Project Title: Mobile Price Classification Using Machine Learning

Project Overview:

The goal of this project is to classify mobile phones into different price ranges based on features such as RAM, battery power, camera specifications, and more. Bob, the founder of a new mobile company, needs a model to help estimate the price category of the phones his company creates. The dataset includes features like RAM, battery power, camera specifications, and other technical attributes, with a target variable price_range indicating the price category (0: low cost, 1: medium cost, 2: high cost, 3: very high cost).

Description:

This project tackles the problem of classifying mobile phones into price ranges based on various features such as hardware specifications and screen dimensions. The dataset consists of 2000 rows and 21 features. The task is approached as a multi-class classification problem, with the target variable price_range consisting of four classes. Data preprocessing steps, including feature renaming, cleaning, and exploratory data analysis (EDA), were performed to prepare the data. Several machine learning models were tested, including Logistic Regression, Random Forest, K-Nearest Neighbors (KNN), and Naive Bayes. Feature engineering techniques, such as creating interaction variables and applying standardization and normalization, were used to enhance model performance. After evaluating the models, Random Forest was selected as the best performer, achieving 89% accuracy on the test set. The trained model was then used to predict the price range of mobile phones in a prediction dataset.

Key Insights and Steps:

1. **Data Import and Preprocessing:**

   - The dataset was loaded into a Pandas DataFrame, and initial inspection revealed that there were no missing values or data type issues.
   - Columns were renamed for better readability, and summary statistics were generated to inspect the data.

2. **Exploratory Data Analysis (EDA):**

   - Target Variable: The target variable price_range had a balanced distribution across all classes, which is ideal for classification.
   - Univariate Analysis: Key numerical features like Battery Power, RAM, and Internal Memory exhibited uniform distributions, while camera-related features showed skewness.
   - Bivariate and Multivariate Analysis: Higher price ranges were associated with better specifications such as higher RAM, battery power, and camera quality. The correlation analysis showed that RAM had the strongest positive correlation with price_range.

3. **Modeling:**

   - Four models were tested: Logistic Regression, Random Forest, K-Nearest Neighbors (KNN), and Naive Bayes.
   - Feature engineering, including the creation of interaction variables like Pixels and Diagonal (Screen Size), improved model performance.

- Model Results: Random Forest outperformed other models with an accuracy of 86.6% during cross-validation, and 89% accuracy on the test set.
- The confusion matrix showed that the model performed well in distinguishing between extreme price categories (Class 0 and Class 3), but some confusion was observed between adjacent classes (Class 1 ↔ Class 2, Class 2 ↔ Class 3).
- Final Predictions: The Random Forest model was used to predict the price ranges for an unseen prediction dataset, showcasing the model's applicability in real-world scenarios.

Part 1: Data Import and Preprocessing

1. Library Imports

The following Python libraries were used for data manipulation, analysis, and visualization:

- **pandas**: For data handling and analysis
- **numpy**: For numerical computations
- **matplotlib** and **seaborn**: For creating data visualizations

2. Data Loading and Initial Inspection

The dataset was downloaded from Kaggle and loaded into a Pandas DataFrame. To understand its structure and quality:

- **.head()** was used to preview the first five rows of the dataset.
- **.info()** provided insights into the data types and the presence of any missing values.
- **Result:**
  - The dataset consists of **2000 rows** and **21 columns**.
  - No missing values were found, and all data types matched the expected format.

3. Shape and Statistical Overview

- **.shape:** Confirmed 2000 rows and 21 columns.
- **.describe():** Generated summary statistics:
  - Minimum, maximum, mean, and quartiles were within logical ranges for all features.
  - No unusual outliers were detected at this stage.

4. Column Renaming

To enhance clarity and readability, column names were renamed. For example:

- 'blue' → 'Bluetooth Support'
- 'dual_sim' → 'Dual SIM Support'
- 'clock_speed' → 'Clock Speed'
- 'fc' → 'Front Camera'

- ‘four_g’ → ‘4G Support’

Other columns were renamed similarly for better understanding. This ensures that the dataset is user-friendly and ready for further exploratory data analysis (EDA).

Conclusion:
The dataset is clean, with no missing values, logical data types, and intuitive column names. It is now prepared for EDA and model building.

Part 2: Exploratory Data Analysis (EDA)

1. Target Variable Analysis

The target variable, price_range, represents the price categories of mobile phones, ranging from 0 (low cost) to 3 (very high cost). Using the .value_counts() function, I examined the distribution of the target classes. All classes were found to have equal counts, indicating a perfectly balanced dataset. This balance simplifies the classification process as no resampling techniques are required. A bar plot was created to visualize the distribution.

2. Univariate Analysis

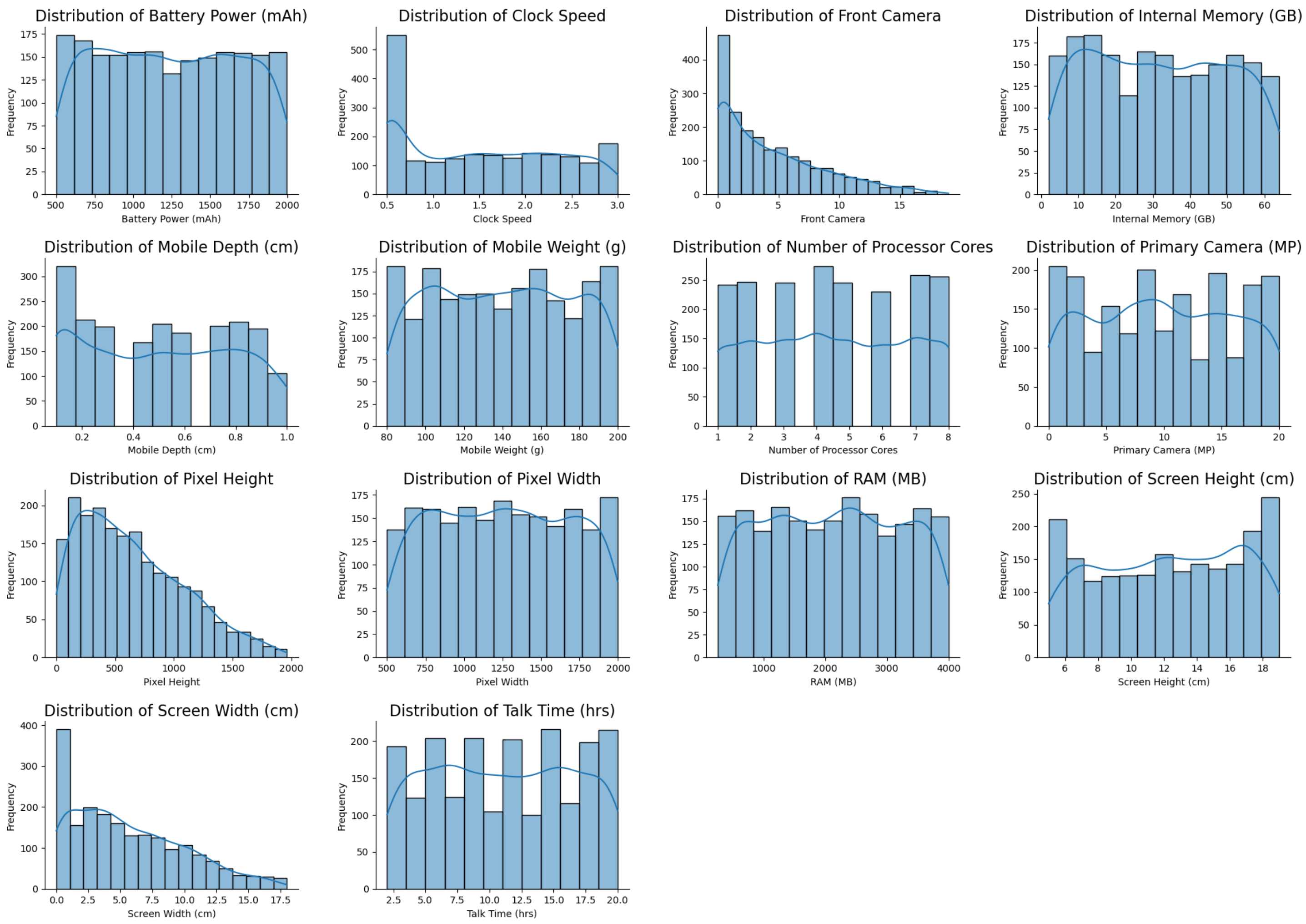
Numerical Features

To understand the distribution of numerical features, I performed a detailed analysis using histograms and Kernel Density Estimate (KDE) plots. Key insights include:

- **Battery Power, RAM, Internal Memory:** These features exhibit uniform distributions, reflecting diverse values across the dataset.
- **Front Camera, Pixel Height, Pixel Width:** The data is skewed toward lower values, indicating that most devices have modest specifications in these categories.
- **Clock Speed, Mobile Depth:** These features show irregular patterns, potentially highlighting variability in their inclusion across devices.
- **Processor Cores, Talk Time:** Discrete values with balanced distributions were observed.
- **Mobile Weight, Screen Dimensions:** These features have a moderate spread, with no extreme concentrations.

KDE overlays provided additional clarity on the spread and concentration of values.

Distribution analysis of numerical features with KDE overlay



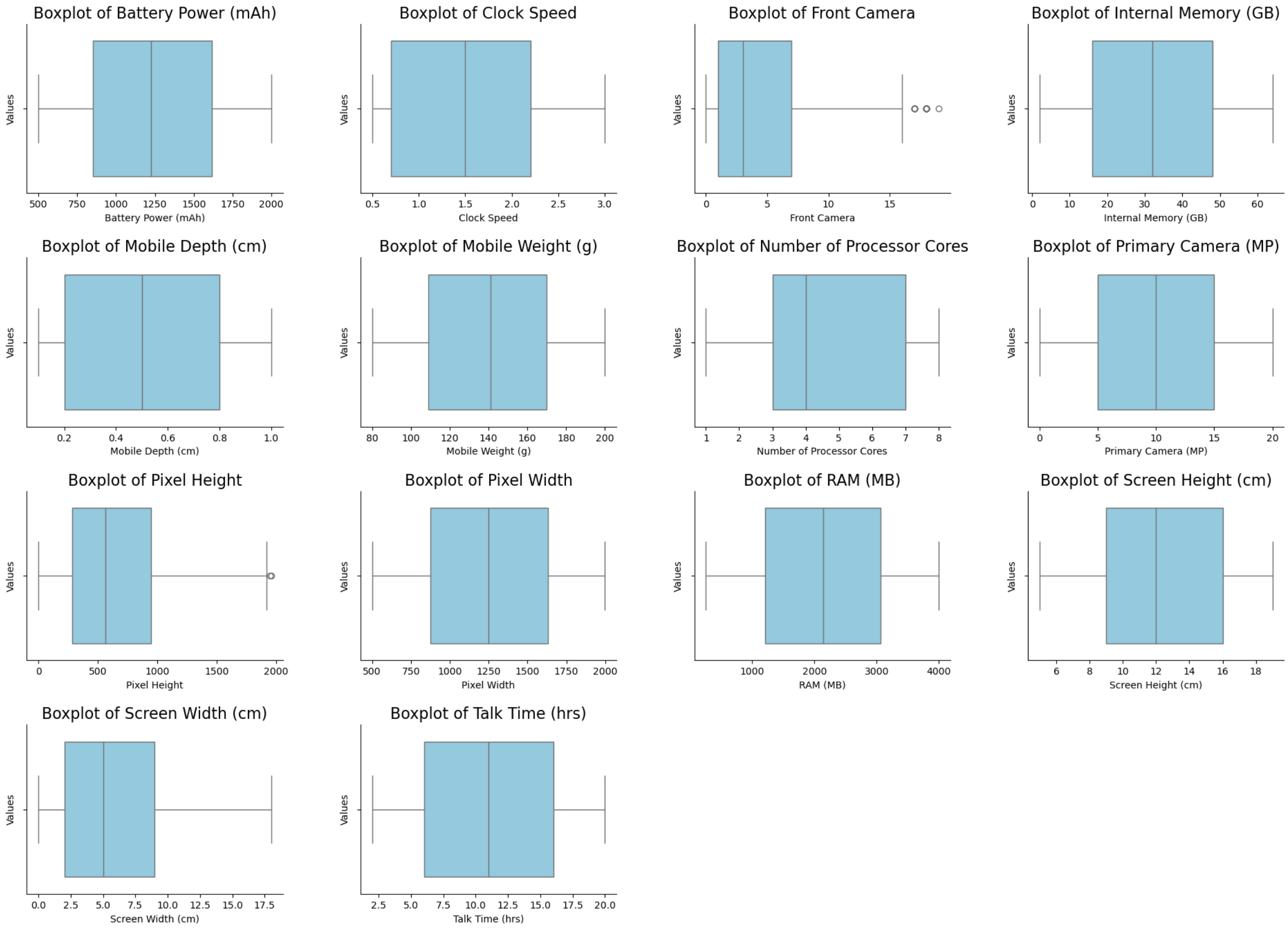
Outlier Detection

Boxplots were used to detect outliers in the numerical features. Observations include:

- **Battery Power, RAM, Internal Memory, Screen Dimensions:** Symmetrical distributions without significant outliers.
- **Front Camera, Pixel Height:** A few mild outliers were detected.
- **Clock Speed, Mobile Weight, Processor Cores, Talk Time:** No major outliers observed.
- **Mobile Depth, Primary Camera:** Consistently within interquartile ranges.

These findings highlight key areas for feature analysis and potential data cleaning, though no severe outliers were found that required removal.

Boxplots for outlier detection across numerical features



Boolean Features

I analyzed the distribution of Boolean features:

- **Bluetooth Support, Dual SIM Support, 4G Support, Touchscreen Support, WiFi Support:** These features showed uniform distributions.
- **3G Support:** Approximately 25% of devices do not support 3G, while 75% do.

3. Bivariate Analysis

Target vs. Numerical Features

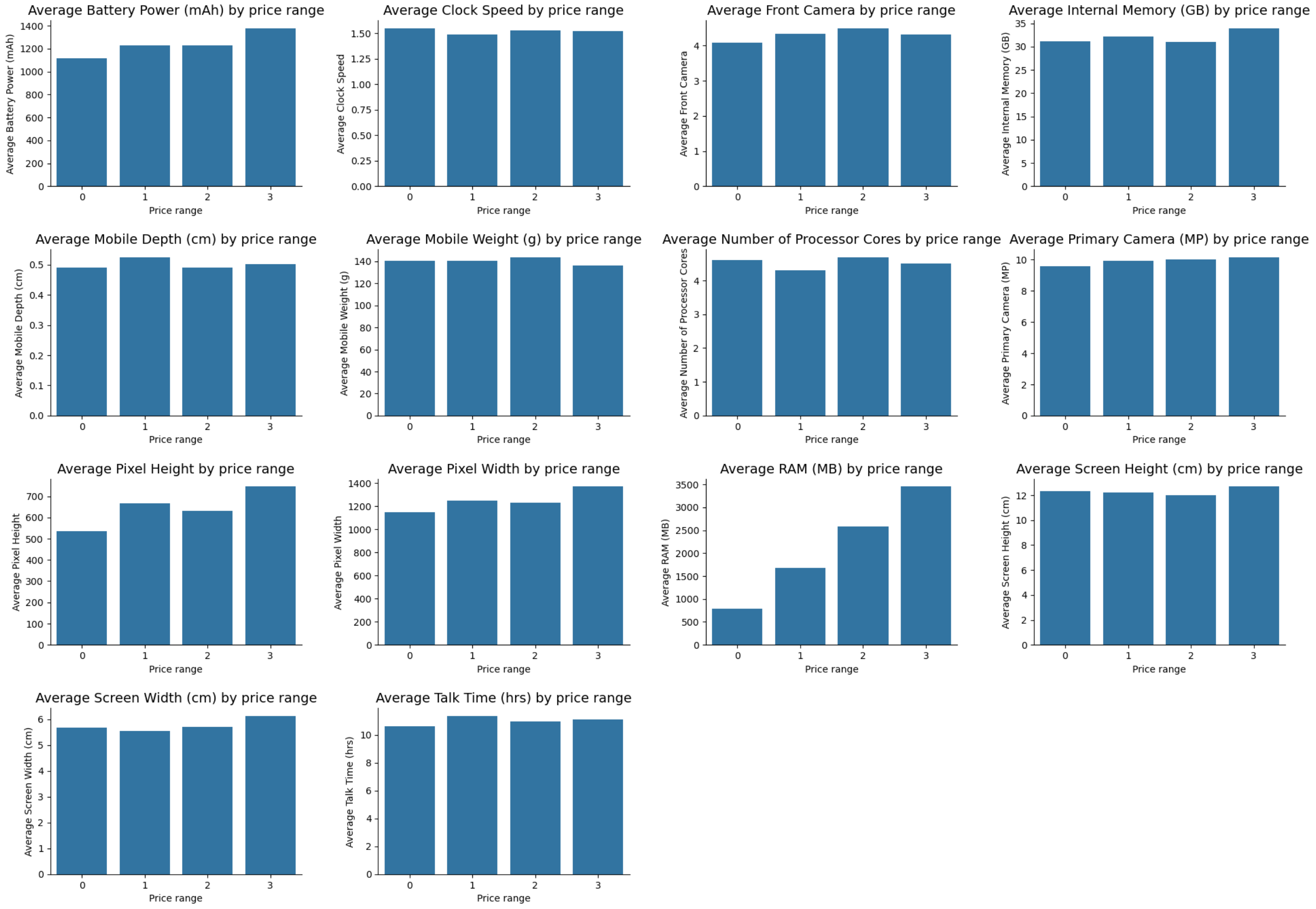
The relationship between the target variable (price_range) and numerical features was examined:

- **Battery Power, RAM, Internal Memory:** Significant increases in these features were observed with higher price ranges, suggesting their importance in premium devices.
- **Clock Speed, Processor Cores, Pixel Dimensions:** Moderate increases indicate their relevance, though trends are less pronounced.

- **Front Camera, Primary Camera:** Gradual improvements with higher price ranges reflect their contribution to increased costs.
- **Screen Dimensions, Talk Time:** These features remained relatively stable across price ranges, indicating minimal variation based on price.
- **Mobile Depth, Mobile Weight:** These features showed minimal changes, suggesting weaker correlations with price.

This analysis highlights that higher-priced devices prioritize performance, storage, and display quality over other features.

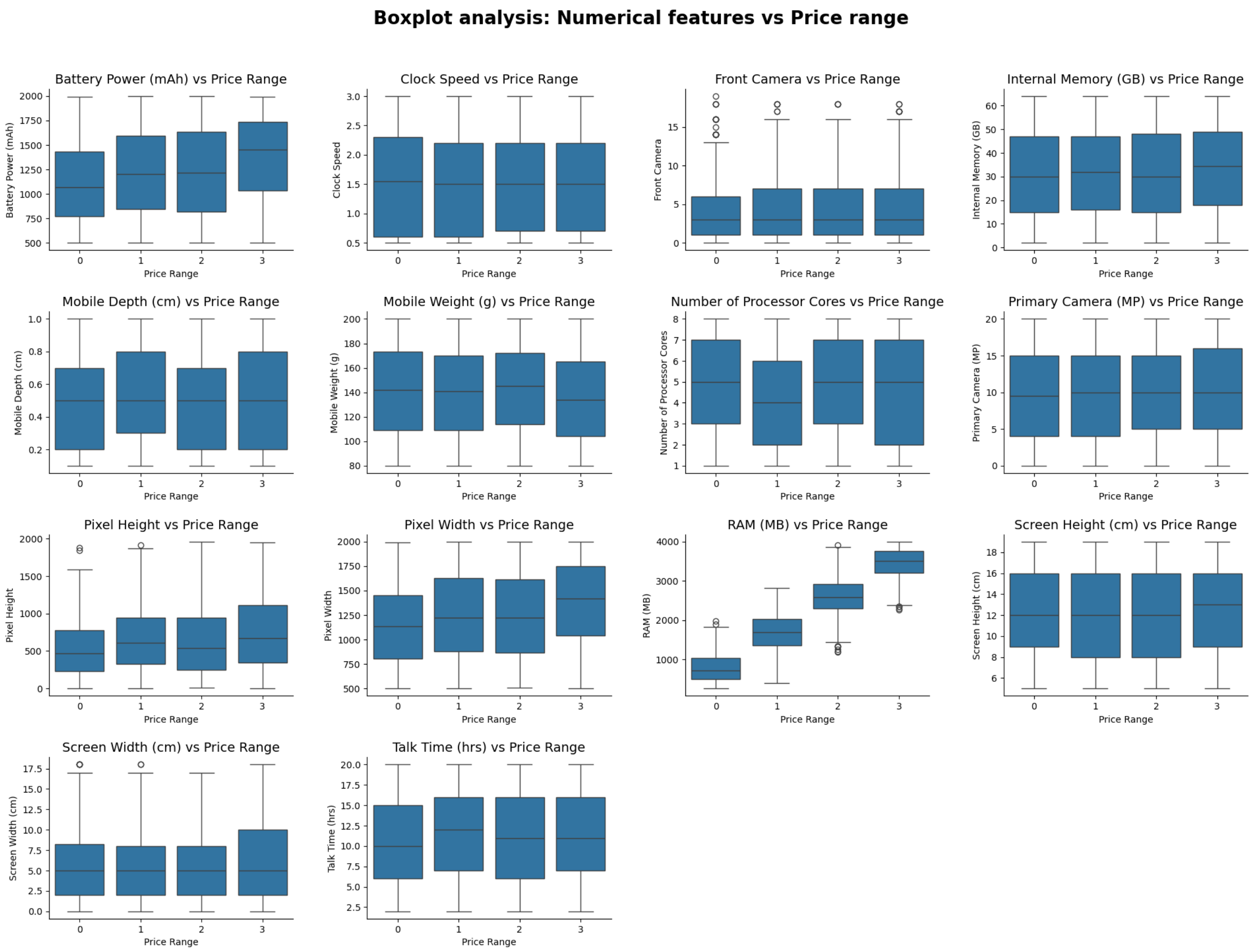
Comparison of average numerical feature values across price ranges



Box Plot Analysis

Box plots comparing numerical features against price_range revealed:

- Higher price ranges are associated with better specifications such as **RAM**, **battery power**, **camera quality**, **internal memory**, and **pixel resolution**.
- Features like **mobile depth**, **talk time**, and **clock speed** showed little variation across price ranges, indicating their lesser impact on pricing.



Boxplot analysis: Numerical features vs Price range

Battery Power (mAh) vs Price Range

Clock Speed vs Price Range

Front Camera vs Price Range

Internal Memory (GB) vs Price Range

Mobile Depth (cm) vs Price Range

Mobile Weight (g) vs Price Range

Number of Processor Cores vs Price Range

Primary Camera (MP) vs Price Range

Pixel Height vs Price Range

Pixel Width vs Price Range

RAM (MB) vs Price Range

Screen Height (cm) vs Price Range

Screen Width (cm) vs Price Range

Talk Time (hrs) vs Price Range

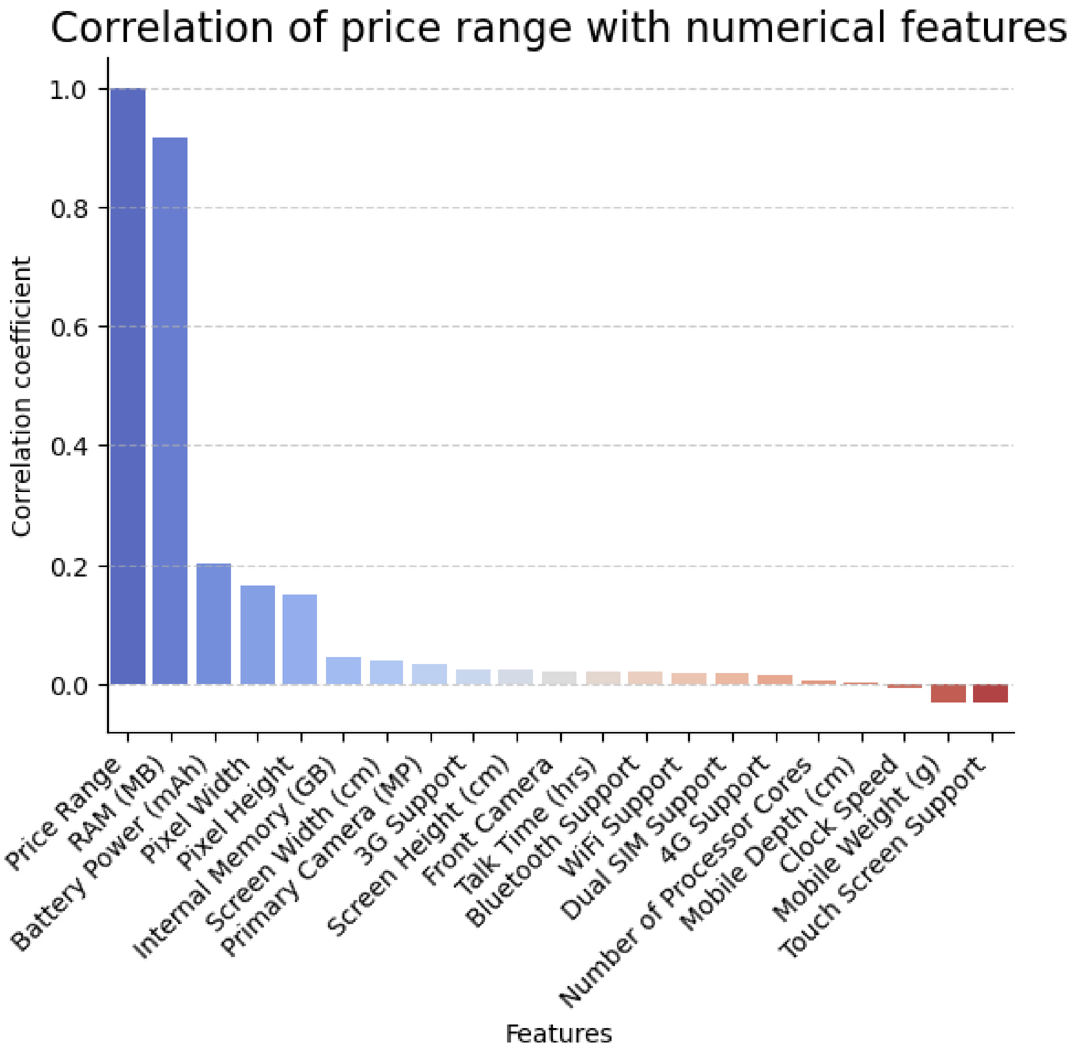
Target vs. Boolean Features

The distribution of each price range within Boolean features was uniform, confirming that these features alone do not significantly influence price categorization.

4. Correlation Analysis

A correlation matrix was generated to identify relationships between features and the target variable (price_range). Key findings:

- **RAM** exhibited the strongest positive correlation (0.9) with price_range, making it a critical feature for predicting price categories.
- **Battery Power, Pixel Width, Pixel Height:** Moderate positive correlations (0.2, 0.19, and 0.18, respectively) were observed.
- **Mobile Weight, Touchscreen Support:** Very small negative correlations were noted, indicating negligible impact on the target variable.



5. Multivariate Analysis

To explore interactions between features, I created a pairplot:

- Diagonal plots displayed distributions of individual variables.
- Off-diagonal scatter plots revealed pairwise relationships, color-coded by the target variable (price_range).
- Clear trends and separations in scatter plots highlighted strong relationships, while random scatter suggested weaker interactions.
- This visualization was particularly useful in identifying clusters, correlations, and potential outliers in the dataset.

Conclusion

The EDA of the mobile price classification dataset revealed several key insights. The dataset is balanced with equal distribution across price categories, eliminating the need for resampling. Numerical features such as RAM, battery power, and internal memory show uniform distributions, while camera-related features are skewed. Mild outliers were observed in some features but didn't require removal. Correlation analysis highlighted that RAM has the strongest positive correlation with the target variable, price_range. Multivariate analysis showed clear relationships between key features and price categories, with performance, storage, and display being the most influential for pricing.

Part 3: Modeling

1. Problem Statement

The target variable, price_range, is a categorical variable with four classes (0 to 3), making this a multi-class classification problem. The objective was to develop a model that accurately predicts price_range based on the given features.

2. Dataset Overview

Two datasets were utilized:

1. **Training Dataset:** Contains the target variable (price_range). This dataset was split into training and testing subsets. Cross-validation was performed on the training subset, while the test subset was reserved for final evaluation.
2. **Prediction Dataset:** Contains 1,000 rows with all features but without the target variable. This dataset was used for class predictions after selecting the best-performing model.

3. Feature Engineering

Feature engineering included the creation of interaction variables to enhance model performance:

- **Pixels:** Product of Pixel Height and Pixel Width.
- **Diagonal (Screen Size):** Calculated using the Pythagorean theorem: Diagonal=Screen Height2+Screen Width2\text{Diagonal} = \sqrt{\text{Screen Height}^2 + \text{Screen Width}^2}Diagonal=Screen Height2+Screen Width2

The original four features (Pixel Height, Pixel Width, Screen Height, and Screen Width) were removed to avoid multicollinearity. Additionally:

- **Standardization** was applied to features for **Logistic Regression**, **Naive Bayes**, and **Random Forest**.
- **Normalization** was applied for the **K-Nearest Neighbors (KNN)** model.

4. Model Selection

Four classification models were tested:

1. **Logistic Regression**
2. **Random Forest**
3. **K-Nearest Neighbors (KNN)**
4. **Naive Bayes**

Model performance was evaluated using the following metrics:

- **Accuracy**
- **Precision (Weighted)**
- **Recall (Weighted)**
- **F1 Score (Weighted)**

Cross-validation was employed.

5. Results

Model Evaluation (Cross-Validation Scores)

Model	Accuracy	Precision (weighted)	Recall (Weighted)	F1 score (Weighted)
Logistic regression	0.8262	0.8209	0.8262	0.8214
Random forest	**0.8662**	**0.8674**	**0.8662**	**0.866**
K-nearest neighbors	0.3969	0.413	0.3969	0.3952
Naïve bayes	0.8006	0.8043	0.8006	0.8013

The **Random Forest** model outperformed all others, with the highest scores across all metrics.

6. Validation on Test Set

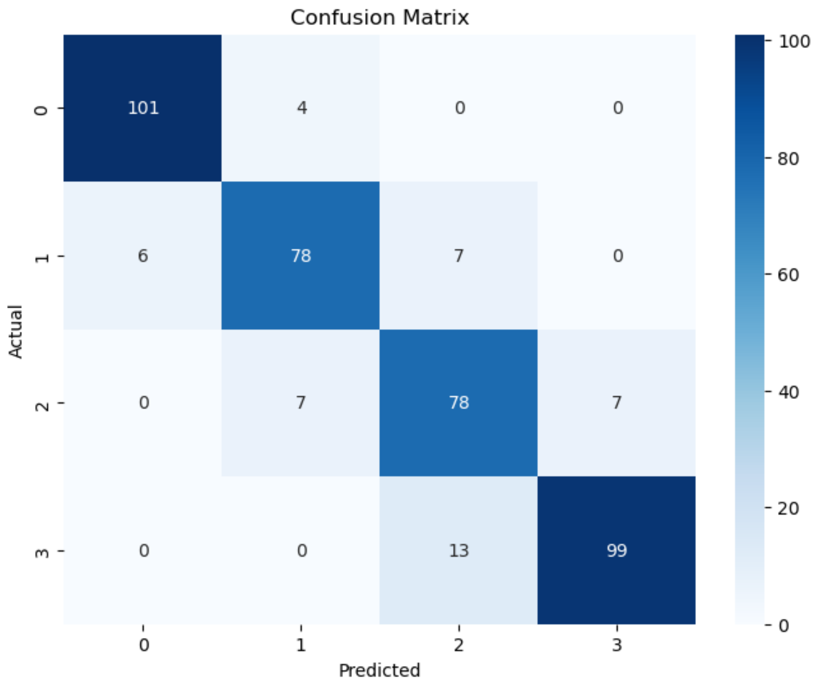
The best-performing Random Forest model was validated on the test set, yielding the following results:

- **Accuracy: 0.8900**
- **Precision (Weighted): 0.8917**
- **Recall (Weighted): 0.8900**
- **F1 Score (Weighted): 0.8904**

Confusion Matrix Analysis

- **Class 0 and Class 3:** High accuracy with minimal misclassification.
- **Class 1 ↔ Class 2 and Class 2 ↔ Class 3:** Some confusion observed, likely due to overlapping feature distributions.
- **Class 0:** Most distinct class with the least misclassifications.

These results demonstrate the model's effectiveness in distinguishing between price categories, with minor expected overlaps in adjacent classes.



7. Final Predictions

The Random Forest model was used to predict the price_range for the prediction dataset (1,000 rows without the target variable). Predictions were successfully generated, providing insights into the price categorization of the given mobile phones.

Conclusion

This project successfully classified mobile phones into four price ranges using machine learning. After evaluating multiple models, **Random Forest** was the best performer, achieving **89% accuracy** on the test set. Key features like **RAM**, **battery power**, and **pixel dimensions** significantly influenced price predictions. Feature engineering, such as creating interaction variables, enhanced model performance. The trained model was also used to predict price ranges for unseen data, demonstrating its practicality. This work showcases the power of machine learning for product categorization and market analysis, with potential for further optimization using advanced techniques.