In [35]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt,seaborn as sns
```

In [36]:
```
df=pd.read_csv(r"C:\Users\user\Downloads\Mobile_Price_Classification_train (1).csv")
df
```

Out[36]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1995	794	1	0.5	1	0	1	2	0.8	106	6	...	1222	1890	668
1996	1965	1	2.6	1	0	0	39	0.2	187	4	...	915	1965	2032
1997	1911	0	0.9	1	1	1	36	0.7	108	8	...	868	1632	3057
1998	1512	0	0.9	0	4	1	46	0.1	145	5	...	336	670	869
1999	510	1	2.0	1	5	1	45	0.9	168	6	...	483	754	3919

2000 rows × 21 columns

In [37]:
```
test_df=pd.read_csv(r"C:\Users\user\Downloads\Mobile_Price_Classification_test.csv")
test_df
```

Out[37]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height	px_width	ram
0	1	1043	1	1.8	1	14	0	5	0.1	193	...	16	226	1412	3476
1	2	841	1	0.5	1	4	1	61	0.8	191	...	12	746	857	3895
2	3	1807	1	2.8	0	1	0	27	0.9	186	...	4	1270	1366	2396
3	4	1546	0	0.5	1	18	1	25	0.5	96	...	20	295	1752	3893
4	5	1434	0	1.4	0	11	1	49	0.5	108	...	18	749	810	1773
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	996	1700	1	1.9	0	0	1	54	0.5	170	...	17	644	913	2121
996	997	609	0	1.8	1	0	0	13	0.9	186	...	2	1152	1632	1933
997	998	1185	0	1.4	0	1	1	8	0.5	80	...	12	477	825	1223
998	999	1533	1	0.5	1	0	0	50	0.4	171	...	12	38	832	2509
999	1000	1270	1	0.5	0	4	1	35	0.1	140	...	19	457	608	2828

1000 rows × 21 columns

In [38]:

```
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 [39]:

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

In [40]:

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

Out[40]:

```
dual_sim
1    1019
0     981
Name: count, dtype: int64
```

```
In [41]: HO={"four_g":{"Yes":1,"No":0}}
         df=df.replace(HO)
         print(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 [42]:
```
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 [43]:
```
test_df.describe()
```

Out[43]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobil
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.0
mean	500.500000	1248.510000	0.516000	1.540900	0.517000	4.593000	0.487000	33.652000	0.517500	139.5
std	288.819436	432.458227	0.499994	0.829268	0.499961	4.463325	0.500081	18.128694	0.280861	34.8
min	1.000000	500.000000	0.000000	0.500000	0.000000	0.000000	0.000000	2.000000	0.100000	80.0
25%	250.750000	895.000000	0.000000	0.700000	0.000000	1.000000	0.000000	18.000000	0.300000	109.7
50%	500.500000	1246.500000	1.000000	1.500000	1.000000	3.000000	0.000000	34.500000	0.500000	139.0
75%	750.250000	1629.250000	1.000000	2.300000	1.000000	7.000000	1.000000	49.000000	0.800000	170.0
max	1000.000000	1999.000000	1.000000	3.000000	1.000000	19.000000	1.000000	64.000000	1.000000	200.0

8 rows × 21 columns

In [44]:
```
test_df['blue'].value_counts()
```

Out[44]:
```
blue
1    516
0    484
Name: count, dtype: int64
```

In [45]:

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

Out[45]:

```
fc
0     210
1     124
2      97
4      80
5      74
3      70
6      59
7      50
9      41
8      38
10     37
11     29
13     21
12     17
14     16
15     12
16     11
18     10
17      2
19      2
Name: count, dtype: int64
```

In [46]:

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

In [47]:

```
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[47]:

```
((1400, 20), (600, 20))
```

In [48]:

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

Out[48]:

```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [49]:

```
rf=RandomForestClassifier()
```

In [50]:

```
param={'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 [51]:

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

Out[51]:

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

In [52]:

```
grid_search.best_score_
```

Out[52]:

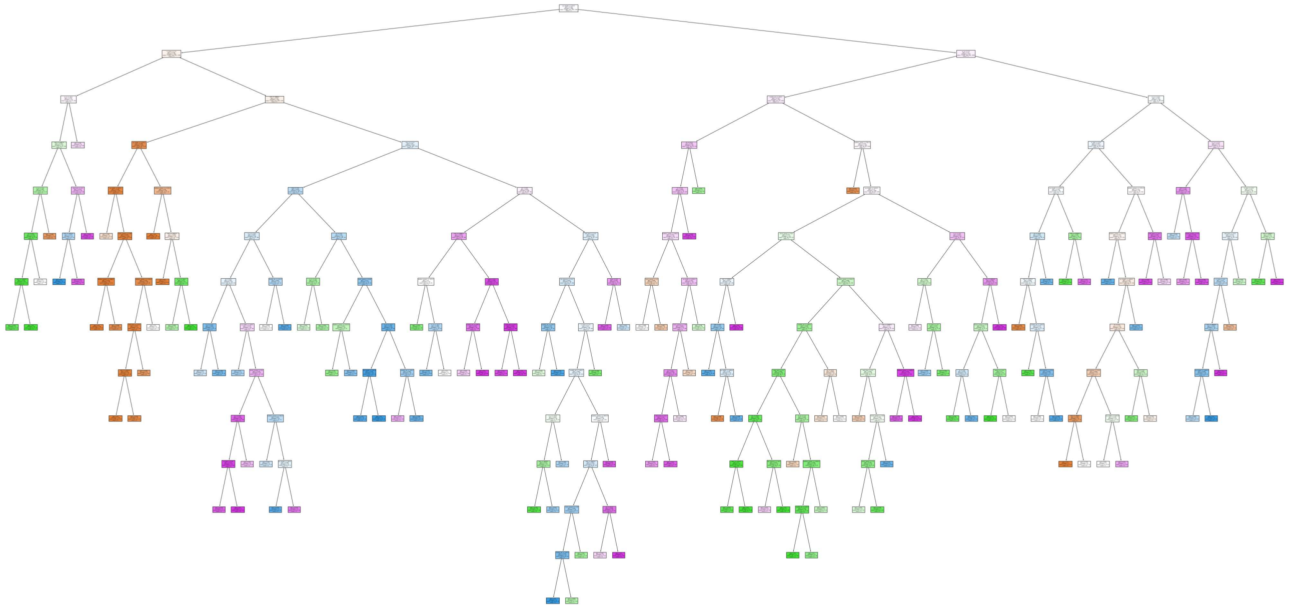
```
0.835
```

In [53]:

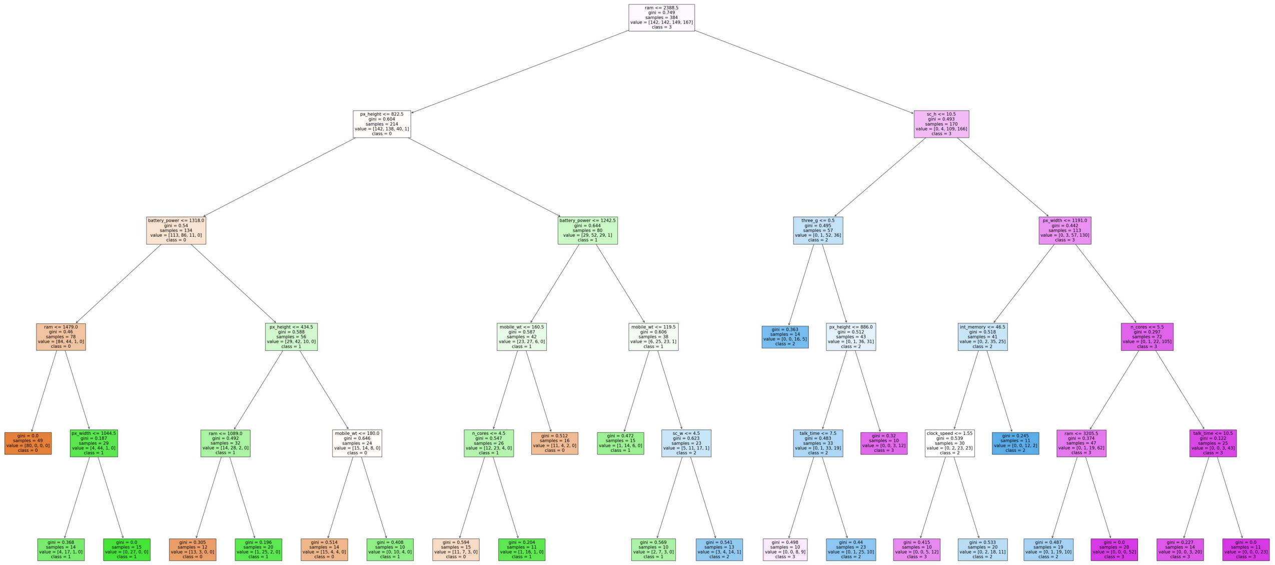
```
rf_best=grid_search.best_estimator_
print(rf_best)
```

```
RandomForestClassifier(max_depth=20, min_samples_leaf=5)
```

In [54]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[5],feature_names=x.columns,class_names=['0','1','2','3'],filled=True);
```



In [73]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[7],feature_names=x.columns,class_names=['0','1','2','3'],filled=True);
```


ram <= 2388.5 gini = 0.749 samples = 384 value = [142, 142, 149, 167] class = 3
px_height <= 822.5 gini = 0.604 samples = 214 value = [142, 138, 40, 1] class = 0
sc_h <= 10.5 gini = 0.493 samples = 170 value = [0, 4, 109, 166] class = 3
battery_power <= 1318.0 gini = 0.54 samples = 134 value = [113, 86, 11, 0] class = 0
battery_power <= 1242.5 gini = 0.644 samples = 80 value = [29, 52, 29, 1] class = 1
three_g <= 0.5 gini = 0.495 samples = 57 value = [0, 1, 52, 36] class = 2
px_width <= 1191.0 gini = 0.442 samples = 113 value = [0, 3, 57, 130] class = 3
ram <= 1479.0 gini = 0.46 samples = 78 value = [84, 44, 1, 0] class = 0
px_height <= 434.5 gini = 0.588 samples = 56 value = [29, 42, 10, 0] class = 1
mobile_wt <= 160.5 gini = 0.587 samples = 42 value = [23, 27, 6, 0] class = 1
mobile_wt <= 119.5 gini = 0.606 samples = 38 value = [6, 25, 23, 1] class = 1
gini = 0.363 samples = 14 value = [0, 0, 16, 5] class = 2
px_height <= 886.0 gini = 0.512 samples = 43 value = [0, 1, 36, 31] class = 2
int_memory <= 46.5 gini = 0.518 samples = 41 value = [0, 2, 35, 25] class = 2
n_cores <= 5.5 gini = 0.297 samples = 72 value = [0, 1, 22, 105] class = 3
gini = 0.0 samples = 49 value = [80, 0, 0, 0] class = 0
px_width <= 1044.5 gini = 0.187 samples = 29 value = [4, 44, 1, 0] class = 1
ram <= 1089.0 gini = 0.492 samples = 32 value = [14, 28, 2, 0] class = 1
mobile_wt <= 180.0 gini = 0.646 samples = 24 value = [15, 14, 8, 0] class = 0
n_cores <= 4.5 gini = 0.547 samples = 26 value = [12, 23, 4, 0] class = 1
gini = 0.512 samples = 16 value = [11, 4, 2, 0] class = 0
gini = 0.472 samples = 15 value = [1, 14, 6, 0] class = 1
sc_w <= 4.5 gini = 0.623 samples = 23 value = [5, 11, 17, 1] class = 2
talk_time <= 7.5 gini = 0.483 samples = 33 value = [0, 1, 33, 19] class = 2
gini = 0.32 samples = 10 value = [0, 0, 3, 12] class = 3
clock_speed <= 1.55 gini = 0.539 samples = 30 value = [0, 2, 23, 23] class = 2
gini = 0.245 samples = 11 value = [0, 0, 12, 2] class = 2
ram <= 3205.5 gini = 0.374 samples = 47 value = [0, 1, 19, 62] class = 3
talk_time <= 10.5 gini = 0.122 samples = 25 value = [0, 0, 3, 43] class = 3
gini = 0.368 samples = 14 value = [4, 17, 1, 0] class = 1
gini = 0.0 samples = 15 value = [0, 27, 0, 0] class = 1
gini = 0.305 samples = 12 value = [13, 3, 0, 0] class = 0
gini = 0.196 samples = 20 value = [1, 25, 2, 0] class = 1
gini = 0.514 samples = 14 value = [15, 4, 4, 0] class = 0
gini = 0.408 samples = 10 value = [0, 10, 4, 0] class = 1
gini = 0.594 samples = 15 value = [11, 7, 3, 0] class = 0
gini = 0.204 samples = 11 value = [1, 16, 1, 0] class = 1
gini = 0.569 samples = 10 value = [2, 7, 3, 0] class = 1
gini = 0.541 samples = 13 value = [3, 4, 14, 1] class = 2
gini = 0.498 samples = 10 value = [0, 0, 8, 9] class = 3
gini = 0.44 samples = 23 value = [0, 1, 25, 10] class = 2
gini = 0.415 samples = 10 value = [0, 0, 5, 12] class = 3
gini = 0.533 samples = 20 value = [0, 2, 18, 11] class = 2
gini = 0.487 samples = 19 value = [0, 1, 19, 10] class = 2
gini = 0.0 samples = 28 value = [0, 0, 0, 52] class = 3
gini = 0.227 samples = 14 value = [0, 0, 3, 20] class = 3
gini = 0.0 samples = 11 value = [0, 0, 0, 23] class = 3

In [56]:
```
rf_best.feature_importances_
```

Out[56]:
```
array([0.06726606, 0.00398477, 0.02021607, 0.00500412, 0.01554225,
       0.00533026, 0.02995187, 0.01812967, 0.02912186, 0.01404124,
       0.02321364, 0.05042497, 0.04602417, 0.59827838, 0.02027905,
       0.01955554, 0.02034547, 0.00343017, 0.00451574, 0.00534467])
```

In [57]:
```
imp_df=pd.DataFrame({"Varname":x_train.columns,"Imp":rf_best.feature_importances_})
imp_df.sort_values(by="Imp",ascending=False)
```

Out[57]:

	Varname	Imp
13	ram	0.598278
0	battery_power	0.067266
11	px_height	0.050425
12	px_width	0.046024
6	int_memory	0.029952
8	mobile_wt	0.029122
10	pc	0.023214
16	talk_time	0.020345
14	sc_h	0.020279
2	clock_speed	0.020216
15	sc_w	0.019556
7	m_dep	0.018130
4	fc	0.015542
9	n_cores	0.014041
19	wifi	0.005345
5	four_g	0.005330
3	dual_sim	0.005004
18	touch_screen	0.004516
1	blue	0.003985
17	three_g	0.003430

In [58]:
```
imp_df=pd.DataFrame({"Varname":x_train.columns,"Imp":rf_best.feature_importances_})
imp_df.sort_values(by="Imp",ascending=False)
```

Out[58]:

	Varname	Imp
13	ram	0.598278
0	battery_power	0.067266
11	px_height	0.050425
12	px_width	0.046024
6	int_memory	0.029952
8	mobile_wt	0.029122
10	pc	0.023214
16	talk_time	0.020345
14	sc_h	0.020279
2	clock_speed	0.020216
15	sc_w	0.019556
7	m_dep	0.018130
4	fc	0.015542
9	n_cores	0.014041
19	wifi	0.005345
5	four_g	0.005330
3	dual_sim	0.005004
18	touch_screen	0.004516
1	blue	0.003985
17	three_g	0.003430

In [59]:
```
X=test_df.drop('dual_sim',axis=1)
Y=test_df['dual_sim']
```

In [60]:
```
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[60]:
```
((1400, 20), (600, 20))
```

In [61]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_test,y_test)
```

Out[61]:
```
▾ RandomForestClassifier

RandomForestClassifier()
```

In [63]:
```
rf=RandomForestClassifier()
```

In [64]:
```
param={'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 [66]:
```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rf,param_grid=param,cv=2,scoring="accuracy")
grid_search.fit(x_test,y_test)
```

Out[66]:
```
▸          GridSearchCV
▸ estimator: RandomForestClassifier
      ▸ RandomForestClassifier
```

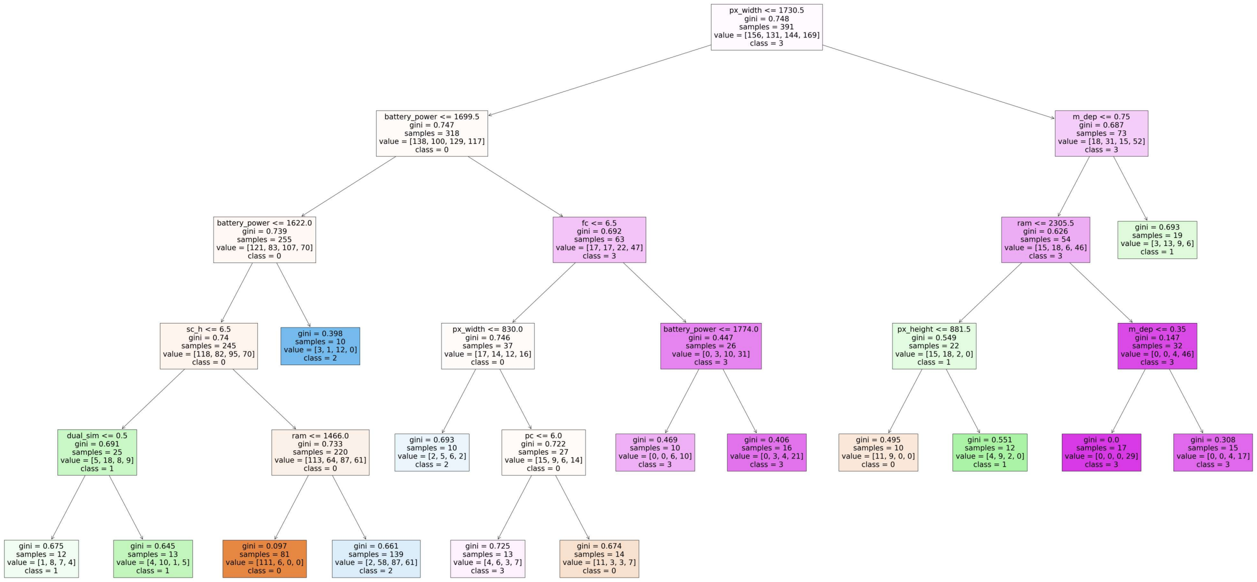
In [67]:
```
grid_search.best_score_
```

Out[67]:
```
0.8166666666666667
```

In [68]:
```
rf_best=grid_search.best_estimator_
print(rf_best)
```

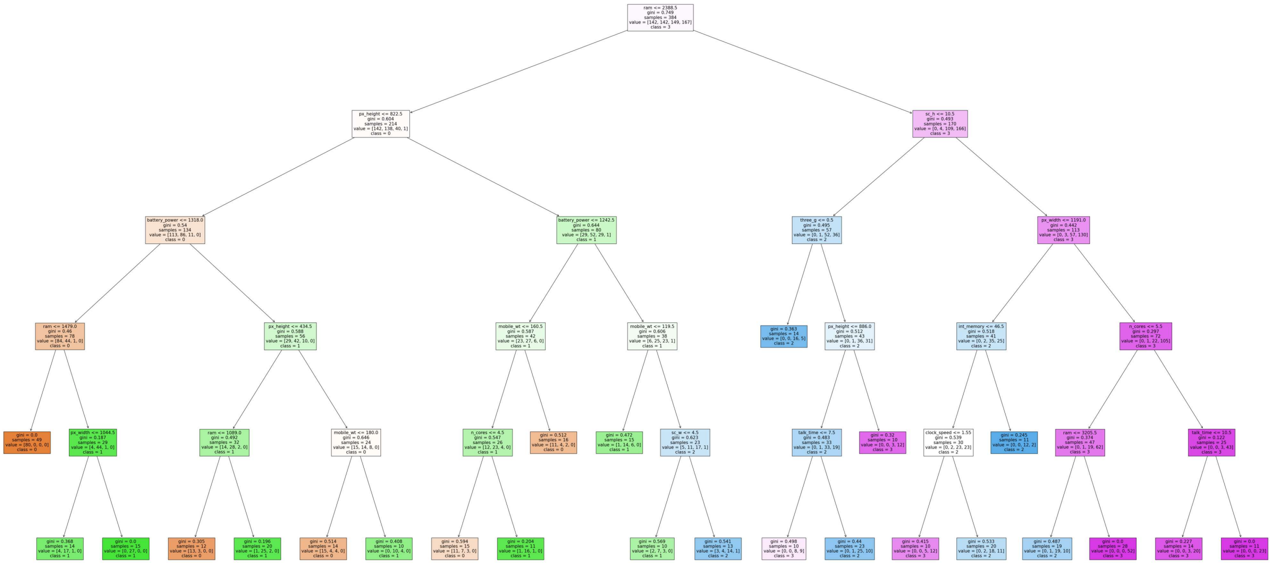
```
RandomForestClassifier(max_depth=5, min_samples_leaf=10, n_estimators=50)
```

In [71]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[5],feature_names=x.columns,class_names=['0','1','2','3'],filled=True);
```



px_width <= 1730.5 gini = 0.748 samples = 391 value = [156, 131, 144, 169] class = 3
battery_power <= 1699.5 gini = 0.747 samples = 318 value = [138, 100, 129, 117] class = 0
m_dep <= 0.75 gini = 0.687 samples = 73 value = [18, 31, 15, 52] class = 3
battery_power <= 1622.0 gini = 0.739 samples = 255 value = [121, 83, 107, 70] class = 0
fc <= 6.5 gini = 0.692 samples = 63 value = [17, 17, 22, 47] class = 3
ram <= 2305.5 gini = 0.626 samples = 54 value = [15, 18, 6, 46] class = 3
gini = 0.693 samples = 19 value = [3, 13, 9, 6] class = 1
sc_h <= 6.5 gini = 0.74 samples = 245 value = [118, 82, 95, 70] class = 0
gini = 0.398 samples = 10 value = [3, 1, 12, 0] class = 2
px_width <= 830.0 gini = 0.746 samples = 37 value = [17, 14, 12, 16] class = 0
battery_power <= 1774.0 gini = 0.447 samples = 26 value = [0, 3, 10, 31] class = 3
px_height <= 881.5 gini = 0.549 samples = 22 value = [15, 18, 2, 0] class = 1
m_dep <= 0.35 gini = 0.147 samples = 32 value = [0, 0, 4, 46] class = 3
dual_sim <= 0.5 gini = 0.691 samples = 25 value = [5, 18, 8, 9] class = 1
ram <= 1466.0 gini = 0.733 samples = 220 value = [113, 64, 87, 61] class = 0
gini = 0.693 samples = 10 value = [2, 5, 6, 2] class = 2
pc <= 6.0 gini = 0.722 samples = 27 value = [15, 9, 6, 14] class = 0
gini = 0.469 samples = 10 value = [0, 0, 6, 10] class = 3
gini = 0.406 samples = 16 value = [0, 3, 4, 21] class = 3
gini = 0.495 samples = 10 value = [11, 9, 0, 0] class = 0
gini = 0.551 samples = 12 value = [4, 9, 2, 0] class = 1
gini = 0.0 samples = 17 value = [0, 0, 0, 29] class = 3
gini = 0.308 samples = 15 value = [0, 0, 4, 17] class = 3
gini = 0.675 samples = 12 value = [1, 8, 7, 4] class = 1
gini = 0.645 samples = 13 value = [4, 10, 1, 5] class = 1
gini = 0.097 samples = 81 value = [111, 6, 0, 0] class = 0
gini = 0.661 samples = 139 value = [2, 58, 87, 61] class = 2
gini = 0.725 samples = 13 value = [4, 6, 3, 7] class = 3
gini = 0.674 samples = 14 value = [11, 3, 3, 7] class = 0

In [70]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[7],feature_names=x.columns,class_names=['0','1','2','3'],filled=True);
```


ram <= 2388.5 gini = 0.749 samples = 384 value = [142, 142, 149, 167] class = 3
px_height <= 822.5 gini = 0.604 samples = 214 value = [142, 138, 40, 1] class = 0
sc_h <= 10.5 gini = 0.493 samples = 170 value = [0, 4, 109, 166] class = 3
battery_power <= 1318.0 gini = 0.54 samples = 134 value = [113, 86, 11, 0] class = 0
battery_power <= 1242.5 gini = 0.644 samples = 80 value = [29, 52, 29, 1] class = 1
three_g <= 0.5 gini = 0.495 samples = 57 value = [0, 1, 52, 36] class = 2
px_width <= 1191.0 gini = 0.442 samples = 113 value = [0, 3, 57, 130] class = 3
ram <= 1479.0 gini = 0.46 samples = 78 value = [84, 44, 1, 0] class = 0
px_height <= 434.5 gini = 0.588 samples = 56 value = [29, 42, 10, 0] class = 1
mobile_wt <= 160.5 gini = 0.587 samples = 42 value = [23, 27, 6, 0] class = 1
mobile_wt <= 119.5 gini = 0.606 samples = 38 value = [6, 25, 23, 1] class = 1
gini = 0.363 samples = 14 value = [0, 0, 16, 5] class = 2
px_height <= 886.0 gini = 0.512 samples = 43 value = [0, 1, 36, 31] class = 2
int_memory <= 46.5 gini = 0.518 samples = 41 value = [0, 2, 35, 25] class = 2
n_cores <= 5.5 gini = 0.297 samples = 72 value = [0, 1, 22, 105] class = 3
gini = 0.0 samples = 49 value = [80, 0, 0, 0] class = 0
px_width <= 1044.5 gini = 0.187 samples = 29 value = [4, 44, 1, 0] class = 1
ram <= 1089.0 gini = 0.492 samples = 32 value = [14, 28, 2, 0] class = 1
mobile_wt <= 180.0 gini = 0.646 samples = 24 value = [15, 14, 8, 0] class = 0
n_cores <= 4.5 gini = 0.547 samples = 26 value = [12, 23, 4, 0] class = 1
gini = 0.512 samples = 16 value = [11, 4, 2, 0] class = 0
gini = 0.472 samples = 15 value = [1, 14, 6, 0] class = 1
sc_w <= 4.5 gini = 0.623 samples = 23 value = [5, 11, 17, 1] class = 2
talk_time <= 7.5 gini = 0.483 samples = 33 value = [0, 1, 33, 19] class = 2
gini = 0.32 samples = 10 value = [0, 0, 3, 12] class = 3
clock_speed <= 1.55 gini = 0.539 samples = 30 value = [0, 2, 23, 23] class = 2
gini = 0.245 samples = 11 value = [0, 0, 12, 2] class = 2
ram <= 3205.5 gini = 0.374 samples = 47 value = [0, 1, 19, 62] class = 3
talk_time <= 10.5 gini = 0.122 samples = 25 value = [0, 0, 3, 43] class = 3
gini = 0.368 samples = 14 value = [4, 17, 1, 0] class = 1
gini = 0.0 samples = 15 value = [0, 27, 0, 0] class = 1
gini = 0.305 samples = 12 value = [13, 3, 0, 0] class = 0
gini = 0.196 samples = 20 value = [1, 25, 2, 0] class = 1
gini = 0.514 samples = 14 value = [15, 4, 4, 0] class = 0
gini = 0.408 samples = 10 value = [0, 10, 4, 0] class = 1
gini = 0.594 samples = 15 value = [11, 7, 3, 0] class = 0
gini = 0.204 samples = 11 value = [1, 16, 1, 0] class = 1
gini = 0.569 samples = 10 value = [2, 7, 3, 0] class = 1
gini = 0.541 samples = 13 value = [3, 4, 14, 1] class = 2
gini = 0.498 samples = 10 value = [0, 0, 8, 9] class = 3
gini = 0.44 samples = 23 value = [0, 1, 25, 10] class = 2
gini = 0.415 samples = 10 value = [0, 0, 5, 12] class = 3
gini = 0.533 samples = 20 value = [0, 2, 18, 11] class = 2
gini = 0.487 samples = 19 value = [0, 1, 19, 10] class = 2
gini = 0.0 samples = 28 value = [0, 0, 0, 52] class = 3
gini = 0.227 samples = 14 value = [0, 0, 3, 20] class = 3
gini = 0.0 samples = 11 value = [0, 0, 0, 23] class = 3

In [74]:
```
rf_best.feature_importances_
```

Out[74]:
```
array([0.07271496, 0.00606811, 0.02160265, 0.00715252, 0.01269713,
       0.00455432, 0.03302288, 0.01138988, 0.03179086, 0.02156046,
       0.02872272, 0.05255705, 0.05551974, 0.56294726, 0.01892744,
       0.01927508, 0.0222952 , 0.00118176, 0.00812567, 0.00789429])
```

In [75]:
```
imp_df=pd.DataFrame({"Varname":x_train.columns,"Imp":rf_best.feature_importances_})
imp_df.sort_values(by="Imp",ascending=False)
```

Out[75]:

	Varname	Imp
13	ram	0.562947
0	battery_power	0.072715
12	px_width	0.055520
11	px_height	0.052557
6	int_memory	0.033023
8	mobile_wt	0.031791
10	pc	0.028723
16	talk_time	0.022295
2	clock_speed	0.021603
9	n_cores	0.021560
15	sc_w	0.019275
14	sc_h	0.018927
4	fc	0.012697
7	m_dep	0.011390
18	touch_screen	0.008126
19	wifi	0.007894
3	dual_sim	0.007153
1	blue	0.006068
5	four_g	0.004554
17	three_g	0.001182

```
In [ ]:
```