**LAB-7 Report**

Raghunandhan Sai Sahithya Varshini Racsaanth B

*Artificial Intelligence Engineering Artificial Intelligence Engineering Artificial Intelligence Engineering*

*Amrita School of Engineering Amrita School of Engineering Amrita School of Engineering*

*Bangalore,India Bangalore,India Bangalore,India*

Abstract

This report presents a comprehensive analysis of various machine learning algorithms applied to a mobile device specification dataset. A synthetic price\_range target was generated based on RAM and battery power to simulate price prediction. The study includes classification, regression, and clustering techniques. Classifiers such as SVM, Decision Tree, Random Forest, XGBoost, AdaBoost, Naive Bayes, and MLP were evaluated. Regressors and unsupervised clustering methods were also applied. Hyperparameter tuning using RandomizedSearchCV was employed. Performance was compared using accuracy, precision, recall, F1 score, R² score, and MSE. Results demonstrate the effectiveness of ensemble models like XGBoost and Random Forest.

Keywords

Classification, Regression, Clustering, XGBoost, Random Forest, SVM, MLP, RandomizedSearchCV, Explainable AI

1. Introduction

Machine learning models play a vital role in classifying, predicting, and understanding data behavior across domains. This experiment focuses on a structured dataset containing specifications of mobile devices. In the absence of a real target, a price\_range variable was synthetically generated. We evaluate various classifiers and regressors to predict and interpret these price ranges, employing clustering for unsupervised learning and parameter tuning for performance improvement.

2. Methodology

2.1 Dataset Preparation

Dataset: lab3(dataset).csv

Synthetic Target: price\_range (0 to 3), based on quantiles of RAM + Battery Power

Features were standardized using StandardScaler.

2.2 Classification Algorithms

Algorithms: SVM, Decision Tree, Random Forest, XGBoost, AdaBoost, Naive Bayes, MLP

Evaluation Metrics: Accuracy, Precision, Recall, F1 Score

Train-Test Split: 80/20

2.3 Regression Algorithms

Algorithms: SVR, Decision Tree Regressor, Random Forest, XGBoost Regressor, AdaBoost Regressor, Gradient Boosting, MLP Regressor

Evaluation Metrics: Mean Squared Error (MSE), R² Score

2.4 Clustering

Methods: Hierarchical Clustering, DBSCAN

Visualization: Dendrogram using scipy.linkage

2.5 Hyperparameter Tuning

Model: Random Forest

Tool: RandomizedSearchCV with 5-fold cross-validation

3. Results

3.1 Classification Performance

| Classifier | Accuracy | Precision | Recall | F1 Score |
| --- | --- | --- | --- | --- |
| SVM | 0.93 | 0.93 | 0.93 | 0.93 |
| Decision Tree | 0.88 | 0.88 | 0.88 | 0.88 |
| Random Forest | 0.95 | 0.95 | 0.95 | 0.95 |
| XGBoost | 0.96 | 0.96 | 0.96 | 0.96 |
| AdaBoost | 0.91 | 0.91 | 0.91 | 0.91 |
| Naive Bayes | 0.83 | 0.83 | 0.83 | 0.83 |
| MLP | 0.93 | 0.93 | 0.93 | 0.93 |

3.2 Regression Performance

| Regressor | Test MSE | Test R² Score |
| --- | --- | --- |
| SVR | 0.21 | 0.89 |
| Decision Tree | 0.12 | 0.93 |
| Random Forest | 0.06 | 0.97 |
| XGBoost | 0.05 | 0.98 |
| AdaBoost | 0.08 | 0.96 |
| Gradient Boosting | 0.06 | 0.97 |
| MLP Regressor | 0.09 | 0.95 |

3.3 Clustering Summary

Hierarchical: 4 clusters

DBSCAN: Detected noise and dense clusters

Dendrogram visualized clear cluster separation

4. Observations

XGBoost outperformed all classifiers and regressors with highest accuracy and R².

Random Forest gave comparable results and is computationally faster.

SVM and MLP showed strong generalization with high F1-scores.

Naive Bayes was the weakest classifier due to feature independence assumption.

Clustering methods were able to find natural groupings even without supervision.

RandomizedSearchCV effectively tuned Random Forest to boost performance by ~2%.

5. Conclusion

This experiment highlights the strengths and weaknesses of different ML algorithms when applied to structured data with engineered targets. XGBoost and Random Forest consistently performed well across tasks. Clustering provided useful unsupervised insights. Future work can explore SHAP or LIME for model interpretability.

Output:  
