

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sea
```

```
In [3]: df = pd.read_csv(r"C:\Users\user\Downloads\C8_loan-test.csv")
df
```

Out[3]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	Loan/
0	LP001015	Male	Yes	0	Graduate	No	5720	0	
1	LP001022	Male	Yes	1	Graduate	No	3076	1500	
2	LP001031	Male	Yes	2	Graduate	No	5000	1800	
3	LP001035	Male	Yes	2	Graduate	No	2340	2546	
4	LP001051	Male	No	0	Not Graduate	No	3276	0	
...
362	LP002971	Male	Yes	3+	Not Graduate	Yes	4009	1777	
363	LP002975	Male	Yes	0	Graduate	No	4158	709	
364	LP002980	Male	No	0	Graduate	No	3250	1993	
365	LP002986	Male	Yes	0	Graduate	No	5000	2393	
366	LP002989	Male	No	0	Graduate	Yes	9200	0	

367 rows × 12 columns



```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 367 entries, 0 to 366
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Loan_ID               367 non-null   object
1   Gender                356 non-null   object
2   Married               367 non-null   object
3   Dependents            357 non-null   object
4   Education             367 non-null   object
5   Self_Employed         344 non-null   object
6   ApplicantIncome       367 non-null   int64
7   CoapplicantIncome     367 non-null   int64
8   LoanAmount            362 non-null   float64
9   Loan_Amount_Term      361 non-null   float64
10  Credit_History         338 non-null   float64
11  Property_Area          367 non-null   object
dtypes: float64(3), int64(2), object(7)
memory usage: 34.5+ KB
```

```
In [5]: df1 = df.fillna(0)
df1
```

Out[5]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	Loan/
0	LP001015	Male	Yes	0	Graduate	No	5720	0	
1	LP001022	Male	Yes	1	Graduate	No	3076	1500	
2	LP001031	Male	Yes	2	Graduate	No	5000	1800	
3	LP001035	Male	Yes	2	Graduate	No	2340	2546	
4	LP001051	Male	No	0	Not Graduate	No	3276	0	
...
362	LP002971	Male	Yes	3+	Not Graduate	Yes	4009	1777	
363	LP002975	Male	Yes	0	Graduate	No	4158	709	
364	LP002980	Male	No	0	Graduate	No	3250	1993	
365	LP002986	Male	Yes	0	Graduate	No	5000	2393	
366	LP002989	Male	No	0	Graduate	Yes	9200	0	

367 rows × 12 columns

```
In [6]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 367 entries, 0 to 366
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Loan_ID               367 non-null    object
1   Gender                367 non-null    object
2   Married               367 non-null    object
3   Dependents            367 non-null    object
4   Education             367 non-null    object
5   Self_Employed         367 non-null    object
6   ApplicantIncome       367 non-null    int64
7   CoapplicantIncome     367 non-null    int64
8   LoanAmount            367 non-null    float64
9   Loan_Amount_Term      367 non-null    float64
10  Credit_History        367 non-null    float64
11  Property_Area         367 non-null    object
dtypes: float64(3), int64(2), object(7)
memory usage: 34.5+ KB
```

```
In [7]: df1.columns
```

```
Out[7]: Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
              'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
              'Loan_Amount_Term', 'Credit_History', 'Property_Area'],
              dtype='object')
```

```
In [8]: df2 = df1[['Married', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
                  'Loan_Amount_Term', 'Credit_History']]
```

In [9]: df2.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 367 entries, 0 to 366
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Married                367 non-null    object
1   ApplicantIncome        367 non-null    int64
2   CoapplicantIncome      367 non-null    int64
3   LoanAmount             367 non-null    float64
4   Loan_Amount_Term       367 non-null    float64
5   Credit_History         367 non-null    float64
dtypes: float64(3), int64(2), object(1)
memory usage: 17.3+ KB
```

```
In [10]: x = df1[['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan_Amount_Term', 'Credit_History']]
         y = df1['Married']
```

```
In [11]: e1 = {"Married":{'No':0, 'Yes':1}}
         df2 = df2.replace(e1)
         df2
```

Out[11]:

	Married	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
0	1	5720	0	110.0	360.0	1.0
1	1	3076	1500	126.0	360.0	1.0
2	1	5000	1800	208.0	360.0	1.0
3	1	2340	2546	100.0	360.0	0.0
4	0	3276	0	78.0	360.0	1.0
...
362	1	4009	1777	113.0	360.0	1.0
363	1	4158	709	115.0	360.0	1.0
364	0	3250	1993	126.0	360.0	0.0
365	1	5000	2393	158.0	360.0	1.0
366	0	9200	0	98.0	180.0	1.0

367 rows × 6 columns

In [12]: from sklearn.model_selection import train_test_split

In [13]: x_train,x_test,y_train,y_test = train_test_split(x,y,train_size=0.70)

In [14]: from sklearn.ensemble import RandomForestClassifier

In [15]: rfc = RandomForestClassifier()
 rfc.fit(x_train,y_train)

Out[15]: RandomForestClassifier()

```
In [16]: parameters = {  
    'max_depth':[11,12,13,14,15],  
    'min_samples_leaf':[15,20,25,30,35],  
    'n_estimators':[10,20,30,40,50]  
}
```

```
In [17]: from sklearn.model_selection import GridSearchCV
```

```
In [18]: grid_search = GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring='accuracy')  
grid_search.fit(x_train,y_train)
```

```
Out[18]: GridSearchCV(cv=2, estimator=RandomForestClassifier(),  
    param_grid={'max_depth': [11, 12, 13, 14, 15],  
    'min_samples_leaf': [15, 20, 25, 30, 35],  
    'n_estimators': [10, 20, 30, 40, 50]},  
    scoring='accuracy')
```

```
In [19]: grid_search.best_score_
```

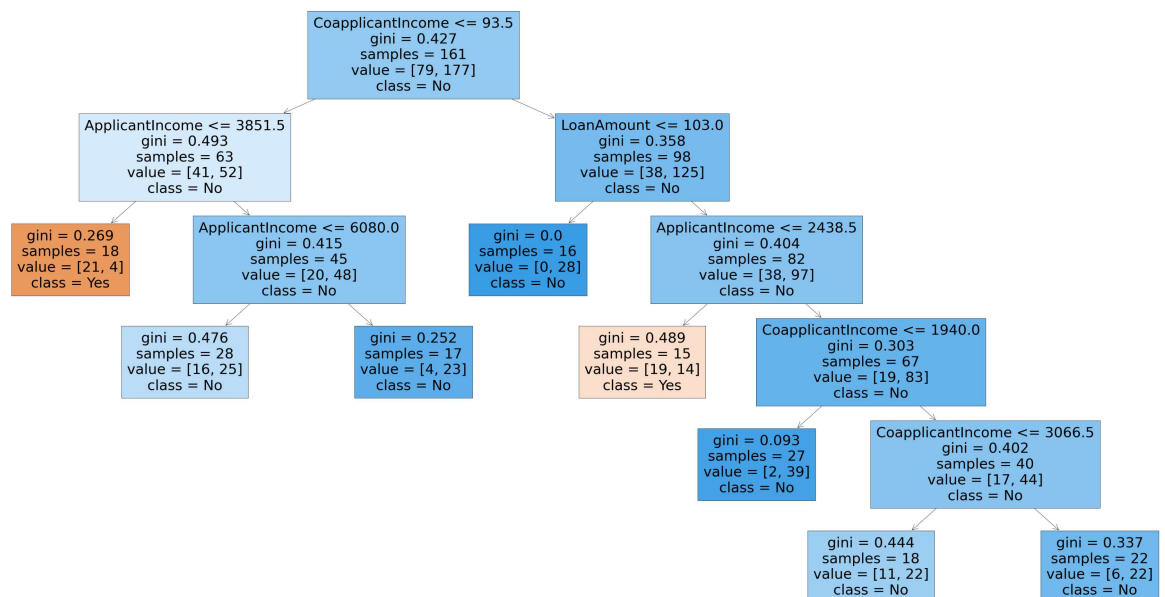
```
Out[19]: 0.703125
```

```
In [20]: from sklearn.tree import plot_tree
```

```
In [21]: rfc_best= grid_search.best_estimator_
```

```
In [22]: plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=["Yes", "No"],filled=True)
```

```
Out[22]: [Text(1623.27272727273, 1993.2, 'CoapplicantIncome <= 93.5\ngini = 0.427\nsamples = 161\nvalue = [79, 177]\n\nclass = No'),
Text(811.6363636363636, 1630.8000000000002, 'ApplicantIncome <= 3851.5\ngini = 0.493\nsamples = 63\nvalue = [41, 52]\n\nclass = No'),
Text(405.8181818181818, 1268.4, 'gini = 0.269\nsamples = 18\nvalue = [21, 4]\n\nclass = Yes'),
Text(1217.4545454545455, 1268.4, 'ApplicantIncome <= 6080.0\ngini = 0.415\nsamples = 45\nvalue = [20, 48]\n\nclass = No'),
Text(811.6363636363636, 906.0, 'gini = 0.476\nsamples = 28\nvalue = [16, 25]\n\nclass = No'),
Text(1623.27272727273, 906.0, 'gini = 0.252\nsamples = 17\nvalue = [4, 23]\n\nclass = No'),
Text(2434.909090909091, 1630.8000000000002, 'LoanAmount <= 103.0\ngini = 0.358\nsamples = 98\nvalue = [38, 125]\n\nclass = No'),
Text(2029.090909090909, 1268.4, 'gini = 0.0\nsamples = 16\nvalue = [0, 28]\n\nclass = No'),
Text(2840.7272727272725, 1268.4, 'ApplicantIncome <= 2438.5\ngini = 0.404\nsamples = 82\nvalue = [38, 97]\n\nclass = No'),
Text(2434.909090909091, 906.0, 'gini = 0.489\nsamples = 15\nvalue = [19, 14]\n\nclass = Yes'),
Text(3246.5454545454545, 906.0, 'CoapplicantIncome <= 1940.0\ngini = 0.303\nsamples = 67\nvalue = [19, 83]\n\nclass = No'),
Text(2840.7272727272725, 543.5999999999999, 'gini = 0.093\nsamples = 27\nvalue = [2, 39]\n\nclass = No'),
Text(3652.3636363636365, 543.5999999999999, 'CoapplicantIncome <= 3066.5\ngini = 0.402\nsamples = 40\nvalue = [17, 44]\n\nclass = No'),
Text(3246.5454545454545, 181.19999999999998, 'gini = 0.444\nsamples = 18\nvalue = [11, 22]\n\nclass = No'),
Text(4058.181818181818, 181.19999999999998, 'gini = 0.337\nsamples = 22\nvalue = [6, 22]\n\nclass = No')]
```



In []: