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\loan test.csv")
df
```

Out[2]:

| | Loan_ID | Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantI |
|---|---|---|---|---|---|---|---|---|
| **0** | LP001015 | Male | Yes | 0 | Graduate | No | 5720 | |
| **1** | LP001022 | Male | Yes | 1 | Graduate | No | 3076 | |
| **2** | LP001031 | Male | Yes | 2 | Graduate | No | 5000 | |
| **3** | LP001035 | Male | Yes | 2 | Graduate | No | 2340 | |
| **4** | LP001051 | Male | No | 0 | Not Graduate | No | 3276 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **362** | LP002971 | Male | Yes | 3+ | Not Graduate | Yes | 4009 | |
| **363** | LP002975 | Male | Yes | 0 | Graduate | No | 4158 | |
| **364** | LP002980 | Male | No | 0 | Graduate | No | 3250 | |
| **365** | LP002986 | Male | Yes | 0 | Graduate | No | 5000 | |
| **366** | LP002989 | Male | No | 0 | Graduate | Yes | 9200 | |

367 rows × 12 columns

In [3]:
```
df.columns
```

Out[3]:
```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       '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 [5]:
```
df['Property_Area'].value_counts()
```

Out[5]:
```
Urban        140
Semiurban    116
Rural        111
Name: Property_Area, dtype: int64
```

In [6]:
```
x=df[['ApplicantIncome', 'CoapplicantIncome']]
y=df['Property_Area']
```

In [7]:
```
g1={"Property_Area":{'Urban':1,'Semiurban':2,'Rural':3,}}
df=df.replace(g1)
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       110.0             360.0
1               3076               1500       126.0             360.0
2               5000               1800       208.0             360.0
3               2340               2546       100.0             360.0
4               3276                  0        78.0             360.0
..               ...                ...         ...               ...
362             4009               1777       113.0             360.0
363             4158                709       115.0             360.0
364             3250               1993       126.0             360.0
365             5000               2393       158.0             360.0
366             9200                  0        98.0             180.0

     Credit_History  Property_Area
0               1.0              1
1               1.0              1
2               1.0              1
3               NaN              1
4               1.0              1
..              ...            ...
362             1.0              1
363             1.0              1
364             NaN              2
365             1.0              3
366             1.0              3
```

```
[367 rows x 12 columns]
```

In [8]:
```
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 [9]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[9]:
```
RandomForestClassifier()
```

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

In [11]:
```
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[11]:
```
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 [12]:
```
grid_search.best_score_
```

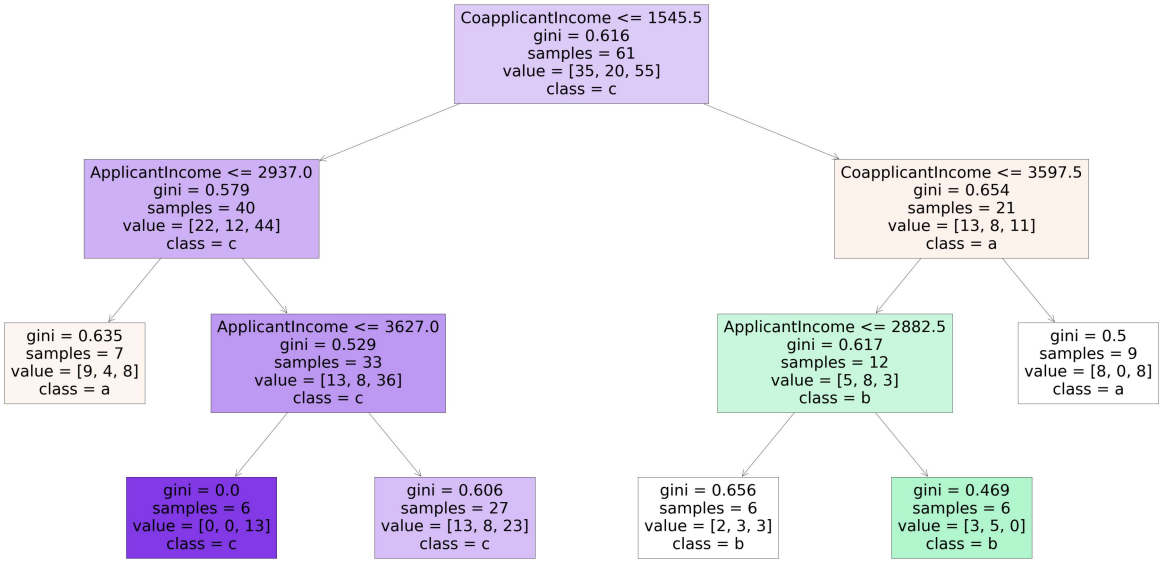
Out[12]:
```
0.4818181818181818
```

In [13]:
```
rfc_best=grid_search.best_estimator_
```

In [14]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['a','b','c'],fil
```

Out[14]:
```
[Text(2232.0, 1902.6000000000001, 'CoapplicantIncome <= 1545.5\ngini = 0.616\nsamples =
61\nvalue = [35, 20, 55]\nclass = c'),
 Text(892.8, 1359.0, 'ApplicantIncome <= 2937.0\ngini = 0.579\nsamples = 40\nvalue = [2
2, 12, 44]\nclass = c'),
 Text(446.4, 815.4000000000001, 'gini = 0.635\nsamples = 7\nvalue = [9, 4, 8]\nclass =
a'),
 Text(1339.1999999999998, 815.4000000000001, 'ApplicantIncome <= 3627.0\ngini = 0.529\ns
amples = 33\nvalue = [13, 8, 36]\nclass = c'),
 Text(892.8, 271.79999999999995, 'gini = 0.0\nsamples = 6\nvalue = [0, 0, 13]\nclass =
c'),
 Text(1785.6, 271.79999999999995, 'gini = 0.606\nsamples = 27\nvalue = [13, 8, 23]\nclas
s = c'),
 Text(3571.2, 1359.0, 'CoapplicantIncome <= 3597.5\ngini = 0.654\nsamples = 21\nvalue =
[13, 8, 11]\nclass = a'),
 Text(3124.7999999999997, 815.4000000000001, 'ApplicantIncome <= 2882.5\ngini = 0.617\ns
```

```
amples = 12\nvalue = [5, 8, 3]\nclass = b'),
 Text(2678.3999999999996, 271.79999999999995, 'gini = 0.656\nsamples = 6\nvalue = [2, 3,
3]\nclass = b'),
 Text(3571.2, 271.79999999999995, 'gini = 0.469\nsamples = 6\nvalue = [3, 5, 0]\nclass =
b'),
 Text(4017.6, 815.4000000000001, 'gini = 0.5\nsamples = 9\nvalue = [8, 0, 8]\nclass =
a')]
```




In [ ]: