

In [1]:

```

1 import numpy as np
2 import pandas as pd
3 import seaborn as sns
4 from sklearn.model_selection import train_test_split
5 from sklearn.tree import DecisionTreeClassifier

```

In [2]:

```

1 df=pd.read_csv(r"C:\Users\91955\Downloads\loan1.csv")
2 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]:

```

1 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]:

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

Out[4]:

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

In [5]:

```
1 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]:

```
1 convert={'Marital Status':{'Single':1,"Married":2,"Divorced":3}}  
2 df=df.replace(convert)  
3 df
```

Out[6]:

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

In [7]:

```

1 convert={'Home Owner':{'Yes':1,"No":2}}
2 df=df.replace(convert)
3 df

```

Out[7]:

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

In [11]:

```

1 x=["Home Owner","Marital Status","Annual Income"]
2 y=["Yes","No"]
3 all_inputs=df[x]
4 all_classes=df["Defaulted Borrower"]

```

In [18]:

```

1 (x_train,x_test,y_train,y_test)=train_test_split(all_inputs,all_classes,test_size=0.

```

In [19]:

```

1 clf=DecisionTreeClassifier(random_state=0)

```

In [20]:

```

1 clf.fit(x_train,y_train)

```

Out[20]:

```

DecisionTreeClassifier
DecisionTreeClassifier(random_state=0)

```

In [17]:

```

1 score=clf.score(x_test,y_test)
2 print(score)

```

0.625

