Mobile Price Range Prediction

Name:	**Chavan Abhijeet Balaji**
Registration No./Roll No.:	19084
Institute/University Name:	IISER Bhopal
Program/Stream:	BS-MS Economic Sciences
Problem Release date:	August 17, 2023
Date of Submission:	November 19, 2023

1 Introduction

Navigating the rapidly evolving mobile phone market requires precise classification tools to predict price ranges based on product features. Machine learning techniques are increasingly employed in decision-making problems, including price prediction scenarios like mobile phones, where various classification algorithms are applied [1]. This analysis leverages a well-structured dataset with 2000 training instances and 1000 test instances, categorically divided into 'cheap', 'moderate', 'economical', and 'expensive' price ranges. Each class is equally represented in the training data, eliminating the need for complex data balancing techniques. The dataset boasts 20 descriptive features, including battery power, Bluetooth availability, and clock speed, among others.

The robustness of the dataset is evident from the absence of missing values, facilitating a smooth preprocessing phase. The task at hand is a clear-cut classification problem: assigning mobile phones to one of four predefined price categories, distinct from predicting continuous values as in regression problems.

This study extends the initial approach of employing traditional machine learning models like Decision Trees, Logistic Regression, and k-Nearest Neighbors (kNN), to more sophisticated techniques such as Random Forests and Gradient Boosted Trees, examined in Phase I. Additionally, the potential of neural networks to detect intricate data patterns is assessed.

Model performance is primarily gauged by accuracy, reflecting the proportion of correctly predicted instances. However, a deeper dive into precision, recall, and F1-scores allows for a multifaceted evaluation, highlighting the models' precision in class identification and their generalization capabilities. This comprehensive analysis not only benchmarks our models against contemporary techniques but also sets the stage for future advancements in predictive analytics within the mobile industry.

2 Methods

This report explores a range of established machine learning algorithms, from basic to more sophisticated ones, to address our classification problem. We start with Logistic Regression and K-Nearest Neighbors, which are fundamental yet powerful in their simplicity. The study then extends to more complex methods, including Support Vector Machines, Decision Trees, Random Forest, and advanced Gradient Boosting techniques like XGBoost. Each of these models has unique characteristics that make them suitable for different scenarios in data analysis. Following the methodologies applied in contemporary studies, we explored a variety of classification techniques to predict mobile price ranges [2]. n aligning with established methodologies in machine learning for classification tasks, we utilized a range of classifiers including Logistic Regression, kNN, SVM, and ensemble methods [3].

A key part of employing these methods effectively is parameter tuning, a process where we adjust specific settings within each algorithm to optimize performance. This includes not only tweaking algorithmic parameters but also preprocessing steps such as one-hot encoding of binary variables,

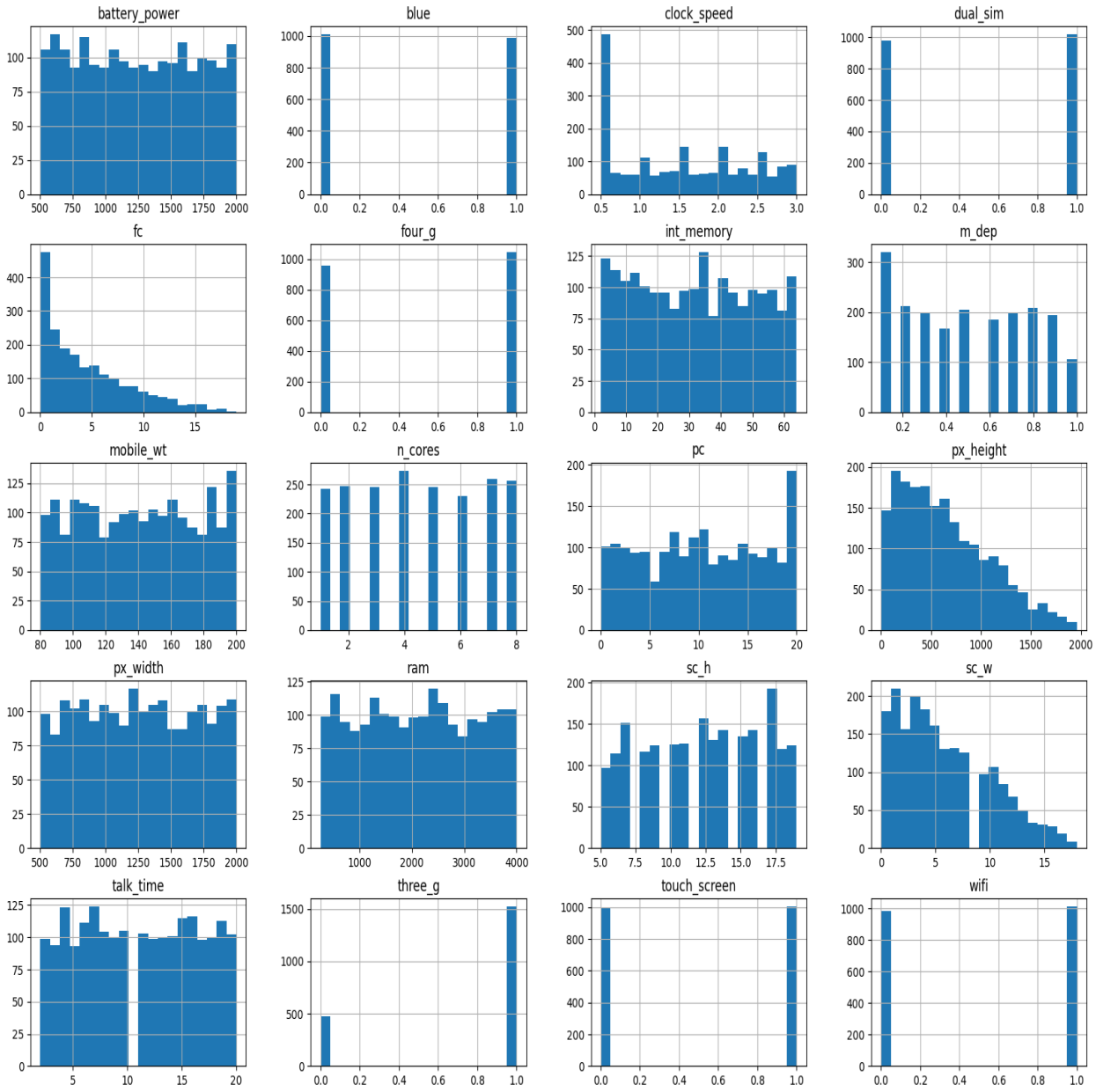

Figure 1: Overview of Data Set

ensuring that each model is finely adapted to our dataset. The incorporation of one-hot encoding is essential for handling categorical features effectively, especially in models where this representation can significantly impact performance. This crucial step of parameter tuning and data preprocessing is guided by a blend of practical experimentation and theoretical insights, ensuring that each model is optimally configured and prepared to handle the intricacies of our dataset.

In addressing the multi-class classification problem inherent in our dataset, Multinomial Logistic Regression was employed as a key analytical method. This model extends the traditional binary logistic regression to handle multiple classes, making it particularly suited for scenarios with more than two outcomes. A core aspect of this approach is the use of the 'multi_class' parameter set to 'multinomial', which adapts the model for multi-class classification by utilizing a softmax function. This function calculates the probabilities of each class over all possible classes, thus providing a more refined classification mechanism. The model was implemented using scikit-learn, a robust and widely-used Python library for machine learning, ensuring a reliable and efficient computational process. Parameter tuning, especially the 'max_iter' parameter, was crucial to ensure model convergence given the complexity of multi-class data. This parameter was set to 1000 iterations, balancing computational efficiency with the need for the algorithm to sufficiently iterate through the optimization process.

In our exploration of classification methods, the k-Nearest Neighbors (kNN) algorithm was employed as a practical and effective model for multi-class classification. The kNN algorithm operates on the premise that similar data points tend to cluster together; thus, it classifies a data point based on the majority class among its nearest neighbors. For our implementation, we leveraged the KNeighborsClassifier from the scikit-learn library. To refine our model, we incorporated a feature selection step using SelectKBest with f_classif to identify the top ten most significant features, thereby reducing dimensionality and potentially enhancing model performance. Subsequently, we conducted a systematic search to find the optimal number of neighbors by evaluating the cross-validated accuracy for different values of k. This was achieved using cross_val_score which provided a robust estimate of model performance. The optimal k, determined to be 11, was the number of neighbors that resulted in the highest cross-validation accuracy, balancing the trade-off between overfitting and underfitting. The final model was then trained with this optimal parameter, ensuring it was finely tuned to the characteristics of our dataset.

In our exploration of machine learning techniques for classification, the Support Vector Machine (SVM) was a key method employed. SVM is a powerful and versatile algorithm known for its ability to handle both linear and non-linear classification through the use of different kernels. In our implementation, the SVC classifier from the scikit-learn library was utilized, with a linear kernel initially chosen for its simplicity and effectiveness in linearly separable data scenarios. The strength of SVM lies in its use of a hyperplane to separate different classes, maximizing the margin between various data points. Hyperparameter tuning played a crucial role in optimizing the SVM model, particularly the parameters 'C', 'gamma', and 'kernel'. 'C' controls the trade-off between smooth decision boundaries and classifying training points correctly, while 'gamma' defines the influence of individual training samples. The choice of kernel (linear, RBF, polynomial) alters the model's capacity to handle complex, non-linear decision boundaries. GridSearchCV was employed to systematically explore a range of parameter combinations, ensuring the selection of optimal parameters for the best classification performance.

In our suite of machine learning models, the Random Forest classifier stands out for its robustness and ability to handle complex classification tasks. As an ensemble learning method, it operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes of the individual trees. This approach inherently mitigates the risk of overfitting, a common challenge with single decision trees. For our implementation, we used the RandomForestClassifier from scikit-learn, leveraging its efficiency and versatility. Hyperparameter tuning, crucial in optimizing the performance of the Random Forest model, was conducted using GridSearchCV. This involved systematically varying parameters such as 'n_estimators', 'max_depth', 'min_samples_split', and 'min_samples_leaf' to identify the most effective combination. The aim was to find a balance between the model's complexity and its generalization ability, ensuring robust performance across diverse data scenarios.

The Decision Tree classifier was selected as part of our machine learning approach due to its interpretability and ease of use. Decision Trees work by splitting the data into subsets based on feature value conditions, forming a tree-like structure of decisions. In our implementation, the DecisionTreeClassifier from scikit-learn was employed. A key strength of Decision Trees is their transparency in decision-making, allowing clear understanding of how predictions are made. To enhance the model's performance, feature importance was evaluated, identifying the most impactful features in the classification task. Based on these importances, we conducted a feature selection process, retaining only the most significant features as determined by a predefined threshold(¿0.05). This step aimed to simplify the model and potentially improve its generalization capability.

In our exploration of advanced machine learning algorithms, XGBoost (eXtreme Gradient Boosting) played a significant role. Recognized for its efficiency and effectiveness in classification tasks, XGBoost is a sophisticated ensemble technique that builds decision trees sequentially, where each tree corrects the errors of its predecessor. This method stands out for its ability to handle a wide range of data types and distributions effectively. In our study, XGBoost was utilized with its key parameters like 'max_depth', 'eta' (learning rate), and 'num_class' carefully selected to optimize its performance for multi-class classification. The dataset was converted into DMatrix, a data structure optimized for

Table 1: Performance Of Different Classifiers Using All Features

Classifier	Precision	Recall	F-measure
Logistic Regression	0.97	0.96	0.96
K-Nearest Neighbor	0.68	0.68	0.68
Support Vector Machine	0.98	0.98	0.97
Decision Tree	0.88	0.88	0.88
Random Forest	0.89	0.90	0.89
Adaptive Boosting	0.91	0.91	0.91

Table 2: Confusion Matrix of Logistic Regression

	Predicted Class			
Actual Class	Cheap	Moderate	Economical	Expensive
Cheap	98	2	0	0
Moderate	1	96	3	0
Economical	0	2	94	4
Expensive	0	0	2	98

XGBoost, enhancing computational efficiency. Training the model involved setting the 'objective' to 'multi:softmax', ideal for multi-class scenarios, and tuning the 'num_boost_round' parameter to control the number of boosting iterations.

3 Experimental Setup

To evaluate and compare the performance of our selected machine learning models, we relied on standard and widely accepted metrics, including precision, recall, and the F1-score. These metrics provide a comprehensive view of each model's accuracy and its ability to handle various aspects of classification tasks. Precision offers insights into the accuracy of positive predictions, recall assesses the model's ability to identify all relevant instances within a class, and the F1-score serves as a harmonic mean of the two, providing a balanced evaluation metric. This methodical approach to model evaluation, utilizing reputable libraries like scikit-learn and XGBoost, ensures that our findings are both reliable and relevant to the classification challenges at hand.

The experimental setup for all models involved splitting the dataset into training and validation sets, ensuring a robust assessment of each model's generalization capabilities. This standardized approach allowed for a fair and thorough evaluation across different models, highlighting their strengths and areas for improvement in multi-class classification tasks.

Specifically, the Multinomial Logistic Regression model's effectiveness was evaluated in a multi-class setting, providing a comprehensive understanding of its performance. Similarly, the kNN model's simplicity, coupled with these metrics, offered valuable insights into its suitability for the classification task. The SVM model's evaluation followed a structured approach, allowing for a detailed assessment of its performance. For the Random Forest and Decision Tree classifiers, the focus was on their capacity to handle multi-class classification and the significance of individual features. Lastly, the XGBoost model's evaluation highlighted its potential in addressing complex classification challenges with high accuracy.

Each model's performance was critically analyzed using these metrics, contributing to a holistic understanding of their applicability and effectiveness in our specific multi-class classification context.

4 Results and Discussion

The performance metrics of various classifiers are consolidated in Table 1. Logistic Regression and Support Vector Machine (SVM) classifiers exhibit superior performance, achieving the highest preci-

Table 3: Confusion Matrix of K Nearest Neighbour

	Predicted Class			
Actual Class	Cheap	Moderate	Economical	Expensive
Cheap	83	17	0	0
Moderate	17	59	24	0
Economical	1	26	53	20
Expensive	0	0	25	75

Table 4: Confusion Matrix of Support Vector Machines

	Predicted Class			
Actual Class	Cheap	Moderate	Economical	Expensive
Cheap	98	2	0	0
Moderate	1	96	3	0
Economical	0	1	97	2
Expensive	0	0	1	99

Table 5: Confusion Matrix of Decision Trees

	Predicted Class			
Actual Class	Cheap	Moderate	Economical	Expensive
Cheap	95	5	0	0
Moderate	7	83	10	0
Economical	0	9	84	7
Expensive	0	0	11	89

Table 6: Confusion Matrix of Random Forest

	Predicted Class			
Actual Class	Cheap	Moderate	Economical	Expensive
Cheap	97	3	0	0
Moderate	6	83	11	0
Economical	0	13	83	4
Expensive	0	0	5	95

Table 7: Confusion Matrix of Gradient Boosting (XGBoost)

	Predicted Class			
Actual Class	Cheap	Moderate	Economical	Expensive
Cheap	97	3	0	0
Moderate	6	87	7	0
Economical	0	9	88	3
Expensive	0	0	9	91

sion, recall, and F-measure scores of approximately 0.97. In contrast, the K-Nearest Neighbor (KNN) model shows a significantly lower performance, with all scores around 0.68. Decision Tree and Random Forest classifiers present moderately high scores, around 0.88 and 0.89, respectively, while Adaptive Boosting achieves a score of 0.91 across all metrics.

Table 2 through Table 7 detail the confusion matrices for each classifier. Logistic Regression and SVM, consistent with their high overall scores, show a strong ability to correctly predict the 'Cheap' and 'Expensive' classes, with the SVM model slightly outperforming Logistic Regression in the 'Expensive' class. The KNN classifier struggles with the 'Moderate' and 'Economical' classes, as reflected by the higher misclassification rates observed in Table 3. Decision Trees and Random Forest exhibit a balanced performance across all classes but with some misclassifications in the 'Moderate' and 'Economical' categories. Notably, the Gradient Boosting (XGBoost) model demonstrates a robust ability to classify the 'Economical' class, with fewer misclassifications compared to other models, as seen in Table 7. Our results, much like those found in Pramanik et al., indicate a variance in classifier performance, with metrics like accuracy, precision, recall, and F1-Score offering insights into each model's strengths and weaknesses [3].

The merits of Logistic Regression and SVM are evident in their high precision and recall scores, indicating reliable predictability and a strong fit to the dataset. KNN's limitations are highlighted by its lower scores, potentially due to its sensitivity to the choice of k and the curse of dimensionality. Decision Trees provide transparency in model decisions but can overfit without careful tuning. Random Forest mitigates this overfitting and improves predictive power, while Adaptive Boosting further refines performance by focusing on misclassified instances. XGBoost's strength is its efficiency and accuracy, particularly for complex datasets, but it requires careful tuning to avoid overfitting.

5 Conclusion

Our study has elucidated that among the classifiers tested, Multinomial Logistic Regression and Support Vector Machines (SVM) have excelled, demonstrating a particularly strong performance with an accuracy of 0.96. This outcome suggests their potential aptitude for effectively navigating the complexities of high-dimensional data spaces, making them valuable tools for classification tasks that involve rich feature sets. The study's conclusions are in agreement with recent findings, where SVM, and Logistic Regression classifiers have shown promising accuracy in mobile price classification tasks [4]. Furthermore, the commendable performance of ensemble techniques such as Random Forest and Gradient Boosting methods, notably XGBoost with an accuracy of 0.91, underscores their capability in managing the intricacies of multi-class classification challenges. These results significantly enhance our comprehension of the operational characteristics and strengths of various classifiers within multi-class domains. Our findings resonate with [3], who found SVM to be highly effective for high-dimensional data, suggesting potential applications in mobile app or website integrations for price prediction.

Looking forward, there is a promising avenue for integrating ensemble methods with sophisticated feature engineering strategies to potentially elevate model performance even further. The applicability of these classifiers across different fields and datasets beckons for further exploration, which could shed light on their versatility and adaptability. For KNN, which lagged in accuracy at 0.68, delving into feature selection and dimensionality reduction might offer avenues for improvement. These directions not only pave the way for advancing classifier technology but also open doors to innovative applications that could transform data analysis across various sectors.

References

[1] Pritish Arora, Sudhanshu Srivastava, and Bindu Garg. Mobile price prediction using weka. 2020.

[2] Muhammad Asim and Zafar Khan. Mobile price class prediction using machine learning techniques. *International Journal of Computer Applications*, 179(29), 2018.

[3] Rwittika Pramanik, Rakshit Agrawal, Mahendra Kumar Gourisaria, and Pradeep Kumar Singh. Comparative analysis of mobile price classification using feature engineering techniques. In *2021*

5th International Conference on Information Systems and Computer Networks (ISCON), pages 1–7, 2021.

[4] Keval Pipalia and Rahul Bhadja. Performance evaluation of different supervised learning algorithms for mobile price classification. *International Journal for Research in Applied Science Engineering Technology (IJRASET)*, 8, 2020.