AIM

Program to implement decision trees using any standard dataset available in the public domain and find the accuracy of the algorithm.

Programming code:

Dataset used: iris

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris

data=load_iris()
X=data.data
y=data.target
print(X.shape,y.shape)
```

OUTPUT:

```
(150, 4) (150,)
```

Programming code:

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier#for checking testin
g results
from sklearn.metrics import classification_report, confusion_matrix
#for visualizing tree
from sklearn.tree import plot_tree
X_train, X_test, y_train, y_test = train_test_split(X , y, test_siz
e = 25, random_state = 10)

clf=DecisionTreeClassifier()
clf.fit(X_train,y_train)
```

OUTPUT:

```
DecisionTreeClassifier()
```

Programming code:

```
y_pred =clf.predict(X_test)
print("Classification report - \n", classification_report(y_test,y_
pred))
```

OUTPUT:

```
Classification report -
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         9
           1       1.00      0.90      0.95        10
           2       0.86      1.00      0.92         6

    accuracy                           0.96        25
   macro avg       0.95      0.97      0.96        25
weighted avg       0.97      0.96      0.96        25
```

Programming code:

```
cm = confusion_matrix(y_test, y_pred)
print(cm)
from sklearn import tree
fig,axes = plt.subplots(nrows=1,ncols=1,figsize =(3,3),dpi=200)
tree.plot_tree(clf,feature_names=data.feature_names,class_names=dat
a.target_names,filled=True)
plt.show()
fig.savefig("/content/iris_tree.png")
```

OUTPUT:

```
[[9 0 0]
 [0 9 1]
 [0 0 6]]
```