# In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

### In [2]:

traindf=pd.read\_csv(r"C:\Users\lenovo\Downloads\Mobile\_Price\_Classification\_train.csv")
traindf

# 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]:

 $testdf=pd.read\_csv(r"C:\Users\lenovo\Downloads\Mobile\_Price\_Classification\_test.csv") \\ testdf$ 

# Out[3]:

ower	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	 рс	px_height	px_width	ram	sc_h
1043	1	1.8	1	14	0	5	0.1	193	 16	226	1412	3476	12
841	1	0.5	1	4	1	61	0.8	191	 12	746	857	3895	6
1807	1	2.8	0	1	0	27	0.9	186	 4	1270	1366	2396	17
1546	0	0.5	1	18	1	25	0.5	96	 20	295	1752	3893	10
1434	0	1.4	0	11	1	49	0.5	108	 18	749	810	1773	15
						•••			 				
1700	1	1.9	0	0	1	54	0.5	170	 17	644	913	2121	14
609	0	1.8	1	0	0	13	0.9	186	 2	1152	1632	1933	8
1185	0	1.4	0	1	1	8	0.5	80	 12	477	825	1223	5
1533	1	0.5	1	0	0	50	0.4	171	 12	38	832	2509	15
1270	1	0.5	0	4	1	35	0.1	140	 19	457	608	2828	9
mns													

#### In [4]:

```
traindf.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
                    Non-Null Count Dtype
#
    Column
                    -----
0
     battery_power
                    2000 non-null
                                    int64
1
     blue
                    2000 non-null
                                    int64
     clock_speed
                    2000 non-null
                                    float64
2
3
     dual_sim
                    2000 non-null
                                    int64
4
     fc
                    2000 non-null
                                    int64
5
                    2000 non-null
     four_g
                                    int64
6
     int_memory
                    2000 non-null
                                    int64
7
                    2000 non-null
     m dep
                                    float64
8
     mobile wt
                    2000 non-null
                                    int64
                    2000 non-null
9
     n_cores
                                    int64
                    2000 non-null
                                    int64
10
    рс
11
    px height
                    2000 non-null
                                    int64
    px width
                    2000 non-null
                                    int64
12
                    2000 non-null
13
     ram
                                    int64
In [5]:
testdf.info()
<class 'pandas.core.frame.DataFrame'>
```

```
<class 'pandas.core.frame.DataFrame'
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):</pre>
```

```
Non-Null Count
#
    Column
                                   Dtype
0
     id
                    1000 non-null
                                    int64
     battery_power 1000 non-null
                                    int64
1
2
     blue
                    1000 non-null
                                    int64
3
     clock_speed
                    1000 non-null
                                    float64
                    1000 non-null
4
     dual_sim
                                    int64
5
     fc
                    1000 non-null
                                    int64
                    1000 non-null
                                    int64
6
     four_g
7
     int_memory
                    1000 non-null
                                    int64
8
     m_dep
                    1000 non-null
                                    float64
                    1000 non-null
9
     mobile_wt
                                    int64
10
                    1000 non-null
                                    int64
    n_cores
11
     рс
                    1000 non-null
                                    int64
12
     px_height
                    1000 non-null
                                     int64
                    1000 non-null
13
     px_width
                                    int64
                    1000 non-null
14
                                    int64
    ram
                    1000 non-null
15
    sc h
                                    int64
16
    SC_W
                    1000 non-null
                                    int64
                    1000 non-null
17 talk_time
                                    int64
18
    three_g
                    1000 non-null
                                    int64
                    1000 non-null
                                    int64
19
    touch_screen
20 wifi
                    1000 non-null
                                    int64
dtypes: float64(2), int64(19)
memory usage: 164.2 KB
```

#### In [6]:

```
traindf.shape,testdf.shape
```

# Out[6]:

```
((2000, 21), (1000, 21))
```

```
In [7]:
```

```
traindf=traindf.head(1000)
traindf
```

### Out[7]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	 px_height	р
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	
995	1456	0	1.6	1	5	0	49	0.2	193	3	 1285	
996	774	0	0.5	1	2	1	10	0.5	188	2	 1480	
997	1068	0	0.5	1	0	1	19	0.9	197	8	 322	
998	1373	1	1.9	1	1	1	29	0.9	141	6	 1220	
999	1777	1	3.0	0	3	0	20	0.6	188	6	 511	

1000 rows × 21 columns

**→** 

### In [8]:

```
traindf.shape,testdf.shape
```

### Out[8]:

```
((1000, 21), (1000, 21))
```

# In [9]:

```
X=testdf
y=traindf['price_range']
X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.7,random_state=42)
```

#### In [10]:

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

# Out[10]:

```
r RandomForestClassifier
RandomForestClassifier()
```

#### In [11]:

```
rf=RandomForestClassifier()
```

# In [12]:

```
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")
```

```
In [14]:
```

```
grid_search.fit(X_train,y_train)
```

### Out[14]:

```
► GridSearchCV

► estimator: RandomForestClassifier

► RandomForestClassifier
```

## In [15]:

```
grid_search.best_score_
```

#### Out[15]:

0.28714285714285714

### In [16]:

```
rf_best=grid_search.best_estimator_
rf_best
```

### Out[16]:

```
RandomForestClassifier
RandomForestClassifier(max_depth=20, min_samples_leaf=50, n_estimators=10)
```

#### In [17]:

```
traindf['price_range'].value_counts()
```

### Out[17]:

price\_range

- 3 276
- 2 248
- 0 242
- 1 234

Name: count, dtype: int64

# In [18]:

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[4],feature_names=X.columns,class_names=['3','2','1','0'],filled=True);
```

```
clock speed \le 2.35
                                                                                             gini = 0.749
                                                                                    samples = 431
value = [170, 167, 188, 175]
class = 1
                                                                        px_width <= 1457.0
gini = 0.748
samples = 333
                                                                                                               gini = 0.728
                                                                                                         samples = 98
value = [28, 38, 47, 64]
                                                                   value = [142, 129, 141, 111]
                                                                                                                 class = 0
                                                                              class = 3
                                  battery_power <= 1643.0
                                                                                                           mobile_wt <= 140.5
                                       gini = 0.746
samples = 224
                                                                                                               gini = 0.727
                                                                                                         samples = 109
value = [65, 29, 49, 36]
                                  value = [77, 100, 92, 75]
                                           class = 2
                                                                                                                 class = 3
                       sc_h <= 13.5
                                                           gini = 0.728
                                                                                              gini = 0.734
                                                                                                                                  gini = 0.68
                        g\bar{i}ni = 0.739
                                                          samples = 50
                                                                                             samples = 58
                                                                                                                                samples = 51
                 samples = 174
value = [63, 86, 75, 48]
                                                    value = [14, 14, 17, 27]
                                                                                        value = [26, 14, 30, 26]
                                                                                                                          value = [39, 15, 19, 10]
                                                            class = 0
                                                                                               class = 1
                                                                                                                                  class = 3
                         class = 2
                                         gini = 0.749
       gini = 0.72
      samples = 98
                                        samples = 76
value = [32, 55, 41, 20]
                                   value = [31, 31, 34, 28]
```

#### In [19]:

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

```
px_width <= 984.0
gini = 0.747
                                                                                                           samples = 445
value = [179, 161, 154, 206]
class = 0
                              pc <= 7.5
gini = 0.711
samples = 147
value = [55, 68, 23, 86]
class = 0
                                                                                                                                                                                             clock_speed <= 1.85
gini = 0.746
samples = 298
value = [124, 93, 131, 120]
                                                                                                                                                                                                                 class = 1
                                                                                                                               ram <= 1802.0
gini = 0.742
samples = 180
value = [86, 59, 77, 57]
class = 3
                                                                                                                                                                                                                                                                  fc \le 3.5
gini = 0.734
samples = 118
value = [38, 34, 54, 63]
gini = 0.679
samples = 70
value = [29, 44, 4, 33]
class = 2
                                                                gini = 0.703
samples = 77
value = [26, 24, 19, 53]
class = 0
                                                                                                                                                                                                                                                                                  class = 0
                                                                                                                                                                 touch_screen <= 0.5
gini = 0.733
samples = 108
value = [50, 23, 49, 43]
class = 3
                                                                                                                                                                                                                                                                                                   gini = 0.712
samples = 64
value = [21, 15, 25, 42]
class = 0
                                                                                               gini = 0.725
samples = 72
value = [36, 36, 28, 14]
class = 3
                                                                                                                                                                                                                                             gini = 0.739
                                                                                                                                                                                                                                 samples = 54
value = [17, 19, 29, 21]
class = 1
                                                                                                                                            aini = 0.705
                                                                                                                                                                                                             aini = 0.738
                                                                                                                                  samples = 54
value = [33, 8, 23, 20]
class = 3
                                                                                                                                                                                                  samples = 54
value = [17, 15, 26, 23]
                                                                                                                                                                                                                 class = 1
```

### In [20]:

```
rf_best.feature_importances_
```

#### Out[20]:

```
array([0.0420621 , 0.04145842, 0. , 0.13109558, 0. , 0.05955225, 0.03281126, 0.04569771, 0. , 0.05751693, 0.07205376, 0.05736547, 0.20151945, 0.12142944, 0.06031637, 0.01960383, 0.01099444, 0. , 0. , 0.03328097, 0.01324201])
```

## In [21]:

```
imp_df=pd.DataFrame({"Varname":X_train.columns,"Imp":rf_best.feature_importances_})
```

# In [22]:

```
imp_df.sort_values(by="Imp",ascending=False)
```

# Out[22]:

	Varname	lmp
12	px_height	0.201519
3	clock_speed	0.131096
13	px_width	0.121429
10	n_cores	0.072054
14	ram	0.060316
5	fc	0.059552
9	mobile_wt	0.057517
11	рс	0.057365
7	int_memory	0.045698
0	id	0.042062
1	battery_power	0.041458
19	touch_screen	0.033281
6	four_g	0.032811
15	sc_h	0.019604
20	wifi	0.013242
16	sc_w	0.010994
8	m_dep	0.000000
4	dual_sim	0.000000
2	blue	0.000000
17	talk_time	0.000000
18	three_g	0.000000

# In [ ]: