In [20]:

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

In [21]:

```python
train_df=pd.read_csv(r"C:\Users\DHEEPAK\Desktop\Mobile_Price_Classification_train.csv")
train_df
```

Out[21]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_
0	842	0	2.2	0	1	0	7	0.6	188	
1	1021	1	0.5	1	0	1	53	0.7	136	
2	563	1	0.5	1	2	1	41	0.9	145	
3	615	1	2.5	0	0	0	10	0.8	131	
4	1821	1	1.2	0	13	1	44	0.6	141	
...	...	...	...	...	...	...	...	...	...	
1995	794	1	0.5	1	0	1	2	0.8	106	
1996	1965	1	2.6	1	0	0	39	0.2	187	
1997	1911	0	0.9	1	1	1	36	0.7	108	
1998	1512	0	0.9	0	4	1	46	0.1	145	
1999	510	1	2.0	1	5	1	45	0.9	168	

2000 rows × 21 columns

In [22]:

```
test_df=pd.read_csv(r"C:\Users\DHEEPAK\Desktop\Mobile_Price_Classification_test.csv")
test_df
```

Out[22]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_w
0	1	1043	1	1.8	1	14	0	5	0.1	19
1	2	841	1	0.5	1	4	1	61	0.8	19
2	3	1807	1	2.8	0	1	0	27	0.9	18
3	4	1546	0	0.5	1	18	1	25	0.5	9
4	5	1434	0	1.4	0	11	1	49	0.5	10
...	...	...	...	...	...	...	...	...	...	
995	996	1700	1	1.9	0	0	1	54	0.5	17
996	997	609	0	1.8	1	0	0	13	0.9	18
997	998	1185	0	1.4	0	1	1	8	0.5	8
998	999	1533	1	0.5	1	0	0	50	0.4	17
999	1000	1270	1	0.5	0	4	1	35	0.1	14

1000 rows × 21 columns

In [23]:

```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   battery_power  2000 non-null   int64  
 1   blue           2000 non-null   int64  
 2   clock_speed    2000 non-null   float64
 3   dual_sim       2000 non-null   int64  
 4   fc             2000 non-null   int64  
 5   four_g         2000 non-null   int64  
 6   int_memory     2000 non-null   int64  
 7   m_dep          2000 non-null   float64
 8   mobile_wt      2000 non-null   int64  
 9   n_cores        2000 non-null   int64  
 10  pc             2000 non-null   int64  
 11  px_height      2000 non-null   int64  
 12  px_width       2000 non-null   int64  
 13  ram            2000 non-null   int64  
 14  sc_h           2000 non-null   int64  
 15  sc_w           2000 non-null   int64  
 16  talk_time      2000 non-null   int64  
 17  three_g        2000 non-null   int64  
 18  touch_screen   2000 non-null   int64  
 19  wifi           2000 non-null   int64  
 20  price_range    2000 non-null   int64  
dtypes: float64(2), int64(19)
memory usage: 328.3 KB
```

In [24]:

```
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 [25]:

```
x=train_df.drop('dual_sim',axis=1)
y=train_df['dual_sim']
```

In [26]:

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

In [27]:

```
train_df['blue'].value_counts()
```

Out[27]:

```
blue
0    1010
1     990
Name: count, dtype: int64
```

In [28]:

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

Out[28]:

```
blue
1    516
0    484
Name: count, dtype: int64
```

In [29]:

```
T={"three_g":{'Yes':1,'No':0}}
train_df=train_df.replace(T)
print(train_df)
```

```
      battery_power  blue  clock_speed  dual_sim  fc  four_g  int_memory
0               842     0          2.2         0   1       0           7  \
1              1021     1          0.5         1   0       1          53
2               563     1          0.5         1   2       1          41
3               615     1          2.5         0   0       0          10
4              1821     1          1.2         0  13       1          44
...             ...   ...          ...       ...  ..     ...         ...
1995            794     1          0.5         1   0       1           2
1996           1965     1          2.6         1   0       0          39
1997           1911     0          0.9         1   1       1          36
1998           1512     0          0.9         0   4       1          46
1999            510     1          2.0         1   5       1          45

      m_dep  mobile_wt  n_cores  ...  px_height  px_width   ram  sc_h  sc_w
0       0.6        188        2  ...         20       756  2549     9     7  \
1       0.7        136        3  ...        905      1988  2631    17     3
2       0.9        145        5  ...       1263      1716  2603    11     2
3       0.8        131        6  ...       1216      1786  2769    16     8
4       0.6        141        2  ...       1208      1212  1411     8     2
...     ...        ...      ...  ...        ...       ...   ...   ...   ...
1995    0.8        106        6  ...       1222      1890   668    13     4
1996    0.2        187        4  ...        915      1965  2032    11    10
1997    0.7        108        8  ...        868      1632  3057     9     1
1998    0.1        145        5  ...        336       670   869    18    10
1999    0.9        168        6  ...        483       754  3919    19     4

      talk_time  three_g  touch_screen  wifi  price_range
0            19        0             0     1            1
1             7        1             1     0            2
2             9        1             1     0            2
3            11        1             0     0            2
4            15        1             1     0            1
...         ...      ...           ...   ...          ...
1995         19        1             1     0            0
1996         16        1             1     1            2
1997          5        1             1     0            3
1998         19        1             1     1            0
1999          2        1             1     1            3

[2000 rows x 21 columns]
```

In [ ]:

```
 
```

In [30]:

```
T={"three_g":{'Yes':1,'No':0}}
test_df=test_df.replace(T)
print(test_df)
```

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

     m_dep  mobile_wt  ...  pc  px_height  px_width   ram  sc_h  sc_w  \
0      0.1        193  ...  16        226      1412  3476    12     7
1      0.8        191  ...  12        746       857  3895     6     0
2      0.9        186  ...   4       1270      1366  2396    17    10
3      0.5         96  ...  20        295      1752  3893    10     0
4      0.5        108  ...  18        749       810  1773    15     8
..     ...        ...  ...  ..        ...       ...   ...   ...   ...
995    0.5        170  ...  17        644       913  2121    14     8
996    0.9        186  ...   2       1152      1632  1933     8     1
997    0.5         80  ...  12        477       825  1223     5     0
998    0.4        171  ...  12         38       832  2509    15    11
999    0.1        140  ...  19        457       608  2828     9     2

     talk_time  three_g  touch_screen  wifi
0            2        0             1     0
1            7        1             0     0
2           10        0             1     1
3            7        1             1     0
4            7        1             0     1
..         ...      ...           ...   ...
995         15        1             1     0
996         19        0             1     1
997         14        1             0     0
998          6        0             1     0
999          3        1             0     1

[1000 rows x 21 columns]
```

In [31]:

```
x=train_df.drop('dual_sim',axis=1)
y=train_df['dual_sim']
```

In [32]:

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

In [33]:

```
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_state=42)
x_train.shape,x_test.shape
```

Out[33]:

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

In [34]:

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

Out[34]:

```
▾ RandomForestClassifier

RandomForestClassifier()
```

In [35]:

```
rf=RandomForestClassifier()
```

In [36]:

```
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 [44]:

```
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[44]:

```
▸             GridSearchCV
▸ estimator: RandomForestClassifier
      ▸ RandomForestClassifier
```

In [40]:

```
grid_search.best_score_
```

Out[40]:

```
0.5557142857142857
```

In [46]:

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

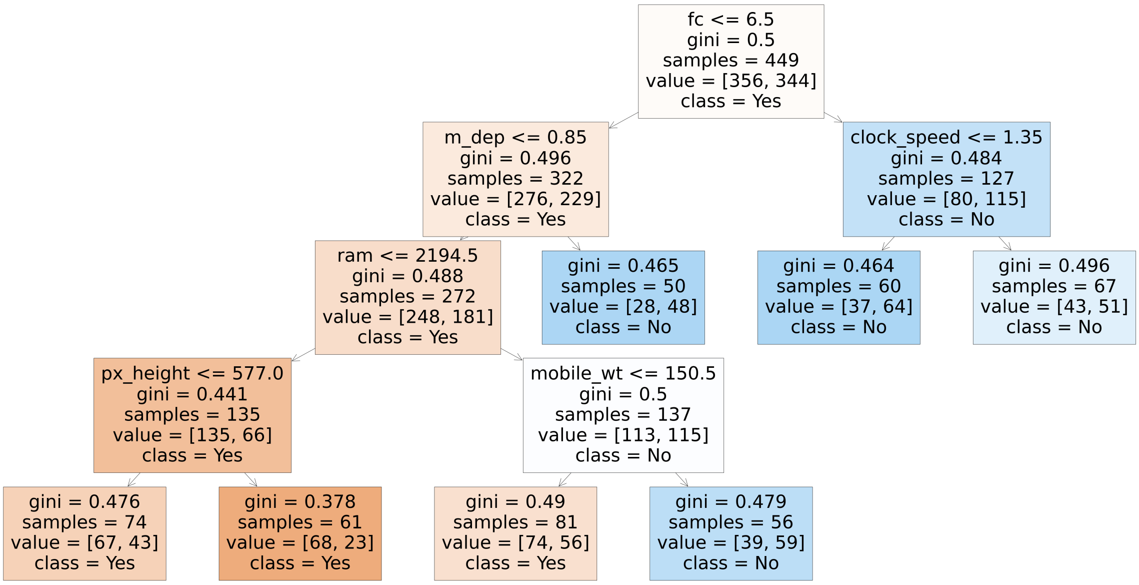
```
RandomForestClassifier(max_depth=20, min_samples_leaf=50, n_estimators=25)
```

In [50]:

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

Out[50]:

```
[Text(0.6363636363636364, 0.9, 'fc <= 6.5\ngini = 0.5\nsamples = 449\nvalue =
[356, 344]\nclass = Yes'),
 Text(0.45454545454545453, 0.7, 'm_dep <= 0.85\ngini = 0.496\nsamples = 322\nv
alue = [276, 229]\nclass = Yes'),
 Text(0.36363636363636365, 0.5, 'ram <= 2194.5\ngini = 0.488\nsamples = 272\nv
alue = [248, 181]\nclass = Yes'),
 Text(0.18181818181818182, 0.3, 'px_height <= 577.0\ngini = 0.441\nsamples = 1
35\nvalue = [135, 66]\nclass = Yes'),
 Text(0.09090909090909091, 0.1, 'gini = 0.476\nsamples = 74\nvalue = [67, 43]
\nclass = Yes'),
 Text(0.2727272727272727, 0.1, 'gini = 0.378\nsamples = 61\nvalue = [68, 23]\n
class = Yes'),
 Text(0.5454545454545454, 0.3, 'mobile_wt <= 150.5\ngini = 0.5\nsamples = 137
\nvalue = [113, 115]\nclass = No'),
 Text(0.45454545454545453, 0.1, 'gini = 0.49\nsamples = 81\nvalue = [74, 56]\n
class = Yes'),
 Text(0.6363636363636364, 0.1, 'gini = 0.479\nsamples = 56\nvalue = [39, 59]\n
class = No'),
 Text(0.5454545454545454, 0.5, 'gini = 0.465\nsamples = 50\nvalue = [28, 48]\n
class = No'),
 Text(0.8181818181818182, 0.7, 'clock_speed <= 1.35\ngini = 0.484\nsamples = 1
27\nvalue = [80, 115]\nclass = No'),
 Text(0.7272727272727273, 0.5, 'gini = 0.464\nsamples = 60\nvalue = [37, 64]\n
class = No'),
 Text(0.9090909090909091, 0.5, 'gini = 0.496\nsamples = 67\nvalue = [43, 51]\n
class = No')]
```

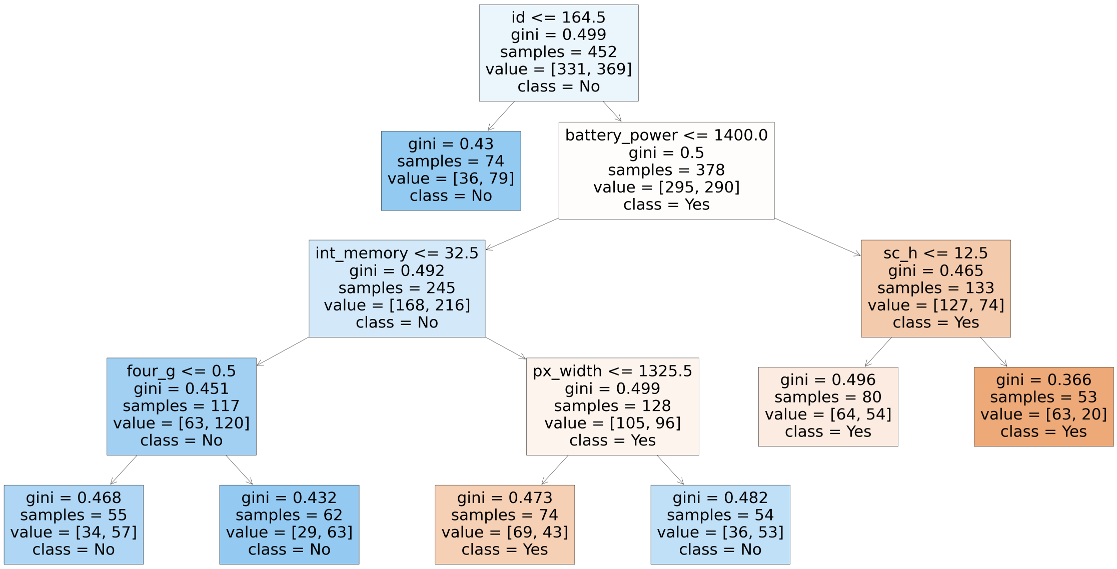


In [51]:

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[7],feature_names=x.columns,class_names=['Yes','No'],filled=Tru
```

Out[51]:

```
[Text(0.5, 0.9, 'id <= 164.5\ngini = 0.499\nsamples = 452\nvalue = [331, 369]
\nclass = No'),
 Text(0.4090909090909091, 0.7, 'gini = 0.43\nsamples = 74\nvalue = [36, 79]\nc
lass = No'),
 Text(0.5909090909090909, 0.7, 'battery_power <= 1400.0\ngini = 0.5\nsamples =
378\nvalue = [295, 290]\nclass = Yes'),
 Text(0.36363636363636365, 0.5, 'int_memory <= 32.5\ngini = 0.492\nsamples = 2
45\nvalue = [168, 216]\nclass = No'),
 Text(0.18181818181818182, 0.3, 'four_g <= 0.5\ngini = 0.451\nsamples = 117\nv
alue = [63, 120]\nclass = No'),
 Text(0.09090909090909091, 0.1, 'gini = 0.468\nsamples = 55\nvalue = [34, 57]
\nclass = No'),
 Text(0.2727272727272727, 0.1, 'gini = 0.432\nsamples = 62\nvalue = [29, 63]\n
class = No'),
 Text(0.5454545454545454, 0.3, 'px_width <= 1325.5\ngini = 0.499\nsamples = 12
8\nvalue = [105, 96]\nclass = Yes'),
 Text(0.45454545454545453, 0.1, 'gini = 0.473\nsamples = 74\nvalue = [69, 43]
\nclass = Yes'),
 Text(0.6363636363636364, 0.1, 'gini = 0.482\nsamples = 54\nvalue = [36, 53]\n
class = No'),
 Text(0.8181818181818182, 0.5, 'sc_h <= 12.5\ngini = 0.465\nsamples = 133\nval
ue = [127, 74]\nclass = Yes'),
 Text(0.7272727272727273, 0.3, 'gini = 0.496\nsamples = 80\nvalue = [64, 54]\n
class = Yes'),
 Text(0.9090909090909091, 0.3, 'gini = 0.366\nsamples = 53\nvalue = [63, 20]\n
class = Yes')]
```



In [52]:

```
rf_best.feature_importances_
```

Out[52]:

```
array([0.10818799, 0.11620779, 0.00616694, 0.0746391 , 0.08398264,
       0.00038537, 0.02412626, 0.04710012, 0.05134397, 0.02683694,
       0.09601713, 0.07501672, 0.05224792, 0.06330306, 0.04853793,
       0.08064983, 0.00584961, 0.        , 0.02331212, 0.01608855])
```

In [54]:

```
imp_df=pd.DataFrame({'Varname':x_train.columns,"Imp":rf_best.feature_importances_})
imp_df.sort_values(by="Imp",ascending=False)
```

Out[54]:

	Varname	Imp
1	battery_power	0.116208
0	id	0.108188
10	pc	0.096017
4	fc	0.083983
15	sc_w	0.080650
11	px_height	0.075017
3	clock_speed	0.074639
13	ram	0.063303
12	px_width	0.052248
8	mobile_wt	0.051344
14	sc_h	0.048538
7	m_dep	0.047100
9	n_cores	0.026837
6	int_memory	0.024126
18	touch_screen	0.023312
19	wifi	0.016089
2	blue	0.006167
16	talk_time	0.005850
5	four_g	0.000385
17	three_g	0.000000

In [ ]:

```

```