

```
In [2]: 1 import numpy as np
        2 import pandas as pd
        3 import matplotlib.pyplot as plt
        4 import seaborn as sns
```

```
In [72]: 1 test_df=pd.read_csv(r"C:\Users\Welcome\Downloads\Mobile_Price_Classification_test.csv")
        2 test_df
```

Out[72]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height	px
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 [74]: 1 train=pd.read_csv(r"C:\Users\Welcome\Downloads\Mobile_Price_Classification_train.csv")
        2 train_df
```

Out[74]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height	px
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 [43]: 1 train_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 [44]: 1 test_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: 328.2 KB
```

```
In [45]: 1 x=train_df.drop('wifi',axis=1)
         2 y=train_df['wifi']
```

```
In [46]: 1 x=test_df.drop('wifi',axis=1)
        2 y=test_df['wifi']
```

```
In [53]: 1 train_df['dual_sim'].value_counts()
```

```
Out[53]: dual_sim
1      517
0      483
Name: count, dtype: int64
```

```
In [54]: 1 test_df['blue'].value_counts()
```

```
Out[54]: blue
0     1010
1       990
Name: count, dtype: int64
```

```
In [55]: 1 T={"three_g":{'Yes':1,'No':0}}
        2 train_df=train_df.replace(T)
        3 print(train_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 [56]: 1 T={"three_g":{"Yes":1,'No':0}}
2 test_df=test_df.replace(T)
3 print(test_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 [57]: 1 x=train_df.drop('wifi',axis=1)
2 y=train_df['wifi']
```

```
In [58]: 1 x=test_df.drop('wifi',axis=1)
2 y=test_df['wifi']
```

```
In [59]: 1 from sklearn.model_selection import train_test_split
2 x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.7,random_state=42)
3 x_train.shape,x_test.shape
```

Out[59]: ((1400, 20), (600, 20))

```
In [60]: 1 from sklearn.ensemble import RandomForestClassifier
2 rfc=RandomForestClassifier()
3 rfc.fit(x_train,y_train)
```

Out[60]:
 ▾ RandomForestClassifier
 RandomForestClassifier()

```
In [61]: 1 rf=RandomForestClassifier()
```

```
In [62]: 1 params={'max_depth':[2,3,5,10,20],
2           'min_samples_leaf':[5,10,20,50,100,200],
3           'n_estimators':[10,25,30,50,100,200]}
```

```
In [63]: 1 from sklearn.model_selection import GridSearchCV
2 grid_search=GridSearchCV(estimator=rf,param_grid=params,cv=2,scoring='accuracy')
3 grid_search.fit(x_train,y_train)
```

```
Out[63]: GridSearchCV
  estimator: RandomForestClassifier
    RandomForestClassifier
```

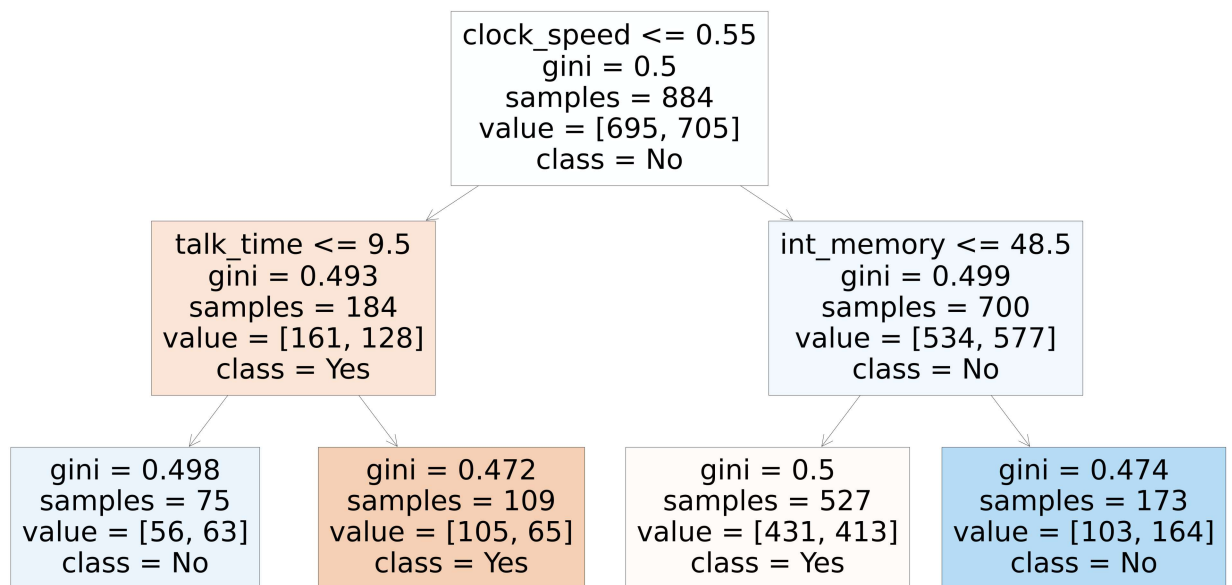
```
In [64]: 1 grid_search.best_score_
```

```
Out[64]: 0.5192857142857144
```

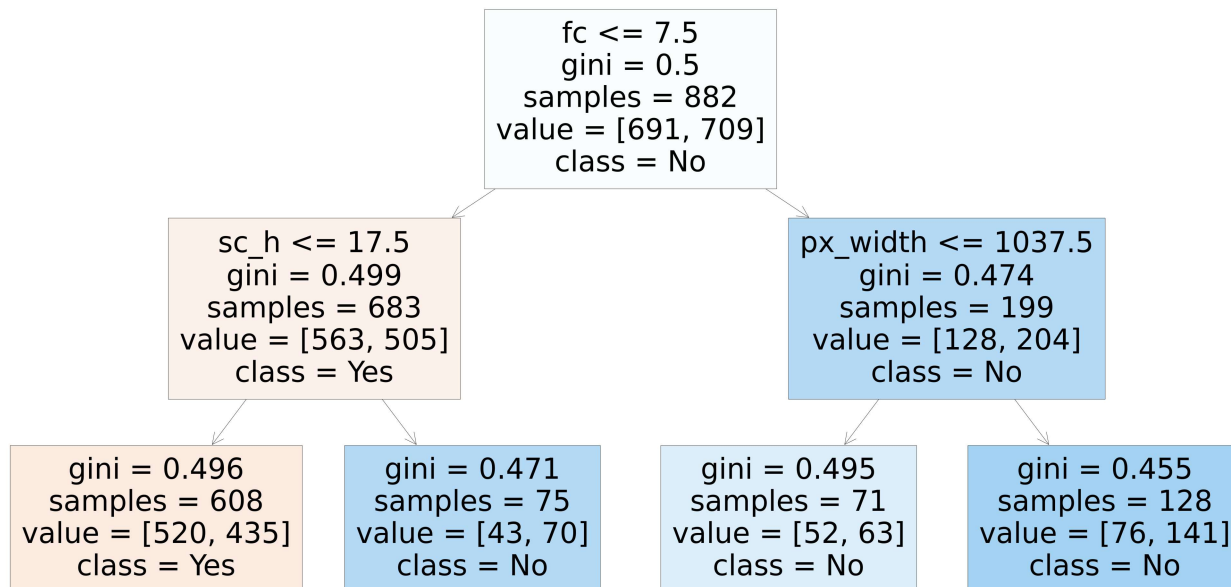
```
In [65]: 1 rf_best=grid_search.best_estimator_
2 print(rf_best)
```

```
RandomForestClassifier(max_depth=2, min_samples_leaf=50, n_estimators=30)
```

```
In [66]: 1 from sklearn.tree import plot_tree
2 plt.figure(figsize=(80,40))
3 plot_tree(rf_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled=True)
```



```
In [67]: 1 from sklearn.tree import plot_tree
2 plt.figure(figsize=(80,40))
3 plot_tree(rf_best.estimators_[7],feature_names=x.columns,class_names=['Yes','No'],filled=True)
```



```
In [68]: 1 rf_best.feature_importances_
```

```
Out[68]: array([0.08753906, 0.00142447, 0.05805406, 0.          , 0.13543381,
0.          , 0.06383638, 0.03497927, 0.04090854, 0.01435646,
0.10806625, 0.231702  , 0.05454053, 0.10889435, 0.01055837,
0.00573444, 0.01341599, 0.          , 0.          , 0.03055602])
```

```
In [69]: 1 imp_df=pd.DataFrame({'Varname':x_train.columns,"Imp":rf_best.feature_importances_})
          2 imp_df.sort_values(by="Imp",ascending=False)
```

Out[69]:

	Varname	Imp
11	px_height	0.231702
4	fc	0.135434
13	ram	0.108894
10	pc	0.108066
0	battery_power	0.087539
6	int_memory	0.063836
2	clock_speed	0.058054
12	px_width	0.054541
8	mobile_wt	0.040909
7	m_dep	0.034979
19	price_range	0.030556
9	n_cores	0.014356
16	talk_time	0.013416
14	sc_h	0.010558
15	sc_w	0.005734
1	blue	0.001424
5	four_g	0.000000
3	dual_sim	0.000000
17	three_g	0.000000
18	touch_screen	0.000000

In []:

1