In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

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

Out[2]:

ried	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	L
No	0	Graduate	No	5849	0.0	NaN	
Yes	1	Graduate	No	4583	1508.0	128.0	
Yes	0	Graduate	Yes	3000	0.0	66.0	
Yes	0	Not Graduate	No	2583	2358.0	120.0	
No	0	Graduate	No	6000	0.0	141.0	
...	...	...	...	...	...	...	
No	0	Graduate	No	2900	0.0	71.0	
Yes	3+	Graduate	No	4106	0.0	40.0	
Yes	1	Graduate	No	8072	240.0	253.0	
Yes	2	Graduate	No	7583	0.0	187.0	
No	0	Graduate	Yes	4583	0.0	133.0	

In [11]:

```
ge=df[['ApplicantIncome','Loan_Status']]
```

In [12]:

```
d=ge.fillna(20)
```

In [13]:

```
ge['Loan_Status'].value_counts()
```

Out[13]:

```
Y    422
N    192
Name: Loan_Status, dtype: int64
```

In [14]:

```
x=ge.drop('Loan_Status',axis=1)
y=ge['Loan_Status']
```

In [15]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [16]:

```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[16]:

```
RandomForestClassifier()
```

In [17]:

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

In [18]:

```
from sklearn.model_selection import GridSearchCV
grid_search= GridSearchCV(estimator = rfc,param_grid=paramets,cv=2,scoring="acc
grid_search.fit(x_train,y_train)
```

Out[18]:

```
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 [19]:

```
grid_search.best_score_
```

Out[19]:

```
0.6923060204303413
```

In [20]:

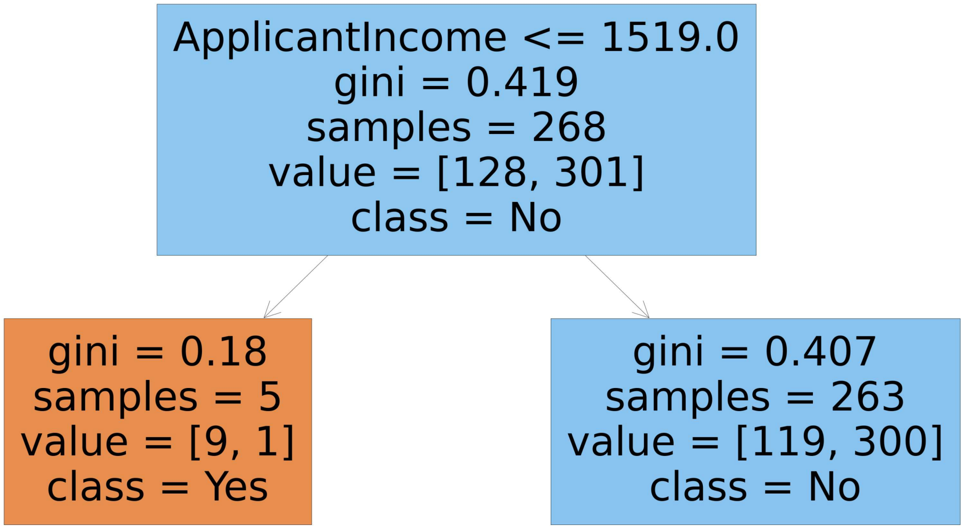
```
rfc_best=grid_search.best_estimator_
```

In [21]:

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','N
```

Out[21]:

```
[Text(2232.0, 1630.8000000000002, 'ApplicantIncome <= 1519.0\ngini = 0.419\ns
amples = 268\nvalue = [128, 301]\nclass = No'),
 Text(1116.0, 543.5999999999999, 'gini = 0.18\nsamples = 5\nvalue = [9, 1]\nc
lass = Yes'),
 Text(3348.0, 543.5999999999999, 'gini = 0.407\nsamples = 263\nvalue = [119,
300]\nclass = No')]
```



```
In [ ]:
```