In [1]:

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

In [2]:

```
df=pd.read_csv('loan_test.csv')
df
```

```
  0  LP001015  Male  Yes   0   Graduate      No   5720
  1  LP001022  Male  Yes   1   Graduate      No   3076   15
  2  LP001031  Male  Yes   2   Graduate      No   5000   18
  3  LP001035  Male  Yes   2   Graduate      No   2340   25
                                Not
  4  LP001051  Male  No    0   Graduate      No   3276
...       ...   ...  ...  ...      ...      ...    ...
                                Not
362  LP002971  Male  Yes  3+   Graduate     Yes   4009   17
363  LP002975  Male  Yes   0   Graduate      No   4158    7
364  LP002980  Male  No    0   Graduate      No   3250   19
365  LP002986  Male  Yes   0   Graduate      No   5000   23
366  LP002989  Male  No    0   Graduate     Yes   9200
```

In [3]:

```
df.columns
```

Out[3]:

```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmoun
t',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area'],
      dtype='object')
```

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 [16]:

```
df['Property_Area'].value_counts()
```

Out[16]:

```
Urban        140
Semiurban    116
Rural        111
Name: Property_Area, dtype: int64
```

In [17]:

```
x=df[['ApplicantIncome']]
y=df['Property_Area']
```

In [19]:

```
d={"Property_Area":{'Urban':1,'Semiurban':2,'Rural':3}}
df=df.replace(df)
print(df)
```

```
      Loan_ID Gender Married Dependents     Education Self_Employed  \
0    LP001015   Male     Yes          0      Graduate            No   
1    LP001022   Male     Yes          1      Graduate            No   
2    LP001031   Male     Yes          2      Graduate            No   
3    LP001035   Male     Yes          2      Graduate            No   
4    LP001051   Male      No          0  Not Graduate            No   
..        ...    ...     ...        ...           ...           ...   
362  LP002971   Male     Yes         3+  Not Graduate           Yes   
363  LP002975   Male     Yes          0      Graduate            No   
364  LP002980   Male      No          0      Graduate            No   
365  LP002986   Male     Yes          0      Graduate            No   
366  LP002989   Male      No          0      Graduate           Yes   

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0               5720                  0       100.0             360.0   
1               3076               1500       150.0             360.0   
2               5000               1800       140.0             360.0   
3               2340               2546       150.0             360.0   
4               3276                  0         NaN             360.0   
..               ...                ...         ...               ...   
362             4009               1777       110.0             360.0   
363             4158                709        75.0             360.0   
364             3250               1993       150.0             360.0   
365             5000               2393       200.0             360.0   
366             9200                  0       108.0             360.0   

     Credit_History Property_Area  
0               1.0         Urban  
1               1.0         Urban  
2               1.0         Urban  
3               NaN         Urban  
4               1.0         Urban  
..              ...           ...  
362             1.0         Urban  
363             1.0         Urban  
364             NaN     Semiurban  
365             1.0         Rural  
366             1.0         Rural  

[367 rows x 12 columns]
```

In [20]:

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

In [21]:

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

Out[21]:

```
RandomForestClassifier()
```

Depth of Tree

In [22]:

```
parameters={"max_depth":[1,2,3,4,5],"min_samples_leaf":[5,23,45,76,78],'n_estimators':[10
```

Cross Validate

In [23]:

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

Out[23]:

```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4, 5],
                         'min_samples_leaf': [5, 23, 45, 76, 78],
                         'n_estimators': [10, 23, 45, 65, 7]},
             scoring='accuracy')
```

Score

In [24]:

```
grid_search.best_score_
```

Out[24]:

```
0.45454545454545453
```

In [25]:

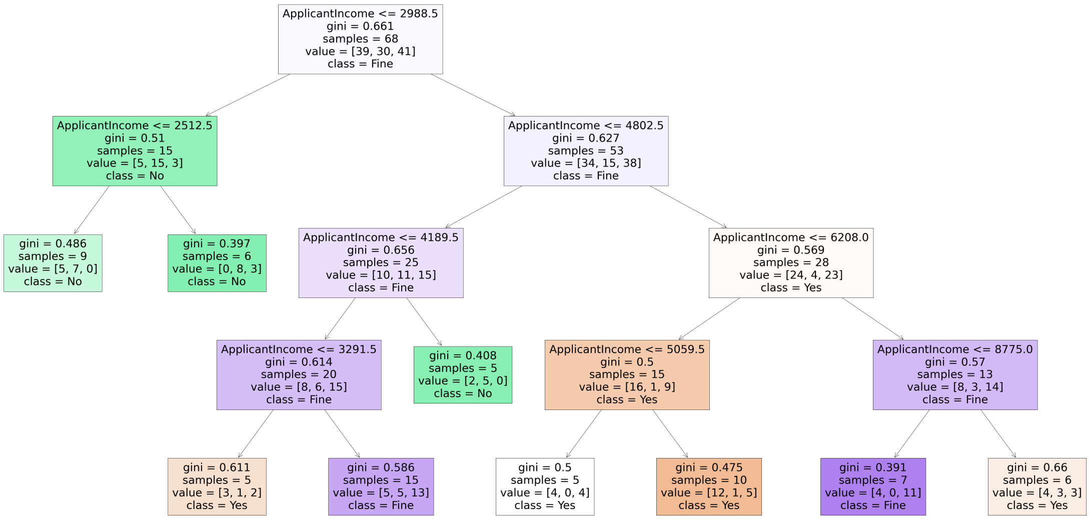
```
rfc_best=grid_search.best_estimator_
```

In [26]:

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No','Fine']
```

Out[26]:

```
[Text(1514.5714285714284, 1956.96, 'ApplicantIncome <= 2988.5\ngini = 0.66
1\nsamples = 68\nvalue = [39, 30, 41]\nclass = Fine'),
 Text(637.7142857142857, 1522.0800000000002, 'ApplicantIncome <= 2512.5\ng
ini = 0.51\nsamples = 15\nvalue = [5, 15, 3]\nclass = No'),
 Text(318.85714285714283, 1087.2, 'gini = 0.486\nsamples = 9\nvalue = [5,
7, 0]\nclass = No'),
 Text(956.5714285714284, 1087.2, 'gini = 0.397\nsamples = 6\nvalue = [0,
8, 3]\nclass = No'),
 Text(2391.428571428571, 1522.0800000000002, 'ApplicantIncome <= 4802.5\ng
ini = 0.627\nsamples = 53\nvalue = [34, 15, 38]\nclass = Fine'),
 Text(1594.2857142857142, 1087.2, 'ApplicantIncome <= 4189.5\ngini = 0.656
\nsamples = 25\nvalue = [10, 11, 15]\nclass = Fine'),
 Text(1275.4285714285713, 652.3200000000002, 'ApplicantIncome <= 3291.5\ng
ini = 0.614\nsamples = 20\nvalue = [8, 6, 15]\nclass = Fine'),
 Text(956.5714285714284, 217.44000000000005, 'gini = 0.611\nsamples = 5\nv
alue = [3, 1, 2]\nclass = Yes'),
 Text(1594.2857142857142, 217.44000000000005, 'gini = 0.586\nsamples = 15
\nvalue = [5, 5, 13]\nclass = Fine'),
 Text(1913.1428571428569, 652.3200000000002, 'gini = 0.408\nsamples = 5\nv
alue = [2, 5, 0]\nclass = No'),
 Text(3188.5714285714284, 1087.2, 'ApplicantIncome <= 6208.0\ngini = 0.569
\nsamples = 28\nvalue = [24, 4, 23]\nclass = Yes'),
 Text(2550.8571428571427, 652.3200000000002, 'ApplicantIncome <= 5059.5\ng
ini = 0.5\nsamples = 15\nvalue = [16, 1, 9]\nclass = Yes'),
 Text(2232.0, 217.44000000000005, 'gini = 0.5\nsamples = 5\nvalue = [4, 0,
4]\nclass = Yes'),
 Text(2869.7142857142853, 217.44000000000005, 'gini = 0.475\nsamples = 10
\nvalue = [12, 1, 5]\nclass = Yes'),
 Text(3826.2857142857138, 652.3200000000002, 'ApplicantIncome <= 8775.0\ng
ini = 0.57\nsamples = 13\nvalue = [8, 3, 14]\nclass = Fine'),
 Text(3507.428571428571, 217.44000000000005, 'gini = 0.391\nsamples = 7\nv
alue = [4, 0, 11]\nclass = Fine'),
 Text(4145.142857142857, 217.44000000000005, 'gini = 0.66\nsamples = 6\nva
lue = [4, 3, 3]\nclass = Yes')]
```



In [ ]: