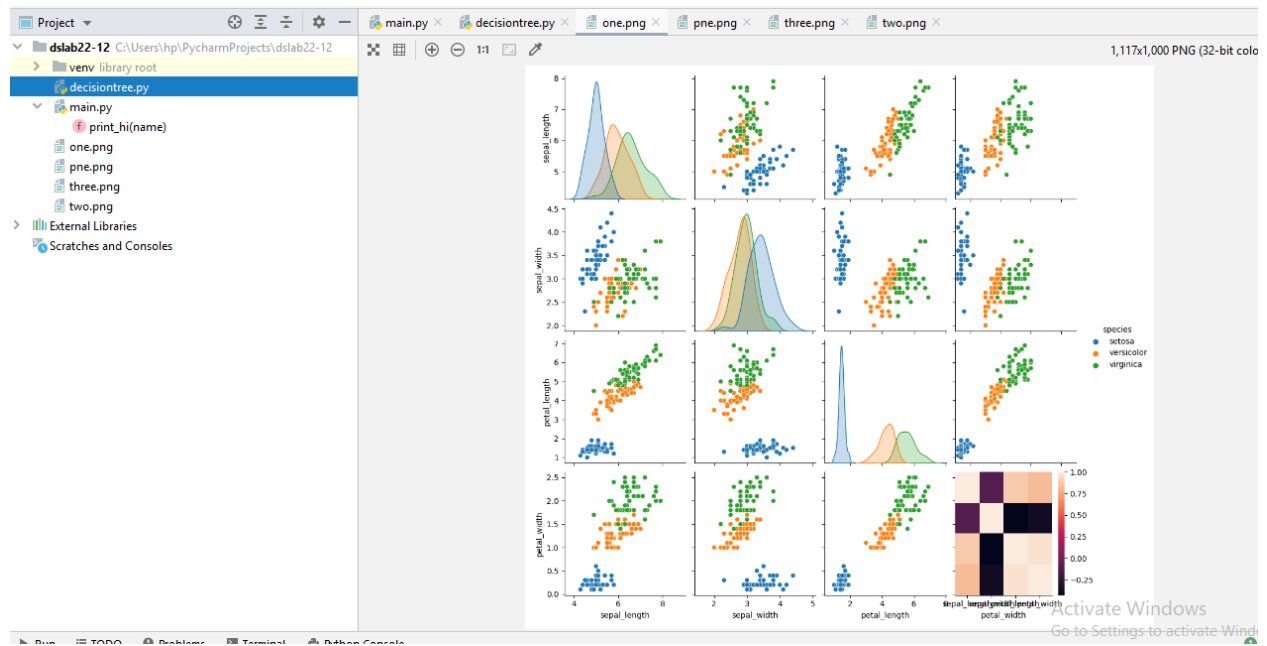
Output:

```

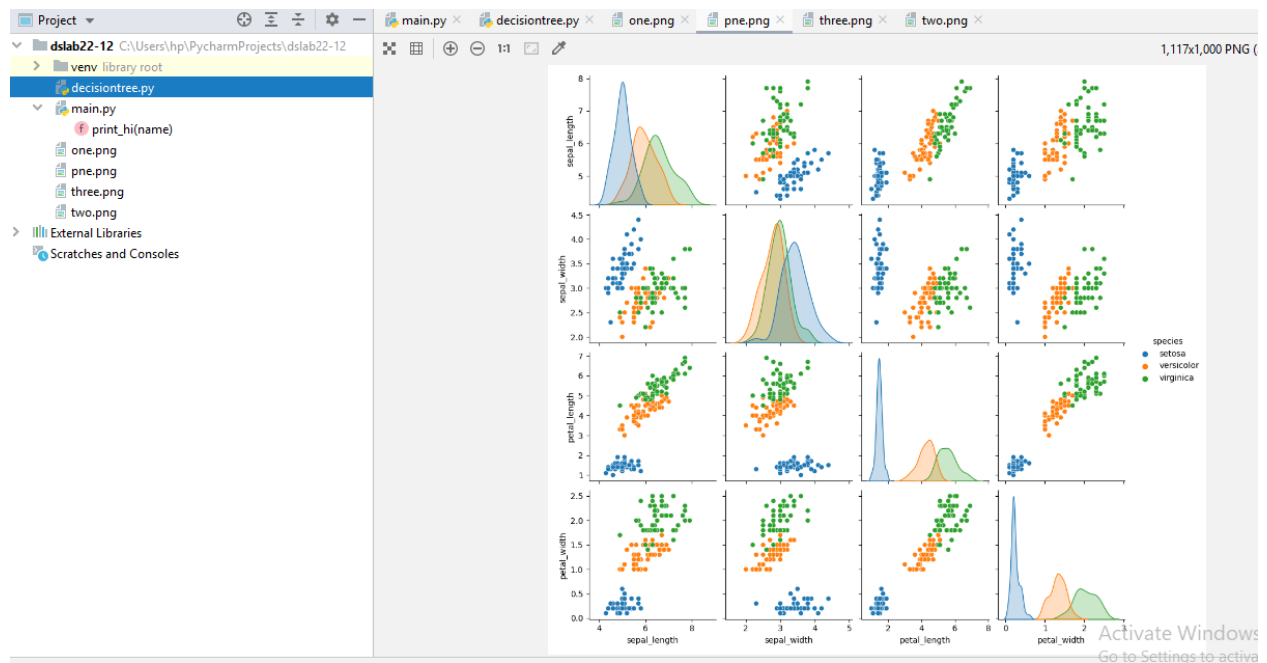
C:\Users\hp\PycharmProjects\ds\lab22-12\venv\Scripts\python.exe C:/Users/hp/PycharmProjects/ds\lab22-12/decisiontree.py
  sepal_length  sepal_width  petal_length  petal_width  species
0           5.1           3.5           1.4           0.2  setosa
1           4.9           3.0           1.4           0.2  setosa
2           4.7           3.2           1.3           0.2  setosa
3           4.6           3.1           1.5           0.2  setosa
4           5.0           3.6           1.4           0.2  setosa
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   sepal_length    150 non-null   float64
1   sepal_width     150 non-null   float64
2   petal_length    150 non-null   float64
3   petal_width     150 non-null   float64
4   species         150 non-null   object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
(150, 5)
(150, 4)
  sepal_length  sepal_width  petal_length  petal_width
0           5.1           3.5           1.4           0.2
1           4.9           3.0           1.4           0.2
2           4.7           3.2           1.3           0.2
3           4.6           3.1           1.5           0.2

```

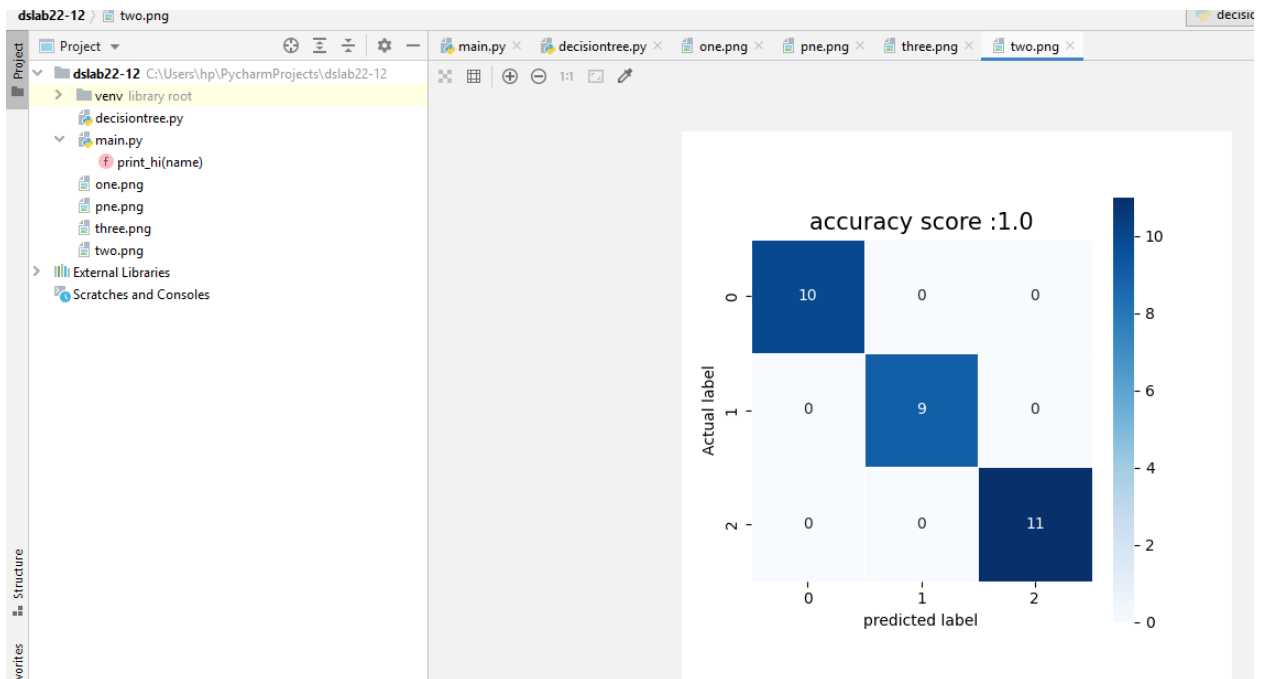

One.png



Pne.png



Two.png



Three.png

