Mobile Price Prediction Using Logistic Regression

Name:	**Mehar Khurana**
Registration No./Roll No.:	20322
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	January 12, 2023
Date of Submission:	April 16, 2023

1 Introduction

This research seeks to predict the pricing range for mobile phones given the varied specs of various phones. The price range is divided into four groups: cheap, moderate, economical and expensive i.e. 0, 1, 2 and 3 respectively. Using the attributes shown in Figure 2, we wish to estimate the price range for the mobile devices included in the dataset. The dataset provided is equally divided among the four classes i.e. each class has 500 mobiles out of given 2000 mobiles.

```
dataset.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
```

Figure 1: Overview of features

2 Methods

2.1 Train-Test Split

The dataset is randomly split into training and testing sets by a factor of 0.8, i.e., 70rows) is selected for training and 30

2.2 Models Used For Classification

This problem is clearly a classification problem as we have four discrete classes (0, 1, 2 and 3). Hence, the main classification algorithms used for this problem are:
• Logistic Regression
• K-Nearest Neighbors
• Gaussian Naive Bayes
• Decision Tree Classifier
• Random Forest Classifier
• Support Vector Classifier

The complete project can be found here.

2.3 Hyperparameter Tuning

Hyperparameters are specified parameters that can be used to tune the behavior of a machine learning algorithm. These are initialized before the training and supplied to the model. A hyperparameter is a parameter whose value governs the learning process. To perform better and improve on the evaluation metric, hyperparameters are tuned by selecting the ideal values. Grid search is one of the most basic hyper-parameter tuning techniques. Hence its implementation is very straightforward. To tune models, all feasible permutations of the hyperparameters for a specific model are used, and the bestperforming ones are chosen.

Table 1: Hyper-parameters of different models

Models	Hyper-parameters Space	Best Hyper-parameter features
K-NN	• n-neighbours: [5,7,9,11,13,15,16,17,18,19] • weights : ['uniform','distance'], • 'metric' : ['minkowski','euclidean','manhattan']	• n-neighbours: [15] • weights : ['distance'], • 'metric' : ['manhattan']
Decision tree	• criterion: ['gini', 'entropy'] • max features: ['auto', 'sqrt', 'log2',None] • max depth: [15, 30, 45, 60] • ccp alpha:[0.009,0.005,0.05]	• criterion: ['entropy'] • max features: [None] • max depth: [15] • ccp alpha:[0.005]
Random forest	• criterion: ['gini', 'entropy'] • n estimators: [int(x) for x in np.linspace(start = 200, stop = 300, num = 100)] • max depth: [10,20,30,50,100,200]	• criterion: ['entropy'] • n estimators: [251] • max depth: [200]
Gaussian Naive bayes	• var smoothing :np.linspace(0,-13,num=100)	• var smoothing :[0.0]
Logistic Regression	• C:np.linspace(start = 0.1, stop = 10, num = 100) • penalty:["l1","l2",'elasticnet'] •solver:['newton-cg','lbfgs','liblinear']	• C:[0.1] •solver:['newton-cg']
SVM	• C :np.logspace(-2,7,num=25,base=2) • gamma: [1,0.1,0.01,0.001] • kernel:('linear','rbf','polynomial','sigmoid')	• C :[1.189207115002721] • gamma: [1] • kernel:['linear']

3 Evaluation Criteria

In this problem, the performance metrics used are accuracy, macro-averaged precision, recall, and f-measure since this is a classification problem. These measures are described as:

- Precision $= \dfrac{\text{No of correctly predicted positive points}}{\text{total predicted positive points}} = \dfrac{\text{TP}}{\text{TP} + \text{FP}}$
- Recall $= \dfrac{\text{No of correctly predicted positive points}}{\text{total actual positive points}} = \dfrac{\text{TP}}{\text{TP} + \text{FN}}$
- Accuracy $= \dfrac{\text{No of correctly predicted data points}}{\text{total number of data points}} = \dfrac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$
- f1_score $= \dfrac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$

If there are two classes Positive and Negative then the following are defined :

- True Positives(TP): It is the case where we predicted Positive and the real output was also Positive.

- True Negatives(TN): It is the case where we predicted Negative and the real output was also Negative.

- False Positives(FP): It is the case where we predicted Positive but it was actually Negative.

- False Negatives(FN): It is the case where we predicted Negative but it was actually Positive

4 Analysis of Results

Table 2 shows the recall, precision, accuracy and f measure for all the classification models used in this project

Table 2: Performance Of Different Classifiers Using All Terms

Classifier	Precision	Recall	Accuracy	F-measure
K-NN	0.94761	0.94765	0.94833	0.94754
Decision tree	0.83914	0.83939	0.84166	0.83885
Random forest	0.89999	0.90098	0.90166	0.90013
Gaussian Naive bayes	0.81804	0.81767	0.82000	0.81772
Logistic Regression	0.98166	0.98119	0.98166	0.98166
SVM	0.97283	0.97290	0.97333	0.97272

5 Discussions and Conclusion

I found out that Logistic Regression provides the best f-measure as well accuracy and hence I'll use it to build my model.

5.1 Result on Test Data Sample

On using the Logistic Regression I got the following class labels predicted for the test sample.

```
array([3, 3, 2, 3, 1, 3, 3, 1, 3, 0, 3, 3, 0, 0, 2, 0, 2, 1, 3, 2, 1, 3,
       1, 1, 3, 0, 2, 0, 3, 0, 2, 0, 3, 0, 1, 1, 3, 1, 2, 1, 1, 2, 0, 0,
       0, 1, 0, 3, 1, 2, 1, 0, 3, 0, 3, 1, 3, 1, 1, 3, 3, 3, 0, 1, 1, 1,
       2, 3, 1, 2, 1, 2, 2, 3, 3, 0, 2, 0, 2, 3, 0, 3, 3, 0, 3, 0, 3, 1,
       3, 0, 1, 2, 2, 1, 2, 2, 1, 2, 1, 2, 1, 0, 0, 3, 0, 2, 1, 1, 2, 3,
       3, 2, 1, 3, 3, 3, 3, 2, 3, 0, 0, 3, 2, 1, 2, 0, 3, 2, 3, 2, 0, 2,
       2, 1, 3, 1, 1, 0, 3, 2, 1, 2, 1, 2, 2, 3, 3, 3, 2, 3, 2, 3, 1, 0,
       3, 2, 3, 3, 3, 3, 2, 2, 3, 3, 3, 3, 1, 0, 3, 0, 0, 0, 2, 1, 0, 1,
       0, 0, 1, 2, 1, 0, 0, 1, 1, 2, 2, 1, 0, 0, 0, 1, 0, 3, 1, 0, 2, 2,
       3, 3, 1, 2, 2, 2, 3, 2, 2, 1, 1, 0, 1, 2, 0, 2, 3, 3, 0, 2, 0, 3,
       2, 3, 3, 1, 0, 1, 0, 3, 0, 1, 0, 2, 2, 1, 3, 1, 3, 0, 3, 1, 2, 0,
       0, 2, 1, 3, 3, 3, 1, 1, 3, 0, 0, 2, 3, 3, 1, 3, 1, 1, 3, 2, 1, 2,
       3, 3, 3, 1, 0, 0, 2, 3, 1, 1, 3, 2, 0, 3, 0, 0, 2, 1, 0, 3, 2, 3,
       3, 2, 1, 3, 3, 2, 3, 1, 2, 1, 2, 0, 2, 3, 1, 0, 0, 3, 0, 3, 0, 1,
       2, 0, 2, 3, 1, 3, 2, 2, 1, 2, 0, 0, 0, 1, 3, 2, 0, 0, 0, 3, 2, 0,
       2, 3, 1, 2, 2, 2, 3, 1, 3, 3, 2, 2, 2, 3, 3, 0, 3, 0, 3, 1, 3, 1,
       2, 3, 0, 1, 0, 3, 1, 3, 2, 3, 0, 0, 0, 0, 2, 0, 0, 2, 2, 1, 2, 2,
       2, 0, 1, 0, 0, 3, 2, 0, 3, 1, 2, 2, 1, 2, 3, 1, 1, 2, 2, 1, 2, 0,
       1, 1, 0, 3, 2, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 2, 2, 3, 2, 3, 0, 3,
       0, 3, 0, 1, 1, 0, 2, 0, 3, 2, 3, 3, 1, 3, 1, 3, 1, 3, 2, 0, 1, 2,
       1, 1, 0, 0, 0, 1, 2, 1, 0, 3, 2, 0, 2, 2, 0, 0, 3, 1, 2, 0, 2, 3,
       3, 0, 3, 0, 2, 3, 2, 3, 0, 2, 0, 2, 3, 0, 1, 1, 0, 0, 1, 1, 1, 3,
       3, 3, 2, 3, 1, 2, 2, 3, 3, 3, 2, 0, 2, 1, 2, 2, 1, 0, 2, 2, 0, 0,
       0, 3, 1, 0, 2, 2, 2, 0, 3, 1, 2, 2, 1, 3, 0, 2, 3, 0, 1, 1, 3, 3,
       1, 1, 2, 3, 2, 0, 3, 1, 2, 0, 3, 3, 1, 3, 2, 2, 3, 0, 1, 2, 3, 1,
       3, 2, 3, 1, 1, 0, 0, 3, 1, 0, 3, 2, 3, 2, 1, 3, 3, 3, 2, 3, 3, 1,
       2, 0, 2, 3, 3, 1, 0, 1, 1, 2, 2, 2, 0, 0, 2, 2, 3, 2, 0, 2, 1, 3,
       3, 0, 1, 3, 0, 2, 1, 1, 0, 0, 2, 1, 0, 1, 1, 2, 2, 0, 2, 2, 1, 0,
       3, 0, 0, 3, 2, 0, 0, 0, 0, 0, 3, 0, 3, 1, 3, 2, 1, 3, 3, 0, 1, 1,
       3, 2, 2, 2, 0, 3, 0, 2, 0, 2, 0, 0, 1, 1, 1, 2, 1, 3, 1, 3, 2, 2,
       1, 3, 2, 0, 2, 2, 0, 3, 3, 0, 2, 1, 1, 2, 0, 3, 2, 0, 3, 2, 3, 0,
       0, 3, 0, 2, 2, 3, 2, 2, 2, 2, 1, 2, 3, 0, 1, 0, 1, 2, 1, 0, 0, 1,
       0, 0, 3, 1, 1, 2, 0, 1, 1, 1, 3, 0, 3, 2, 3, 0, 0, 1, 2, 2, 1, 0,
       1, 2, 0, 1, 1, 0, 0, 3, 3, 0, 3, 1, 1, 3, 0, 1, 0, 2, 2, 0, 3, 1,
       0, 3, 0, 1, 0, 3, 3, 3, 2, 3, 0, 3, 2, 0, 0, 0, 3, 3, 2, 0, 2, 1,
       3, 0, 0, 2, 2, 0, 3, 1, 2, 1, 1, 2, 3, 1, 1, 1, 2, 1, 0, 2, 2, 0,
       2, 0, 0, 0, 0, 2, 3, 3, 3, 0, 1, 2, 1, 1, 0, 0, 2, 1, 0, 2, 0, 3,
       2, 2, 1, 2, 0, 2, 1, 3, 0, 0, 3, 2, 3, 0, 0, 2, 3, 3, 1, 3, 2, 1,
       0, 0, 3, 3, 1, 3, 0, 0, 0, 2, 2, 1, 2, 0, 3, 2, 1, 2, 3, 3, 0, 1,
       1, 2, 1, 2, 2, 0, 1, 3, 1, 1, 3, 0, 2, 3, 2, 1, 1, 1, 3, 3, 0, 2,
       3, 0, 2, 3, 2, 2, 2, 3, 2, 0, 1, 2, 1, 2, 1, 1, 2, 2, 2, 1, 2, 1,
       1, 1, 3, 1, 0, 1, 2, 3, 1, 0, 0, 3, 2, 2, 3, 0, 3, 2, 2, 1, 3, 0,
       1, 3, 1, 1, 1, 2, 3, 2, 0, 3, 0, 2, 3, 0, 3, 1, 3, 3, 1, 0, 2, 3,
       1, 0, 2, 1, 2, 1, 2, 0, 2, 2, 0, 2, 3, 2, 3, 0, 2, 1, 1, 2, 2, 3,
       3, 0, 2, 1, 2, 1, 3, 1, 1, 3, 0, 1, 0, 0, 3, 3, 2, 0, 0, 0, 0, 3,
       2, 3, 3, 0, 0, 2, 1, 0, 2, 2])
```

Figure 2: Overview of features

5.2 Future Plans

In the longer run, this project can be expanded in several ways such as:

- Making similar models for finding which mobile phones can give better features and still are cheaper as compared to the other mobiles with similar feature but being expensive.

The model can even be made better with certain enhancements

- To increase efficiency and make it easier for the algorithm to detect patterns in the data, feature engineering can be employed.
- Artifical Neural Networks can also be used along with some Deep Learning Techniques.

6 References

- Lecture notes by Dr.Tanmay Basu.
- Machine Learning by Tom M. Mitchell.
- `https://www.kaggle.com`
- `https://scikit-learn.org`
- `https://towardsdatascience.com`