**Mobile Price Prediction Analysis Using KNN Model**

**(Euclidean Distance)**

**Full Name:** Lourin Ejiuwa  
**Reg No.:** CST/19/COM/00284

**Introduction:**

This project aims to analyze a mobile price prediction dataset and develop a K-Nearest Neighbors (KNN) model to predict mobile price ranges. The analysis includes data cleaning, exploratory data analysis (EDA), feature engineering, model training, and evaluation.

**About the Dataset:**

The dataset contains information on various mobile phone features, such as battery power, RAM, screen dimensions, and camera quality. The goal is to predict the mobile's price range into four categories:

* **0:** Low cost
* **1:** Medium cost
* **2:** High cost
* **3:** Very high cost

The dataset comprises **2,000 entries** and **21 features**, making it suitable for multi-class classification tasks.

**Data Cleaning:**

* Checked for missing values: **No missing values found**.
* Removed unnecessary spaces in column names.
* Replaced unrealistic values below thresholds for features like m\_dep and px\_height.
* Dropped unnecessary columns that had no impact on prediction.

**Exploratory Data Analysis (EDA) - Key Findings:**

* **RAM and Price Range:** Strong positive correlation — higher RAM typically leads to a higher price range.
* **Battery Power:** Devices with higher battery power tend to fall into higher price ranges.
* **Screen Dimensions:** sc\_h and sc\_w show a gradual increase with price range, but the correlation is weaker.
* **Camera Features:** fc (front camera) and pc (primary camera) contribute modestly to price prediction.
* **Connectivity Features:** four\_g, three\_g, wifi, and bluetooth show less impact on price prediction.

**Data Preprocessing:**

* **Outlier Detection:** Applied the Interquartile Range (IQR) method to detect and handle outliers.
* **Feature Engineering:** Added interaction terms such as screen\_area (sc\_h × sc\_w) and polynomial features for ram and battery\_power to improve model performance.
* **Normalization:** Scaled all numerical features to ensure uniformity in distance calculations.

**Data Splitting and Model Training:**

* Split the dataset into **80% training** and **20% testing**.
* Trained a **K-Nearest Neighbors (KNN) model** with **k = 41**.
* Used **weighted KNN** to give higher importance to closer neighbors, improving accuracy.

**Model Evaluation:**

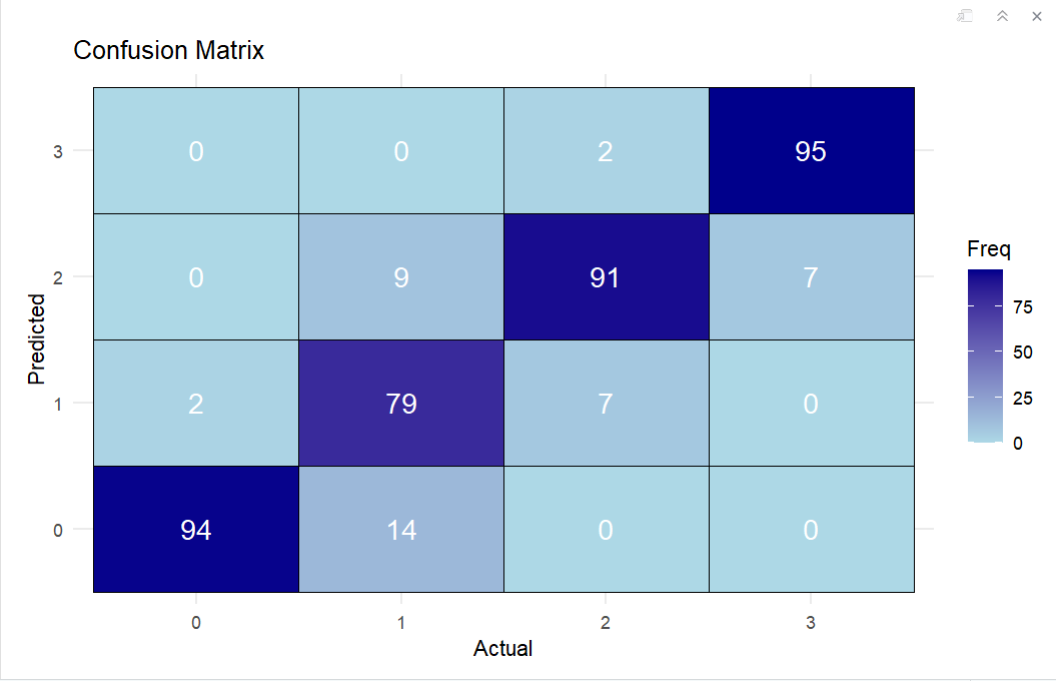
**Overall Performance:**

* **Accuracy:** **89.75%** — The model correctly predicts mobile price ranges in 89.75% of cases.
* **Kappa:** **0.8634** — Indicates strong agreement between predictions and actual values.

**Class-wise Metrics:**

* **Class 0 (Low Cost):** Sensitivity: **97.92%**, Precision: **87.04%** — The model accurately identifies most low-cost devices.
* **Class 1 (Medium Cost):** Sensitivity: **77.45%**, Precision: **89.77%** — Slightly lower performance compared to other classes, suggesting room for improvement.
* **Class 2 (High Cost):** Sensitivity: **91.00%**, Precision: **85.05%** — Shows balanced performance in identifying high-cost devices.
* **Class 3 (Very High Cost):** Sensitivity: **93.14%**, Precision: **97.94%** — Excellent performance in predicting very high-cost devices.

**Confusion Matrix:**



* **True Positives:** Correctly classified instances per class improved after feature engineering.
* **False Positives:** Reduced for Classes 2 and 3, indicating fewer misclassifications.

**Conclusion:**

The K-Nearest Neighbors (KNN) model developed for mobile price prediction demonstrates **strong performance and reliability**:

* **High Accuracy:** Achieved **89.75% accuracy**, indicating excellent predictive power.
* **Balanced Performance:** The model maintains a **good balance between sensitivity and specificity**, especially for Classes 0, 2, and 3.
* **Key Features:** The most influential features were found to be **RAM**, **battery power**, and **screen dimensions**.
* **Class 1 Performance:** Performance for Class 1 (medium-cost devices) could be improved with further feature engineering or exploring other classification models.