Mobile Price Prediction

Name:	**Gondikar Anvit Abhay Apu Chakraborty Shivansh Singh**
Registration No./Roll No.:	20118, 20053, 20258
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	January 12, 2023
Date of Submission:	April 16, 2023

1 Introduction

Mobile price prediction is a machine learning project that involves predicting the price range of a mobile device based on certain features. In this project, we have been provided with a data set that contains 20 features and 2000 datapoints. The aim of the project is to build a model that can accurately predict the price range of a mobile device based on these attributes.
After the initial look at the data, the plan of action is to try out some feature selection methods,proceeded by different data normalization techniques. then test those different methods independently with different classifiers to see which combination gives us the best results. I plan to use SVM, Random Forest, K Nearest Neighbors, Logistic Regression and Decision Tree Classifier.

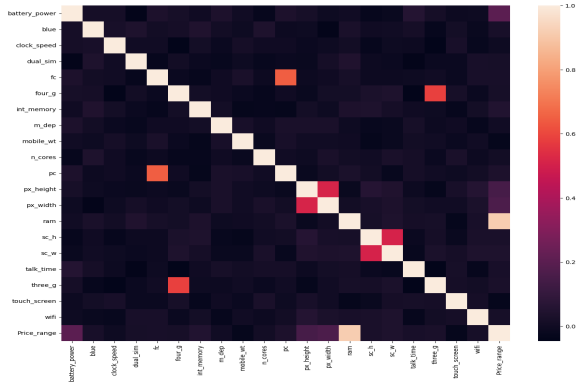

Figure 1: Corelation matrix of data

2 Methods

We have divided our task into three parts
- Data Pre-processing
- Feature Selection
- Training the Classifier

2.1 Data Pre-Processing

The data we have provided with is very balanced data. The data does not contain any categorical values. All the values are in numerical form. So there is no need of data Pre-processing.

2.2 Feature Selection

Now that our data is on the same scale, we can move to feature selection. In our dataset, we have 20 features. While it can be possible that all of them are important, we have to test and see if the results are improved if we select the most important of the features, as suggested by different algorithm. While training the data we will use the trimmed and the original data with all features and see which of them give better results.
As you can see from above co-relation matrix some of the features are nearly independent on target variables.So we have to check whether the dropping out that features is affecting the macro average and F1 score or not.

2.3 Training the Classifiers

As mentioned earlier we have used five Classifiers. Namely Random Forest, Decision Tree, Logistic Regression, Support Vector and KNN. We have used the hyper-parameter tuning for all the above mentioned 5 models.
1) **K - Nearest neighbour** : We have used 'n neighbors': [3, 5, 7, 9, 11], 'weights': ['uniform', 'distance'] these parameters in grid search and we got 'n neighbors': 11, 'weights': 'distance' as best parameter
2) **Support Vector** : We have used 'C': [0.1, 1, 10, 100],
'gamma':['scale', 'auto'],
'kernel': ['linear','poly','rbf','sigmoid']
These parameters in grid search cv to train model and we got 'C': 0.1, 'gamma': 'scale', 'kernel': 'linear'
3) **Decision Tree** : We have used following parameter in grid search cv to train the model 'criterion': ['gini', 'entropy'],
'max depth': [3, 5, 6, 8, 10],
'min samples split': [2, 4, 6, 8, 10],
'min samples leaf': [1, 2, 3, 4, 5]
and we got 'criterion': 'gini', 'max depth': 8, 'min samples leaf': 5, 'min samples split': 4 as best parameter.
4) **Random Forest** : We have used following parameter in grid search cv to train the model 'criterion': ['gini', 'entropy'],
'n estimators': [10, 5, 100],
'max depth': [None, 5, 10],
'min samples split': [2, 5, 10]
and we got 'criterion': 'entropy', 'max depth': None, 'min samples split': 5, 'n estimators': 100 as best parameter.
5) **Logistic Regression** : We have used the following parameter in grid search cv to train the model 'penalty': ['l1', 'l2'],
'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000],
'solver': ['liblinear', 'saga']
out of these we got 'C': 10, 'penalty': 'l1', 'solver': 'liblinear' as best parameter.

3 Evaluation Criteria

We have a classification problem. For classification problem the evaluation criteria are Precision, Recall, F1 score, Macro average and Micro average. Out of that we are using the F1 score and Macro average as a Evaluation criteria. We have used **F1 score** as our Evaluation criteria because it is harmonic mean of Precision and recall.

F measure is the harmonic mean of Precision and Recall.
F Measure =(2 × Precision × Recall)/(Precision + Recall)
The Macro Averaged measure finds the precision and recall score each class from the confusion matrix and then the these scores for all classes are averaged. so, Total Macro Averaged = Macro Averaged

Precision + Macro Averaged Recall.

$$\text{Macro Averaged Precision} = \frac{1}{m}\sum_{i=1}^{m}\frac{TP_i}{TP_i + FP_i} \tag{1}$$

$$\text{Macro Averaged Recall} = \frac{1}{m}\sum_{i=1}^{m}\frac{TP_i}{TP_i + FN_i} \tag{2}$$

4 Analysis of Results

Here we have taken mean of value of F1 score of 4 class labels for each model

Table 1: Performance Of Different Classifiers without Feature Engineering

Classifier	F1 score	Macro average
SVM	0.967	0.97
KNN	0.937	0.94
RF	0.865	0.86
LR	0.880	0.88
DT	0.855	0.85

Table 2: Performance Of Different Classifiers with Feature Engineering

Classifier	F1 score	Macro average
SVM	0.97	0.97
KNN	0.937	0.94
RF	0.89	0.89
LR	0.875	0.87
DT	0.85	0.85

We have done here the Feature Engineering.So first we have run whole models without dropping any columns and got the results as shown above. As we can see from above co relation matrix that some of the features are nearly independent to the target variables such as clock speed, mobile weight and touch screen etc. So we test the model by dropping all features one by one. By dropping the features mobile wt and touch screen we tend to know that F1 score of most of the model decreases so we keep that features although they are independent.
But After dropping the clock speed we got results that value of f1 score and macro average is either remains as it is or increasing for most of the models as we can see from the tables.

For SVM: For parameter kernal = linear
we got the Macro average 0.97 and
'kernel': ['linear','poly','rbf','sigmoid']
we got Macro average 0.97 means by applying these sub-parameters macro average does not changes.

For Random Forest:
Without using criterion we have got the Macro average 0.87 but after applying 'criterion': ['gini', 'entropy'] we got Macro average 0.89. that means with using another parameter macro average increases.

For KNN:
We took n neighbors:[3, 5, 7, 9, 11,13,15] for cross cheaking whether n=13 is a good parameter or not and grid search cv take as best parameter n neighbour=13 but F1 score and Macro Averaged decreases . so we take n neighbours value [3,5,7,9,11] which normally present in sklearn website and we got n neighbour = 11 as best parameter.

5 Discussions and Conclusion

Support Vector Machine is so far the best performing classifier on our dataset. So Support Vector machine as our final model. We have predicted the class label of test data using support vector machine as our model and it predict more number of data getting the class for 0 and 3 as a classlabel for given test dataset.

Advantages of our model

1.Improved decision-making: A mobile price prediction model can help consumers and retailers make more informed decisions about the purchase or sale of mobile devices.

2.Increased efficiency: By predicting the price of mobile devices, retailers can better manage their inventory, adjust their pricing strategies, and optimize their sales and marketing efforts.

Disadvantages of our model

1.Complexity: Developing and implementing a mobile price prediction model can be complex and require significant resources, including data, computing power, and expertise.

Future Improvement

Real-time updates: Mobile price prediction models can be made more useful by providing real-time updates on pricing information, allowing consumers and retailers to make informed decisions on the spot.

6 Reference

•Tanmay Sir Class notes
•https://scikit-learn.org/stable/
•Tutorials of machine learning

7 Github Link

Anvit Gondikar