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

In [2]:
```
df = pd.read_csv(r"C:\Users\user\Downloads\C8_loan-train.csv")
df
```

Out[2]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amou
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	
...	...	...	...	...	...	...	...	...	...	
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	

614 rows × 13 columns

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Loan_ID            614 non-null    object 
 1   Gender             601 non-null    object 
 2   Married            611 non-null    object 
 3   Dependents         599 non-null    object 
 4   Education          614 non-null    object 
 5   Self_Employed      582 non-null    object 
 6   ApplicantIncome    614 non-null    int64  
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object 
 12  Loan_Status        614 non-null    object 
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

In [4]:
```
df2 = df.fillna(0)
```

In [5]:
```
df2
```

Out[5]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amou
0	LP001002	Male	No	0	Graduate	No	5849	0.0	0.0	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	
...	...	...	...	...	...	...	...	...	...	...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	

614 rows × 13 columns

In [6]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Loan_ID            614 non-null    object 
 1   Gender             614 non-null    object 
 2   Married            614 non-null    object 
 3   Dependents         614 non-null    object 
 4   Education          614 non-null    object 
 5   Self_Employed      614 non-null    object 
 6   ApplicantIncome    614 non-null    int64  
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         614 non-null    float64
 9   Loan_Amount_Term   614 non-null    float64
 10  Credit_History     614 non-null    float64
 11  Property_Area      614 non-null    object 
 12  Loan_Status        614 non-null    object 
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

In [7]:
```
df3 = df2[['ApplicantIncome','CoapplicantIncome','LoanAmount','Loan_Amount_Term','Credit_History','Loan_Status
df3
```

Out[7]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Loan_Status
0	5849	0.0	0.0	360.0	1.0	Y
1	4583	1508.0	128.0	360.0	1.0	N
2	3000	0.0	66.0	360.0	1.0	Y
3	2583	2358.0	120.0	360.0	1.0	Y
4	6000	0.0	141.0	360.0	1.0	Y
...	...	...	...	...	...	...
609	2900	0.0	71.0	360.0	1.0	Y
610	4106	0.0	40.0	180.0	1.0	Y
611	8072	240.0	253.0	360.0	1.0	Y
612	7583	0.0	187.0	360.0	1.0	Y
613	4583	0.0	133.0	360.0	0.0	N

614 rows × 6 columns

In [8]:
```
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   ApplicantIncome    614 non-null    int64  
 1   CoapplicantIncome  614 non-null    float64
 2   LoanAmount         614 non-null    float64
 3   Loan_Amount_Term   614 non-null    float64
 4   Credit_History     614 non-null    float64
 5   Loan_Status        614 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 28.9+ KB
```

In [10]:
```
df3['Loan_Status'].value_counts()
```

Out[10]:
```
Y    422
N    192
Name: Loan_Status, dtype: int64
```

In [11]:
```
ch = {"Loan_Status":{'Y':1,'N':0}}
df3 = df3.replace(ch)
df3
```

Out[11]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Loan_Status
0	5849	0.0	0.0	360.0	1.0	1
1	4583	1508.0	128.0	360.0	1.0	0
2	3000	0.0	66.0	360.0	1.0	1
3	2583	2358.0	120.0	360.0	1.0	1
4	6000	0.0	141.0	360.0	1.0	1
...	...	...	...	...	...	...
609	2900	0.0	71.0	360.0	1.0	1
610	4106	0.0	40.0	180.0	1.0	1
611	8072	240.0	253.0	360.0	1.0	1
612	7583	0.0	187.0	360.0	1.0	1
613	4583	0.0	133.0	360.0	0.0	0

614 rows × 6 columns

In [14]:
```
x = df3[['ApplicantIncome','CoapplicantIncome','LoanAmount','Loan_Amount_Term','Credit_History']]
y = df3['Loan_Status']
```

In [15]:
```
from sklearn.model_selection import train_test_split
```

In [16]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,train_size=0.70)
```

In [17]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [18]:
```
rfc = RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[18]:
```
RandomForestClassifier()
```

In [19]:
```
parameters = {
    'max_depth':[1,2,3,4,5],
    'min_samples_leaf':[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

In [20]:
```
from sklearn.model_selection import GridSearchCV
```

In [21]:
```
grid_search = GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring='accuracy')
grid_search.fit(x_train,y_train)
```

Out[21]:
```
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 [22]:
```
grid_search.best_score_
```

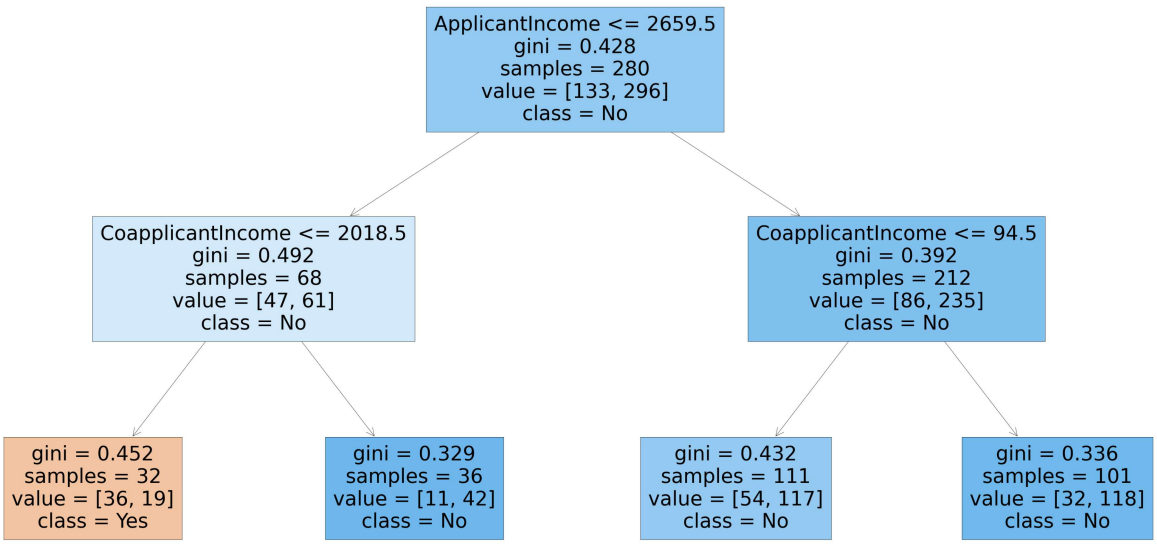
Out[22]:
```
0.7809280591175831
```

In [23]:
```
from sklearn.tree import plot_tree
```

In [24]:
```
rfc_best= grid_search.best_estimator_
```

In [25]:
```
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=["Yes","No"],filled=True)
```

Out[25]:
```
[Text(2232.0, 1812.0, 'ApplicantIncome <= 2659.5\ngini = 0.428\nsamples = 280\nvalue = [133, 296]\nclass = N
o'),
 Text(1116.0, 1087.2, 'CoapplicantIncome <= 2018.5\ngini = 0.492\nsamples = 68\nvalue = [47, 61]\nclass = N
o'),
 Text(558.0, 362.39999999999986, 'gini = 0.452\nsamples = 32\nvalue = [36, 19]\nclass = Yes'),
 Text(1674.0, 362.39999999999986, 'gini = 0.329\nsamples = 36\nvalue = [11, 42]\nclass = No'),
 Text(3348.0, 1087.2, 'CoapplicantIncome <= 94.5\ngini = 0.392\nsamples = 212\nvalue = [86, 235]\nclass = N
o'),
 Text(2790.0, 362.39999999999986, 'gini = 0.432\nsamples = 111\nvalue = [54, 117]\nclass = No'),
 Text(3906.0, 362.39999999999986, 'gini = 0.336\nsamples = 101\nvalue = [32, 118]\nclass = No')]
```



In [ ]:
```
```