Mobile Price Prediction using different Classification Techniques using Machine Leraning

Name:	**Saba Ruhsana**
Registration No./Roll No.:	20350
Institute/University Name:	IISER Bhopal
Program/Stream:	Economic Science
Problem Release date:	August 17, 2023
Date of Submission:	November 19, 2023

1 Introduction

The project aims at developing a model to accurately classify diffrent category of mobile phone categorized with respect to their price.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 20 features.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 target column is price range. There is no missing value but there are categorical values.

2 Methods

In this project i used diffrent classifier including kNN , Decision Tree , Logistic Regression, Gaussian Naive Bayes, Random Forest and SVC. We used the following steps :

2.1 2.1 One Hot Encoding

We use this method of one hot encoding to represent categorical values as our data contains categorical value. Categorical featues in our dataset are - bluetooth,dual sim,front camera,four G, pc,three g,touch screen,wifi. I have used one hot encoding for both train data and test data.

2.2 2.2 Train and Test data split

To apply these models on the dataset, I have divided the training dataset into train and validation set in the ratio 60:40 with the help of train test split from scikit learn.

2.3 2.3 HyperParameter Tuning

Before applying any feature selection and feature scaling, I applied these classification models on the raw dataset along with hyperparameter tuning using GridSearchCV.Hyperparameters are used to tune the behaviour of a machine learning algorithm.These are given to the model and initialized prior to training. To carry out By choosing the best values, hyperparameters are adjusted to perform better and enhance the evaluation metric.All possible combinations of the hyperparameters for a given model are utilized to fine-tune it, and the top performers are selected.

2.4 2.4 Classification methods

a)k nearest neighbour For different values of k from 3 to 100 , f measure is found and we choose k=16 as it gives the highest f measure.
b)Decision tree Hyperparameters used - criterion: ('gini', 'entropy'),max features: ('auto', 'sqrt', 'log2', None),max depth: (15, 30, 45, 60)'ccp alpha: (0.009, 0.005, 0.05).Grid search is used to find the values for the criteria, maximum feature, and maximum depth. These values are then held constant while the f-measure is measured for a range of ccp alpha values, from 0 to 0.1. The value of ccp alpha that yielded the maximum f-measure is selected.
c)random forest modelling:Hyperparameters used are:'criterion: ('gini', 'entropy'),n estimators: [int(x) for x in np.linspace (start = 200, stop = 2000, num = 100)],max depth: (10, 20, 30, 50, 100, 200). Values for criterion, max feature, and max depth are found through grid search. The values for these parameters, which gave the maximum value of f measure, is chosen.
d)Gaussian Naive Bayes:Hyperparameter used is:var smoothing:np.logspace(0, -13, num = 100). Value for var smoothing is found through grid search, and the value which gave the maximum value of f measure is chosen.
e) support vector machine: Hyperparameters used are:C: [0.01,0.1,1,10,100],'gamma': [1,0.1,0.01,0.001] kernel:('linear', 'rbf', 'polynomial', 'sigmoid'). Values for 'C', 'gamma', and 'kernel' are found through grid search. The values for these parameters, which gave the maximum value of f measure are chosen
f)logistic regression: Hyperparameters used are:C: np.logspace(-6, 6, num = 50, base = 2), penalty:["l1", "l2", 'elasticnet'], solver:['newton-cg', 'lbfgs', 'liblinear']. Values for 'C', 'penalty', and 'solver' are found through grid search. The values for these parameters, which gave the maximum value of f-measure, are chosen.

2.5 2.5 Improving Hyperparameter Tuning

I fixed other parameters the same to acquire a local maximum, and I adjusted a parameter around the value I obtained from the grid search to enhance performance without adding more values to the grid search.I applied this methodology in the decision tree, SVM, and logistic regression models.We can Perform a grid search to get the best values for the parameters from the inputted values of parameters and also Plot a graph between the value of the chosen parameter and the model's performance around the best value of that parameter we got from the grid search and also finding graph's peak, the corresponding parameter value is considered for that model.

3 Evaluation Criteria

In this project, f1_score and accuracy are considered to evaluate the performance of the models.

- Precision is defined as the ratio of correctly classified positive samples to the total number of classified positive samples.

$$\text{precision} = \frac{\text{No of correctly predicted positive points}}{\text{total predicted positive points}} = \frac{\text{TP}}{\text{TP + FP}}$$

- Recall is the ratio of correctly predicted observations to the all observations in the actual class.

$$\text{Recall} = \frac{\text{No of correctly predicted positive points}}{\text{total actual positive points}} = \frac{\text{TP}}{\text{TP + FN}}$$

- Accuracy is defined as the number of samples correctly predicted to the desired classes divided by the total number os samples in the dataset.

$$\text{Accuracy} = \frac{\text{No of correctly predicted data points}}{\text{total number of data points}} = \frac{\text{TP + TN}}{\text{TP + FP + FN + TN}}$$

- F1 Score is the harmonic mean between Precision and Recall.

$$\text{f1\_score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Since f1_score takes both false positives and false negatives into account, This can be the best criteria to evaluate the perfomance of this model.

4 GitHub link

GitHub link for the python program is:
`https://github.com/SabaRuhsana/MOBILE-PRICE-RANGE-PREDICTION.git`

5 Experimental Setup

State different evaluation criteria e.g., precision, recall etc. that are used to evaluate various models. Mention significant parameters or hyper-parameters of the state of the arts which needs to be tuned, if necessary. The libraries used to implement different models may be noted here. Ideally this section and methods section should not be more than one and half pages long individually.

6 Analysis of Results

Best parameters obtained after tuning are:

- kNN : Value of k = 16.
- Decision tree : ccp_alpha = 0.00425, criterion = 'entropy', max_depth = 15, max_features = None
- Random forest : criterion = 'entropy', max_depth = 100, n_estimators = 279
- gaussian Naive bayes : var_smoothing=3.430469286314912e-05
- svm :C=0.01, class weight='balanced', gamma=1, kernel='linear', probability=True
- logistic regression : C=0.021941347171432164, multiclass='multinomial', solver='newton-cg'

Table 1 shows the accuracy and f measure for all the classification models used in this project

Table 1: Performance Of Different Classifiers Using All Terms

Classifier	Accuracy	F-measure
K-NN	0.85875	0.9328801791350829
Decision tree	0.8625	0.855246013038413
Random forest	0.85375	0.8506651104919882
Gaussian Naive bayes	0.80375	0.8000446600361341
logistic regression	0.9725	0.9716894606706078
SVM	0.97625	0.9755959148492781

7 Discussion and Conclusion

Our analysis reveals that logistic regression yields the highest F-score, indicating its suitability for classifying the test data. Therefore, we have decided to employ logistic regression for the final classification of the test dataset.

7.1 Methods Improvement

To further enhance the performance of our methods, we propose the following improvements:

- Feature Engineering: Implementation of feature engineering can potentially boost efficiency, facilitating the algorithm in detecting patterns within the data more effectively.

7.2 Future Scope

Looking ahead, we envision several possibilities for extending and refining our model:

- Incorporating Additional Features: As technology evolves, new features are continually introduced in mobile phones. By adding these features to the dataset, our model can be adapted to predict mobile prices in the future.

References

We acknowledge the following sources for guidance and data:

- Tutorial codes provided by the tutors.

1 scikit-learn documentation: `https://scikit-learn.org/stable/`

2 Kaggle dataset: `https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-classificatio`