

```
In [285]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [287]: df=pd.read_csv(r"C:\USERS\user\Downloads\C10_loan1 - C10_loan1.csv")
```

Out[287]:

	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 [288]:

Out[288]: Index(['Home Owner', 'Marital Status', 'Annual Income', 'Defaulted Borrower'], dtype='object')

```
In [289]: df=df.head(15)
```

Out[289]:

	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 [323]: a=df[['Home Owner','Annual Income']]
```

```
Out[323]:
```

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

```
In [324]:
```

```
Out[324]: No      7
          Yes     3
          Name: Home Owner, dtype: int64
```

```
In [325]: x=a.drop('Home Owner',axis=1)
```

```
In [326]: g1={"Home Owner":{"No":1,'Yes':2}}
          a=a.replace(g1)
```

	Home Owner	Annual Income
0	2	125
1	1	100
2	1	70
3	2	120
4	1	95
5	1	60
6	2	220
7	1	85
8	1	75
9	1	90

```
In [327]: from sklearn.model_selection import train_test_split
```

```
In [328]: from sklearn.ensemble import RandomForestClassifier
```

```
rfc=RandomForestClassifier()
```

```
Out[328]: RandomForestClassifier()
```

```
In [329]: parameters={'max_depth':[1,2,3,4,5],
                      'min_samples_leaf':[5,10,15,20,25],
```

```
In [330]: from sklearn.model_selection import GridSearchCV
```

```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="acc
```

```
Out[330]: GridSearchCV(cv=2, estimator=RandomForestClassifier(),
                      param_grid={'max_depth': [1, 2, 3, 4, 5],
                                   'min_samples_leaf': [5, 10, 15, 20, 25],
                                   'n_estimators': [10, 20, 30, 40, 50]},
                      scoring='accuracy')
```

```
In [331]:
```

```
Out[331]: 0.7083333333333333
```

```
In [332]:
```

```
In [333]: from sklearn.tree import plot_tree
```

```
plt.figure(figsize=(80,40))
```

```
Out[333]: [Text(2232.0, 1087.2, 'gini = 0.408\nsamples = 5\nvalue = [5, 2]\nnclass = Yes\n')]
```

gini = 0.408
samples = 5
value = [5, 2]
class = Yes

```
In [ ]:
```

```
In [ ]:
```