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

In [2]:
```
traind=pd.read_csv(r"C:\Users\teppa\Downloads\Mobile_Price_Classification_train.csv")
traind
```

Out[2]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603	11
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769	16
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
1995	794	1	0.5	1	0	1	2	0.8	106	6	...	1222	1890	668	13
1996	1965	1	2.6	1	0	0	39	0.2	187	4	...	915	1965	2032	11
1997	1911	0	0.9	1	1	1	36	0.7	108	8	...	868	1632	3057	9
1998	1512	0	0.9	0	4	1	46	0.1	145	5	...	336	670	869	18
1999	510	1	2.0	1	5	1	45	0.9	168	6	...	483	754	3919	19

2000 rows × 21 columns

In [3]:
```
testd=pd.read_csv(r"C:\Users\teppa\Downloads\Mobile_Price_Classification_test.csv")
testd
```

Out[3]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height	px_width	ram	sc_h
0	1	1043	1	1.8	1	14	0	5	0.1	193	...	16	226	1412	3476	12
1	2	841	1	0.5	1	4	1	61	0.8	191	...	12	746	857	3895	6
2	3	1807	1	2.8	0	1	0	27	0.9	186	...	4	1270	1366	2396	17
3	4	1546	0	0.5	1	18	1	25	0.5	96	...	20	295	1752	3893	10
4	5	1434	0	1.4	0	11	1	49	0.5	108	...	18	749	810	1773	15
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
995	996	1700	1	1.9	0	0	1	54	0.5	170	...	17	644	913	2121	14
996	997	609	0	1.8	1	0	0	13	0.9	186	...	2	1152	1632	1933	8
997	998	1185	0	1.4	0	1	1	8	0.5	80	...	12	477	825	1223	5
998	999	1533	1	0.5	1	0	0	50	0.4	171	...	12	38	832	2509	15
999	1000	1270	1	0.5	0	4	1	35	0.1	140	...	19	457	608	2828	9

1000 rows × 21 columns

In [4]:
```
traind.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 [5]:
```
testd.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 [28]:
```
x=traind.drop('wifi',axis=1)
y=traind["wifi"]
```

In [7]:
```
traind['dual_sim'].value_counts()
```

Out[7]:
```
dual_sim
1    1019
0     981
Name: count, dtype: int64
```

In [8]:
```
m={"three_g":{"yes":1,"No":0}}
td=traind.replace(m)
print(td)
```

```
      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 [9]:
```
x=traind.drop('wifi',axis=1)
y=traind['wifi']
```

In [10]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
x_train.shape,x_test.shape
```

Out[10]: ((1600, 20), (400, 20))

In [11]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[11]:
```
RandomForestClassifier
RandomForestClassifier()
```

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

In [36]:
```
params={'max_depth':[2,35,10,20],'min_samples_leaf':[5,10,20,50,100,200],'n_estimators':[10,25,30,50,100,200]
```

In [14]:
```
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[14]:

GridSearchCV

estimator: RandomForestClassifier

RandomForestClassifier

In [15]:
```
grid_search.best_score_
```

Out[15]:
```
0.5387500000000001
```

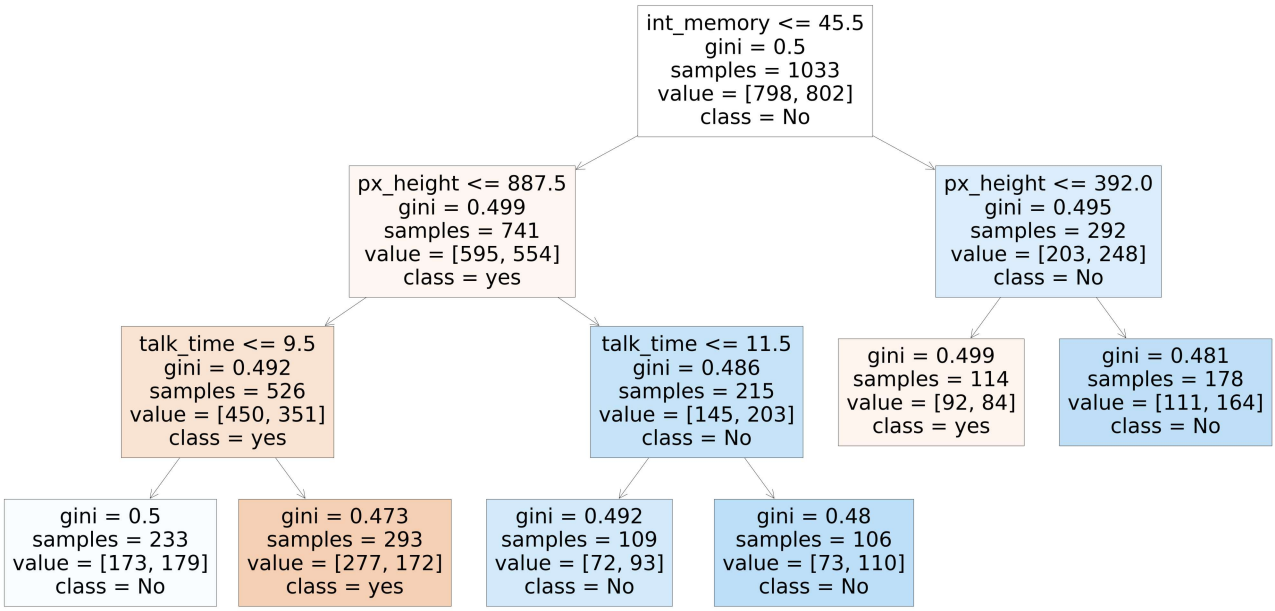
In [18]:
```
rf_best=grid_search.best_estimator_
print(rf_best)
```

```
RandomForestClassifier(max_depth=3, min_samples_leaf=100, n_estimators=50)
```

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

```
RandomForestClassifier(max_depth=3, min_samples_leaf=100, n_estimators=50)
```

In [21]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[5],feature_names=x.columns,class_names=["yes","No"],filled=True);
```



In [23]:
```
rf_best.feature_importances_
```

Out[23]:
```
array([0.06249034, 0.0075286 , 0.04696915, 0.00864533, 0.06713393,
       0.01704392, 0.06086025, 0.04680869, 0.05561469, 0.01160255,
       0.02707365, 0.2379613 , 0.09129798, 0.07013714, 0.04267532,
       0.01711047, 0.08518967, 0.00324524, 0.02976883, 0.01084295])
```

In [27]:
```
imp_df=pd.DataFrame({"Varience":x_train.columns,"Imp":rf_best.feature_importances_})
imp_df.sort_values(by="Imp",ascending=False)
```

Out[27]:

	Varience	Imp
11	px_height	0.237961
12	px_width	0.091298
16	talk_time	0.085190
13	ram	0.070137
4	fc	0.067134
0	battery_power	0.062490
6	int_memory	0.060860
8	mobile_wt	0.055615
2	clock_speed	0.046969
7	m_dep	0.046809
14	sc_h	0.042675
18	touch_screen	0.029769
10	pc	0.027074
15	sc_w	0.017110
5	four_g	0.017044
9	n_cores	0.011603
19	price_range	0.010843
3	dual_sim	0.008645
1	blue	0.007529
17	three_g	0.003245

Test Data

In [29]:
```
x=testd.drop('wifi',axis=1)
y=testd["wifi"]
```

In [30]:
```
testd['dual_sim'].value_counts()
```

Out[30]:
```
dual_sim
1    517
0    483
Name: count, dtype: int64
```

In [31]:
```
m={"three_g":{"yes":1,"No":0}}
td=traind.replace(m)
print(td)
```

```
      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 [32]:
```
x=testd.drop('wifi',axis=1)
y=testd['wifi']
```

In [33]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
x_train.shape,x_test.shape
```

Out[33]: ((800, 20), (200, 20))

In [34]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[34]:
```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [35]:
```
rf=RandomForestClassifier()
```

In [37]:
```
params={'max_depth':[2,35,10,20],'min_samples_leaf':[5,10,20,50,100,200],'n_estimators':[10,25,30,50,100,200]
```

In [38]:
```
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[38]:

GridSearchCV

estimator: RandomForestClassifier

RandomForestClassifier

In [39]:
```
grid_search.best_score_
```

Out[39]: 0.53

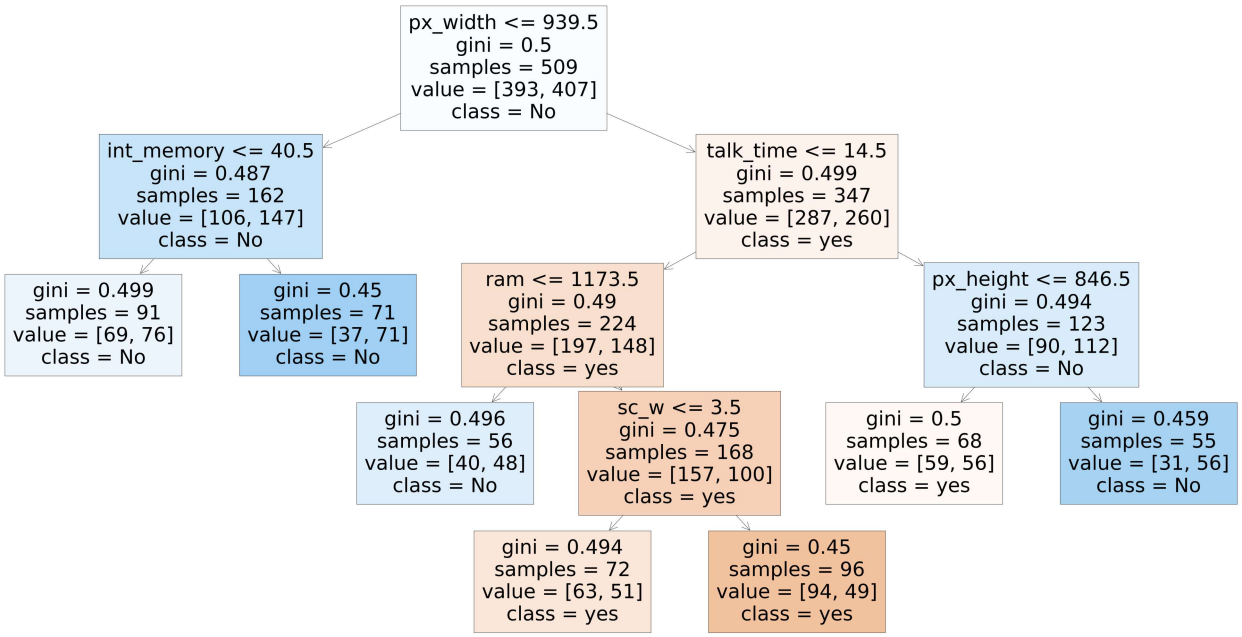
In [40]:
```
rf_best=grid_search.best_estimator_
print(rf_best)
```

```
RandomForestClassifier(max_depth=35, min_samples_leaf=50, n_estimators=50)
```

In [41]:
```
rf_best=grid_search.best_estimator_
print(rf_best)
```

```
RandomForestClassifier(max_depth=35, min_samples_leaf=50, n_estimators=50)
```

In [42]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[5],feature_names=x.columns,class_names=["yes","No"],filled=True);
```



In [43]:
```
rf_best.feature_importances_
```

Out[43]:
```
array([0.04405736, 0.08051273, 0.01745645, 0.05571116, 0.01453586,
       0.06371533, 0.03197881, 0.06969091, 0.01912393, 0.11489556,
       0.01938361, 0.04267464, 0.06589819, 0.09438956, 0.08943337,
       0.02961427, 0.02508144, 0.10637251, 0.00143634, 0.01403797])
```

In [44]:
```
imp_df=pd.DataFrame({"Varience":x_train.columns,"Imp":rf_best.feature_importances_})
imp_df.sort_values(by="Imp",ascending=False)
```

Out[44]:

	Varience	Imp
9	mobile_wt	0.114896
17	talk_time	0.106373
13	px_width	0.094390
14	ram	0.089433
1	battery_power	0.080513
7	int_memory	0.069691
12	px_height	0.065898
5	fc	0.063715
3	clock_speed	0.055711
0	id	0.044057
11	pc	0.042675
6	four_g	0.031979
15	sc_h	0.029614
16	sc_w	0.025081
10	n_cores	0.019384
8	m_dep	0.019124
2	blue	0.017456
4	dual_sim	0.014536
19	touch_screen	0.014038
18	three_g	0.001436

In [ ]:
```

```