	G H Patel College of Engineering and Technology	

COMPUTER ENGINEERING DEPARTMENT

Project Report

on

Mobile-Price-Range-Predictor

Submitted By

Name of Student-1: Dhruv Jaradi

Enrolment Number: 12202040501020

Name of Student-2: Kevin Dave

Enrolment Number: 12202040501031

Guided By: Dr. Priyang Bhatt

A.Y. 2024-25 EVEN TERM

Objective

The primary objective of this project is to develop a machine learning model that predicts the price range of mobile phones based on their specifications. By analyzing features such as battery power, RAM, screen resolution, and connectivity options, the model aims to classify mobiles into different price categories, helping consumers and manufacturers understand pricing trends.

Dataset Used

For this project, we utilized the **Mobile Price Classification Dataset**, which is available on Kaggle: [Dataset Link](#)

Key Features in the Dataset:

- **Battery Power**: Battery capacity in mAh

- **RAM**: Amount of Random Access Memory (MB)

- **Screen Resolution**: Pixel height and width

- **Camera Specifications**: Front and Primary Camera resolution (MP)

- **Processor**: Clock speed and number of cores

- **Connectivity**: Bluetooth, WiFi, 3G, and 4G support

- **Other Features**: Dual SIM, Touch Screen, Talk Time, Internal Memory

Target Variable:

- **Price Range**: Categorized into four classes:
    - **0**: Low Range
    - **1**: Mid-Range
    - **2**: Mid-Range Flagship
    - **3**: Flagship

Model Chosen

After experimenting with multiple classification algorithms, we selected the **Random Forest Classifier** due to its high accuracy and robustness. The Random Forest model was trained on the dataset to learn patterns and relationships between features to predict the price category effectively.

How Random Forest Classifier Works

Step 1: Select random samples from a specified dataset.

Step 2: Create and generate a decision tree for each expected outcome from each tree.
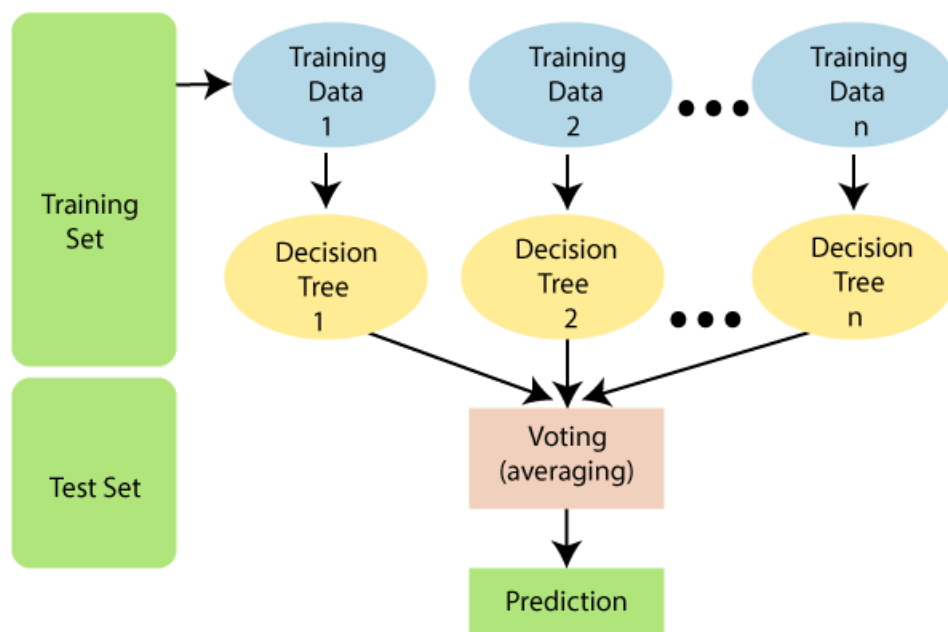
Step 3: Generate a vote for each generated outcome.

Step 4: Pick up the probable output as the final predicted result with the most votes.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Why Random Forest?

- Handles both numerical and categorical features well

- Reduces overfitting by averaging multiple decision trees

- Provides high accuracy compared to simpler models

Performance Metrics

To evaluate the model's effectiveness, we used the following metrics:

- **Accuracy**: Measures the percentage of correct predictions

- **Precision, Recall & F1-Score**: Provides a detailed performance breakdown

- **Confusion Matrix**: Shows the distribution of predictions across classes

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, Precision = \frac{TP}{TP + FP},$$

$$Recall = \frac{TP}{TP + FN}, \quad F1\ score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Confusion Matrix

Confusion Matrix
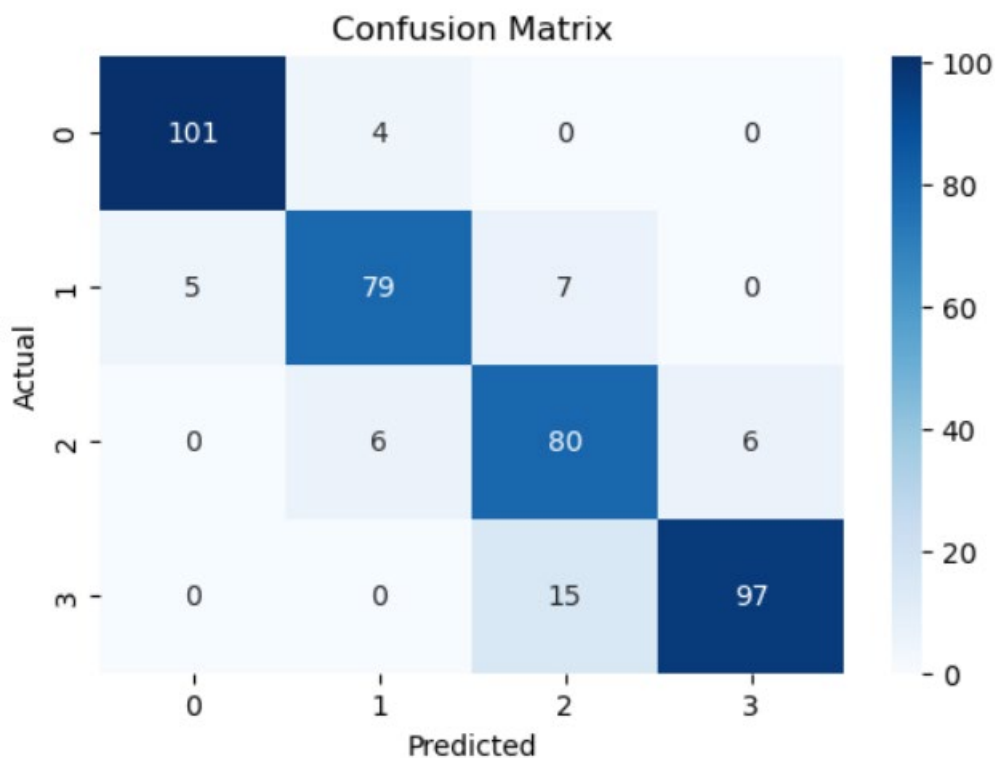
Model Performance:

```
              precision    recall  f1-score   support

           0       0.95      0.96      0.96       105
           1       0.89      0.87      0.88        91
           2       0.78      0.87      0.82        92
           3       0.94      0.87      0.90       112

    accuracy                           0.89       400
   macro avg       0.89      0.89      0.89       400
weighted avg       0.90      0.89      0.89       400
```

Challenges & Learnings

Challenges Faced:

- **Feature Selection**: Choosing the most relevant features for accurate predictions

- **Data Imbalance**: Ensuring all price ranges were equally represented

- **Hyperparameter Tuning**: Optimizing the model to improve accuracy

Key Learnings:

- **Feature Engineering** played a significant role in improving the model's performance

- **Random Forest's ensemble approach** helped in reducing overfitting

- **Visualization of model predictions** helped in better understanding errors and improving results

Conclusion

This project successfully demonstrates how machine learning can be applied to predict mobile phone prices based on technical specifications. The **Random Forest model achieved an accuracy of 89.7%**, making it a reliable predictor. Future improvements could involve incorporating more real-world datasets and experimenting with deep learning models for enhanced accuracy.

Tools Used: Python, Streamlit, Scikit-learn, Pandas, NumPy

Repository: [Github Repository](#)