Predicting mobile phone price using Machine Learning

1. Introduction

1.1 Problem Formulation

The goal is to predict the price category label of the mobile phone samples from the test set based on different mobile phone specifications. Since all data points in the train set have their respective labels, the task will be supervised learning.

1.2 Data description

The dataset is downloaded from kaggle.com. The dataset offers a wide range of features: 14 numeric features, six binary features (Details of the feature description are attached in the appendix.), and categorical label – price range in 0, 1, 2, and 3. The dataset contains a training set of 2000 samples. Datasets are well balanced, with 500 samples in each label category. No missing features or labels were found.

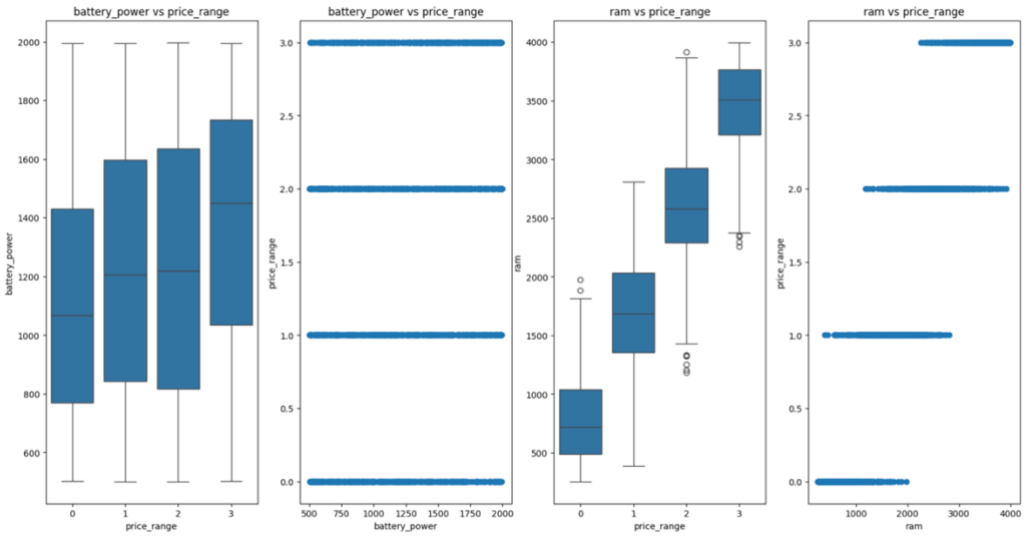
1.3 Data processing

The dataset is split into training and testing sets at a threshold of 0.33, a standard practice for splitting to ensure enough data points for training.

Due to varying ranges of numeric features, StandardScaler is used for standardization, transforming the numeric features to have a mean of 0 and a standard deviation of 1.

As one can observe in the feature correlation matrix (Figure 1 in the appendix), some features are moderately correlated. Therefore, every two features with a correlation higher than 0.5 are combined (Table 1, Appendix).

Scatter and box plots are plotted to observe correlations between some features and the label. Feature ‘ram’ is observed to have a strong correlation with price range.



Due to the ordinal nature of the label data, Pearson correlation is used to observe the linearity in the relation between each feature and the label. The result further confirmed the strong linear correlation between ‘ram ’ and ‘price_range.’

Kdeplot is used to observe data distribution for each feature. As seen in Figure 2, the distribution of data points about price range is very similar for some features. However, all features are kept for more accurate analysis.

2. Method

2.1 LinearRegression Model

Linear regression(LR) is suitable for predicting continuous labels. Nevertheless, LR is used as the baseline model for several benefits. Despite being categoric, the label data has an ordinal nature: 0 being the cheapest and 3 being the most expensive. With a squared residual loss function, linear regression can provide helpful information about feature-label linearity. Besides, Linear Regression models are generally fast to train and predict.

The Validation metrics I chose for the linear regression model are Accuracy Score, mean squared error (MSE), and R-squared score. The Accuracy Score is an intuitive measure of overall model performance. It calculates the ratio of correct predictions to the total cases evaluated. MSE measures the average squared difference between predicted and actual values and is sensitive to outliers. R-squared measures the proportion of variance explained by the model in the data.

The Linear Regression yields a mean-squared error of 0.34 and an R-squared error of 0.91, which signals good prediction. The continuous predictions are discretized by converting into [0,1,2,3] bins to calculate the accuracy score, which measures the proportion of correctly classified instances. However, the result is merely 0.5.

2.2 LogisticsRegression Model

Multinomial Logistic Regression is designed to predict probabilities for categorical outcomes and show how each feature influences the probability of each price range. Unlike Linear regression, Logistic regression does not assume a linear relationship between features and the label. The model is configured to use a multinomial logistic regression known as Softmax Regression. Softmax Regression fits one model for all classes at once, making it more probable to capture the relationships accurately.

I chose the accuracy score, classification report, cross-entropy loss, ROC AUC Score, and confusion matrix as evaluation criteria. The classification report comprehensively shows precision, recall, and F1 scores for each class and weighted averages. Cross-entropy loss is a suitable metric for evaluating models with 0-1 probability outputs. It measures the difference between the predictions and the label and is a standard metric for multi-class classification problems. The ROC AUC curve measures the model's ability to distinguish between classes. 1.0 presents a 100% accuracy in prediction, while 0.5 presents random guessing. The confusion matrix shows the counts of true positives, false positives, true negatives, and false negatives for each class.

The model yields an accuracy score of 0.92 and a cross-entropy loss of 0.24. The accuracy score indicates that 92% of the test samples were classified correctly. Cross-entropy loss of 0.24 is relatively low, showing that the model predicts most labels correctly. Precision measures the portion of true positives out of predicted positive samples. Recall measures the portion of true positives out of actual positive samples. An average F1 score of 0.91 indicates that the model is doing well in predicting positive samples while not missing much of the actual positive ones. The ROC AUC Score of 0.99 suggests the model's high accuracy in distinguishing between different price ranges.

3. Summary

The evaluation metrics suggested that a great model was trained, which can very accurately predict phone prices based on given features. Here are some key findings:

- The logistics regression model performed clearly better than the linear regression model.
- Discretization of continuous results does not yield accurate predictions despite having ordinal label values.
- In cases where value ranges of features vary greatly, StandardScaler() can improve the performance significantly.

Limitations and Potential Improvements

Logistic Regression assumes a linear relationship between the features and the log odds of a phone being in one phone price range. LR can fail to capture the true relationship when this assumption is violated to a certain degree. Further improvements can be made by plotting the log odds of each class against individual features to inspect the linearity assumption. Any non-linear pattern suggests a violation of the linearity assumption. It is also

worth checking if the relationship is consistent for each feature across classes. If non-linear patterns are detected, the features(such as polynomial features) can be transformed, or more flexible models, such as Random Forest or Neural Networks, can be used to capture non-linear relationships better.

Furthermore, the dataset size is relatively small. Splitting 20 percent of 2000 data entries results in a test set that can be too small to evaluate the model's performance accurately.

References

Abhishek Sharma. Mobile Phone price classification. Kaggle.

Available at https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-classification/data?select=train.csv

Appendix

GitHub code

https://github.com/xin4869/mobile_phone_price_predict.git

Dataset description

- Numeric features:

battery_power	Total battery energy in one charge(mAh)	501 - 1998
clock_speed	Microprocessor execution speed	0.5 - 3
fc	Front camera(Megapixels)	0-19
pc	Primary camera(Megapixels)	0-20
int_memory	Internal memory(Gigabytes)	2-64
m_dep	Mobile depth (cm)	0.1 - 1
mobile_wt	Mobile width(cm)	80-200
n_cores	Number of processor cores	1-8
px_height	Pixel resolution height	0-1960
px_width	Pixel resolution width	500-1998
ram	Random access memory (Megabytes)	256-3998
sc_h	Screen size – height(cm)	5-19
sc_w	Screen size – width(cm)	0-18
talk_time	Time that one full charge could last for voice call	2-20

- Binary features:

blue	Has bluetooth or not	0-1
dual_sim	Support double sim car or not	0-1
four_g	Support 4G or not	0-1
three_g	Support 3G or not	0-1
touch_screen	Has touch screen or not	0-1
wifi	Has wifi or not	0-1

- Categorical label

price_range	(Taget variable) categorized mobile phone selling price(0, 1, 2, 3)	0 (low cost), 1(medium) 2(medium-high), 3(high)

Feature correlation matrix

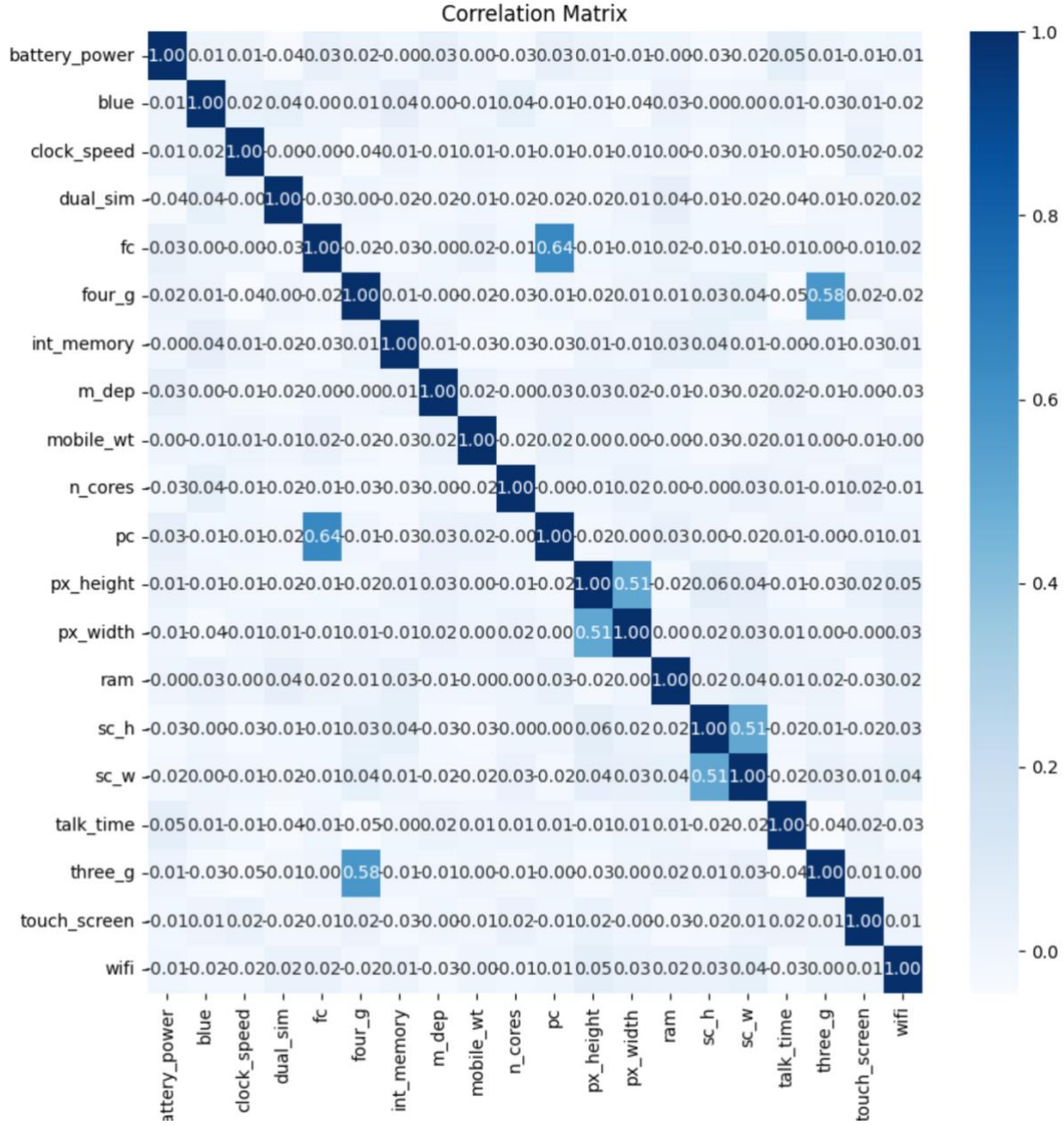


Figure 1

Feature combination

New feature	Original feature
camera	fc + pc
sreen_pixels	Px_width * px_height
screen_size	sc_w * sc_h

Table 1

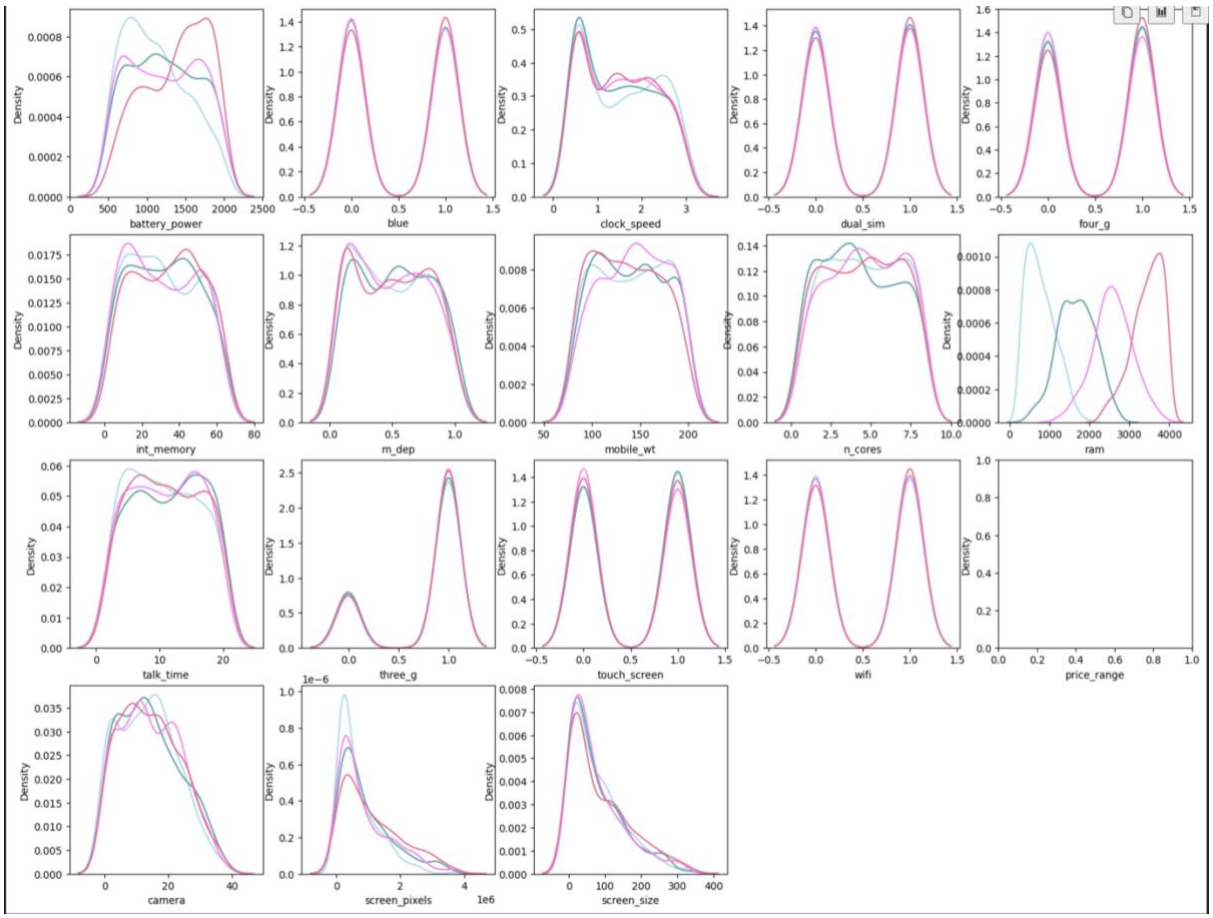

Density
battery_power
blue
clock_speed
dual_sim
four_g
int_memory
m_dep
mobile_wt
n_cores
ram
talk_time
three_g
touch_screen
wifi
price_range
camera
screen_pixels
1e6
1e−6
screen_size

Figure 2