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\Teju\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]:

```
Single      4
Married     4
Divorced    2
Name: Marital Status, dtype: int64
```

In [5]:

```
df['Annual Income'].value_counts()
```

Out[5]:

```
125    1
100    1
70     1
120    1
95     1
60     1
220    1
85     1
75     1
90     1
Name: Annual Income, dtype: int64
```

In [6]:

```
convert={'Home Owner':{"Yes":1,"No":0}}
df=df.replace(convert)
df
```

Out[6]:

	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)
df
```

Out[7]:

	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]:

```
x=['Home Owner','Marital Status','Annual Income']
y=["Yes","No"]
all_inputs=df[x]
all_classes=df["Defaulted Borrower"]
```

In [9]:

```
(x_train,x_test,y_train,y_test)=train_test_split(all_inputs,all_classes,test_size=0.5)
```

In [10]:

```
clf=DecisionTreeClassifier(random_state=0)
```

In [11]:

```
clf.fit(x_train,y_train)
```

Out[11]:

```
▾       DecisionTreeClassifier
DecisionTreeClassifier(random_state=0)
```

In [12]:

```
score=clf.score(x_test,y_test)
print(score)
```

```
0.8
```

In [24]:

```
df=pd.read_csv(r"C:\Users\Teju\Downloads\drug200.csv")
df
```

Out[24]:

	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 [25]:

```
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 [26]:

```
df['Cholesterol'].value_counts()
```

Out[26]:

```
HIGH      103
NORMAL     97
Name: Cholesterol, dtype: int64
```

In [27]:

```
df['Drug'].value_counts()
```

Out[27]:

```
drugY    91
drugX    54
drugA    23
drugC    16
drugB    16
Name: Drug, dtype: int64
```

In [28]:

```
convert={'Sex':{"F":1,"M":0}}
df=df.replace(convert)
df
```

Out[28]:

	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 × 6 columns

In [29]:

```
convert={'BP':{"LOW":1,"NORMAL":2,"HIGH":3}}
df=df.replace(convert)
df
```

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

In [30]:

```
convert={'Cholesterol':{"NORMAL":0,"HIGH":1}}
df=df.replace(convert)
df
```

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

In [36]:

```
x=['Age','Sex','BP','Cholesterol','Na_to_K']
y=["drugY","drugX","drugA","drugC","drugB"]
all_inputs=df[x]
all_classes=df["Drug"]
```

In [37]:

```
(x_train,x_test,y_train,y_test)=train_test_split(all_inputs,all_classes,test_size=0.5)
```

In [38]:

```
clf=DecisionTreeClassifier(random_state=0)
```

In [39]:

```
clf.fit(x_train,y_train)
```

Out[39]:

```
▾        DecisionTreeClassifier
DecisionTreeClassifier(random_state=0)
```

In [40]:

```
score=clf.score(x_test,y_test)
print(score)
```

```
0.98
```

In [ ]:

```
```