Mobile_price_classification_with_ML

December 23, 2025

1 Mobile Price Classification with Machine Learning

[2]:

```
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.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
sns.set()
```

[7]:

```
data = pd.read_csv("mobile_prices.csv")
print(data.head())
```

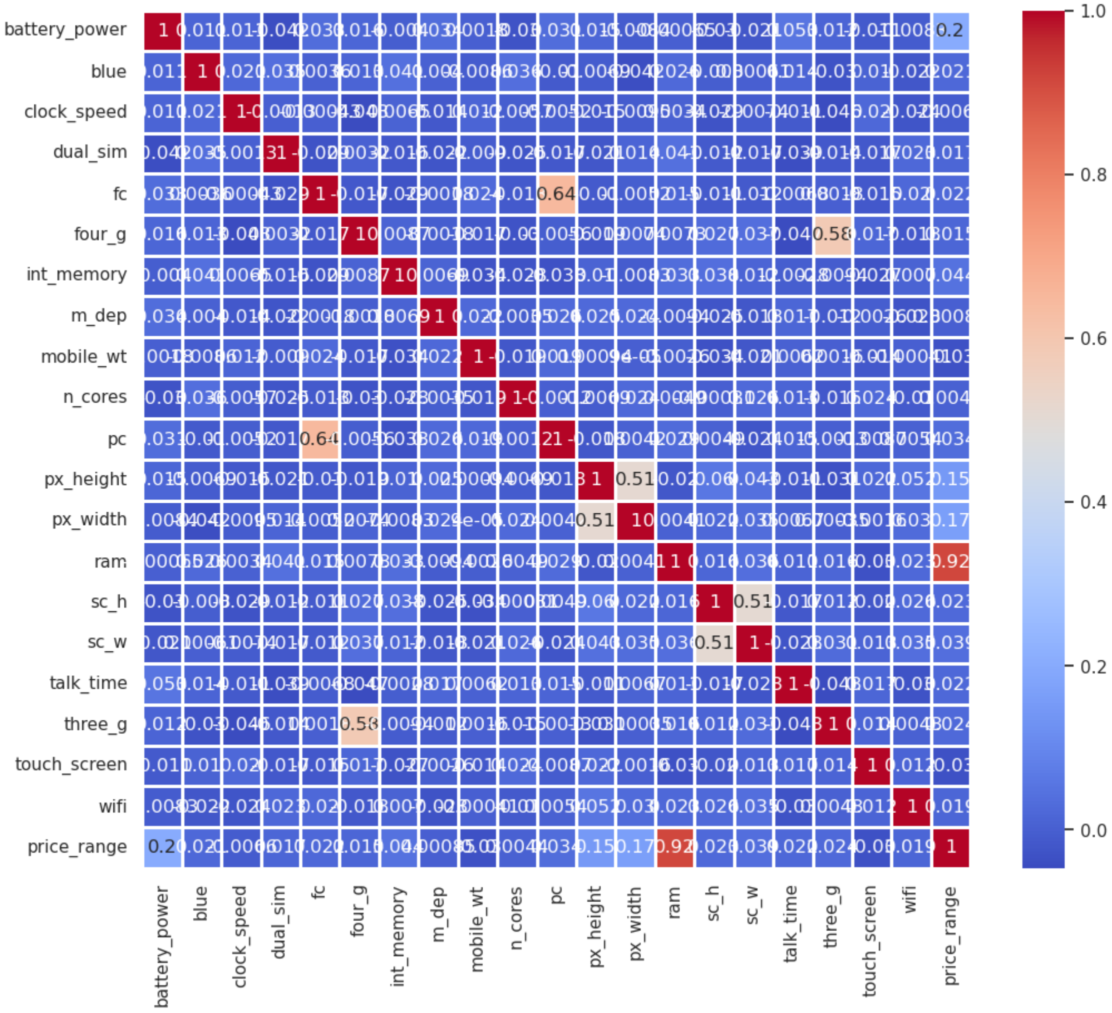
```
   battery_power  blue  clock_speed  dual_sim  fc  four_g  int_memory  m_dep
mobile_wt  n_cores  pc  px_height  px_width   ram  sc_h  sc_w  talk_time
three_g  touch_screen  wifi  price_range
0            842     0          2.2         0   1       0           7    0.6
188        2   2          20       756  2549     9     7          19        0
0     1            1
1           1021     1          0.5         1   0       1          53    0.7
136        3   6         905      1988  2631    17     3           7        1
1     0            2
2            563     1          0.5         1   2       1          41    0.9
145        5   6        1263      1716  2603    11     2           9        1
1     0            2
3            615     1          2.5         0   0       0          10    0.8
131        6   9        1216      1786  2769    16     8          11        1
0     0            2
4           1821     1          1.2         0  13       1          44    0.6
141        2  14        1208      1212  1411     8     2          15        1
1     0            1
```

[8]:

```
plt.figure(figsize=(12,10))
sns.heatmap(data.corr(), annot=True,␣
 ↪cmap="coolwarm",linecolor="white",linewidths=1)
```

[8]:

```
<Axes: >
```



1.1 Data Preparation

[10]:

```
x = data.iloc[:, :-1].values
y = data.iloc[:, -1].values
x = StandardScaler().fit_transform(x)
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20,␣
  ↪random_state=0)
```

1.2 Mobile Price Classification Model

[14]:

```
from sklearn.linear_model import LogisticRegression
lreg = LogisticRegression()
lreg.fit(x_train, y_train)
```

```
y_pred = lreg.predict(x_test)
```

[16]:
```
accuracy = accuracy_score(y_test, y_pred) * 100
print("Accuracy of the Logistics Regression Model: ",accuracy)
```

```
Accuracy of the Logistics Regression Model:  95.5
```

[17]:
```
print(y_pred)
```

```
[3 0 2 2 3 0 0 3 3 1 1 3 0 2 3 0 3 2 2 1 0 0 3 1 2 2 3 1 3 1 1 0 2 0 2 3 0
 0 3 3 3 1 3 3 1 3 0 1 3 1 1 3 0 3 0 2 2 2 0 3 3 1 3 2 1 2 3 2 2 2 3 2 1 0
 1 3 2 2 1 2 3 3 3 0 0 0 2 1 2 3 1 2 2 1 0 3 3 3 0 3 1 1 3 1 3 2 2 3 2 3 3
 0 0 1 3 3 0 0 1 0 0 3 2 2 1 2 1 1 0 2 1 3 3 3 3 3 3 2 0 1 1 2 1 3 0 3 0 0
 2 0 1 1 1 1 3 0 0 3 1 3 2 1 3 1 2 3 3 2 1 0 3 1 2 3 3 0 2 2 3 1 2 1 0 1 2
 2 2 0 3 3 1 1 0 2 3 0 1 2 2 0 3 3 3 1 2 3 3 3 0 0 0 2 3 3 0 0 1 3 2 3 3 3
 0 0 2 3 3 1 0 2 0 0 0 3 2 1 2 2 1 1 0 2 3 3 0 0 1 3 3 1 3 0 3 1 1 0 2 3 3
 2 0 0 1 2 3 2 2 3 2 1 0 3 3 2 1 3 2 2 2 1 0 2 2 1 0 0 2 2 2 3 0 1 3 0 2 2
 3 0 2 0 1 1 3 0 0 2 3 1 2 0 2 0 3 0 3 3 2 3 1 2 2 1 1 1 0 1 0 3 1 0 3 0 0
 1 3 0 3 1 1 0 1 3 0 2 1 1 2 1 1 0 2 0 0 3 1 2 3 2 2 0 3 2 2 1 3 2 3 3 3 0
 2 0 3 0 1 1 2 3 1 3 1 2 0 1 2 3 0 0 1 3 0 3 0 2 2 1 1 0 2 0]
```

[18]:
```
(unique, counts) = np.unique(y_pred, return_counts=True)
price_range = np.array((unique, counts)).T
print(price_range)
```

```
[[  0  95]
 [  1  90]
 [  2  97]
 [  3 118]]
```

[ ]:
```
```