

```
import numpy as np
import pandas as pd
from numpy import log2 as log
```

```
dataset = [
    ['<21', 'High', 'Male', 'Single', 'No'],
    ['<21', 'High', 'Male', 'Married', 'No'],
    ['21-35', 'High', 'Male', 'Single', 'Yes'],
    ['>35', 'Medium', 'Male', 'Single', 'Yes'],
    ['>35', 'Low', 'Female', 'Single', 'Yes'],
    ['>35', 'Low', 'Female', 'Married', 'No'],
    ['21-35', 'Low', 'Female', 'Married', 'Yes'],
    ['<21', 'Medium', 'Male', 'Single', 'No'],
    ['<21', 'Low', 'Female', 'Married', 'Yes'],
    ['>35', 'Medium', 'Female', 'Single', 'Yes'],
    ['<21', 'Medium', 'Female', 'Married', 'Yes'],
    ['21-35', 'Medium', 'Male', 'Married', 'Yes'],
    ['21-35', 'High', 'Female', 'Single', 'Yes'],
    ['>35', 'Medium', 'Male', 'Married', 'No']
]
```

```
columns = ['Age', 'Income', 'Gender', 'Marital Status', 'Buys']
df = pd.DataFrame(dataset, columns=columns)
df
```

	Age	Income	Gender	Marital Status	Buys
0	<21	High	Male	Single	No
1	<21	High	Male	Married	No
2	21-35	High	Male	Single	Yes
3	>35	Medium	Male	Single	Yes
4	>35	Low	Female	Single	Yes
5	>35	Low	Female	Married	No
6	21-35	Low	Female	Married	Yes
7	<21	Medium	Male	Single	No
8	<21	Low	Female	Married	Yes
9	>35	Medium	Female	Single	Yes
10	<21	Medium	Female	Married	Yes
11	21-35	Medium	Male	Married	Yes
12	21-35	High	Female	Single	Yes
13	>35	Medium	Male	Married	No

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
for i in range(5):
    df[columns[i]] = le.fit_transform(df[columns[i]])
df
```

	Age	Income	Gender	Marital Status	Buys
0	1	0	1	1	0
1	1	0	1	0	0
2	0	0	1	1	1
3	2	2	1	1	1
4	2	1	0	1	1
5	2	1	0	0	0
6	0	1	0	0	1
7	1	2	1	1	0
8	1	1	0	0	1
9	2	2	0	1	1
10	1	2	0	0	1
11	0	2	1	0	1
12	0	0	0	1	1
13	2	2	1	0	0

```
test_data=[[0, 0, 0, 0]]
test = pd.DataFrame(test_data, columns=['Age', 'Income', 'Gender', 'Marital Status'])
test
```

	Age	Income	Gender	Marital Status
0	0	0	0	0

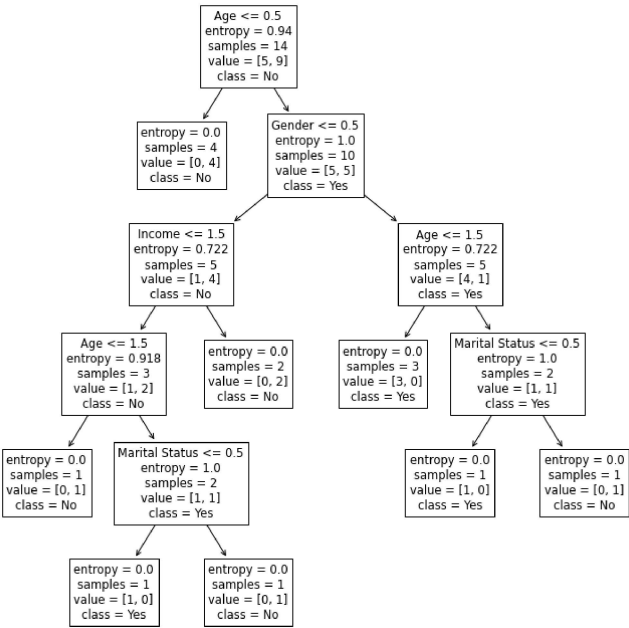
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.tree import plot_tree
sklearn_dt=DecisionTreeClassifier(criterion="entropy")
```

```
df1 = df.copy()
df1.drop('Buys', axis=1, inplace=True)
X=df1
```

```
sklearn_dt.fit(X, df['Buys'])
sklearn_dt.predict(test)
```

array([1])

```
import matplotlib.pyplot as plt
plt.figure(figsize=(12,12))
dec_tree = plot_tree(decision_tree=sklearn_dtrees, feature_names = df.columns, class_names=["Yes", "No"])
plt.show()
```



```
dtree
{
  'Age': {0: 1,
    1: {'Gender': {0: 1, 1: 0}},
    2: {'Marital Status': {0: 0, 1: 1}}}}
}
```