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.head()
```

Out[3]:

	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

In [4]:

```
df.tail()
```

Out[4]:

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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 [5]:

```
df.describe()
```

Out[5]:

	Annual Income
count	10.000000
mean	104.000000
std	45.631373
min	60.000000
25%	77.500000
50%	92.500000
75%	115.000000
max	220.000000

In [6]:

```
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 [7]:

```
df.isnull().sum()
```

Out[7]:

```
Home Owner           0
Marital Status       0
Annual Income        0
Defaulted Borrower   0
dtype: int64
```

In [8]:

```
df['Marital Status'].value_counts()
```

Out[8]:

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

In [9]:

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

Out[9]:

```
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 [10]:

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

Out[10]:

	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 [11]:

```
x=["Home Owner","Annual Income"]
y=["Yes","No"]
all_inputs=df[x]
all_classes=df["Defaulted Borrower"]
(x_train,x_test,y_train,y_test)=train_test_split(all_inputs,all_classes,test_size=0.03)
```

In [12]:

```
clf=DecisionTreeClassifier(random_state=0)
clf.fit(x_train,y_train)
```

Out[12]:

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

In [13]:

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

```
1.0
```