In [1]:

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

In [2]:

```
df=pd.read_csv(r"C:\Users\Arshad Shaik\Downloads\Mobile_Price_Classification_test.csv")
df
```

Out[2]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height
0	1	1043	1	1.8	1	14	0	5	0.1	193	...	16	226
1	2	841	1	0.5	1	4	1	61	0.8	191	...	12	746
2	3	1807	1	2.8	0	1	0	27	0.9	186	...	4	1270
3	4	1546	0	0.5	1	18	1	25	0.5	96	...	20	295
4	5	1434	0	1.4	0	11	1	49	0.5	108	...	18	749
...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	996	1700	1	1.9	0	0	1	54	0.5	170	...	17	644
996	997	609	0	1.8	1	0	0	13	0.9	186	...	2	1152
997	998	1185	0	1.4	0	1	1	8	0.5	80	...	12	477
998	999	1533	1	0.5	1	0	0	50	0.4	171	...	12	38
999	1000	1270	1	0.5	0	4	1	35	0.1	140	...	19	457

1000 rows × 21 columns

In [3]:

```
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]:

```
df.describe()
```

Out[4]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000
mean	500.500000	1248.510000	0.516000	1.540900	0.517000	4.593000	0.487000	33.652000	0
std	288.819436	432.458227	0.499994	0.829268	0.499961	4.463325	0.500081	18.128694	0
min	1.000000	500.000000	0.000000	0.500000	0.000000	0.000000	0.000000	2.000000	0
25%	250.750000	895.000000	0.000000	0.700000	0.000000	1.000000	0.000000	18.000000	0
50%	500.500000	1246.500000	1.000000	1.500000	1.000000	3.000000	0.000000	34.500000	0
75%	750.250000	1629.250000	1.000000	2.300000	1.000000	7.000000	1.000000	49.000000	0
max	1000.000000	1999.000000	1.000000	3.000000	1.000000	19.000000	1.000000	64.000000	1

8 rows × 21 columns

In [5]:

```
df.tail()
```

Out[5]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height
995	996	1700	1	1.9	0	0	1	54	0.5	170	...	17	644
996	997	609	0	1.8	1	0	0	13	0.9	186	...	2	1152
997	998	1185	0	1.4	0	1	1	8	0.5	80	...	12	477
998	999	1533	1	0.5	1	0	0	50	0.4	171	...	12	38
999	1000	1270	1	0.5	0	4	1	35	0.1	140	...	19	457

5 rows × 21 columns

In [6]:

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

In [7]:

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

Out[7]:

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

In [8]:

```
m={"three_g":{"yes":1,"No":0}}
df=df.replace(m)
print(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 [9]:

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

In [10]:

```
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[10]:

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

In [11]:

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

Out[11]:

```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [12]:

```
rf=RandomForestClassifier()
```

In [13]:

```
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 [14]:

```
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[14]:

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

In [15]:

```
grid_search.best_score_
```

Out[15]:

```
0.5585714285714286
```

In [16]:

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

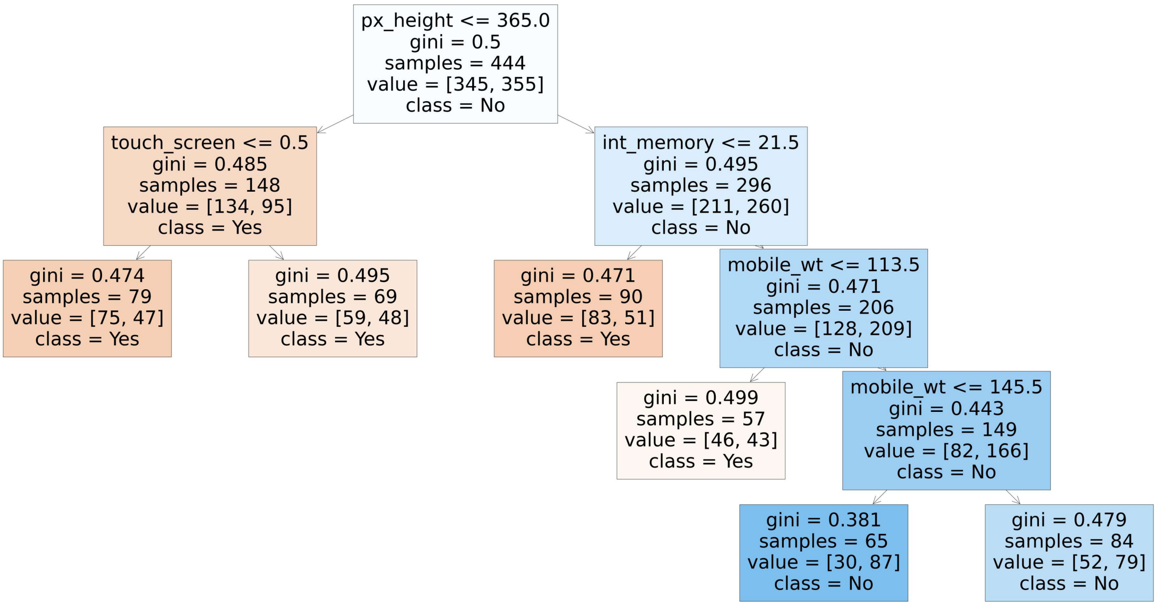
```
RandomForestClassifier(max_depth=10, min_samples_leaf=50, n_estimators=25)
```

In [20]:

```
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=True)
```

Out[20]:

```
[Text(0.4, 0.9, 'px_height <= 365.0\ngini = 0.5\nsamples = 444\nvalue = [345, 355]\nclass = N
o'),
 Text(0.2, 0.7, 'touch_screen <= 0.5\ngini = 0.485\nsamples = 148\nvalue = [134, 95]\nclass = Ye
s'),
 Text(0.1, 0.5, 'gini = 0.474\nsamples = 79\nvalue = [75, 47]\nclass = Yes'),
 Text(0.3, 0.5, 'gini = 0.495\nsamples = 69\nvalue = [59, 48]\nclass = Yes'),
 Text(0.6, 0.7, 'int_memory <= 21.5\ngini = 0.495\nsamples = 296\nvalue = [211, 260]\nclass = N
o'),
 Text(0.5, 0.5, 'gini = 0.471\nsamples = 90\nvalue = [83, 51]\nclass = Yes'),
 Text(0.7, 0.5, 'mobile_wt <= 113.5\ngini = 0.471\nsamples = 206\nvalue = [128, 209]\nclass = N
o'),
 Text(0.6, 0.3, 'gini = 0.499\nsamples = 57\nvalue = [46, 43]\nclass = Yes'),
 Text(0.8, 0.3, 'mobile_wt <= 145.5\ngini = 0.443\nsamples = 149\nvalue = [82, 166]\nclass = N
o'),
 Text(0.7, 0.1, 'gini = 0.381\nsamples = 65\nvalue = [30, 87]\nclass = No'),
 Text(0.9, 0.1, 'gini = 0.479\nsamples = 84\nvalue = [52, 79]\nclass = No')]
```



In [18]:

```
rf_best.feature_importances_
```

Out[18]:

```
array([0.0241424 , 0.09968909, 0.02228753, 0.06057503, 0.02304026,
       0.03701829, 0.01597702, 0.09365474, 0.08441055, 0.08530544,
       0.00997564, 0.03481966, 0.0673099 , 0.11056347, 0.07005003,
       0.04598406, 0.04883798, 0.06560401, 0.        , 0.0007549 ])
```

In [19]:

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

Out[19]:

	Varname	IMP
13	px_width	0.110563
1	battery_power	0.099689
7	int_memory	0.093655
9	mobile_wt	0.085305
8	m_dep	0.084411
14	ram	0.070050
12	px_height	0.067310
17	talk_time	0.065604
3	clock_speed	0.060575
16	sc_w	0.048838
15	sc_h	0.045984
5	fc	0.037018
11	pc	0.034820
0	id	0.024142
4	dual_sim	0.023040
2	blue	0.022288
6	four_g	0.015977
10	n_cores	0.009976
19	touch_screen	0.000755
18	three_g	0.000000

In [ ]:

```
```