

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

```

```

import seaborn as sns
iris = sns.load_dataset("iris")
iris

```

	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
...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

```
[150 rows x 5 columns]
```

```
iris.shape
```

```
(150, 5)
```

```

le=LabelEncoder()
iris['species']=le.fit_transform(iris['species'])

```

```

x=iris[['sepal_length','sepal_width','petal_length','petal_width']]
y=iris['species']

```

```

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,
random_state=42)

```

```

from sklearn.ensemble import RandomForestClassifier
clf=RandomForestClassifier(n_estimators=10,criterion='gini',random_state=1,max_depth=3)
clf.fit(x_train,y_train)

```

```
y_pred = clf.predict(x_test)
```

```

from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

```

```
Accuracy: 1.0
```

```
def future_pred(sample):
    prediction=clf.predict(sample)
    if prediction==0:
        print("It is the Flower Setosa")
    elif prediction==1:
        print("The Flower is Versicolor")
    else:
        print("The Flower is Virginica")
```

```
future_pred([[6.7,3,5.2,2.3]])
```

The Flower is Virginica

c:\Users\chakr\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:409: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with feature names

```
warnings.warn(
```

```
from sklearn.datasets import load_iris
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import f1_score,classification_report
iris = load_iris()
X_train, X_test, y_train, y_test = train_test_split(iris.data,
iris.target, test_size=0.3, random_state=42)
```

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
rfc = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
rfc.fit(X_train, y_train)
```

```
y_pred = rfc.predict(X_test)
```

```
#print("Accuracy:", rfc.score(y_test, y_pred))
print("Accuracy:", classification_report(y_test,y_pred))
```

Accuracy:		precision	recall	f1-score	support
	0	1.00	1.00	1.00	19
	1	1.00	1.00	1.00	13
	2	1.00	1.00	1.00	13
	accuracy			1.00	45
	macro avg	1.00	1.00	1.00	45

weighted avg	1.00	1.00	1.00	45
--------------	------	------	------	----