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

In [2]:
```
test_df=pd.read_csv(r"C:\Users\Sudheer\AppData\Local\Microsoft\Windows\INetCac
test_df
```

Out[2]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile
0	1	1043	1	1.8	1	14	0	5	0.1	
1	2	841	1	0.5	1	4	1	61	0.8	
2	3	1807	1	2.8	0	1	0	27	0.9	
3	4	1546	0	0.5	1	18	1	25	0.5	
4	5	1434	0	1.4	0	11	1	49	0.5	
...	...	...	...	...	...	...	...	...	...	
995	996	1700	1	1.9	0	0	1	54	0.5	
996	997	609	0	1.8	1	0	0	13	0.9	
997	998	1185	0	1.4	0	1	1	8	0.5	
998	999	1533	1	0.5	1	0	0	50	0.4	
999	1000	1270	1	0.5	0	4	1	35	0.1	

1000 rows × 21 columns

In [3]:

```
test_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   id             1000 non-null   int64  
 1   battery_power  1000 non-null   int64  
 2   blue           1000 non-null   int64  
 3   clock_speed    1000 non-null   float64
 4   dual_sim       1000 non-null   int64  
 5   fc             1000 non-null   int64  
 6   four_g         1000 non-null   int64  
 7   int_memory     1000 non-null   int64  
 8   m_dep          1000 non-null   float64
 9   mobile_wt      1000 non-null   int64  
 10  n_cores        1000 non-null   int64  
 11  pc             1000 non-null   int64  
 12  px_height      1000 non-null   int64  
 13  px_width       1000 non-null   int64  
 14  ram            1000 non-null   int64  
 15  sc_h           1000 non-null   int64  
 16  sc_w           1000 non-null   int64  
 17  talk_time      1000 non-null   int64  
 18  three_g        1000 non-null   int64  
 19  touch_screen   1000 non-null   int64  
 20  wifi           1000 non-null   int64  
dtypes: float64(2), int64(19)
memory usage: 164.2 KB
```

In [4]:

```
x=test_df.drop('wifi',axis=1)
y=test_df['wifi']
```

In [5]:

```
test_df['dual_sim'].value_counts()
```

Out[5]:

```
dual_sim
1    517
0    483
Name: count, dtype: int64
```

In [6]:

```
m={"three_g":{"Yes":1,"No":0}}
test_df=test_df.replace(m)
test_df
```

Out[6]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile
0	1	1043	1	1.8	1	14	0	5	0.1	
1	2	841	1	0.5	1	4	1	61	0.8	
2	3	1807	1	2.8	0	1	0	27	0.9	
3	4	1546	0	0.5	1	18	1	25	0.5	
4	5	1434	0	1.4	0	11	1	49	0.5	
...	...	...	...	...	...	...	...	...	...	
995	996	1700	1	1.9	0	0	1	54	0.5	
996	997	609	0	1.8	1	0	0	13	0.9	
997	998	1185	0	1.4	0	1	1	8	0.5	
998	999	1533	1	0.5	1	0	0	50	0.4	
999	1000	1270	1	0.5	0	4	1	35	0.1	

1000 rows × 21 columns

In [7]:

```
x=test_df.drop('wifi',axis=1)
y=test_df['wifi']
```

In [8]:

```
from sklearn.model_selection import train_test_split
(x_train,x_test,y_train,y_test)=train_test_split(x,y,train_size=0.7,random_sta
x_train.shape,x_test.shape
```

Out[8]:

```
((700, 20), (300, 20))
```

In [9]:

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

Out[9]:

```
RandomForestClassifier()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [25]:

```
rf=RandomForestClassifier()
```

In [26]:

```
params={'max_depth':[2,3,5,10,20],
       'min_samples_leaf':[5,10,20,50,100,200],
       'n_estimators':[10,25,30,50,100,200]}
```

In [27]:

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

Out[27]:

```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [2, 3, 5, 10, 20],
                         'min_samples_leaf': [5, 10, 20, 50, 100, 200],
                         'n_estimators': [10, 25, 30, 50, 100, 200]},
             scoring='accuracy')
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [28]:

```
grid_search.best_score_
```

Out[28]:

```
0.5571428571428572
```

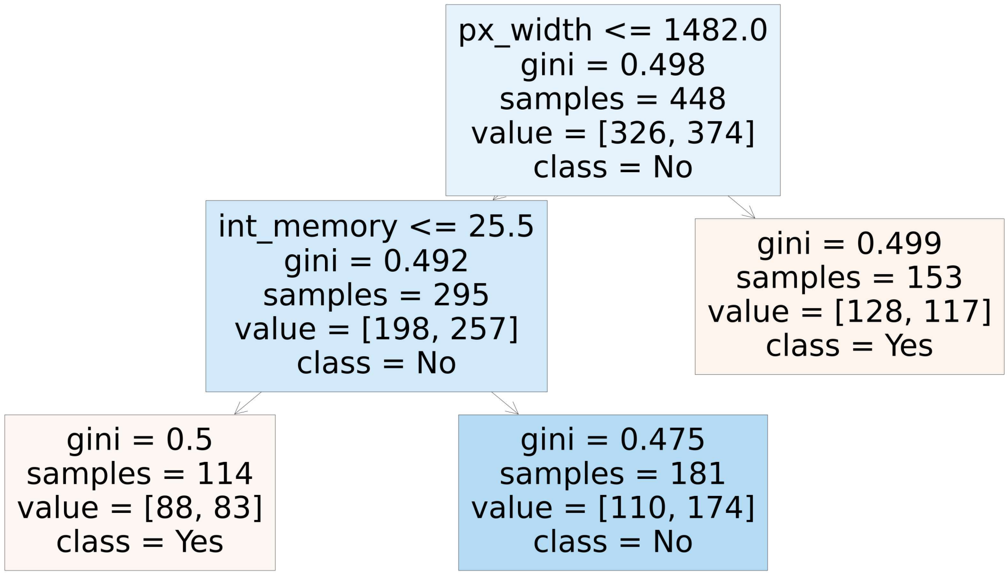
In [29]:

```
rf_best=grid_search.best_estimator_
print(rf_best)
```

```
RandomForestClassifier(max_depth=3, min_samples_leaf=100, n_estimators=200)
```

In [30]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[5],feature_names=x.columns,class_names=["Yes","N
```

Out[30]:
```
[Text(0.6, 0.8333333333333334, 'px_width <= 1482.0\ngini = 0.498\nsamples = 4
48\nvalue = [326, 374]\nclass = No'),
 Text(0.4, 0.5, 'int_memory <= 25.5\ngini = 0.492\nsamples = 295\nvalue = [19
8, 257]\nclass = No'),
 Text(0.2, 0.16666666666666666, 'gini = 0.5\nsamples = 114\nvalue = [88, 83]
\nclass = Yes'),
 Text(0.6, 0.16666666666666666, 'gini = 0.475\nsamples = 181\nvalue = [110, 1
74]\nclass = No'),
 Text(0.8, 0.5, 'gini = 0.499\nsamples = 153\nvalue = [128, 117]\nclass = Ye
s')]
```



In [31]:
```
rfc_best.feature_importances_
```

Out[31]:
```
array([0.02959573, 0.05730982, 0.        , 0.04376253, 0.00900582,
       0.09457378, 0.05896247, 0.06613749, 0.01158459, 0.13761976,
       0.01220874, 0.09131765, 0.06455098, 0.12141692, 0.09691083,
       0.01230052, 0.        , 0.09274237, 0.        , 0.        ])
```

In [32]:
```
imp_df=pd.DataFrame({'Varnames':x_train.columns,"Imp":rf_best.feature_importan
imp_df.sort_values(by="Imp",ascending=False)
```

Out[32]:
```
         Varnames       Imp
13       px_width  0.149878
 9      mobile_wt  0.108896
 3    clock_speed  0.104175
 1  battery_power  0.090303
 8          m_dep  0.089869
 7     int_memory  0.062564
 5             fc  0.056642
14            ram  0.055700
17      talk_time  0.054610
12      px_height  0.053771
11             pc  0.045400
 0             id  0.032350
 6         four_g  0.024107
16           sc_w  0.022403
15           sc_h  0.017338
10        n_cores  0.014274
 4       dual_sim  0.006316
19   touch_screen  0.004878
 2           blue  0.003912
18        three_g  0.002614
```

In [ ]:

In [ ]: