

Mobile Price Predictor with Random Forest



A
ADM Course Project Report
In
partial fulfilment of the degree

Bachelor of Technology
in
Computer Science & Engineering

By

Name: V.Hruthika	HTNo (2303A51543)
Name: S.Shryitha	HTNo (2303A51637)
Name: A.Vaishnavi	HTNo (2303A51554)
Name: R. Rupa Sri	HTNo (2303A51918)
Name: B. Sai Spurthi	HTNo (2303A51886)

Under the guidance of

Bediga Sharan
Assistant
Professor

Submitted to

School of Computer Science and Artificial Intelligence



CERTIFICATE

This is to certify that the **APPLICATIONS OF DATA MINING– Course Project** Report entitled “**Mobile Price Predictor with Random Forest**” is a record of bonafide work carried out by the student(s) V.HRUTHIKA,S.SHRYITHA,A.VAISHNAVI,R.RUPASRI,B.SAI SPURTHI bearing Hallticket No(s) 2303A51543,2303A51637,2303A51554,2303A51918,2303A51886 during the academic year 2024-25 in partial fulfillment of the award of the degree of *Bachelor of Technology* in **Computer Science & Engineering** by the SR University, Warangal.

Supervisor

(Mr. Bediga Sharan)
Assistant Professor

Head of the Department

(Dr. M. Sheshikala)
Professor

TABLE OF CONTENTS

Topic	PageNo
Title page	1
Certificate	2
Table of Contents	3
Abstract	4
Objective	5
Definitions of the Elements used in the project	6-8
Design	9
Implementation	10-15
Result Screens	16-23
Conclusion	24

Abstract:

The rapid evolution of mobile phone technology has led to a diverse and competitive marketplace where pricing strategies play a critical role in consumer decision-making and product positioning. This project investigates the correlation between mobile phone specifications and their market prices using data mining and machine learning techniques. By leveraging regression, classification, and clustering models, the study aims to predict phone prices based on key features such as RAM, screen size, battery capacity, processor type, storage, and camera quality.

A comprehensive dataset was curated and preprocessed to ensure consistency and relevance. Regression models, including Linear Regression and Random Forest Regression, were employed to estimate continuous price values, while classification algorithms such as K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) categorized devices into defined price segments. Additionally, clustering techniques like K-Means were applied to identify natural groupings of devices based on feature similarity, revealing hidden patterns and market trends.

The analysis yielded significant insights into how specific hardware features influence pricing, highlighting trends such as the growing importance of camera performance and high RAM in premium smartphones. These findings have practical implications for manufacturers, retailers, and consumers, aiding in strategic decision-making and market analysis.

The project culminates in the development of an interactive prototype that allows users to input mobile phone specifications and receive price predictions along with visual representations of feature-price correlations and clustering outcomes. Accompanied by thorough documentation, this tool serves as a practical application of the research and demonstrates the potential of data-driven approaches in understanding and navigating the mobile phone market.

Objective of the Project:

The objectives of this project are:

- Develop an intelligent machine learning system capable of accurately predicting the launch price of mobile phones based on key specifications such as RAM, screen size, battery capacity, camera specifications, and brand, aiding in product positioning and competitive analysis.
- Perform thorough data preprocessing and exploratory data analysis (EDA) to clean and transform the dataset—handling missing values, outliers, and non-numeric formats while revealing key market trends, feature distributions, and correlations.
- Implement and evaluate multiple machine learning models including Random Forest Regressor (for price prediction) and various classifiers (for price category classification), using metrics such as R^2 Score, MAE, MSE, Accuracy, Precision, and F1-Score to ensure model accuracy and reliability.
- Visualize results and model behavior using histograms, KDE plots, heatmaps, trend lines, and confusion matrices, enhancing interpretability and offering insights into how different features affect price and demand.
- Deliver a user-friendly, web-based prediction tool built using Gradio, allowing users to input mobile specifications interactively and instantly receive predicted prices, thereby promoting the use of explainable AI in consumer electronics decision-making and market forecasting.

Definitions of the Elements Used in the Project:

This section outlines the key technical elements, data features, and machine learning concepts applied throughout the development of the “Mobile Price Prediction” model. They are grouped by their role in the data science workflow.

Data elements

Dataset:

A structured CSV file containing specifications and launch prices of mobile phones, including features like “RAM, battery capacity, screen size, camera specs, and brand name”.

Feature:

An individual input variable used to train the model. Examples include “RAM (GB), Battery (mAh), Front Camera (MP),” and “Company Name”.

Target Variable (Price):

The value the model is trying to predict — in this case, the “Launched Price (India)”, represented as a continuous numerical value (₹).

Label Encoding:

Used to convert the “Company Name” (categorical variable) into a numeric format to be processed by machine learning models.

Data Preprocessing & Cleaning

Missing Values:

Empty entries handled using “forward-fill imputation”, which propagates the previous non-null value forward.

Outliers:

Detected using the “Interquartile Range (IQR)” method. Helps improve model reliability by filtering extreme values.

Data Normalization:

Features like RAM, Battery, etc., were normalized using “Min-Max Scaling”, bringing them into a 0–1 range to ensure balanced feature weightage.

Feature Extraction:

Numeric values were extracted from mixed-type fields like '8 GB RAM' or '₹15,000' using regular expressions.

Exploratory data analysis (EDA)

Histogram & KDE Plots:

Used to visualize price distribution and identify concentration of data around common values.

Line Plots:

Show trends such as “average RAM growth over years” or “average price changes by launch year”, highlighting evolving market trends.

Outlier Visualization:

Combined plots overlay outlier points (in red) on standard distribution charts for easy comparison.

Correlation Heatmap:

Helps visualize relationships between features like RAM, screen size, and price — useful for identifying which inputs affect price most.

Machine Learning Concepts

Supervised Learning:

The model is trained on labeled data (features + price) to learn patterns and predict unseen outcomes.

Regression:

A supervised learning task where the model predicts “continuous values” like price. Implemented using “Random Forest Regressor”.

Classification (Extended):

For advanced evaluation, prices were grouped into 4 categories (Low, Medium, High, Very High), and classification models were tested with a “confusion matrix”.

Random Forest Regressor:

An ensemble method using multiple decision trees to output the average prediction — robust, accurate, and handles non-linear data well.

Random Forest Classifier (optional):

Used when prices were classified into categories — enables use of confusion matrix and classification metrics.

Model evaluation metrics:

R² Score:

Indicates how well the regression model explains the variance in the data (1.0 = perfect prediction).

MAE (Mean Absolute Error):

Measures the average error between predicted and actual prices in ₹ — lower is better.

MSE (Mean Squared Error):

Emphasizes larger errors by squaring them — helpful for identifying models with fewer major mistakes.

Confusion Matrix:

Used in the classification extension — a 4x4 matrix showing actual vs. predicted categories.

Classification Report:

Displays Precision, Recall, and F1-Score for each price category when classification is applied.

Model interpretability:**Feature Importance (built-in):**

Random Forest models provide insights into which features (e.g., RAM or Brand) most influence predictions.

Data Visualizations (EDA):

Help communicate insights and justify model behavior visually, especially during trend analysis.

Tools & environment:**Google Colab:**

A free, cloud-based Python coding environment used for developing, training, and testing the ML model with GPU/CPU support.

Pandas & NumPy:

Libraries used for data loading, cleaning, numeric operations, and manipulation of DataFrames.

Scikit-learn:

Main ML library used to build the “Random Forest Regressor/Classifier”, perform preprocessing, scaling, encoding, and evaluation.

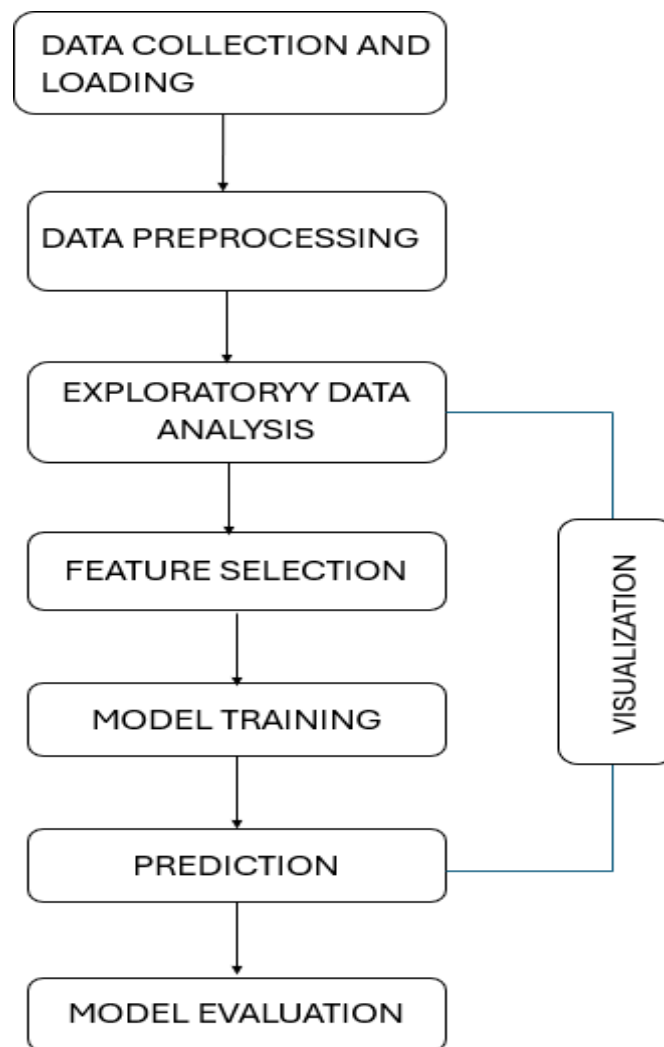
Matplotlib & Seaborn:

Used for creating attractive and informative visualizations like histograms, line plots, KDEs, and heatmaps.

Gradio:

An open-source tool for building web UIs. Used to deploy a “real-time mobile price prediction interface”, where users can input features and see the predicted price.

DESIGN:



GITHUB LINKS:

1. <https://github.com/SHRYITHA/ADM-PROJECT>
2. https://github.com/Hruthikaveldi/ADM_PROJECT
3. <https://github.com/VAISHNAVI-ACHI/ADM-PROJECT>
4. https://github.com/spurthi1886/ADM_PROJECT
5. https://github.com/Rupasri0105/ADM_PROJECT

Implementation:

Interactive Prediction Web Interface with Gradio:

Gradio Web Interface for prediction

```
!pip install gradio --quiet
```

```
import gradio as gr
```

```
# Generate sorted list of brand names for dropdown
```

```
company_list = sorted(df['Company Name'].unique().tolist())
```

```
# Prediction function using user inputs
```

```
def predict_price(ram, screen, battery, front_cam, back_cam, company_name):
```

```
    encoded_company = le.transform([company_name])[0]
```

```
    input_data = np.array([[ram, screen, battery, front_cam, back_cam, encoded_company]])
```

```
    input_scaled = scaler.transform(input_data[:, :5]) # Normalize numerical part
```

```
    final_input = np.concatenate((input_scaled, input_data[:, 5:]), axis=1) # Combine with encoded company
```

```
    predicted_price = rf.predict(final_input)[0]
```

```
    return f" Estimated Price in India:
```

The implementation of the Mobile Price Prediction project using Random Forest followed a structured data science workflow, involving several critical steps from data preparation to model deployment. This section provides an in-depth explanation of each phase for clarity and academic understanding.

1. Importing Required Libraries

The first step involved importing all the necessary Python libraries essential for data handling, preprocessing, visualization, and machine learning. Libraries such as pandas and numpy were used for data manipulation, matplotlib and seaborn for visualization, scikit-learn for modeling and evaluation, and gradio for building a web-based interface. These libraries provided the foundational tools required to

perform end-to-end data science operations.

2. Loading and Understanding the Dataset

We loaded the dataset into a Pandas DataFrame using `read_csv()`. The dataset contained mobile specifications such as RAM, battery capacity, screen size, camera specifications, and corresponding launch prices across various regions. Initial inspection using `df.head()` and `df.info()` helped understand the structure, identify null values, data types, and assess the overall readiness for analysis.

3. Data Cleaning and Transformation

Many columns contained mixed data formats. For instance, RAM values included units (e.g., "8 GB"), prices had symbols (e.g., "₹15,000"), and camera specs listed multiple lens values. We applied regular expressions to extract numeric values from such strings, converting them into float-type values. Specific operations included:

- Removing "₹" and commas from price columns.
- Extracting numeric parts from "8 GB RAM", "5000 mAh Battery".
- Cleaning camera values by choosing the maximum megapixel in the case of multiple lenses.

Duplicates were removed using `drop_duplicates()`, and missing values were forward-filled to maintain data continuity.

4. Feature Engineering and Encoding

We created a new feature called `Company_Encoded` by applying Label Encoding to the Company Name column. This transformation is crucial because machine learning models do not handle categorical (string) data directly. Numerical encoding allows us to preserve brand information in a machine-readable form.

Additionally, we filtered out extreme cases (e.g., devices with RAM over 12GB) to reduce outlier effects on the model's performance.

5. Exploratory data analysis (EDA)

Exploratory Data Analysis (EDA) is a crucial step in any data science project. In this mobile price prediction project, EDA helped uncover hidden patterns, trends, and anomalies in the data, allowing for better feature selection and a deeper understanding of the dataset's structure.

i. Price Distribution Analysis

We used both histograms and Kernel Density Estimation (KDE) plots to understand how mobile prices are distributed in the dataset.

- Histogram: Revealed the frequency of different price ranges.
- KDE Plot: Provided a smooth curve representing the probability distribution of prices, helping us visualize data density and skewness.

To enhance the analysis, we plotted outliers in red on top of the normal distribution to highlight devices with unusually high prices.

ii. Outlier Detection (RAM & Price)

Using the Interquartile Range (IQR) method, we identified devices that fall outside the normal range for RAM and price.

- IQR detects values below $Q1 - 1.5IQR$ or above $Q3 + 1.5IQR$.
- These outliers were separated and plotted to understand how extreme values influence the data.

Outliers were especially important in pricing, as high-end phones with luxury features skewed the average price upward.

iii. Trend Analysis: RAM Over the Years

We plotted average RAM per launch year using a line plot with data points.

- This revealed a consistent upward trend in average RAM, showcasing how mobile hardware has evolved over time.
- The plot helped us understand the technological shift in consumer devices, influencing how features affect pricing.

iv. Launch Year vs. Average Price

A line plot with mean price by year was used to observe pricing trends.

- We noticed that prices fluctuate year-to-year based on market innovation, release of flagship devices, and demand.
- Red scatter points were overlaid to highlight outliers — phones priced unusually high in their respective years.

v. Brand Distribution

A countplot was created to display the number of mobile models offered by each brand in the dataset.

- This gave a clear picture of brand presence and market dominance.
- Rotated x-axis labels ensured all brand names were visible and readable.

vi. Correlation Heatmap

We generated a correlation matrix heatmap to examine relationships between numeric features:

- Strong positive correlations were found between price and features like RAM, Battery, and Camera specs.
- We used coolwarm color palette and annotations to make the heatmap visually interpretable.
- This analysis supported feature selection for the machine learning model.

6. Feature Scaling (Normalization)

We normalized key numeric features like RAM, Battery Capacity, and Screen Size using the MinMaxScaler to bring all values into a consistent range of 0 to 1. This step ensures that larger-scale features don't dominate the learning process and improves overall model accuracy.

7. Regression Model Training and Evaluation

We selected the following features as inputs: RAM, Screen Size, Battery Capacity, Front Camera, Back Camera, and Company_Encoded. The target variable was the Launched Price (India).

Using an 80/20 train-test split, we trained a Random Forest Regressor with 200 estimators. The model was evaluated using:

- R^2 Score to measure model accuracy.
- MAE (Mean Absolute Error) for average deviation.
- MSE (Mean Squared Error) to penalize large errors.

The regression model performed well and was capable of making accurate price predictions based on the input features.

8. Classification Extension

To evaluate the model using classification techniques, we divided the price into four ranges (Low, Medium, High, Very High) and converted the continuous price into a categorical variable. A Random

Forest Classifier was trained on the same features, and the results were evaluated using:

- Confusion Matrix
- Classification Report (Precision, Recall, F1-score)

This step allowed us to understand model behavior in terms of categorization and supported comparative analysis.

9. Multi-Model Classification Comparison

We extended the classification model further by comparing six algorithms:

- Random Forest
- Logistic Regression
- Decision Tree
- SVM
- Naive Bayes
- K-Nearest Neighbors

Each model was trained and tested on the same dataset, and their performance was compared using accuracy scores and visualized using a bar chart. This provided insight into which algorithm best suited the mobile price category prediction task.

10. Gradio Interface

Gradio is a Python library that allows developers to quickly create user-friendly web interfaces for machine learning models. In this project, it was used to build a real-time mobile price prediction tool.

1. Purpose of Using Gradio

- To allow users to input mobile specifications such as RAM, Battery, Camera, and Brand via a clean interface.
- To return the predicted mobile price instantly using our trained Random Forest model.
- To bridge the gap between technical backend code and real-world usability.

2. User Input Widgets

Gradio supports multiple input types. For our use case:

- Sliders were used for numerical features like RAM, Screen Size, Battery, Front Camera, and Back

Camera.

- Dropdown Menu was used to select the mobile's brand name from a sorted list.

This design makes it easy for non-technical users to interact with the model without understanding the underlying code.

3. Prediction Function

- The function receives user inputs, encodes the brand name using LabelEncoder, and normalizes the numeric inputs using the same MinMaxScaler used during model training.
- These processed values are combined and passed to the trained Random Forest Regressor to generate the price prediction.
- The output is formatted as a clean, readable price in Indian Rupees (₹).

4. Launching the Interface

Once the interface is defined, `iface.launch()` is called to:

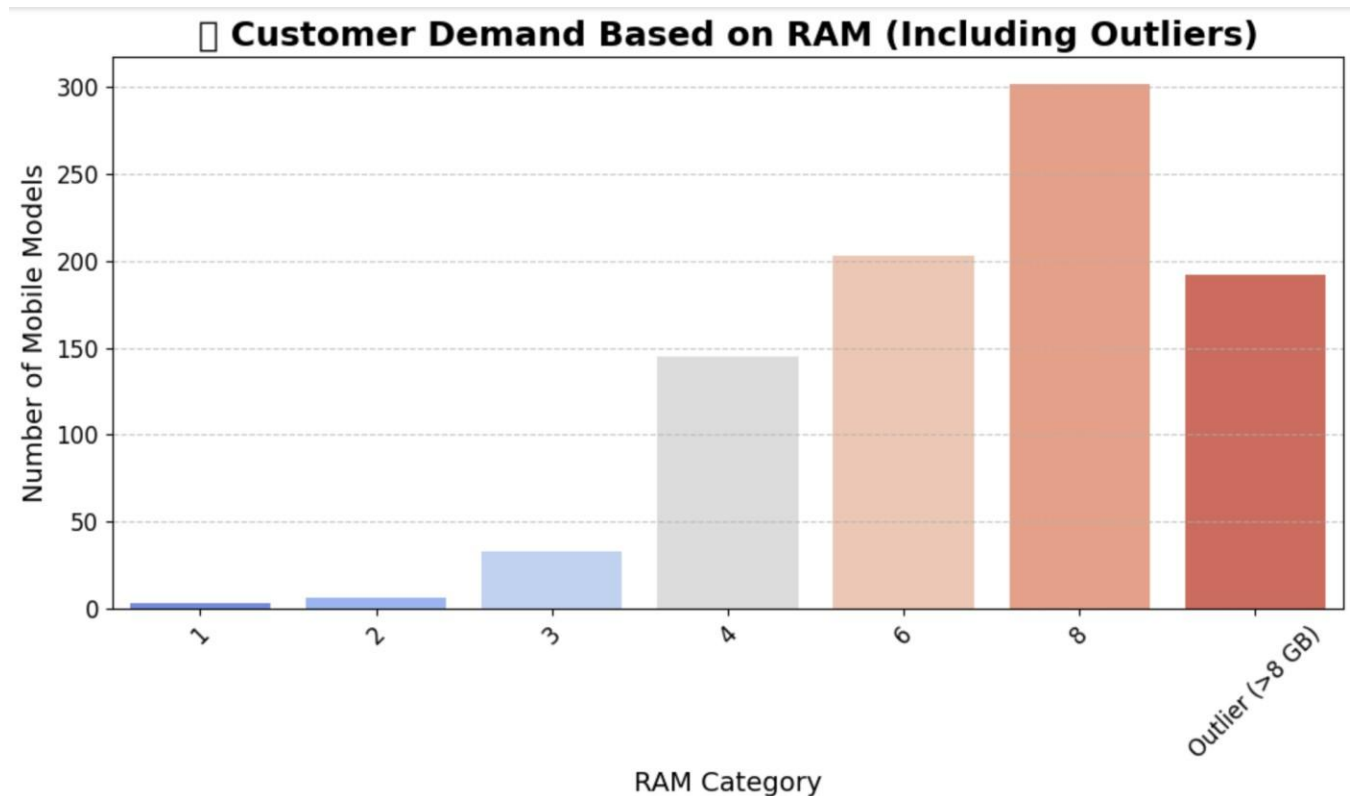
- Start a temporary local server.
- Open the interface in the browser.
- Allow real-time interaction with the model in a web-based format.

5. Benefits of Gradio

- Requires minimal code and setup.
- Excellent for demoing machine learning projects.
- Great for user testing, model deployment, or classroom presentations.
- Can be deployed on cloud platforms like Hugging Face Spaces.

Together, EDA and Gradio form the analytical and interactive backbone of the project. While EDA ensures informed model design and transparency, Gradio transforms the model into a usable tool with real-world applications.

Result Screens:



Feature Scaling for Modeling:

-->Applies StandardScaler to selected numeric columns, transforming them to have zero mean and unit variance. This prepares the data for algorithms that are sensitive to scale (e.g., KMeans, SVM, Linear Regression).

Simplifying RAM Values for Visualization:

-->Rounds off RAM to the nearest whole number for clearer categorization and grouping in plots.

Outlier Categorization:

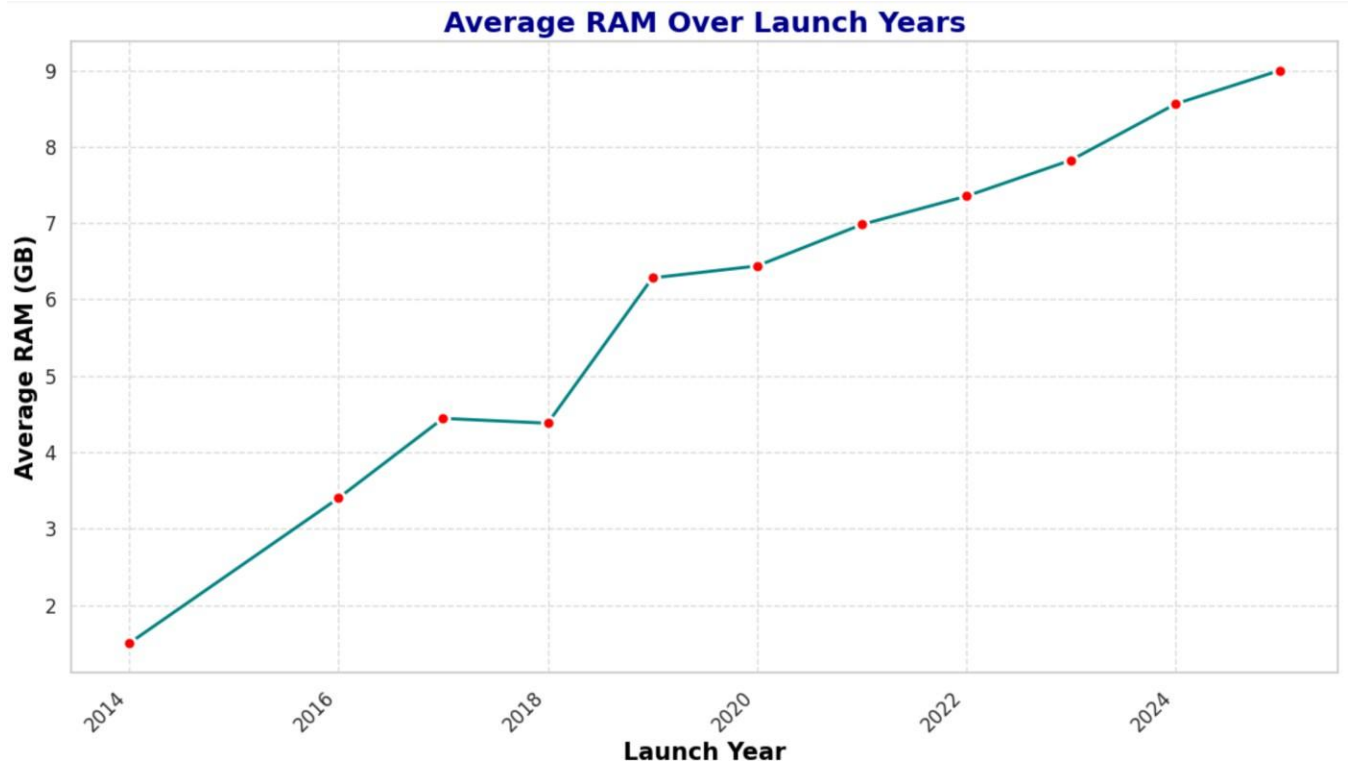
-->Groups RAM sizes above 8 GB under a common label (Outlier (>8 GB)), treating them as less common.

Category Frequency Calculation:

-->Counts how many mobile models fall under each RAM category.

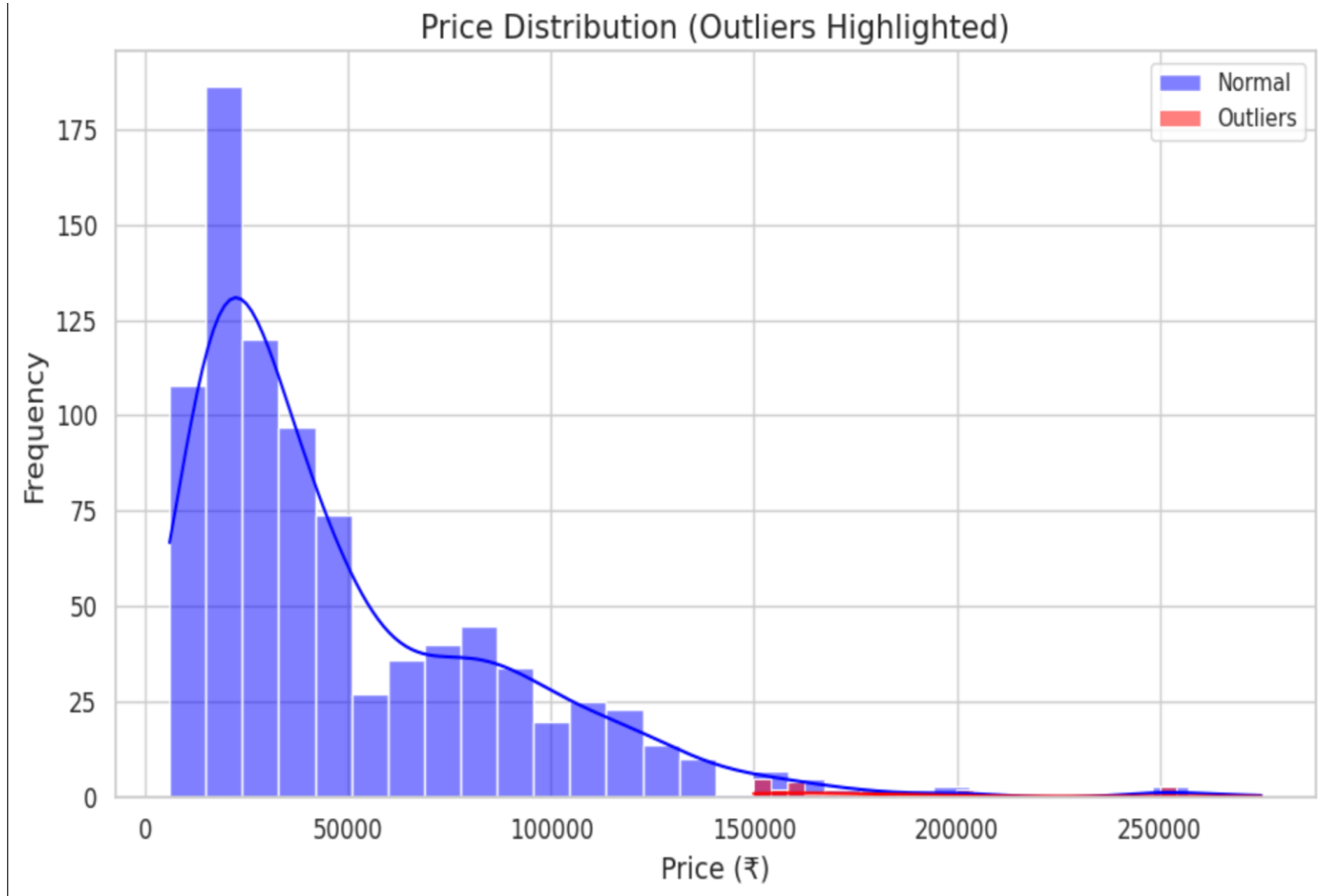
Data Visualization – RAM Demand Analysis:

-->Creates a styled bar chart showing how many models exist in each RAM category, visually representing customer demand and market distribution.



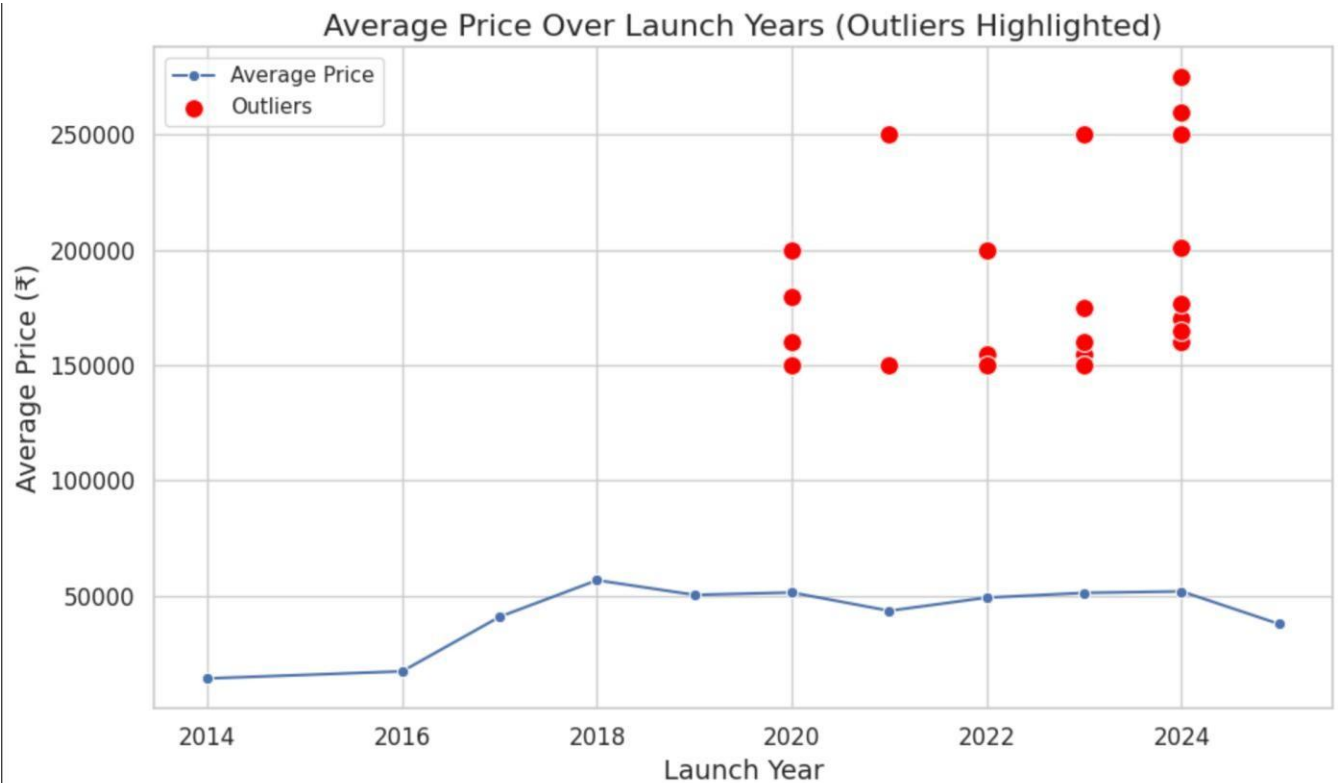
Aim: To visualize the trend of average mobile RAM capacity over the years by plotting a line graph of RAM vs. Launched Year. This helps identify whether smartphone RAM has increased over time and by how much.

Insight: This plot allows analysts or businesses to observe market evolution and technological growth in mobile memory. It's useful for product planning, benchmarking, and understanding customer expectations over



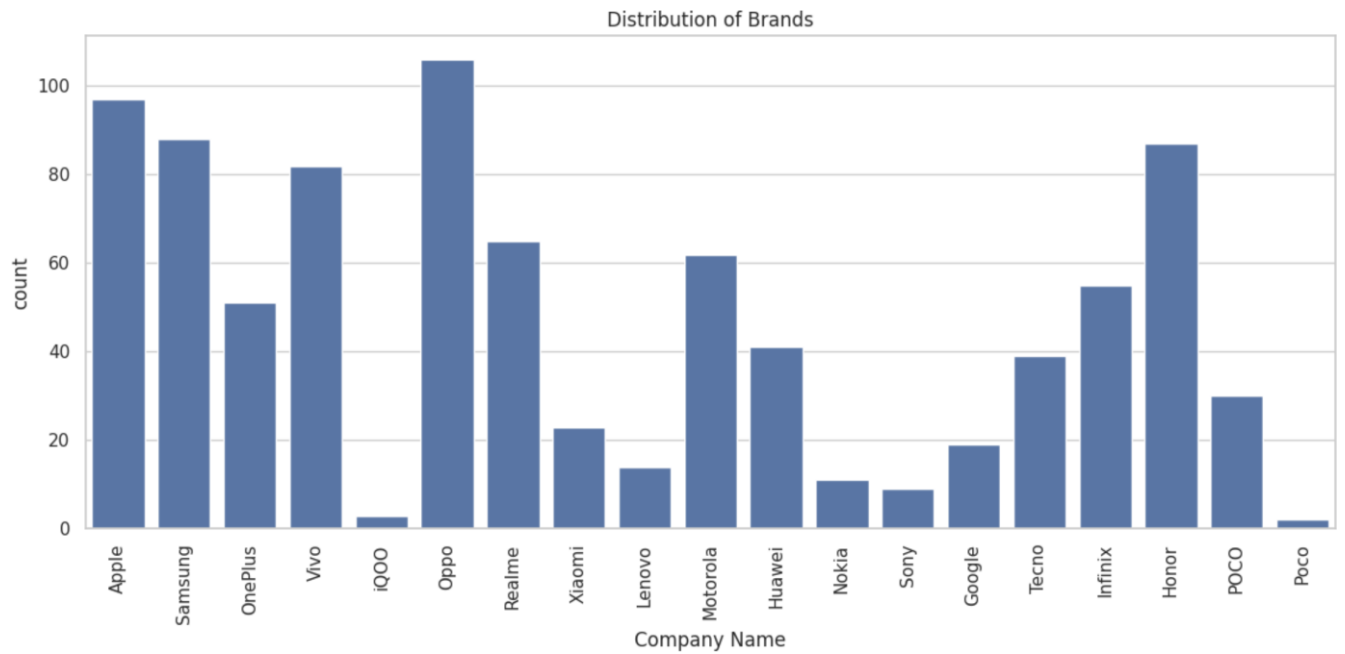
Aim: To visualize the distribution of mobile launch prices in India and clearly highlight the outlier price points using a histogram with kernel density estimation (KDE).

Insight: This helps in understanding the price spread in the market and identifies any unusual pricing trends (e.g., extremely cheap or expensive phones) that may affect analysis or model training.



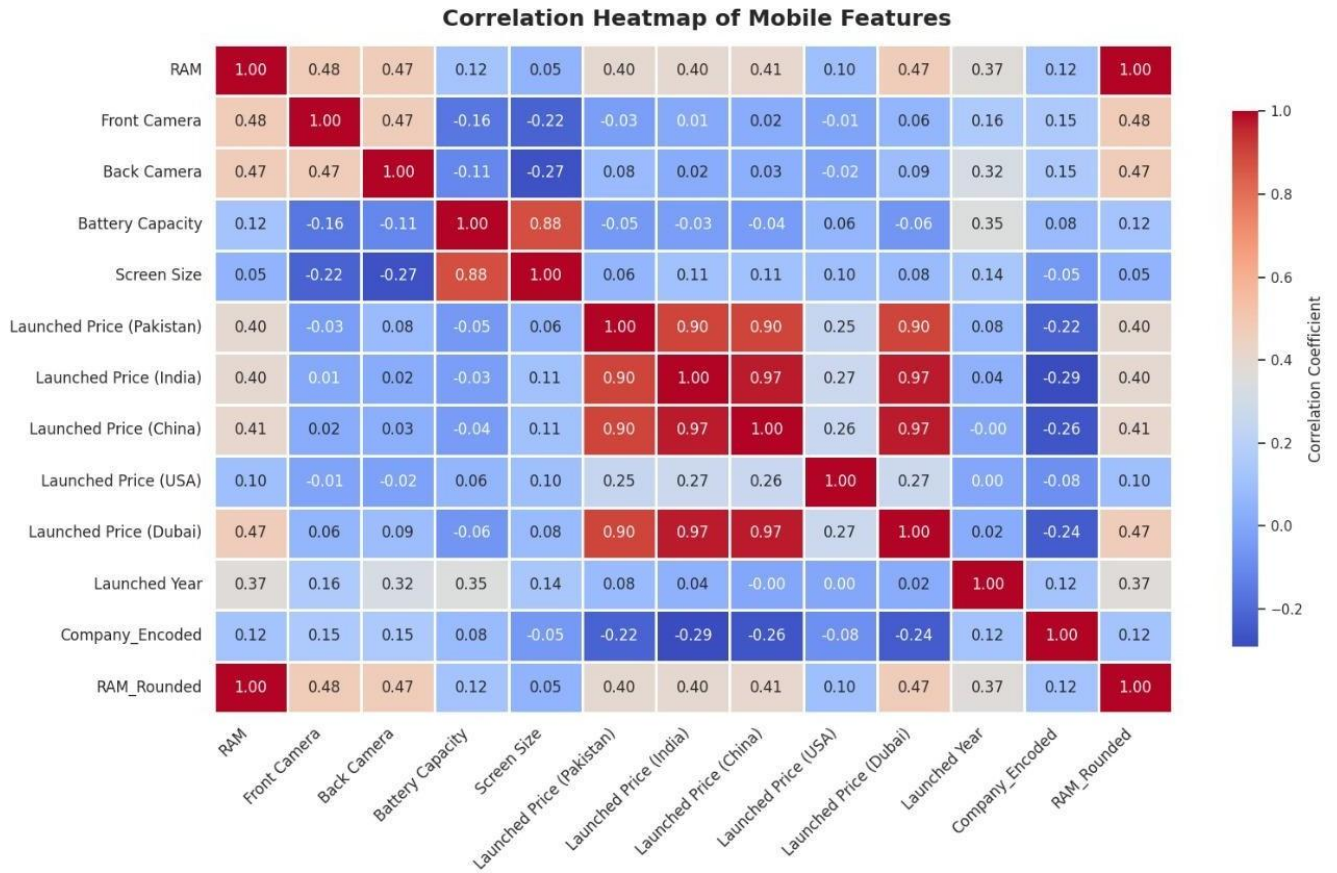
Aim: To visualize how the average launch price of mobiles has changed over different years, and highlight price outliers to detect anomalies or luxury product launches.

Insight: This plot shows both the general pricing trend over time and emphasizes any unusually priced models that may skew analysis or reflect premium segments. It helps in time-based trend analysis for pricing strategy or forecasting.



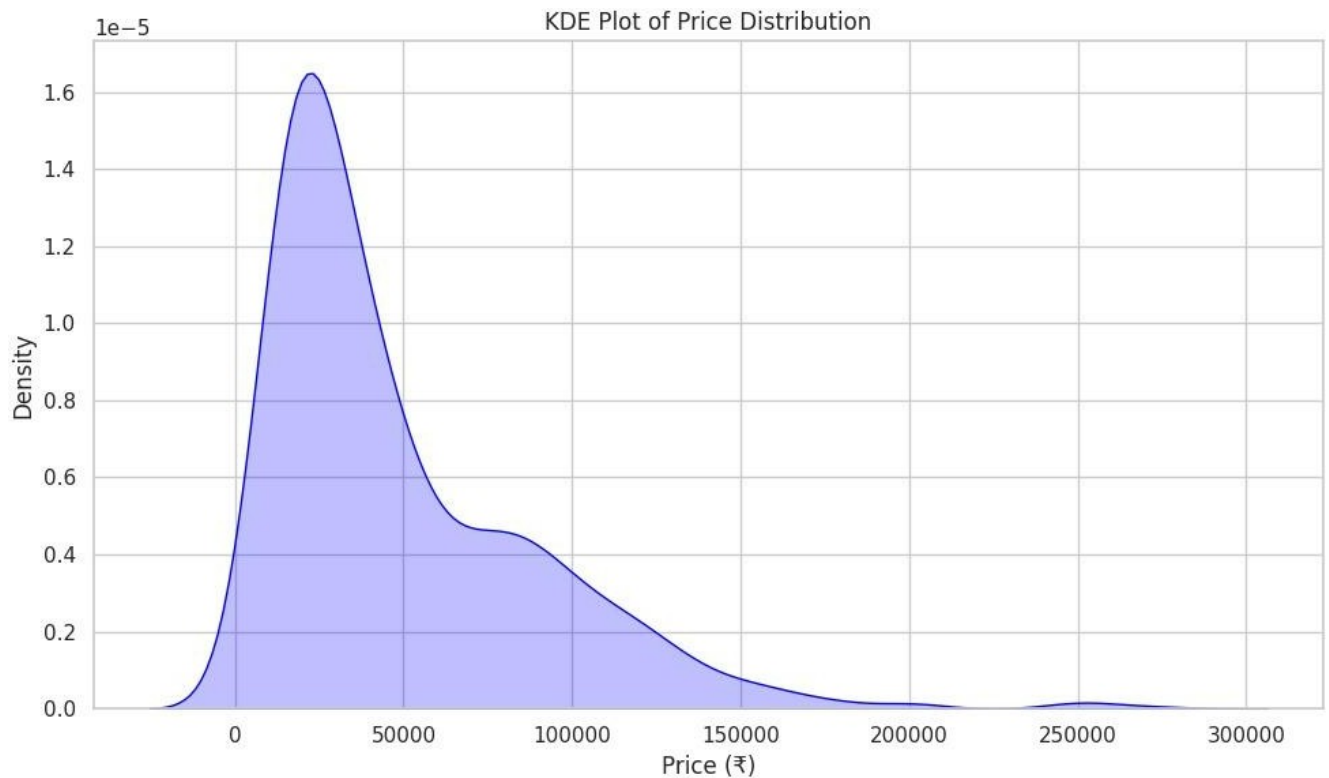
Aim: To display the number of mobile models launched by each brand/company using a count plot

Insight: This visualization helps identify which brands dominate the market, which ones are less active, and offers a quick overview of brand-wise distribution in the dataset.



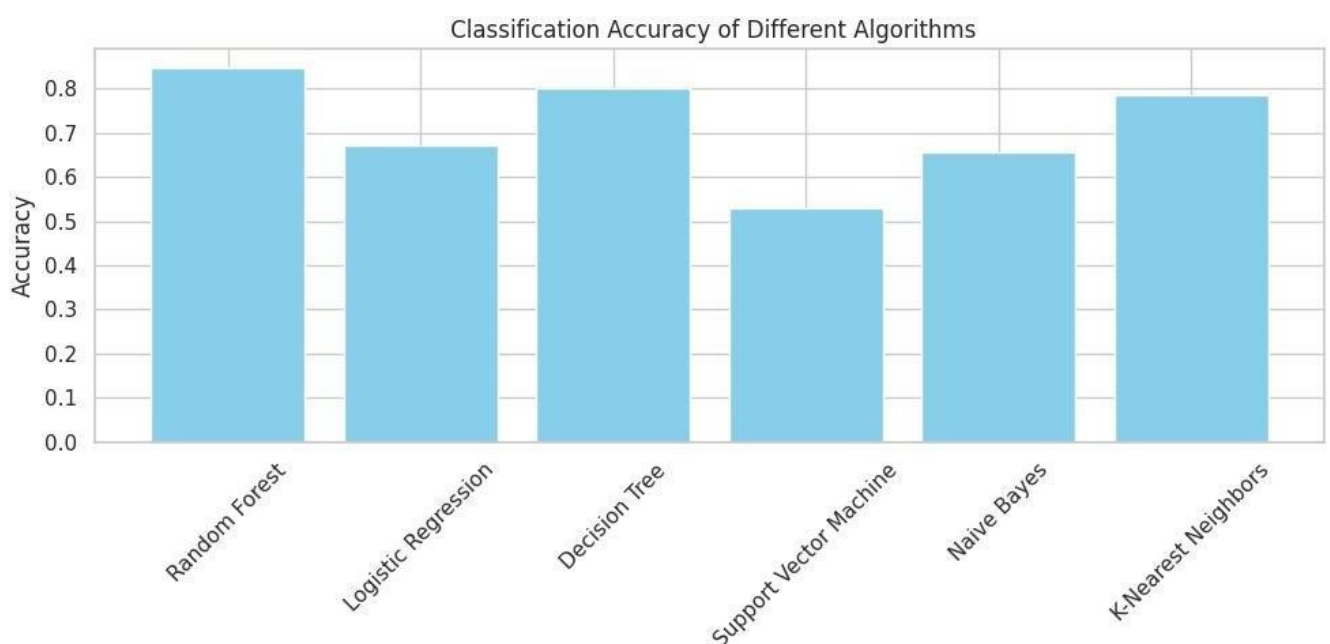
Aim: To visualize the correlation strength between numeric mobile features (e.g., RAM, Battery, Price, Camera specs) using a heatmap.

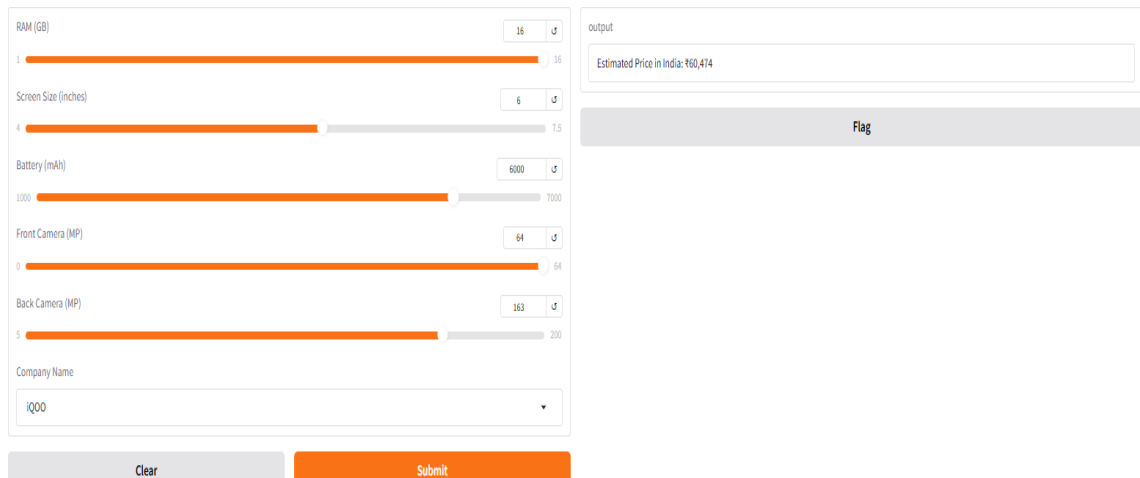
Insight: This helps identify strong positive or negative relationships (e.g., price vs. RAM or screen size), which are useful for feature selection in machine learning or understanding feature importance in analysis.



Aim: To plot a KDE (Kernel Density Estimation) curve for the Launched Price (India) column, showing a smooth probability distribution of price values.

Insight: KDE plots help to visualize the shape of the data distribution, identify skewness, and understand where prices are concentrated (e.g., budget, midrange, premium segments).





The image shows a Gradio web interface for a mobile price prediction model. The interface is divided into two main sections: input and output. The input section on the left contains several sliders and a dropdown menu. The sliders are for RAM (GB), Screen Size (inches), Battery (mAh), Front Camera (MP), and Back Camera (MP). The dropdown menu is for Company Name. The output section on the right displays the estimated price in India. Below the output section is a 'Flag' button. At the bottom of the interface are 'Clear' and 'Submit' buttons.

Input	Value
RAM (GB)	16
Screen Size (inches)	6
Battery (mAh)	6000
Front Camera (MP)	64
Back Camera (MP)	163
Company Name	iqoo

output

Estimated Price in India: ₹60,474

Flag

Clear Submit

Aim: This code launches a Gradio web interface for the mobile price prediction model. The app takes multiple inputs from the user, including RAM, screen size, battery capacity, front and back camera specifications, and company name, and displays an estimated launch price for a mobile in India. Gradio allows users to interactively provide inputs via sliders and dropdown menus and get the predicted output in real time.

Insight: Gradio provides a simple and quick way to create a front-end interface for machine learning models, making them accessible to non-technical users. The app is designed to help users input mobile specifications and instantly get a price estimate based on the trained regression model. This can be helpful for users looking to make informed decisions about mobile purchases or for e-commerce platforms that need a price prediction system.

Conclusion:

This project demonstrates the practical and effective use of machine learning and data science techniques in analyzing and predicting mobile phone launch prices based on core device specifications. By applying a combination of data preprocessing, feature engineering, exploratory analysis, and predictive modeling, the project successfully identifies and quantifies the key factors that influence mobile pricing across brands and configurations.

The use of a Random Forest Regressor enabled robust and accurate prediction of continuous price values, while the classification extension offered deeper insights into market segmentation through the categorization of devices into Low, Medium, High, and Very High price ranges. Visual analytics such as trend plots, outlier detection, heatmaps, and brand distribution further enriched the interpretability of results and supported strategic analysis of evolving mobile technology trends.

The development of a user-friendly Gradio interface adds functional value to the system, transforming it into an interactive price prediction tool. This allows users to input mobile specifications in real-time and receive immediate, data-driven pricing insights, making the model both accessible and actionable.

Looking forward, the project holds strong potential for future enhancement. Expanding the dataset with global pricing data and newer smartphone models would increase the generalizability and accuracy of the model. Further, the integration of advanced algorithms or deep learning models could refine prediction precision, especially for highly dynamic markets. Lastly, deploying the tool as a mobile application or cloud-based API would ensure wider accessibility and real-world usability, positioning this system as a scalable solution for tech companies, consumers, and e-commerce platform