In [6]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt,seaborn as sns
```

In [7]:
```
train_df=pd.read_csv(r"C:\Users\MY HOME\Downloads\Mobile_Price_Classification_train.csv")
df
```

Out[7]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height	px_width	ram	sc
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 [8]:
```
test_df=pd.read_csv(r"C:\Users\MY HOME\Downloads\Mobile_Price_Classification_test.csv")
df
```

Out[8]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height	px_width	ram	sc
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 [9]:
```
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 [10]:
```
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 [12]:
```
x=test_df.drop('wifi',axis=1)
y=test_df['wifi']
```

In [13]:
```
x=train_df.drop('wifi',axis=1)
y=train_df['wifi']
```

In [14]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=50)
x_train.shape,x_test.shape
```

Out[14]: ((1500, 20), (500, 20))

In [15]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[15]:
```
▾ RandomForestClassifier

RandomForestClassifier()
```

In [16]:
```
rf=RandomForestClassifier()
```

In [17]:
```
params={'max_depth':[23,40,3,43,7],'min_samples_leaf':[56,78,34,12,5,67],'n_estimators':[23,56,87,12,5,76]}
```

In [18]:
```
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)
grid_search.best_score_
```

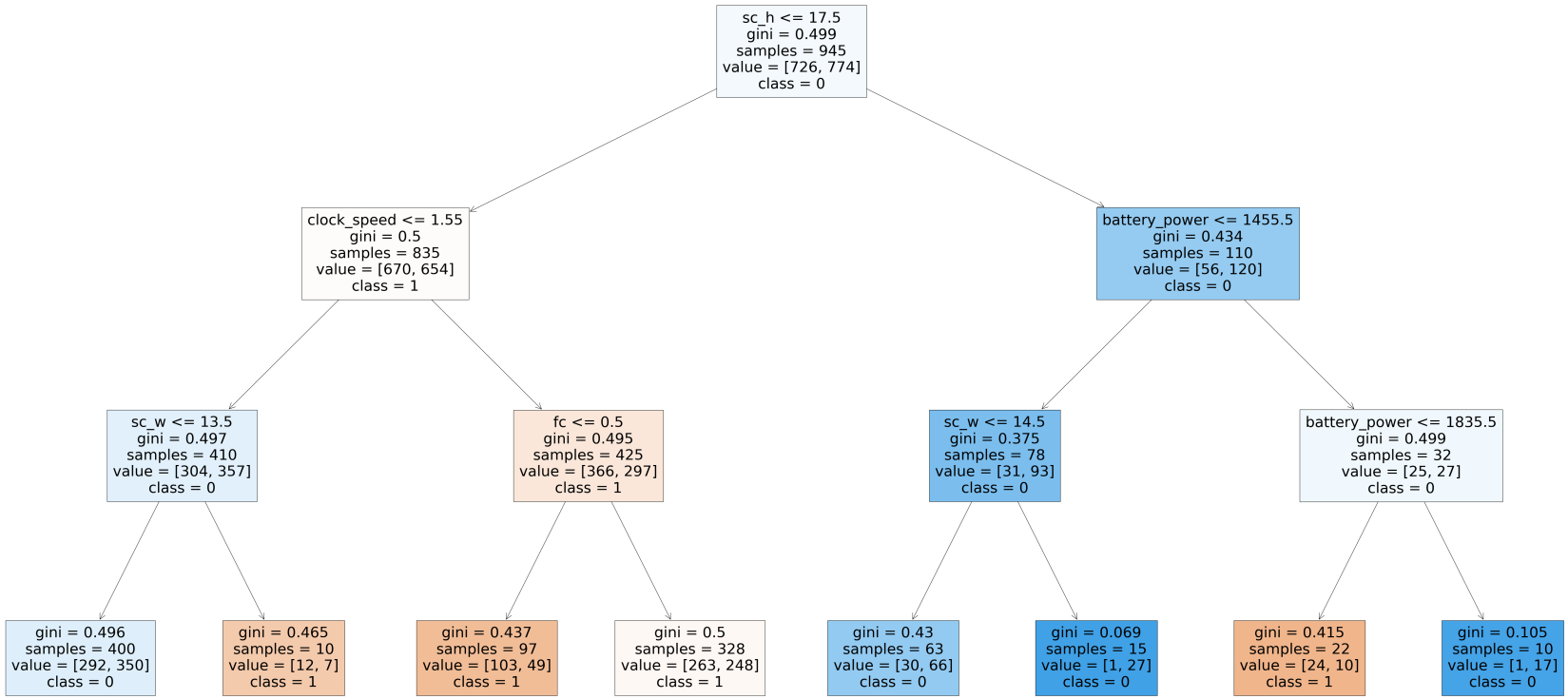
Out[18]: 0.5346666666666666

In [19]:
```
rf_best=grid_search.best_estimator_
print(rf_best)
```

```
RandomForestClassifier(max_depth=3, min_samples_leaf=5, n_estimators=87)
```

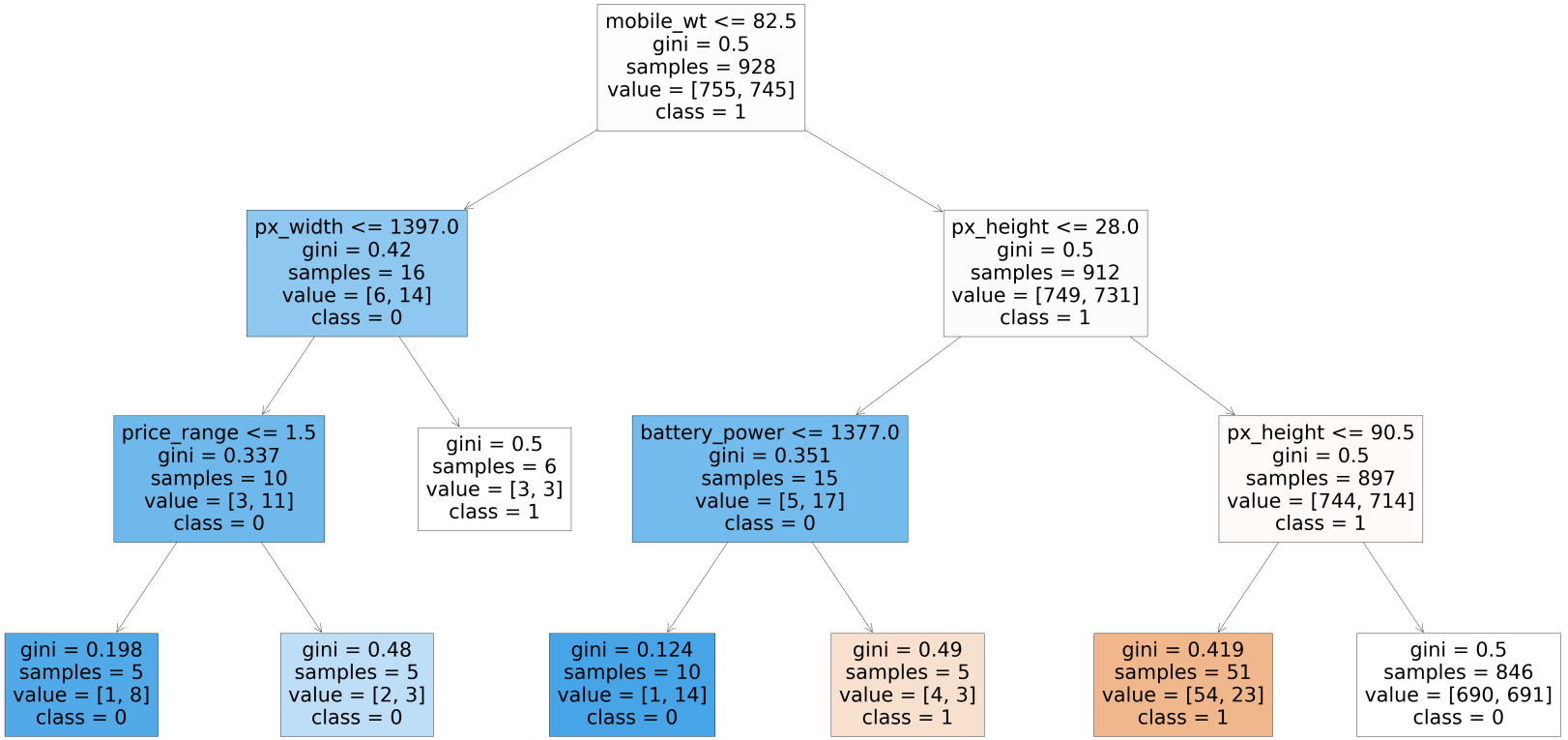
In [20]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[4],feature_names=x.columns,class_names=['1','0'],filled=True)
```

Out[20]:
```
[Text(0.5, 0.875, 'sc_h <= 17.5\ngini = 0.499\nsamples = 945\nvalue = [726, 774]\nclass = 0'),
 Text(0.25, 0.625, 'clock_speed <= 1.55\ngini = 0.5\nsamples = 835\nvalue = [670, 654]\nclass = 1'),
 Text(0.125, 0.375, 'sc_w <= 13.5\ngini = 0.497\nsamples = 410\nvalue = [304, 357]\nclass = 0'),
 Text(0.0625, 0.125, 'gini = 0.496\nsamples = 400\nvalue = [292, 350]\nclass = 0'),
 Text(0.1875, 0.125, 'gini = 0.465\nsamples = 10\nvalue = [12, 7]\nclass = 1'),
 Text(0.375, 0.375, 'fc <= 0.5\ngini = 0.495\nsamples = 425\nvalue = [366, 297]\nclass = 1'),
 Text(0.3125, 0.125, 'gini = 0.437\nsamples = 97\nvalue = [103, 49]\nclass = 1'),
 Text(0.4375, 0.125, 'gini = 0.5\nsamples = 328\nvalue = [263, 248]\nclass = 1'),
 Text(0.75, 0.625, 'battery_power <= 1455.5\ngini = 0.434\nsamples = 110\nvalue = [56, 120]\nclass = 0'),
 Text(0.625, 0.375, 'sc_w <= 14.5\ngini = 0.375\nsamples = 78\nvalue = [31, 93]\nclass = 0'),
 Text(0.5625, 0.125, 'gini = 0.43\nsamples = 63\nvalue = [30, 66]\nclass = 0'),
 Text(0.6875, 0.125, 'gini = 0.069\nsamples = 15\nvalue = [1, 27]\nclass = 0'),
 Text(0.875, 0.375, 'battery_power <= 1835.5\ngini = 0.499\nsamples = 32\nvalue = [25, 27]\nclass = 0'),
 Text(0.8125, 0.125, 'gini = 0.415\nsamples = 22\nvalue = [24, 10]\nclass = 1'),
 Text(0.9375, 0.125, 'gini = 0.105\nsamples = 10\nvalue = [1, 17]\nclass = 0')]
```


sc_h <= 17.5
gini = 0.499
samples = 945
value = [726, 774]
class = 0
clock_speed <= 1.55
gini = 0.5
samples = 835
value = [670, 654]
class = 1
battery_power <= 1455.5
gini = 0.434
samples = 110
value = [56, 120]
class = 0
sc_w <= 13.5
gini = 0.497
samples = 410
value = [304, 357]
class = 0
fc <= 0.5
gini = 0.495
samples = 425
value = [366, 297]
class = 1
sc_w <= 14.5
gini = 0.375
samples = 78
value = [31, 93]
class = 0
battery_power <= 1835.5
gini = 0.499
samples = 32
value = [25, 27]
class = 0
gini = 0.496
samples = 400
value = [292, 350]
class = 0
gini = 0.465
samples = 10
value = [12, 7]
class = 1
gini = 0.437
samples = 97
value = [103, 49]
class = 1
gini = 0.5
samples = 328
value = [263, 248]
class = 1
gini = 0.43
samples = 63
value = [30, 66]
class = 0
gini = 0.069
samples = 15
value = [1, 27]
class = 0
gini = 0.415
samples = 22
value = [24, 10]
class = 1
gini = 0.105
samples = 10
value = [1, 17]
class = 0

In [21]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[3],feature_names=x.columns,class_names=['1','0'],filled=True)
```

Out[21]:
```
[Text(0.4583333333333333, 0.875, 'mobile_wt <= 82.5\ngini = 0.5\nsamples = 928\nvalue = [755, 745]\nclass =
1'),
 Text(0.25, 0.625, 'px_width <= 1397.0\ngini = 0.42\nsamples = 16\nvalue = [6, 14]\nclass = 0'),
 Text(0.16666666666666666, 0.375, 'price_range <= 1.5\ngini = 0.337\nsamples = 10\nvalue = [3, 11]\nclass =
0'),
 Text(0.08333333333333333, 0.125, 'gini = 0.198\nsamples = 5\nvalue = [1, 8]\nclass = 0'),
 Text(0.25, 0.125, 'gini = 0.48\nsamples = 5\nvalue = [2, 3]\nclass = 0'),
 Text(0.3333333333333333, 0.375, 'gini = 0.5\nsamples = 6\nvalue = [3, 3]\nclass = 1'),
 Text(0.6666666666666666, 0.625, 'px_height <= 28.0\ngini = 0.5\nsamples = 912\nvalue = [749, 731]\nclass =
1'),
 Text(0.5, 0.375, 'battery_power <= 1377.0\ngini = 0.351\nsamples = 15\nvalue = [5, 17]\nclass = 0'),
 Text(0.4166666666666667, 0.125, 'gini = 0.124\nsamples = 10\nvalue = [1, 14]\nclass = 0'),
 Text(0.5833333333333334, 0.125, 'gini = 0.49\nsamples = 5\nvalue = [4, 3]\nclass = 1'),
 Text(0.8333333333333334, 0.375, 'px_height <= 90.5\ngini = 0.5\nsamples = 897\nvalue = [744, 714]\nclass =
1'),
 Text(0.75, 0.125, 'gini = 0.419\nsamples = 51\nvalue = [54, 23]\nclass = 1'),
 Text(0.9166666666666666, 0.125, 'gini = 0.5\nsamples = 846\nvalue = [690, 691]\nclass = 0')]
```



In [22]:
```
rf_best.feature_importances_
```

Out[22]:
```
array([0.09272687, 0.0089464 , 0.05029426, 0.01066547, 0.1012335 ,
       0.00688462, 0.08349102, 0.05314514, 0.06224154, 0.02338459,
       0.04472112, 0.12361403, 0.09224796, 0.08734734, 0.05204175,
       0.04008419, 0.04397685, 0.00230973, 0.00767951, 0.01296412])
```

In [23]:
```
imp_df=pd.DataFrame({"varname":x_train.columns,"Imp":rf_best.feature_importances_})
imp_df.sort_values(by="Imp",ascending=False)
```

Out[23]:

	varname	Imp
11	px_height	0.123614
4	fc	0.101234
0	battery_power	0.092727
12	px_width	0.092248
13	ram	0.087347
6	int_memory	0.083491
8	mobile_wt	0.062242
7	m_dep	0.053145
14	sc_h	0.052042
2	clock_speed	0.050294
10	pc	0.044721
16	talk_time	0.043977
15	sc_w	0.040084
9	n_cores	0.023385
19	price_range	0.012964
3	dual_sim	0.010665
1	blue	0.008946
18	touch_screen	0.007680
5	four_g	0.006885
17	three_g	0.002310

In [ ]: