

```
#importing packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
```

```
file_path="/content/drive/MyDrive/Dataset/Iris.csv"
df=pd.read_csv(file_path)
```

```
print(df.head)
```

```
<bound method NDFrame.head of      Id  SepalLengthCm  SepalWidthCm
PetalLengthCm  PetalWidthCm  \
0           1           5.1           3.5           1.4           0.2
1           2           4.9           3.0           1.4           0.2
2           3           4.7           3.2           1.3           0.2
3           4           4.6           3.1           1.5           0.2
4           5           5.0           3.6           1.4           0.2
..      ...           ...           ...           ...           ...
145        146           6.7           3.0           5.2           2.3
146        147           6.3           2.5           5.0           1.9
147        148           6.5           3.0           5.2           2.0
148        149           6.2           3.4           5.4           2.3
149        150           5.9           3.0           5.1           1.8
```

```
      Species
0      Iris-setosa
1      Iris-setosa
2      Iris-setosa
3      Iris-setosa
4      Iris-setosa
..      ...
145  Iris-virginica
146  Iris-virginica
147  Iris-virginica
148  Iris-virginica
149  Iris-virginica
```

```
[150 rows x 6 columns]>
```

```
df.shape
```

```
(150, 6)
```

```
#Handling Missing values
df.isnull().sum()
```

```

Id          0
SepalLengthCm  0
SepalWidthCm  0
PetalLengthCm  0
PetalWidthCm  0
Species      0
dtype: int64

df.Species.unique()

array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'],
      dtype=object)

#Splitting data
x=df.iloc[:,[0,1,2,3]]
y=df.iloc[:, -1]

x.head()

{"summary":{"\n  \"name\": \"x\", \n  \"rows\": 150, \n  \"fields\": [\n    {\n      \"column\": \"Id\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 43, \n        \"min\": 1, \n        \"max\": 150, \n        \"num_unique_values\": 150, \n        \"samples\": [\n          74, \n          19, \n          119\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\"\n      }, \n      {\n        \"column\": \"SepalLengthCm\", \n        \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": 0.8280661279778629, \n          \"min\": 4.3, \n          \"max\": 7.9, \n          \"num_unique_values\": 35, \n          \"samples\": [\n            6.2, \n            4.5, \n            5.6\n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n        }, \n        {\n          \"column\": \"SepalWidthCm\", \n          \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": 0.4335943113621737, \n            \"min\": 2.0, \n            \"max\": 4.4, \n            \"num_unique_values\": 23, \n            \"samples\": [\n              2.3, \n              4.0, \n              3.5\n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\"\n          }, \n          {\n            \"column\": \"PetalLengthCm\", \n            \"properties\": {\n              \"dtype\": \"number\", \n              \"std\": 1.7644204199522617, \n              \"min\": 1.0, \n              \"max\": 6.9, \n              \"num_unique_values\": 43, \n              \"samples\": [\n                6.7, \n                3.8, \n                3.7\n              ], \n              \"semantic_type\": \"\", \n              \"description\": \"\"\n            }, \n            {\n              \"column\": \"PetalWidthCm\", \n              \"properties\": {\n                \"dtype\": \"number\", \n                \"std\": 0.7633997221462228, \n                \"min\": 0.4, \n                \"max\": 2.0, \n                \"num_unique_values\": 35, \n                \"samples\": [\n                  1.3, \n                  0.5, \n                  1.7\n                ], \n                \"semantic_type\": \"\", \n                \"description\": \"\"\n              }\n            }\n          ]\n        }\n      }\n    ]\n  },\n  \"type\": \"dataframe\", \"variable_name\": \"x\"}

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=10)
x_train.shape,x_test.shape,y_train.shape,y_test.shape

((120, 4), (30, 4), (120,), (30,))

```

```

model = DecisionTreeClassifier()
model.fit(x_train,y_train)
y_predict = model.predict(x_test)
accuracy_score(y_test,y_predict)

```

1.0

```

model.get_depth()

```

2

```

model = DecisionTreeClassifier(max_depth=3) # max.depth
model.fit(x_train,y_train)
y_predict = model.predict(x_test)
accuracy_score(y_test,y_predict)

```

1.0

```

model = DecisionTreeClassifier(max_depth=10) # max.depth
model.fit(x_train,y_train)
y_predict = model.predict(x_test)
accuracy_score(y_test,y_predict)

```

1.0

```

print(classification_report(y_test,y_predict))

```

|                 | precision | recall | f1-score | support |
|-----------------|-----------|--------|----------|---------|
| Iris-setosa     | 1.00      | 1.00   | 1.00     | 10      |
| Iris-versicolor | 1.00      | 1.00   | 1.00     | 13      |
| Iris-virginica  | 1.00      | 1.00   | 1.00     | 7       |
| accuracy        |           |        | 1.00     | 30      |
| macro avg       | 1.00      | 1.00   | 1.00     | 30      |
| weighted avg    | 1.00      | 1.00   | 1.00     | 30      |

```

confusion_matrix(y_test,y_predict)

```

```

array([[10,  0,  0],
       [ 0, 13,  0],
       [ 0,  0,  7]])

```

```

from sklearn import tree
from matplotlib import rcParams
rcParams['figure.figsize']=12,12
dt = tree.DecisionTreeClassifier()
dt.fit(x,y)
tree.plot_tree(dt,
                class_names=['Setosa','Versicolor','Virginica'],

```

```
feature_names=['sepal_length','sepal_width','petal_length','petal_width'],  
              filled=True,max_depth=6)  
plt.show()
```

