```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn import tree
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.tree import plot_tree
from sklearn.metrics import classification_report
import graphviz
```

```
data=pd.read_csv('/content/Comp.csv')
data.head()
```

	age	Income	Student	Credit_Rating	Buys_Computer
0	<=30	high	no	fair	no
1	<=30	high	no	excellent	no
2	31...40	high	no	fair	yes
3	>40	medium	no	fair	yes
4	>40	low	yes	fair	yes

```
data=data.replace(['<=30','31...40','>40'],[1,2,3])
data.head()
data=data.replace(['high','medium','low'],[1,2,3])
data.head()
data=data.replace(['no','yes'],[1,2])
data.head()
data=data.replace(['fair','excellent'],[1,2])
data.head()
#X=data.drop(columns=['Outcome'])
#Y=data['Outcome']
#X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.9, random_state =0)
```

	age	Income	Student	Credit_Rating	Buys_Computer
0	1	1	1	1	1
1	1	1	1	2	1
2	2	1	1	1	2
3	3	2	1	1	2
4	3	3	2	1	2

```

```

```
X=data.drop(columns=['Buys_Computer'])
```

```
Y=data['Buys_Computer']
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.9, random_state =0)
print("Entropy")
model = DecisionTreeClassifier(criterion = "entropy")
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Predicted values:")
print(y_pred)
print("Confusion Matrix: ",confusion_matrix(y_test, y_pred))
print("Accuracy : ",accuracy_score(y_test,y_pred)*100)
print("Report : ",classification_report(y_test, y_pred))

dtree = DecisionTreeClassifier()
dtree = dtree.fit(X_test, y_test)
features = ['age', 'Income', 'Student', 'Credit_Rating','Buys_Computer']
tree.plot_tree(dtree, feature_names=features)
```

```
Entropy
Predicted values:
[2 2 2 2 2 2 2 2 2 2 2 2 2]
Confusion Matrix:  [[0 5]
 [0 8]]
Accuracy :  61.53846153846154
Report :                precision    recall  f1-score   support
```

```
X=data.drop(columns=['Buys_Computer'])
Y=data['Buys_Computer']
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.5, random_state =1)
print("Gini")
model=DecisionTreeClassifier(criterion = "gini")
model=model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Predicted values:")
print(y_pred)
print("Confusion Matrix: ",confusion_matrix(y_test, y_pred))
print("Accuracy : ",accuracy_score(y_test,y_pred)*100)
print("Report : ",classification_report(y_test, y_pred))
```

```
Gini
Predicted values:
[2 1 2 2 1 2 1]
Confusion Matrix:  [[2 0]
 [1 4]]
Accuracy :  85.71428571428571
Report :                precision    recall  f1-score   support

           1       0.67      1.00      0.80         2
           2       1.00      0.80      0.89         5

    accuracy                           0.86         7
   macro avg       0.83      0.90      0.84         7
weighted avg       0.90      0.86      0.86         7
```

```
2]'),
```

```
plt.figure(figsize=(10,6))
plot_tree(model)
plt.show()
```