MOBILE PRICE PREDICTION

Name:	**ABHINAV**
Registration No./Roll No.:	21004
Institute/University Name:	IISER Bhopal
Program/Stream:	DSE
Problem Release date:	August 17, 2023
Date of Submission:	November 19,2023

1 Introduction

The aim of this project is to predict a price range for the mobile phones, using given different specifications of various mobiles. The price range can be classified into four categories : cheap (0), moderate (1), economical (2) and expensive (3). Using the features I have to estimate the price range for the mobiles given in the dataset.

The dataset contains 21 attributes in total – 20 features and a class label which is the price range. The features include battery capacity, RAM, weight, camera pixels, etc. The class label is the price range. It has 4 kinds of values – 0,1,2 and 3 which are of ordinal data type representing the increasing degree of price. Higher the value, higher is the price range the mobile falls under. These 4 values can be interpreted as cheap,moderate , economical and expensive. So, despite price traditionally being a numeric problem, the type of ML is classification (not regression) since there are discrete values in the class label.

Number of Training instances = 2000
Number of Test instances = 1000
Number of Features = 20
Number of Classes = 4
There are no Missing values in data set.

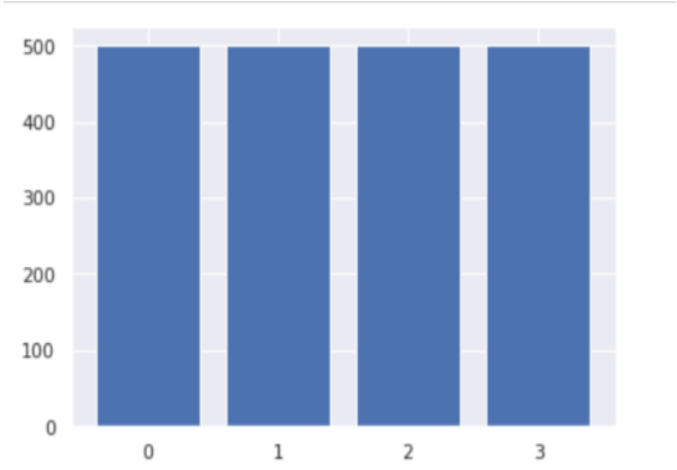


Figure 1: Training instances belong to individual classes

2 Methods

This problem is a classification problem, as the target variable is categorical, having 4 distinct classes to which a new instance may be assigned.[1] The experimentation methodology encompasses training various algorithms on the dataset using k-fold cross-validation and subsequently computing evaluation metrics, including the Confusion Matrix, Precision, Sensitivity, Specificity, and F1-measure for each model iteration. The assessment of performance metrics derived from this cross-validation process serves as the basis for identifying the most fitting model.

Grid search, a widely acknowledged method in machine learning projects, was employed for hyperparameter tuning. This technique systematically explored and evaluated an exhaustive range of hyperparameter combinations to pinpoint the set that yielded superior model performance. This method proved pivotal in optimizing model parameters for heightened predictive accuracy and generalization.

The methodology was founded on the execution of diverse algorithms trained on the dataset using k-fold cross-validation. The assessment involved computing key evaluation metrics such as Confusion Matrix, Precision, Sensitivity, Specificity, and F1-measure for each model iteration. These metrics, derived from the cross-validation process, formed the basis for model selection, allowing the identification of the most suitable model for our objectives.

Prior to training the models, the dataset underwent preprocessing steps to ensure data quality and consistency. One-Hot Encoding was applied to the categorical features (to the 6 columns which are 'wifi', 'four g' ,'dual sim' ,'blue' ,'touch screen' and 'three g'). This encoding method effectively represented categorical data within our dataset. Initially, MinMax scaling was implemented to ensure uniform feature scaling, preventing dominance by larger-magnitude features. However, due to adverse impacts on accuracy, precision, and F1-score, the MinMax scaler was excluded from further use.

The primary goal was to resolve a categorical target variable with four distinct classes using classification methodologies. Specifically, the application focused on leveraging decision trees, K-nearest neighbors (KNN), and Random Forest algorithms. Rigorous comparisons were facilitated using k-fold cross-validation to ascertain the most suitable model. Evaluation metrics, encompassing Confusion Matrix, Precision, Sensitivity, Specificity, and F1-measure, guided the selection process.

Implementation centered on Python, integrating crucial libraries like pandas, scikit-learn, numpy, and matplotlib for robust execution. The code for this project can be found at the link below[1]

3 Experimental Setup

The evaluation criteria employed in assessing the model's performance encompassed four key metrics: F1 measure, accuracy, precision, and recall[2]. F1 measure, the harmonic mean of precision and recall, offers a balanced assessment of a model's ability to simultaneously achieve high precision and recall. Accuracy represents the ratio of correctly predicted instances to the total instances, providing an overall measure of correctness. Precision measures the accuracy of positive predictions, indicating the model's ability to correctly identify true positive instances. Recall, on the other hand, gauges the model's capacity to capture all relevant instances by measuring the ratio of true positive predictions to the total actual positives. Together, these metrics offer a comprehensive evaluation of the model's effectiveness in terms of precision, recall, overall accuracy, and the harmonic balance between precision and recall captured by the F1 measure. In Support Vector Machine (SVM) modeling, several key parameters play a crucial role in shaping its behavior and performance. These parameters include:

- Kernel Selection: The kernel function determines how the data is transformed into a higher-dimensional space, making it easier to separate different classes. Different kernel functions, such as linear, polynomial, radial basis function (RBF), and sigmoid, have different strengths and weaknesses in handling various types of data. For this project, the polynomial kernel proved to be the most effective.

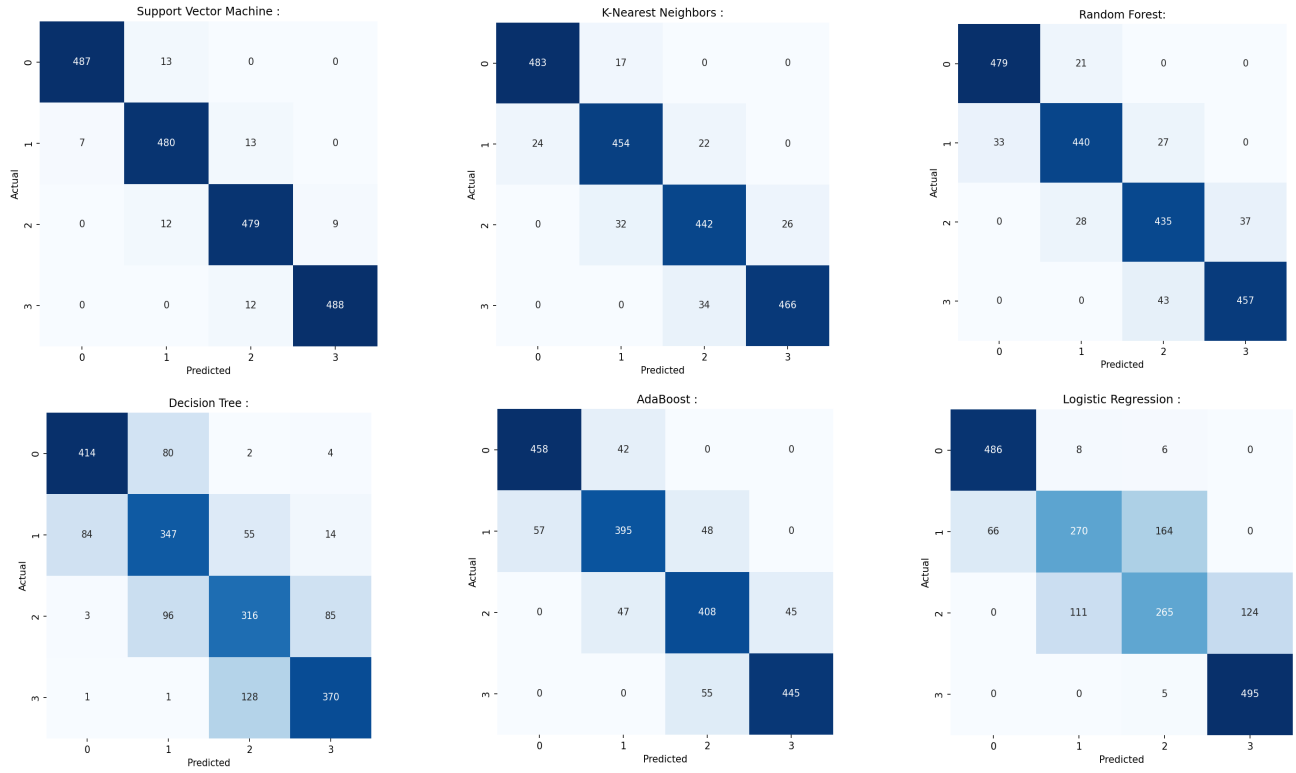
[1] Project Code github link

- Feature Selection: This parameter determines which features are used to train the SVM model. Selecting the right features is crucial for improving the model's performance and generalizability. In this case, using six features yielded the best results.
- C Parameter Calibration: The C parameter balances the trade-off between maximizing the margin and minimizing classification errors. It controls the penalty imposed on misclassified data points during training. A smaller C value leads to a wider margin but may allow more misclassifications. Conversely, a larger C value prioritizes correctly classifying training data points, potentially resulting in a narrower margin. The optimal value of C for this project was found to be 6.

Table 1: Performance Of Different Classifiers Using All Features

Classifier	Accuracy	Precision	Recall	F-measure
Adaptive Boosting	0.85	0.853	0.853	0.852
Decision Tree	0.72	0.717	0.716	0.715
K-Nearest Neighbor	0.92	0.922	0.922	0.922
Multinomial Naive Bayes	0.52	0.501	0.515	0.505
Logistic Regression	0.77	0.755	0.768	0.758
Random Forest	0.91	0.907	0.907	0.90
Support Vector Machine	0.97	0.967	0.967	0.967

Table 2: Confusion Matrix of Different Classifiers



4 Results and Discussion

Metrics used to evaluate the algorithms in this project are confusion matrix, classification report and accuracy score. A confusion matrix has the total count of the accurately grouped occurrences along

its cross and the count of the incorrectly classified instances in the rest of the matrix. We have used 4 class values; so, the matrix generated is a 4*4 matrix.

Different classification algorithms were implemented and their performance assessed using k-fold cross-validation, summarized in Table 1.

The results show that the SVM classifier is the most suitable model for the given problem task. This is likely due to the fact that SVM is a powerful classification algorithm that can handle complex relationships between features and the target variable. The KNN classifier also performed well, and it may be a good choice if computational efficiency is a concern. The other classifiers did not perform as well, and they may not be suitable for this problem task.

5 Conclusion

The SVM classifier is the most suitable model for predicting the price range of mobile phones. This model achieved the highest accuracy, precision, recall, and F-measure among all the classifiers. Future work could focus on improving the performance of the other classifiers by tuning their hyperparameters or using different feature selection techniques.

For future improvements and expansions To enhance accuracy in predicting product prices, utilizing advanced AI methods could be beneficial. Increasing the dataset size by adding more instances and selecting more relevant features can significantly boost accuracy.Therefore, focusing on a larger dataset and choosing better-suited features can lead to higher predictive accuracy.

References

[1] Mustafa Çetın and Yunus Koç. Mobile phone price class prediction using different classification algorithms with feature selection and parameter optimization. In *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 483–487. IEEE, 2021.

[2] Tzu-Tsung Wong and Po-Yang Yeh. Reliable accuracy estimates from k-fold cross validation. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1586–1594, 2019.