In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

In [2]:
```
df=pd.read_csv(r"C:\Users\HP\Downloads\loan1.csv")
df
```

Out[2]:

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
0	Yes	Single	125	No
1	No	Married	100	No
2	No	Single	70	No
3	Yes	Married	120	No
4	No	Divorced	95	Yes
5	No	Married	60	No
6	Yes	Divorced	220	No
7	No	Single	85	Yes
8	No	Married	75	No
9	No	Single	90	Yes

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Home Owner          10 non-null     object
 1   Marital Status      10 non-null     object
 2   Annual Income       10 non-null     int64 
 3   Defaulted Borrower  10 non-null     object
dtypes: int64(1), object(3)
memory usage: 448.0+ bytes
```

In [4]:
```
df['Marital Status'].value_counts()
```

Out[4]:
```
Marital Status
Single      4
Married     4
Divorced    2
Name: count, dtype: int64
```

In [5]:
```
df['Annual Income'].value_counts()
```

Out[5]:
```
Annual Income
125    1
100    1
70     1
120    1
95     1
60     1
220    1
85     1
75     1
90     1
Name: count, dtype: int64
```

In [6]:
```
convert={"Home Owner":{"Yes":1,"No":0}}
df=df.replace(convert)
print(df)
```

```
   Home Owner Marital Status  Annual Income Defaulted Borrower
0           1         Single            125                 No
1           0        Married            100                 No
2           0         Single             70                 No
3           1        Married            120                 No
4           0       Divorced             95                Yes
5           0        Married             60                 No
6           1       Divorced            220                 No
7           0         Single             85                Yes
8           0        Married             75                 No
9           0         Single             90                Yes
```

In [7]:
```
convert={"Marital Status":{"Single":1,"Married":2,"Divorced":3}}
df=df.replace(convert)
print(df)
```

```
   Home Owner  Marital Status  Annual Income Defaulted Borrower
0           1               1            125                 No
1           0               2            100                 No
2           0               1             70                 No
3           1               2            120                 No
4           0               3             95                Yes
5           0               2             60                 No
6           1               3            220                 No
7           0               1             85                Yes
8           0               2             75                 No
9           0               1             90                Yes
```

In [8]:
```
convert={"Default Borrower":{"Yes":0,"No":1}}
df=df.replace(convert)
df
```

Out[8]:

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
0	1	1	125	No
1	0	2	100	No
2	0	1	70	No
3	1	2	120	No
4	0	3	95	Yes
5	0	2	60	No
6	1	3	220	No
7	0	1	85	Yes
8	0	2	75	No
9	0	1	90	Yes

In [9]:
```
x=["Home Owner","Marital Status","Annual Income"]
y=["yes","No"]
all_inputs=df[x]
all_classes=df["Defaulted Borrower"]
```

In [10]:
```
x_train,x_test,y_train,y_test=train_test_split(all_inputs,all_classes,test_siz
```

In [11]:
```
clt=DecisionTreeClassifier(random_state=0)
```

In [12]:
```
clt.fit(x_train,y_train)
```

Out[12]:
```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=0)
```

In [13]:
```
score=clt.score(x_test,y_test)
print(score)
```

```
1.0
```

2).Drug Dataset

In [14]:
```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

In [15]:
```
df=pd.read_csv(r"C:\Users\HP\Downloads\drug200.csv")
df
```

Out[15]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY
...	...	...	...	...	...	...
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

200 rows × 6 columns

In [16]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Age          200 non-null    int64  
 1   Sex          200 non-null    object 
 2   BP           200 non-null    object 
 3   Cholesterol  200 non-null    object 
 4   Na_to_K      200 non-null    float64
 5   Drug         200 non-null    object 
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
```

In [17]:
```
df['BP'].value_counts()
```

Out[17]:
```
BP
HIGH      77
LOW       64
NORMAL    59
Name: count, dtype: int64
```

In [18]:
```
df['Age'].value_counts()
```

Out[18]:

Age	
47	8
23	7
28	7
49	7
39	6
32	6
50	5
37	5
58	5
60	5
22	5
34	4
72	4
51	4
42	4
26	4
24	4
74	4
67	4
68	4
61	4
56	4
20	4
36	4
45	4
41	4
31	4
43	4
65	4
57	4
53	3
40	3
70	3
59	3
16	3
38	3
15	3
69	3
35	3
18	3
64	3
52	2
55	2
62	2
19	2
29	2
66	2
73	2
46	2
48	2
54	1
17	1
33	1
63	1
30	1
21	1

```
25    1
Name: count, dtype: int64
```

In [19]:
```
convert={"Sex":{"F":1,"M":0}}
df=df.replace(convert)
print(df)
```

```
     Age  Sex      BP Cholesterol  Na_to_K   Drug
0     23    1    HIGH        HIGH   25.355  drugY
1     47    0     LOW        HIGH   13.093  drugC
2     47    0     LOW        HIGH   10.114  drugC
3     28    1  NORMAL        HIGH    7.798  drugX
4     61    1     LOW        HIGH   18.043  drugY
..   ...  ...     ...         ...      ...    ...
195   56    1     LOW        HIGH   11.567  drugC
196   16    0     LOW        HIGH   12.006  drugC
197   52    0  NORMAL        HIGH    9.894  drugX
198   23    0  NORMAL      NORMAL   14.020  drugX
199   40    1     LOW      NORMAL   11.349  drugX

[200 rows x 6 columns]
```

In [20]:
```
convert={"BP":{"HIGH":1,"LOW":2,"NORMAL":3}}
df=df.replace(convert)
print(df)
```

```
     Age  Sex  BP Cholesterol  Na_to_K   Drug
0     23    1   1        HIGH   25.355  drugY
1     47    0   2        HIGH   13.093  drugC
2     47    0   2        HIGH   10.114  drugC
3     28    1   3        HIGH    7.798  drugX
4     61    1   2        HIGH   18.043  drugY
..   ...  ...  ..         ...      ...    ...
195   56    1   2        HIGH   11.567  drugC
196   16    0   2        HIGH   12.006  drugC
197   52    0   3        HIGH    9.894  drugX
198   23    0   3      NORMAL   14.020  drugX
199   40    1   2      NORMAL   11.349  drugX

[200 rows x 6 columns]
```

In [21]:
```
convert={"Cholesterol":{"HIGH":0,"NORMAL":1}}
df=df.replace(convert)
df
```

Out[21]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	1	1	0	25.355	drugY
1	47	0	2	0	13.093	drugC
2	47	0	2	0	10.114	drugC
3	28	1	3	0	7.798	drugX
4	61	1	2	0	18.043	drugY
...	...	...	...	...	...	...
195	56	1	2	0	11.567	drugC
196	16	0	2	0	12.006	drugC
197	52	0	3	0	9.894	drugX
198	23	0	3	1	14.020	drugX
199	40	1	2	1	11.349	drugX

200 rows × 6 columns

In [22]:
```
x=["Sex","BP","Age"]
y=["HIGH","NORMAL"]
all_inputs=df[x]
all_classes=df["Cholesterol"]
```

In [23]:
```
x_train,x_test,y_train,y_test=train_test_split(all_inputs,all_classes,test_siz
```

In [24]:
```
clt=DecisionTreeClassifier(random_state=0)
```

In [25]:
```
clt.fit(x_train,y_train)
```

Out[25]:

DecisionTreeClassifier

DecisionTreeClassifier(random_state=0)

In [26]:
```
score=clt.score(x_test,y_test)
print(score)
```

```
0.48
```

In [ ]: