In [1]:

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

In [2]:

```
train_df=pd.read_csv(r"C:\Users\DELL\Downloads\Mobile_Price_Classification_train.csv")
train_df
```

Out[2]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208
...	...	...	...	...	...	...	...	...	...	...	...	...
1995	794	1	0.5	1	0	1	2	0.8	106	6	...	1222
1996	1965	1	2.6	1	0	0	39	0.2	187	4	...	915
1997	1911	0	0.9	1	1	1	36	0.7	108	8	...	868
1998	1512	0	0.9	0	4	1	46	0.1	145	5	...	336
1999	510	1	2.0	1	5	1	45	0.9	168	6	...	483

2000 rows × 21 columns

In [3]:

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

Out[3]:

	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 [4]:

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

In [5]:

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

Out[5]:

```
four_g
1    1043
0     957
Name: count, dtype: int64
```

In [6]:

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

In [7]:

```
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[7]:

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

In [8]:

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

Out[8]:

RandomForestClassifier

RandomForestClassifier()

In [9]:

```
rf=RandomForestClassifier()
```

In [10]:

```
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 [13]:

```
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[13]:

GridSearchCV

estimator: RandomForestClassifier

RandomForestClassifier

In [14]:

```
grid_search.best_score_
```

Out[14]:

```
0.8335714285714286
```

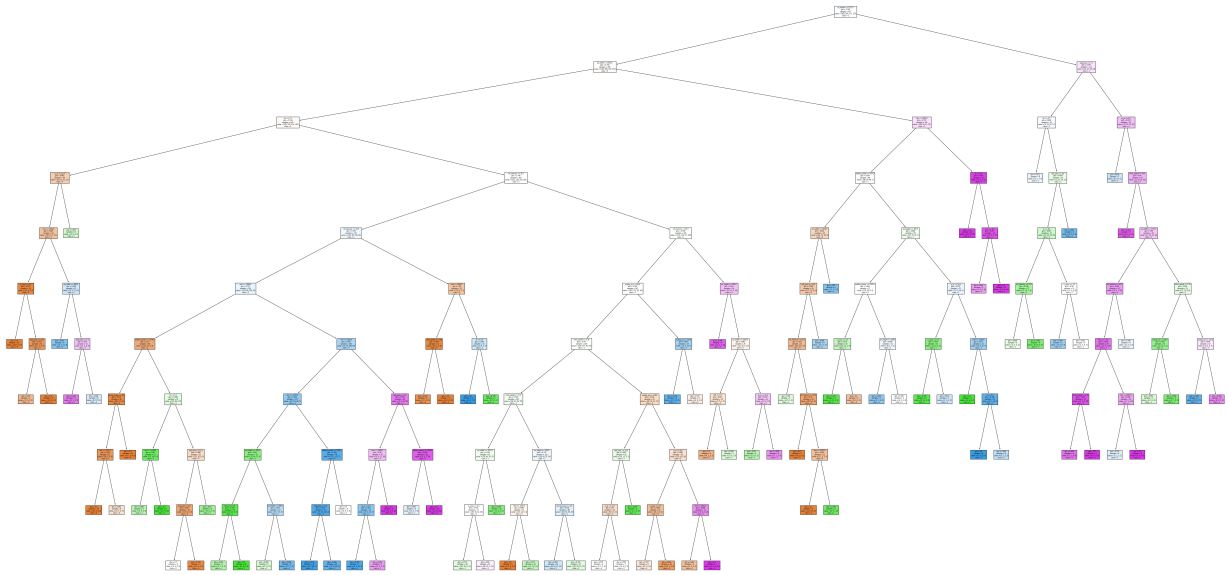
In [15]:

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

```
RandomForestClassifier(max_depth=10, min_samples_leaf=5, n_estimators=200)
```

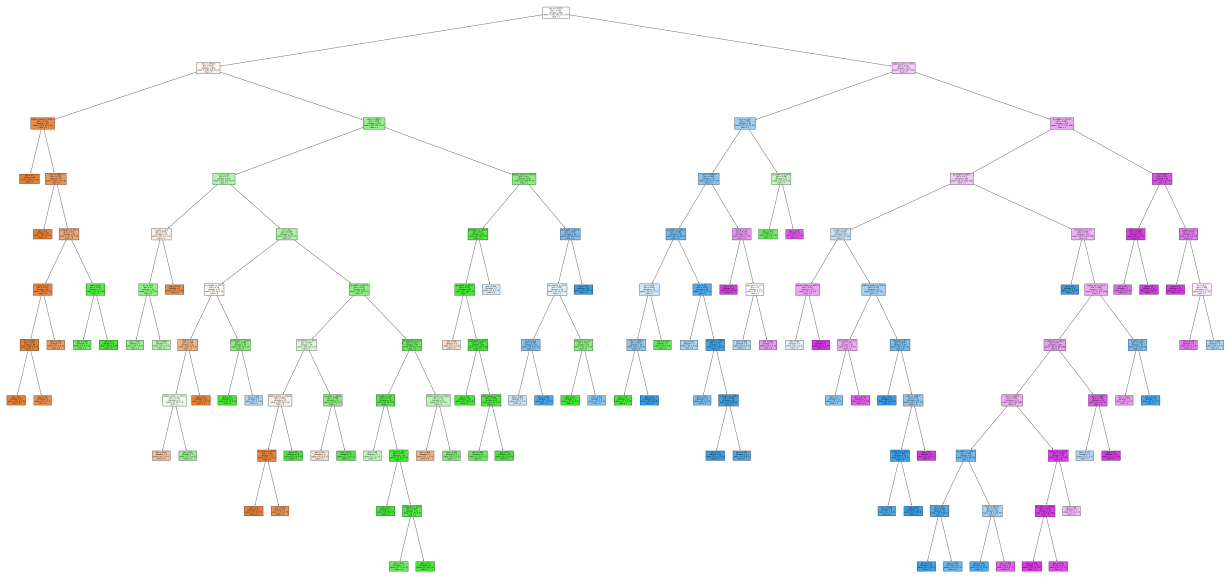
In [16]:

```
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 [17]:

```
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);
```



In [18]:

```
rf_best.feature_importances_
```

Out[18]:

```
array([0.07318864, 0.00391308, 0.01948877, 0.00409401, 0.01604893,
       0.00446461, 0.02866551, 0.01690407, 0.02889539, 0.01494503,
       0.01984308, 0.04843908, 0.04725811, 0.60454   , 0.01700206,
       0.01855947, 0.02154822, 0.00279326, 0.0056337 , 0.00377497])
```

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	ram	0.604540
0	battery_power	0.073189
11	px_height	0.048439
12	px_width	0.047258
8	mobile_wt	0.028895
6	int_memory	0.028666
16	talk_time	0.021548
10	pc	0.019843
2	clock_speed	0.019489
15	sc_w	0.018559
14	sc_h	0.017002
7	m_dep	0.016904
4	fc	0.016049
9	n_cores	0.014945
18	touch_screen	0.005634
5	four_g	0.004465
3	dual_sim	0.004094
1	blue	0.003913
19	wifi	0.003775
17	three_g	0.002793

In [20]:

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

Out[20]:

	Varname	Imp
13	ram	0.604540
0	battery_power	0.073189
11	px_height	0.048439
12	px_width	0.047258
8	mobile_wt	0.028895
6	int_memory	0.028666
16	talk_time	0.021548
10	pc	0.019843
2	clock_speed	0.019489
15	sc_w	0.018559
14	sc_h	0.017002
7	m_dep	0.016904
4	fc	0.016049
9	n_cores	0.014945
18	touch_screen	0.005634
5	four_g	0.004465
3	dual_sim	0.004094
1	blue	0.003913
19	wifi	0.003775
17	three_g	0.002793

In [21]:

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

Out[21]:

```
four_g
0    513
1    487
Name: count, dtype: int64
```

In [22]:

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

In [23]:

```
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[23]:

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

In [25]:

```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_test,y_test)
```

Out[25]:

RandomForestClassifier

RandomForestClassifier()

In [26]:

```
rf=RandomForestClassifier()
```

In [27]:

```
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 [35]:

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

Out[35]:

GridSearchCV

estimator: RandomForestClassifier

RandomForestClassifier

In [29]:

```
grid_search.best_score_
```

Out[29]:

```
0.55
```

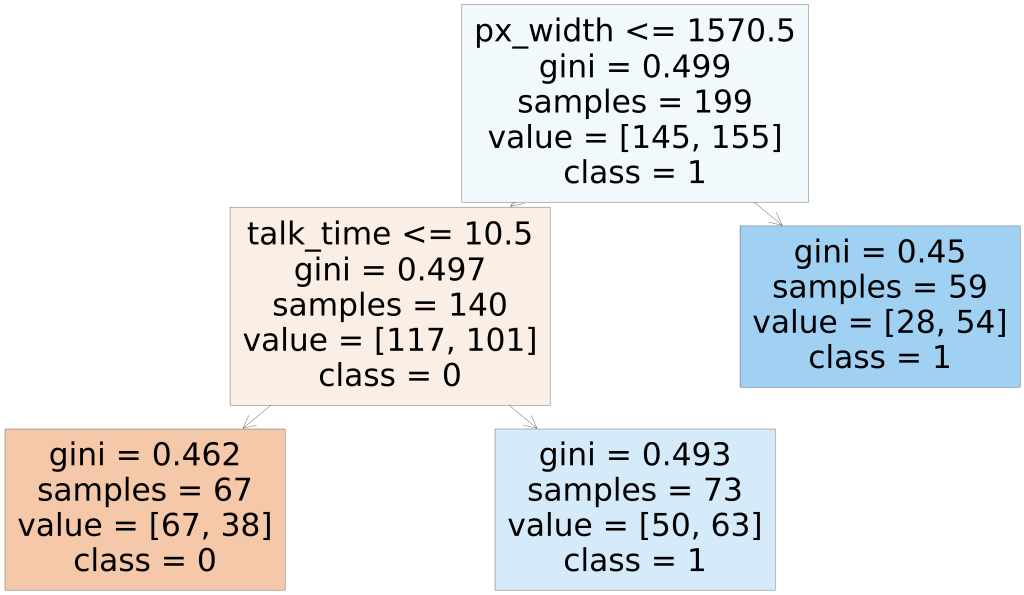
In [30]:

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

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

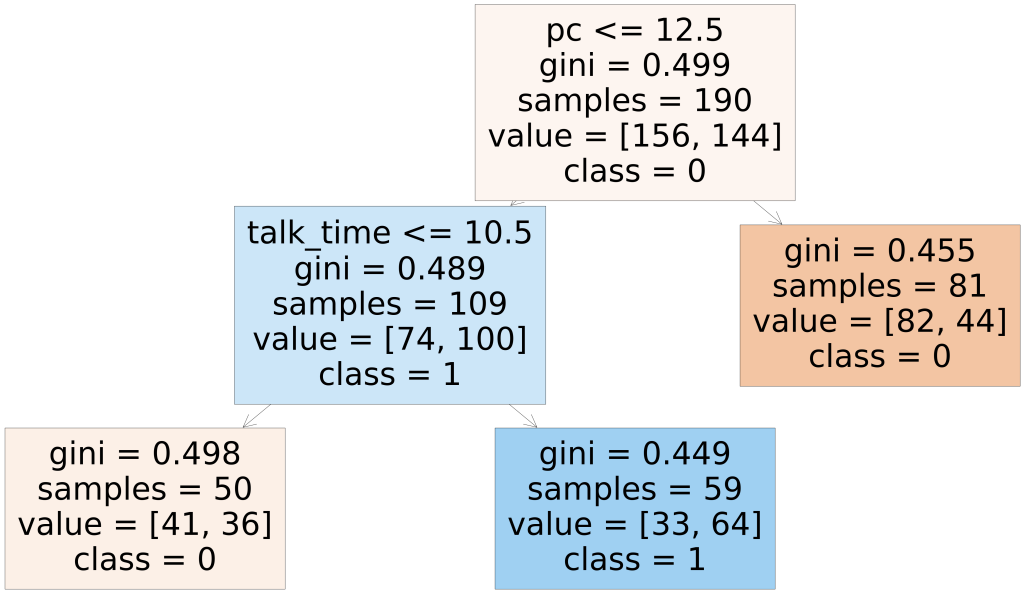
In [31]:

```
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 [32]:

```
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);
```



In [33]:

```
rf_best.feature_importances_
```

Out[33]:

```
array([0.15890982, 0.02075883, 0.07796203, 0.03166757, 0.        ,
       0.04      , 0.        , 0.1390264 , 0.04041441, 0.        ,
       0.02720677, 0.04381743, 0.03960463, 0.12081511, 0.06761667,
       0.08073397, 0.04434856, 0.03110227, 0.        , 0.03601551])
```

In [34]:

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

Out[34]:

	Varname	Imp
0	id	0.158910
7	int_memory	0.139026
13	px_width	0.120815
15	sc_h	0.080734
2	blue	0.077962
14	ram	0.067617
16	sc_w	0.044349
11	pc	0.043817
8	m_dep	0.040414
5	fc	0.040000
12	px_height	0.039605
19	touch_screen	0.036016
3	clock_speed	0.031668
17	talk_time	0.031102
10	n_cores	0.027207
1	battery_power	0.020759
9	mobile_wt	0.000000
6	four_g	0.000000
4	dual_sim	0.000000
18	three_g	0.000000

In [ ]:

```

```

In [ ]:

```

```