

```
In [2]: import pandas as pd
from sklearn.model_selection import train_test_split
```

```
In [3]: from sklearn.ensemble import RandomForestClassifier
import matplotlib.pyplot as plt
from sklearn.model_selection import GridSearchCV
from sklearn.tree import plot_tree
```

```
In [4]: df=pd.read_csv("C10_loan1.csv")
df
```

```
Out[4]:
```

	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 [9]: df["Home Owner"]=df["Home Owner"].replace({"Yes":1,"No":0}).astype(int)
df["Marital Status"]=df["Marital Status"].replace({"Single":1,"Married":2,"Divorced":3}).astype(int)
df["Defaulted Borrower"]=df["Defaulted Borrower"].replace({"Yes":1,"No":0}).astype(int)
```

```
Out[9]:
```

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

In [10]: 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     int32
1   Marital Status         10 non-null     int32
2   Annual Income          10 non-null     int64
3   Defaulted Borrower     10 non-null     int32
dtypes: int32(3), int64(1)
memory usage: 328.0 bytes
```

In [11]: `y=df["Defaulted Borrower"]`
`x=df.drop(["Defaulted Borrower"],axis=1)`
`x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)`

In [12]: `rfc=RandomForestClassifier()`
`rfc.fit(x_train,y_train)`

Out[12]: RandomForestClassifier()

In [13]: `parameter={'max_depth':[1,2,3,4,5],`
`"min_samples_leaf":[5,10,15,20,25],`
`"n_estimators":[10,20,30,40,50]}`

In [14]: `grid_search = GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accuracy")`
`grid_search.fit(x_train,y_train)`

Out[14]: 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')

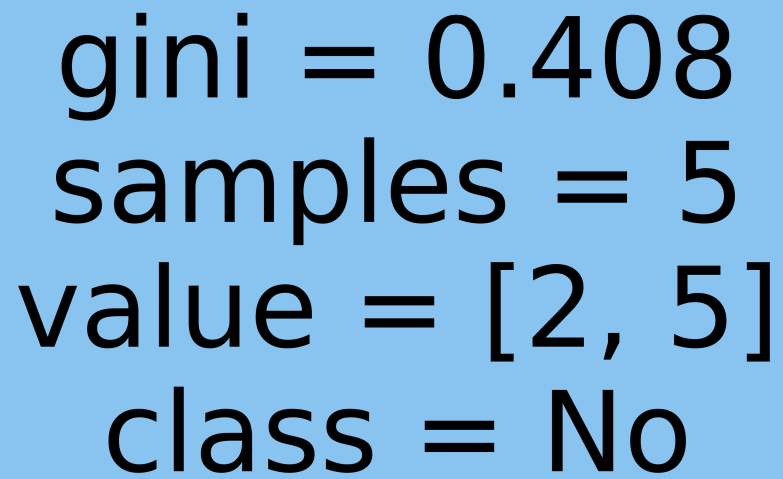
In [15]: `grid_search.best_score_`

Out[15]: 0.5833333333333333

In [16]: `rfc_best=grid_search.best_estimator_`

```
In [18]: plt.figure(figsize=(80,40))  
plot_tree(rfc_best.estimators_[5],class_names=['Yes','No'],filled=True)
```

```
Out[18]: [Text(2232.0, 1087.2, 'gini = 0.408\nsamples = 5\nvalue = [2, 5]\nclass = No')]
```



gini = 0.408
samples = 5
value = [2, 5]
class = No

```
In [ ]:
```