
# *Mobile Price Classification*

## About Dataset

Bob has started his own mobile company. He wants to give tough fights to big companies like Apple,Samsung etc.
He does not know how to estimate the price of mobiles his company creates. In this competitive mobile phone market you cannot simply assume things. To solve this problem he collects sales data of mobile phones of various companies.
Bob wants to find out some relation between features of a mobile phone(eg:- RAM,Internal Memory etc) and its selling price. But he is not so good at Machine Learning. So he needs your help to solve this problem.
In this problem you do not have to predict the actual price but a price range indicating how high the price is.

## 1.Data

Used the data set called"Mobile Price | Multiclass classification" from Kaggle.com, and this dataset contains all the functions of current mainstream mobile phones and the frequency of customer use. I'm going to use  the pandas library, and functions like read_csv() to collect the data from the database.
Website:
https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-classification/data
We will proceed with reading the data, and then perform data analysis. After data analysis, we will find out the data distribution and data types.

## 2.Data Wrangling

```
<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
```

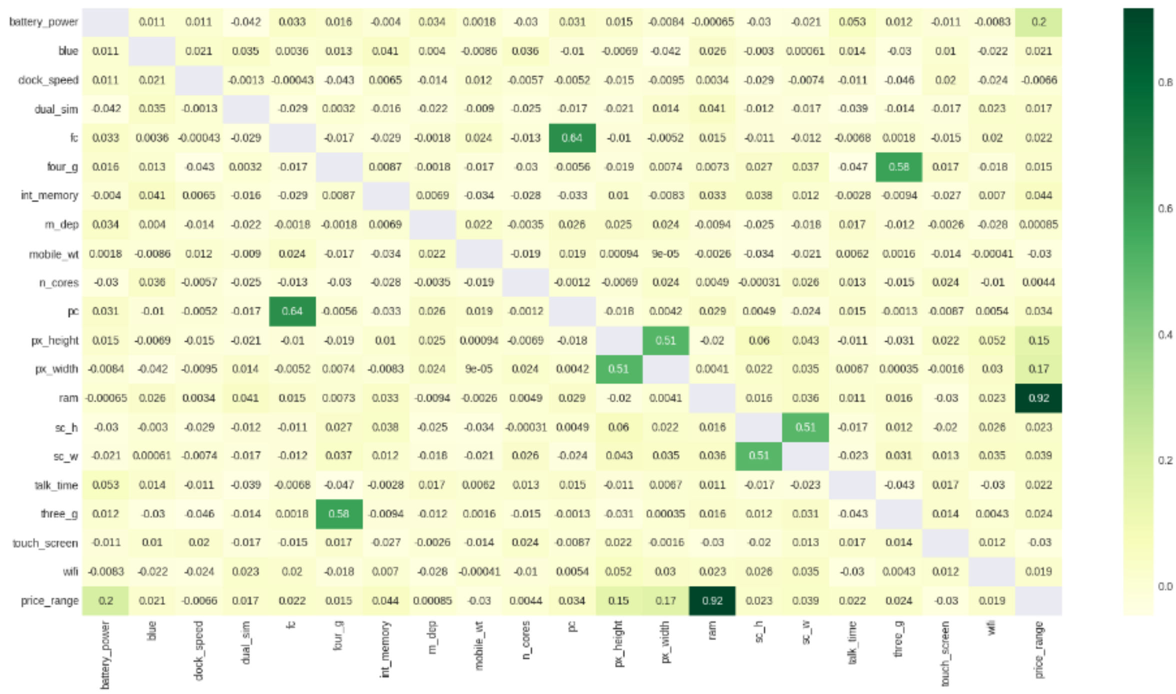
We found that we needed to clean some data and after cleaning the data we got 0 NaN values in the dataframe.

```
battery_power    0
blue             0
clock_speed      0
dual_sim         0
fc               0
four_g           0
int_memory       0
m_dep            0
mobile_wt        0
n_cores          0
pc               0
px_height        0
px_width         0
ram              0
sc_h             0
sc_w             0
talk_time        0
three_g          0
touch_screen     0
wifi             0
price_range      0
dtype: int64
```

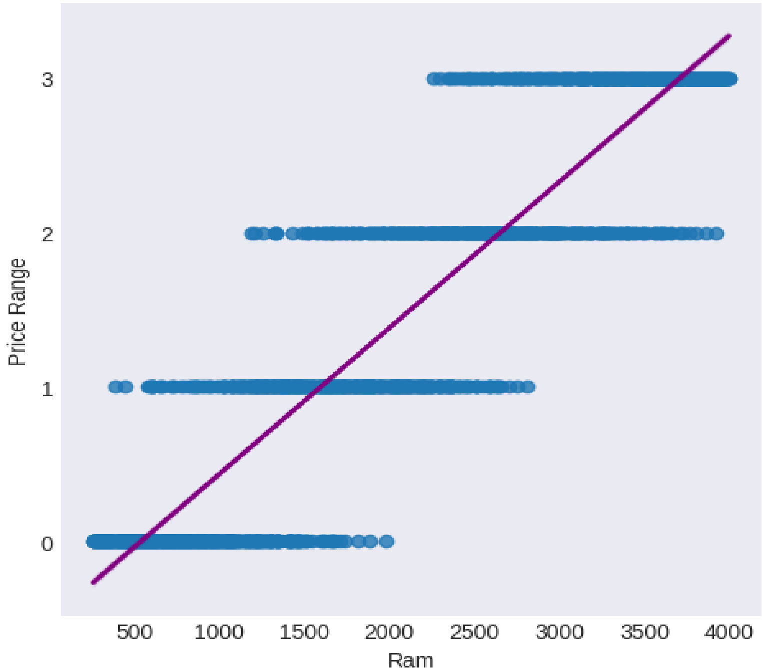
Next, we used EDA to determine the relationship between different features and the main impact of different features on pricing.

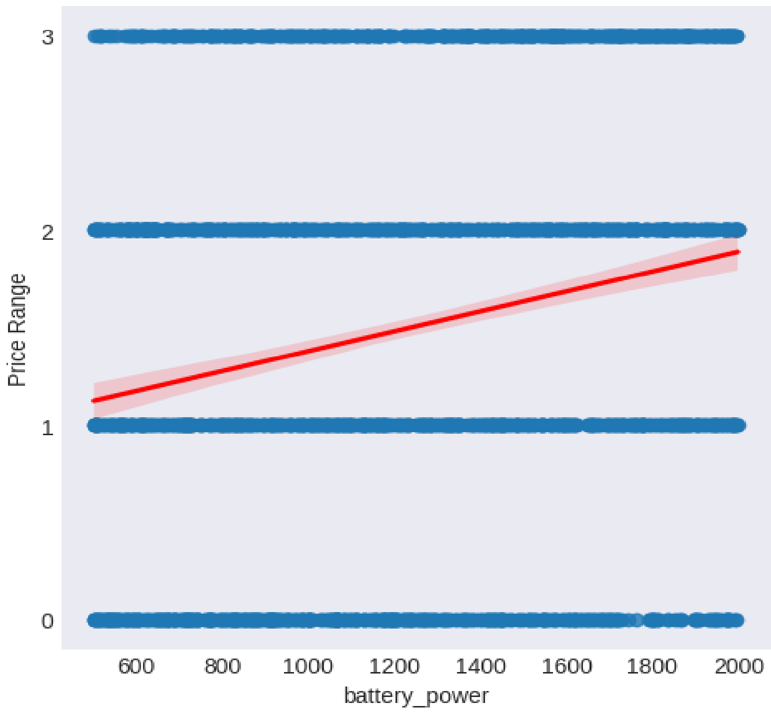
## 3.EDA

```
ram              0.917046
battery_power    0.200723
px_width         0.165818
px_height        0.148858
int_memory       0.044435
sc_w             0.038711
pc               0.033599
touch_screen     0.030411
mobile_wt        0.030302
three_g          0.023611
sc_h             0.022986
fc               0.021998
talk_time        0.021859
blue             0.020573
wifi             0.018785
dual_sim         0.017444
four_g           0.014772
clock_speed      0.006606
n_cores          0.004399
m_dep            0.000853
price_range           NaN
Name: price_range, dtype: float64
```



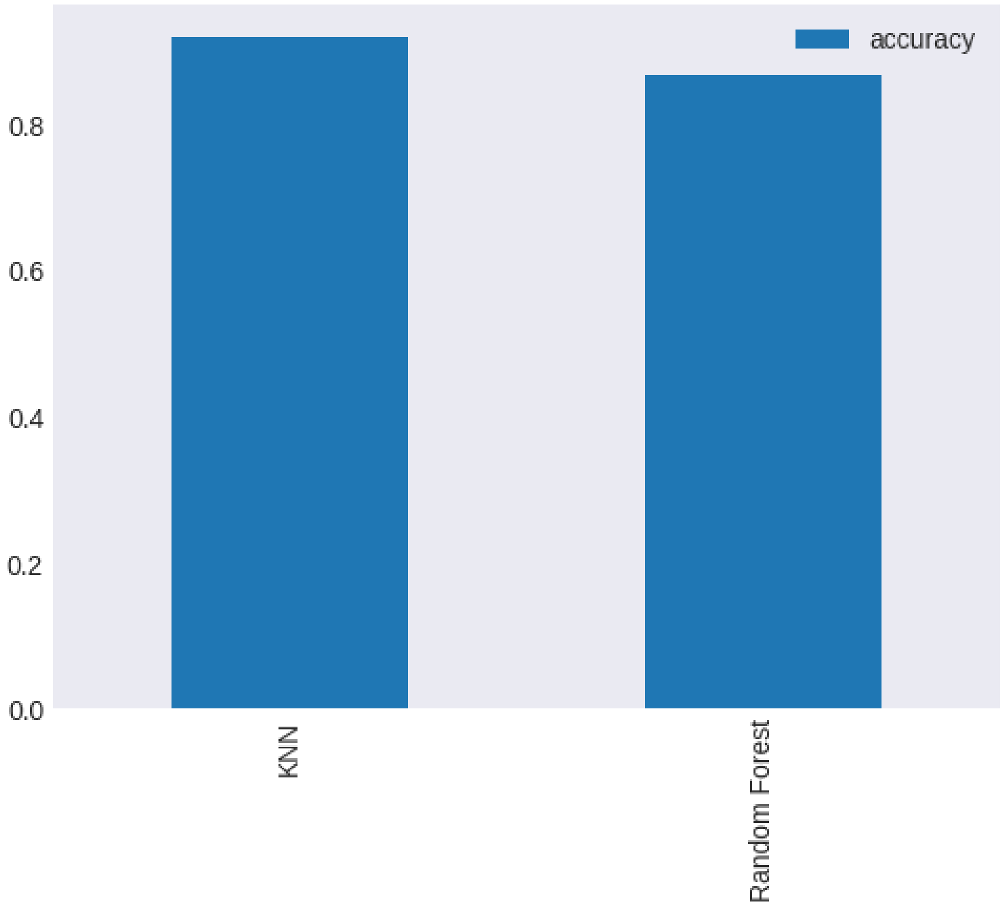

We used a heat map to show the highest correlation between Price_range and other features in the dataset. We found that the correlation between "price_range" and "ram" is very high, which means that in future modeling we may need to use the important feature of "ram" to predict the price range of mobile phones.Moreover, battery power also has a high correlation with our target variable.




We can clearly see that the higher the RAM value, the higher the price of the mobile phone. Therefore, there is an increasing relationship between RAM and mobile phone prices. Let's also see some relations between price range and battery power.We can see that although there is an increasing relationship between battery power and price range, it does not have as big an impact as ram on the price range. This strengthens our reason for choosing the ram feature for modeling.

## 4.Pre-Processing and Training Data




After we did the Train-Test-Split. Two modeling methods, KNN and random forest regression, are established and compared. After comparing the data, we found that the KNN modeling method has more accuracy, so KNN is a more suitable modeling method. Next, we did tune the hyperparameters of our KNN and Random Forest models. Unfortunately, however, Linear Regression has a few hyperparameters which don't affect its overall score, and therefore, our final final score for our Linear Regression model is the score above.

## 5.Modeling

Model comparison alone cannot yield the most accurate score, so we used a hyperparameter method to verify whether the final score of the KNN model can be improved.

## 1. Hyperparemeter tuning for Random Forest:

In [38]:
```
rf_grid = {"n_estimators": np.arange(10, 1000, 50),
           "max_depth": [None, 3, 5, 10],
           "min_samples_split": np.arange(2, 20, 2),
           "min_samples_leaf": np.arange(1, 20, 2)}
```

In [39]:
```
rs_rf = RandomizedSearchCV(RandomForestClassifier(),
                           param_distributions=rf_grid,
                           cv=5,
                           n_iter=20,
                           verbose=True)

rs_rf.fit(X_train, y_train);

rs_rf.best_params_
```

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
```

Out[39]:
```
{'n_estimators': 410,
 'min_samples_split': 8,
 'min_samples_leaf': 1,
 'max_depth': None}
```

In [40]:
```
rs_rf.score(X_test, y_test)
```

Out[40]:
```
0.8818181818181818
```

As we can see the test secor is around 88%, let's see how KNN goes.

## 2. Hyperparemeter tuning KNN:

In [43]:
```
train_scores = []

test_scores = []

neighbors = range(1, 21)

kNN = KNeighborsClassifier()

for i in neighbors:
    kNN.set_params(n_neighbors = i)

    kNN.fit(X_train, y_train)

    train_scores.append(kNN.score(X_train, y_train))

    test_scores.append(kNN.score(X_test, y_test))
```

In [44]:
```
kNN = KNeighborsClassifier(n_neighbors=13)
kNN.fit(X_train, y_train)
y_pred = kNN.predict(X_test)
print(f'KNN Model Score: {kNN.score(X_test, y_test) * 100}%')
```

```
KNN Model Score: 92.42424242424242%
```

That pretty much says that the KNN model is correct to use the best model with the best hyperparameters.After we determined the best model, we did the model evaluation. We used two model evaluation methods: classification report and Cross Validation.

```
              precision    recall  f1-score   support

           0       0.96      0.98      0.97       178
           1       0.91      0.90      0.90       163
           2       0.87      0.88      0.88       161
           3       0.95      0.93      0.94       158

    accuracy                           0.92       660
   macro avg       0.92      0.92      0.92       660
weighted avg       0.92      0.92      0.92       660
```

```
Cross Validation Scores: [0.94   0.935  0.9425 0.935  0.9175]
```

```
]: print(f'Cross Validation Score (Mean): ' + str(np.mean(cross_val_score(kNN, X, y, cv=5))))
```

```
Cross Validation Score (Mean): 0.9339999999999999
```

Through the cross validation method, it is not difficult to see that similar scores can be maintained by performing data segmentation in time and even increasing some accuracy.

## 6.Conclusion

Overall observation shows that the most important features for predicting mobile phone prices are memory and battery power. We solve this problem by using a correlation matrix, specifically looking at the variables that are most correlated with the price range.

Through the KNN model we can see that KNN Model Score: 92%. Compared to a random forest model, the score is clearly higher. This is right for choosing the KNN model for this problem. After comparing the models, we used the hyperparameter method and found that it can improve the score of the KNN model, thus further confirming that KNN is the most suitable model for this problem.

Advice on business:

1. While maintaining almost the same price range and the same phone conditions, appropriately increase the phone RAM. Because in our opinion, mobile phone RAM is an important factor that affects the price range, and companies need to pay attention to it, so that they can compete with big-brand mobile phone companies.
2. If the first method does not achieve good results, we can also try to improve the battery performance while also increasing the RAM, because battery performance, as a factor second only to the mobile phone RAM, will also affect pricing.