Mobile Price Prediction using different Classification Techniques using Machine Leraning

Name:	**Akash Kumar Singh**
Registration No./Roll No.:	19023
Institute/University Name:	IISER Bhopal
Program/Stream:	BS/DSE
Problem Release date:	February 02, 2022
Date of Submission:	April 24, 2022

1 Introduction

Nowadays mobile phone is one of the most common buying devices. Every day new mobiles with new version and more features are launched. Customers are worried first and wonder "whether he/she can buy a mobile with given requirements or not". So, the main goal of this project is to determine whether a mobile phone having certain specifications will be cheap, moderate, economical and expensive. The data set used for this project has various features of mobile phones. The train data set has 2000 rows and 21 columns (including the target class). The test set has 1000 rows and 20 columns. There are 4 class labels namely 0,1,2,3 in the train set, where 0 is for cheap, 1 is for moderate, 2 is for economical and 3 is for expensive price range respectively.

The columns in the data set are : Battery Power in mAh, the phone has BlueTooth or not, has dual sim support or not, Front Camera Megapixels, Primary Camera Megapixels, Has 4G or not, Internal Memory capacity in GB, mobile Depth in cm, Weight of Mobile in gram, Number of cores, Pixel Resolution height, Pixel resolution width, RAM in MB, screen height in cm, screen width in cm, talk time after a single charge in Hour, has 3G or not, has touch screen or not, has wifi or not, Microprocessor clock speed. The target column is 'price_range'.

In this project i have used Python libraries such as Numpy, Pandas, Scikit-Learn, Matplotlib, Seaborn.

2 Methods

The below subsections represent the step by step work in order to complete this project.

2.1 Pre Processing and EDA

Before applying Classification model and feature extraction I did some pre processing work or exploratory data analysis (EDA) and found that the dataset has no Null and Nan values, training dataset is balanced (i.e each class has equal no. of data points) as shown in Figure 1., all columns in the dataset are numerical and there are no Categorical Values in those columns, found the range of values of each columns in the dataset.

2.2 Feature Observation

In this step I found the Correlation between the features and also with the target variable as shown in Figure 2 by computing Correlation Matrix. Then, With the help of Box plot i found that there is no outlier present in the dataset as shown in Figure 3. I also plotted distributions of some features.

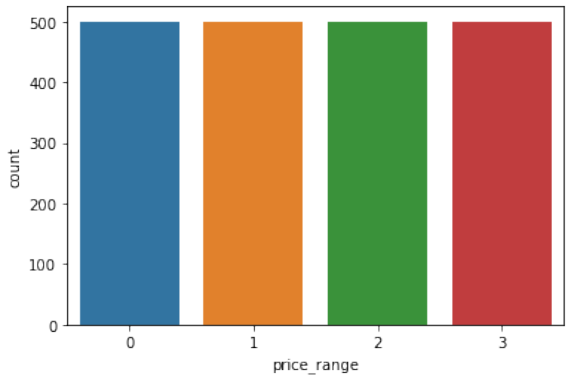

Figure 1: train dataset count

2.3 Classification Models Used & Hyper-Parameter Tuning

In this project I have used some classification models namely : Logistic Regression, Decision Tree, Random Forest, KNN, Ada Boost, Linear Discriminant Analysis (LDA), Gaussian Naive Bayes, Multinomial Naive Bayes, Support Vector Machine (both Linear and Non-Linear), by using Scikit-Learn library. To apply these models on the dataset, I have divided the training dataset into train and validation set in the ratio of 75/25 with the help of train_test_split from scikit learn. Before applying any feature selection and feature scaling, I applied these classification models on the raw dataset along with hyperparameter tuning using GridSearchCV. Hyperparameter tuning is a process of choosing a set of optimal hyperparameters for a learning algorithm which minimizes the loss function on given dataset. This is implemented here by GridSearchCV[1].

2.4 Feature Scaling

Feature Scaling is a technique to normalize the range of features present in the dataset in a fixed range. It is used to handle highly varying values or units. Many distance based algorithms (such as SVM, KNN, logistic regression, etc.) tends to weigh greater values, higher and smaller values as lower, regardless of the unit of the values. To reduce this effect, we need to bring all features to the same level of magnitudes. So, these types of algorithms require features to be normalized. So in this step I applied two feature scaling methods a.) Standardization b.) Normalization with the help of StandardScaler[2] and MinMaxScaler[3] respectively from scikit-learn since we don't know which method will work and which not.

2.5 Feature Selection

Having irrelevant features in the dataset decrease the performance of the models. To overcome this issue i did Feature Selection i.e selecting the best k features among all features present in the dataset. Now in this step I applied two Feature Selection techniques namely Mutual Information and ANNOVA seperately on the dataset obtained after applying the two feature scaling techniques one by one. To get the best k features among the total 20 features available in the train set I used SelectKBest[4] method from scikit-learn and passed the all possible range for k in it using GridSearchCV.

The full code is here on my github repo[5]

[1] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
[2] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
[3] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
[4] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html
[5] https://github.com/AkashSingh215/dsmlproject

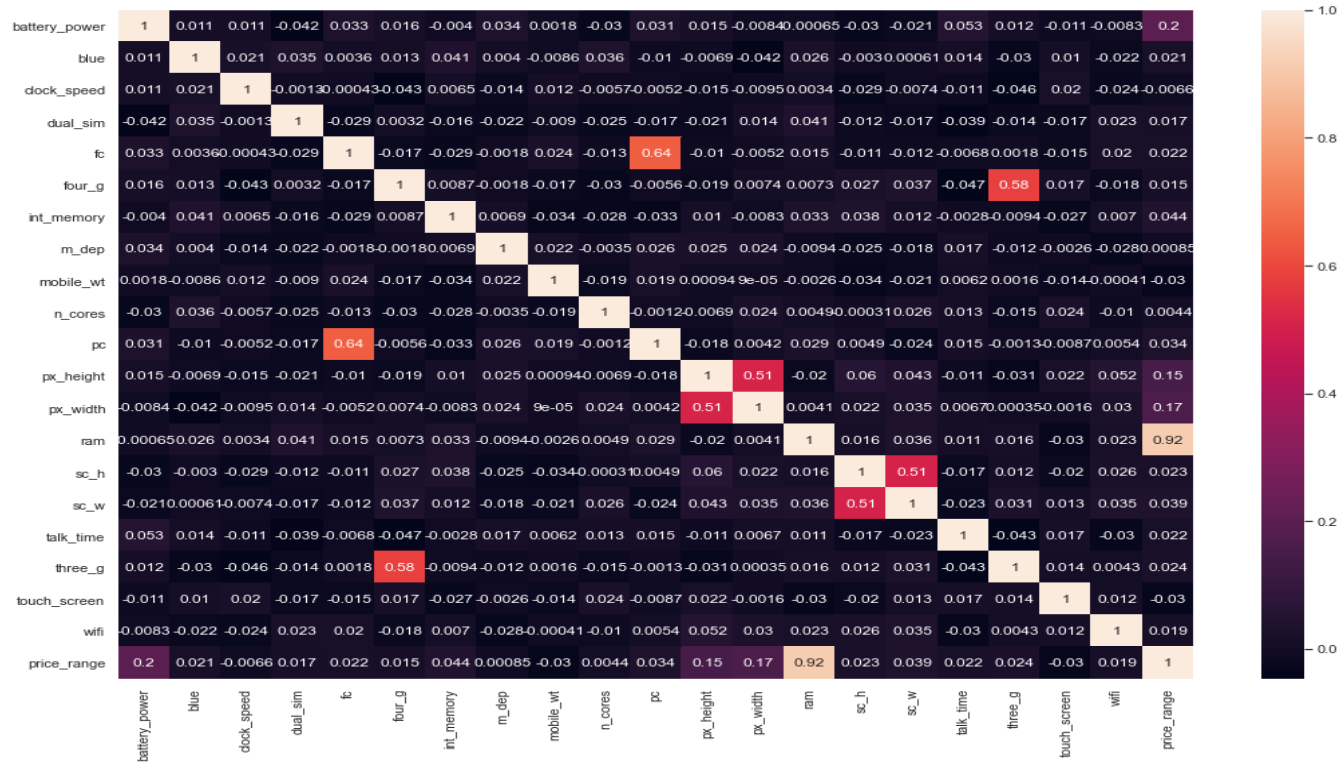

Figure 2: Correlation Matrix

3 Evaluation Criteria

Precision is defined as the number of true positives (Tp) over the number of true positives plus the number of false positives (Fp). $P = Tp/(Tp + Fp)$.

Recall is defined as the number of true positives (Tp) over the number of true positives plus the number of false negatives (Fn). $R = Tp/(Tp + Fn)$.

F1 score is defined as the harmonic mean of precision and recall. $f1 = 2 * P * R/(P + R)$

Micro averaged is defined as sum of true positives for all the classes divided by the sum of all true positives and false positives.

Macro averaged is defined as the arithmetic mean of all the precision scores of different classes.

4 Experimental Analysis:

In this section i have listed the top 5 performing models with best feature selection and scaling techniques. The performance measure of all the methods are here in this google sheet[6]

Model	Feature Selection/Scaling	macro avg. Precision	Recall	f-measure
Logistic Regression	ANOVA/Stand. (k=8)	0.98	0.98	0.98
Logistic Regression	ANOVA/Normali. (k=8)	0.98	0.98	0.98
SVC	ANOVA/Stand. (k=8)	0.98	0.98	0.98
SVC	ANOVA/Normali. (k=8)	0.98	0.98	0.98
Logistic Regression	Raw Data	0.98	0.98	0.98

Table 1: Best 5 Models

[6] https://docs.google.com/spreadsheets/d/1r5yR9jAaB5-9AZccDEXYAdl2Hb-bDVHaHPYi28B0wNI/edit?usp=sharing

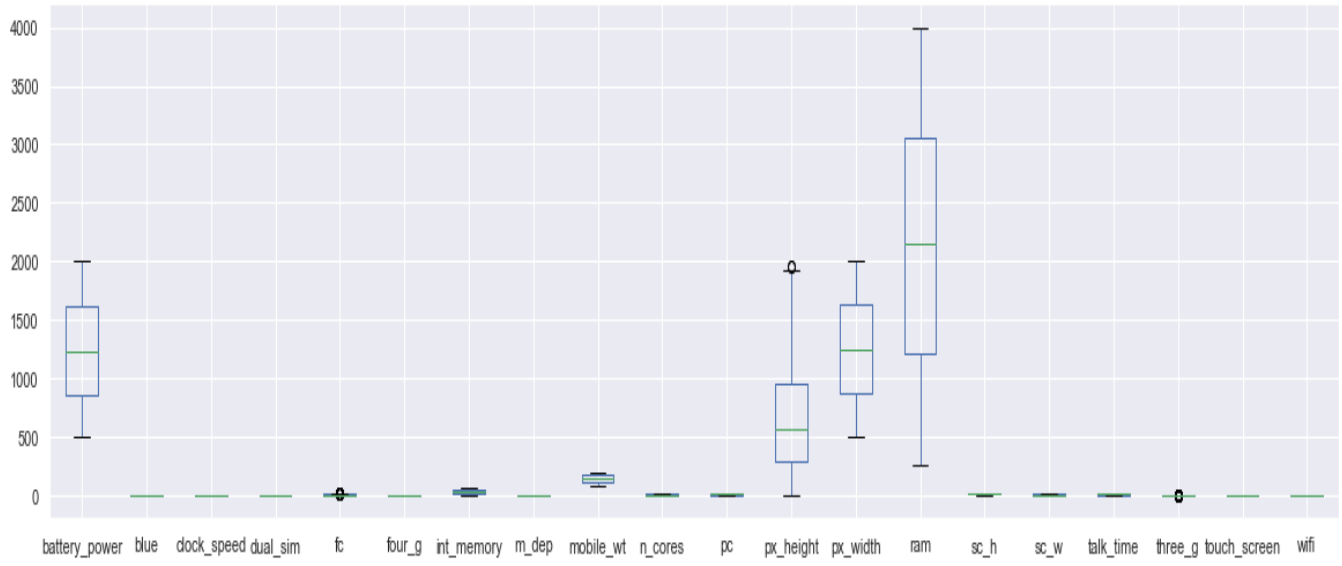

Figure 3: Box Plot

5 Analysis of Results

From the Table 1 I selected the best set of model which is logistic regression with k=8 best feature by ANNOVA and using Standardisation after comparing all sets of methods used based on macro average precision,recall from Classification Report[7] generated in each methods by using scikit-learn. And then run the best set of combinations to predict the class labels of test dataset since the top 5 models have same performance score but the training f1-score is more for logistic regression using ANNOVA and Standardization.

6 Discussions and Conclusion

The performance of the models may further be improved by using more range of hyperparameters and using other classification models which require more computational power.

[7] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html