

ASSIGNMENT : 9

CODE :

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier, plot_tree

# Define the dataset
data = {
    'Age': ['<21', '<21', '21-35', '>35', '>35', '>35', '21-35', '<21', '<21', '>35', '<21', '21-35', '21-35', '>35'],
    'Income': ['High', 'High', 'High', 'Medium', 'Low', 'Low', 'Low', 'Medium', 'Low', 'Medium', 'Medium', 'Medium', 'High', 'Medium'],
    'Gender': ['Male', 'Male', 'Male', 'Male', 'Female', 'Female', 'Female', 'Male', 'Female', 'Female', 'Female', 'Male', 'Female', 'Male'],
    'Marital Status': ['Single', 'Married', 'Single', 'Single', 'Single', 'Married', 'Married', 'Single', 'Married', 'Single', 'Married', 'Married', 'Married', 'Single'],
    'Buys': ['No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'No']
}

# Convert to DataFrame
df = pd.DataFrame(data)

# Encode categorical variables
df.replace(
    {'Age': {'<21': 0, '21-35': 1, '>35': 2},
     'Income': {'Low': 0, 'Medium': 1, 'High': 2},
     'Gender': {'Male': 0, 'Female': 1},
     'Marital Status': {'Single': 0, 'Married': 1},
     'Buys': {'No': 0, 'Yes': 1}}, inplace=True
)
df = df.infer_objects(copy=False) # Explicitly infer object types

# Split features and target
X = df.drop(columns=['Buys'])
y = df['Buys']

# Train Decision Tree Classifier
clf = DecisionTreeClassifier(criterion='entropy', random_state=0)
clf.fit(X, y)

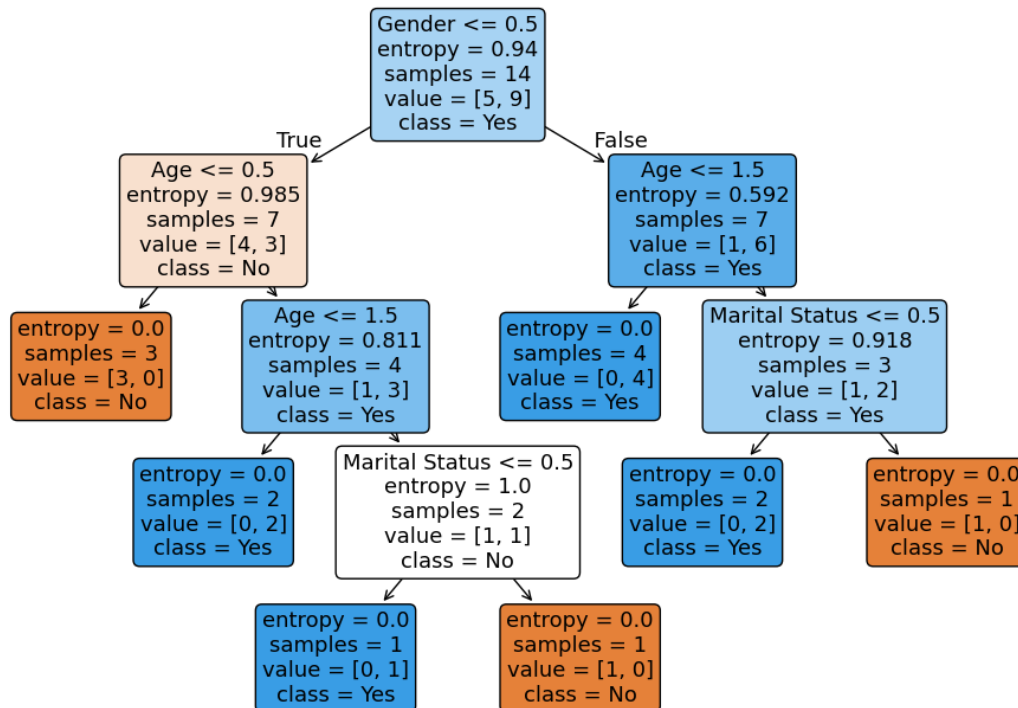
# Visualize the decision tree
plt.figure(figsize=(12, 8))
plot_tree(clf, feature_names=X.columns, class_names=['No', 'Yes'], filled=True, rounded=True)
plt.show()

# Print the textual decision tree representation
from sklearn.tree import export_text
tree_rules = export_text(clf, feature_names=list(X.columns))
print(tree_rules)
```

```
# Convert test_sample to DataFrame to avoid feature name warning
test_sample = pd.DataFrame([[0, 0, 1, 1]], columns=X.columns)
```

```
# Predict
prediction = clf.predict(test_sample)
```

```
print("Predicted Decision:", "Yes" if prediction[0] == 1 else "No")
OUTPUT :
```



```

|--- Gender <= 0.50
| |--- Age <= 0.50
| | |--- class: 0
| |--- Age > 0.50
| | |--- Age <= 1.50
| | | |--- class: 1
| | |--- Age > 1.50
| | | |--- Marital Status <= 0.50
| | | | |--- class: 1
| | | |--- Marital Status > 0.50
| | | | |--- class: 0
|--- Gender > 0.50
| |--- Age <= 1.50
| | |--- class: 1
| |--- Age > 1.50
| | |--- Marital Status <= 0.50
| | | |--- class: 1
| | |--- Marital Status > 0.50
| | | |--- class: 0

```

Predicted Decision: Yes