

```
In [4]: 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 [5]: 1 df=pd.read_csv(r"C:\Users\P. VIJAY KUMAR\Downloads\loan1.csv")
        2 df
```

```
Out[5]:
```

	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 [6]: 1 df.info
```

```
Out[6]: <bound method DataFrame.info of
         Home Owner Marital Status Annual Income Defa
         ulted 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 [7]: 1 df.isnull().any()
```

```
Out[7]: Home Owner      False
         Marital Status  False
         Annual Income   False
         Defaulted Borrower False
         dtype: bool
```

```
In [8]: 1 df['Marital Status'].value_counts()
```

```
Out[8]: Marital Status
Single      4
Married     4
Divorced    2
Name: count, dtype: int64
```

```
In [10]: 1 df["Annual Income"].value_counts()
```

```
Out[10]: 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 [12]: 1 convert={"Home Owner":{"Yes":1,"No":0}}
2 df=df.replace(convert)
3 df
```

```
Out[12]:
```

	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 [14]: 1 convert={"Marital Status":{"Single":1,"Married":2,"Divorced":3}}
          2 df=df.replace(convert)
          3 df
```

```
Out[14]:
```

	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 [17]: 1 x=["Home Owner","Marital Status","Annual Income"]
          2 y=["Yes","No"]
          3 all_inputs=df[x]
          4 all_classes=df["Defaulted Borrower"]
```

```
In [23]: 1 x_train,x_test,y_train,y_test=train_test_split(all_inputs,all_classes,test_s
          2 clf=DecisionTreeClassifier(random_state=0)
          3 clf.fit(x_train,y_train)
```

```
Out[23]: DecisionTreeClassifier(random_state=0)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.

On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [24]: 1 score=clf.score(x_test,y_test)
          2 print(score)
```

```
0.6666666666666666
```

```
In [ ]: 1
```