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\saranya\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

Out[3]:

method	DataFrame.info of	Home Owner	Marital	Status	Annual	Income
Defaulted Borrower						
Yes	Single	125		N	0	
No	Married	100		N	0	
No	Single	70		N	0	
Yes	Married	120		N	0	
No	Divorced	95		Ye	S	
No	Married	60		N	0	
Yes	Divorced	220		N	0	
No	Single	85		Ye	S	
No	Married	75		N	0	
No	Single	90		Ye	s>	
	ted Bori Yes No No Yes No No Yes No	ted Borrower Yes Single No Married No Single Yes Married No Divorced No Married Yes Divorced No Single	Yes Single 125 No Married 100 No Single 70 Yes Married 120 No Divorced 95 No Married 60 Yes Divorced 220 No Single 85 No Married 75	ted Borrower Yes Single 125 No Married 100 No Single 70 Yes Married 120 No Divorced 95 No Married 60 Yes Divorced 220 No Single 85 No Married 75	ted Borrower Yes Single 125 N No Married 100 N No Single 70 N Yes Married 120 N No Divorced 95 Ye No Married 60 N Yes Divorced 220 N No Single 85 Ye No Married 75 N	Yes Single 125 No No Married 100 No No Single 70 No Yes Married 120 No No Divorced 95 Yes No Married 60 No Yes Divorced 220 No No Single 85 Yes No Married 75 No

```
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 1 60 220 1 1 85 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]:

```
\label{lem:continuous} $$(x_train,x_test,y_train,y_test)=train_test_split(all_inputs,all_classes,train_size=0.7)$$ clf=DecisionTreeClassifier(random_state=0)$$ clf.fit(x_train,y_train)$$
```

Out[9]:

DecisionTreeClassifier(random state=0)

In [10]:

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

0.3333333333333333

In []: