National University of Computer and Emerging Sciences

<Products Data Analysis>

Team

Ifra Ejaz 21L-7508 BS(CS)
Abdullah Awan 21L-7713 BS(CS)

FAST School of Computing

National University of Computer and Emerging Sciences

Lahore, Pakistan

December 2024

Abstract

This project explores the relationship between mobile device specifications and their corresponding prices by leveraging a dataset containing product features such as RAM, Internal Memory, Display Size, Brand, and customer-related attributes like the Number of Ratings. The target variable is Price, which is predicted using machine learning and deep learning models.

The study begins with preprocessing the dataset to handle missing values, normalize numerical features, and encode categorical variables. After thorough data cleaning and feature selection, two predictive models are developed: Random Forest Regressor (machine learning) and a Neural Network (deep learning). Both models are trained and evaluated on the processed data to compare their performance based on key metrics, including Mean Squared Error (MSE) and R^2 Score.

The Random Forest model achieves superior performance with an MSE of 0.00302 and an R^2 Score of 0.9028, outperforming the Neural Network's MSE of 0.00549 and R^2 Score of 0.8234. Insights from the feature importance analysis highlight that specifications like Internal Memory, RAM, and Display Size have a significant influence on pricing trends.

The findings demonstrate the effectiveness of machine learning models for pricing prediction in small-to-medium-sized datasets and underline the potential of deep learning for future scalability. This project provides actionable insights for manufacturers and consumers, aiding strategic pricing decisions and improving market understanding.

1. Introduction

Dataset Overview

This project focuses on analyzing a dataset containing mobile device specifications and their corresponding prices. The dataset includes key features such as RAM, Battery Capacity, Display Size, Internal Memory, and Brand, along with the target variable, Price. It also incorporates customer-related attributes like the Number of Ratings.

The dataset provides a comprehensive view of product specifications, enabling insights into pricing trends and market offerings. It includes 1,345 rows and 8 columns, with preprocessing techniques applied to clean and standardize the data for analysis and modeling.

Relevance

Understanding product specifications and their relationship with pricing is critical for both manufacturers and customers. For manufacturers, it helps design competitive products and pricing strategies. For customers, it simplifies decision-making when comparing products. This analysis serves as a bridge between the two, delivering actionable insights into the mobile device market.

Project Objectives

1. **Standardize and Analyze Product Features:**
    - Preprocess raw product data to ensure consistency and comparability across features like RAM, Battery, and Internal Memory.
    - Handle missing values, outliers, and discrepancies to create a clean dataset for analysis.
2. **Uncover Trends:**
    - Identify relationships between product features and pricing trends.
    - Highlight key market insights, such as the impact of Display Size and Internal Memory on pricing.
3. **Develop Predictive Models:**
    - Build machine learning (Random Forest) and deep learning (Neural Network) models to predict prices based on product features.
    - Evaluate and compare model performance using metrics such as Mean Squared Error (MSE) and R^2 Score.

Expected Outcomes

- Development of accurate predictive models to estimate mobile device prices.
- Identification of significant product features influencing pricing, such as Internal Memory, RAM, and Display Size.
- Comprehensive visualizations and insights to guide business strategies and consumer decisions.
- A comparative analysis of machine learning and deep learning models to determine the most effective approach for price prediction.

2. Data Preparation

1. Data Loading

The dataset used in this project contains specifications and pricing details of various mobile devices. It was loaded into a Python environment using libraries like pandas for efficient data manipulation. The dataset initially contained 1,345 rows and 8 columns, including features like RAM, Battery, Internal Memory, Display Size, and the target variable, Price.

- import pandas as pd
- data = pd.read_csv('Products_Data.csv')

2. Data Exploration

The following code was utilized to load and explore the data:

☐ data.head()

```
data.head()
```

	Brand	Model	Price	Number of Ratings	Display Size	RAM	Battery	Internal Memory
0	Infinix	Zero 40 4G	Rs.70,000	23	6.78	8	500.0	256
1	Samsung	Galaxy Z Flip 6	Rs.385,000	39	6.70	12	4000.0	512
2	Samsung	Galaxy Z Fold 6	Rs.605,000	45	7.60	12	4400.0	512
3	Samsung	Galaxy A05	Rs.25,000	56	6.70	4	5000.0	64
4	Tecno	Phantom V Fold 2 5G	Rs.370,000	37	7.85	12	5750.0	512

☐ data.tail()

```
data.tail()
```

	Brand	Model	Price	Number of Ratings	Display Size	RAM	Battery	Internal Memory
1340	gfive	GFive Disco	Rs 3,199	59	2.40	0.031	3000.0	0.031
1341	gfive	GFive Spark	Rs 2,325	3 Ratings	1.80	0.031	3000.0	0.031
1342	e-tachi	E-Tachi E888	Rs 3,749	38	2.80	0.031	3000.0	0.031
1343	sparx	SparX Edge 20	Rs 5,000	24	6.67	8	5000.0	256
1344	gfive	GFive 4G Style	Rs 6,999	39	2.80	2	4000.0	16

- ☐ data.info()

```
[123] data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1345 entries, 0 to 1344
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Brand              1345 non-null   object
 1   Model              1345 non-null   object
 2   Price              1345 non-null   object
 3   Number of Ratings  1345 non-null   object
 4   Display Size       1345 non-null   float64
 5   RAM                1339 non-null   object
 6   Battery            1342 non-null   float64
 7   Internal Memory    1344 non-null   object
dtypes: float64(2), object(6)
memory usage: 84.2+ KB
```

- ☐ data.describe()

	Display Size	Battery
count	1345.000000	1342.000000
mean	6.249314	5044.555887
std	1.179237	13556.005191
min	0.000000	30.000000
25%	6.500000	5000.000000
50%	6.600000	5000.000000
75%	6.700000	5000.000000
max	7.850000	500018.000000

Data Standardization Method Applied

Column	Format Example	Method Applied
Ram	"8GB", "8 GB + 8 GB EXTENDED"	Extract GB values, return first value
Ram	"8+12GBRAM.UPTO20GB	Extract GB values, return max

	DYNAMIC RAM"	
Ram	"2 GB (expandable)"	Extract numeric, return first value
Battery	"4500mAh", "3500 WhBattery"	Extract mAh value, convert Wh to mAh
Battery	"WhBattery"	Fill missing values with 0
Battery	"Erroneous Entries"	Extract first 4 numeric digits
Internal Memory	"128GB", "64GB, 128GB"	Extract GB values, return max
Display	"6.7 InchesDisplay", "6.67Display"	Extract numeric value, convert to float

3. Data Cleaning

- **Handling Missing Values**
  - **Ram**
    - Missing values in the RAM column were filled using the global mean of the column. This approach assumes that missing values are likely to be close to the average RAM for the entire dataset.
  - **Internal Memory**
    - Missing values in the Internal Memory column were filled using the global mean of the column. This approach assumes that missing values are likely to be close to the average internal memory for the entire dataset.
  - **Display Size**
    - The missing values in the Display Size column were also filled by grouping the data by Brand and Model, just like with RAM. The mean display size within each group was calculated, and missing values were filled with this mean.
  - **Price**
    - The For the Price column, the missing values were filled with the global mean of the prices in the dataset, ensuring that no price data remained incomplete.
- **Duplicate Removal**
  - The dataset was checked for duplicates using the drop_duplicates() function. Any identical rows that represented redundant entries were removed to ensure the dataset contained only unique records. This step is important to avoid over- representing specific data points during analysis.
- **Data Type Conversion**

- Several columns in the dataset contained values as strings or in mixed formats. These were converted to numeric types to facilitate proper analysis:
  - RAM, Battery, Internal Memory, Price, and Number of Ratings were all converted using pd.to_numeric(). This function attempts to convert values into numeric types, and any values that couldn't be converted (e.g., non- numeric strings) were coerced to NaN for further processing.

- **Outlier Detection**

  For each column, the following steps were taken to handle outliers:

  1. **Ram**
     - The RAM column was checked for outliers using the Interquartile Range (IQR) method. This method calculates the first quartile (Q1, the 25th percentile) and the third quartile (Q3, the 75th percentile), and then defines the IQR as Q3 - Q1. Outliers are identified as values below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR.
  2. **Internal Memory**
     - Similarly, the Internal Memory column was checked for outliers using the IQR method. This ensures that extreme values.
  3. **Display Size**
     - The Display Size column also underwent outlier detection using the IQR method. Any display sizes that fell outside the defined range of typical values were considered outliers.
  4. **Battery**
     - The Battery column was processed for outliers using the same IQR method. This was especially important since battery capacities could vary widely, and extreme values would distort the analysis.
  5. **Price**
     - Outliers in the Price column were detected using the IQR method. This ensured that the analysis remained focused on typical price ranges, removing any extreme outlier prices that might have represented errors or unusual entries.
  6. **Number of Ratings**
     - The Number of Ratings column, representing how many times a product was rated, was also analyzed for outliers using the IQR method. Outliers here might represent products that received unusually high or low numbers of ratings, potentially due to errors or anomalies.

4. Data Transformation

Scaling: To ensure that all features contribute equally to the analysis, **Min-Max scaling** was applied to the numerical columns: RAM, Internal Memory, Price, Number of Ratings, and Display Size. This transformation scaled the values to a range between 0 and 1.

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

data[['RAM', 'Internal Memory', 'Price', 'Number of Ratings', 'Display Size']] =
scaler.fit_transform(data[['RAM', 'Internal Memory', 'Price', 'Number of Ratings', 'Display
Size']])
```

- By scaling these columns, the dataset becomes more balanced for future modeling or analysis tasks, especially since the data is not normally distributed.
- **Encoding Categorical Variables**: Categorical features such as Brand and Model were converted into numeric values using **one-hot encoding**. This method creates binary columns for each category, allowing the machine learning models to process categorical data efficiently. We applied one-hot encoding with drop_first=True to avoid multicollinearity.
- After scaling the value in the column of battery are zero because all of the value were almost same

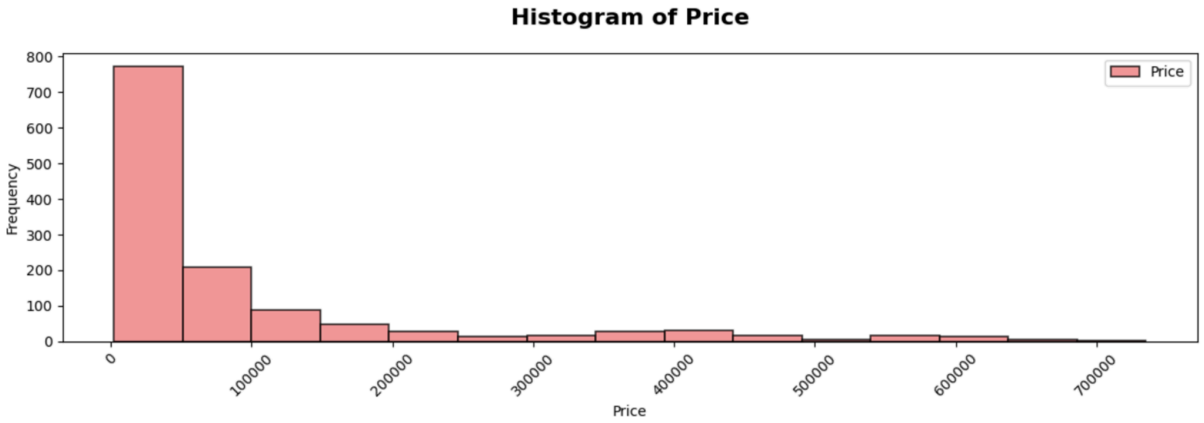
```
data = pd.get_dummies(data, columns=['Brand', 'Model'], drop_first=True)
```

This approach preserves the categorical nature of the data while allowing it to be used effectively in numerical analysis.

3. Data Analysis

1. Univariate Analysis

- **Feature:** Price
- **Graph Type:** Histogram



Description: The histogram visualizes the distribution of the 'Price' feature, with the X- axis representing the price values and the Y-axis representing the frequency of occurrences.

Observation:

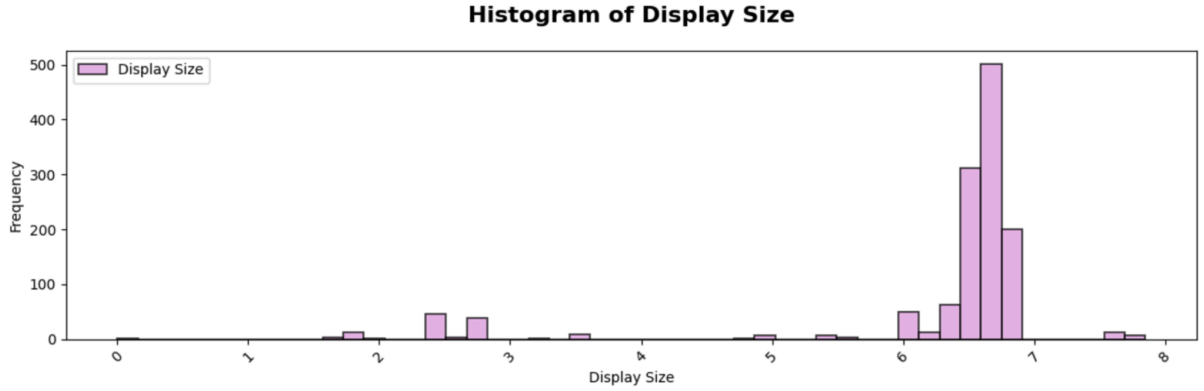
- A long tail extends towards the right of the histogram, indicating that the data is right/positively skewed.
- The highest bar on the left suggests that a large number of products fall into the lower price range.
- As the price increases, the frequency of products in each price range decreases significantly.

Insight:

- The market may consist primarily of low-priced products, with only a few expensive ones.
- Affordability is a dominant factor, but there is a small segment for premium-priced products.

- **Feature:** Display Size

- **Graph Type:** Histogram



Description: The histogram visualizes the distribution of the 'Display Size' feature, with the X-axis representing the display size values and the Y-axis representing the frequency of occurrences

Observation:

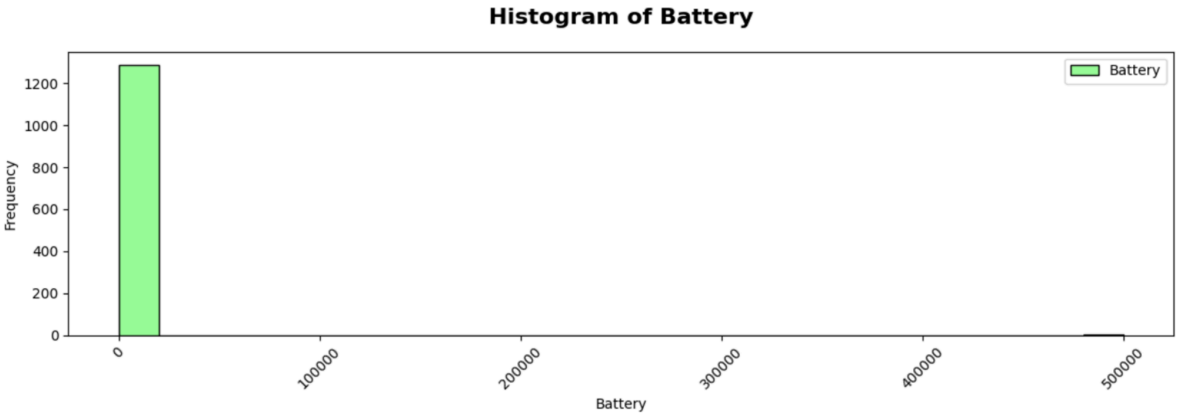
- A long tail extends towards the left of the histogram, indicating that the data is left/negatively skewed.
- Most products have larger display sizes, as evidenced by the concentration of bars toward the right.
- Products with smaller display sizes (the tail on the left) are much less common.

Insight:

- A trend favoring larger screens, which is consistent with market demand for products with bigger displays.

- **Feature:** Battery

- **Graph Type:** Histogram



Description: The histogram visualizes the distribution of the 'Battery' feature, with the X-axis representing the battery capacity values and the Y-axis representing the frequency of occurrences.

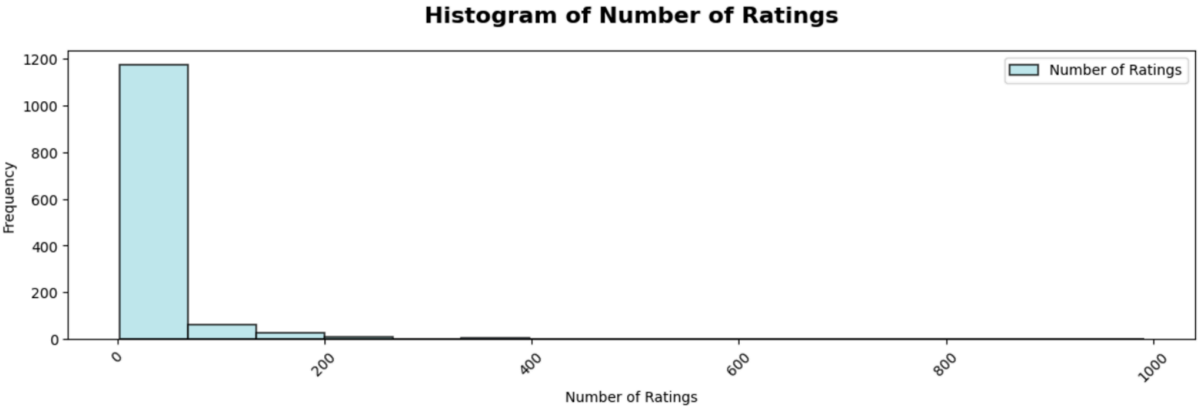
Observation:

- Due to very little variation in the column, almost all the values are concentrated near 0.0 after normalization.
- A lack of diversity in the battery capacities is evident.

Insight:

- The limited variation in battery capacity suggests that most products offer similar battery performance, potentially indicating a standardization in the market.

- **Feature:** Number of Rating
- **Graph Type:** Histogram



Description: The histogram visualizes the distribution of the 'Number of Ratings' feature,

with the X-axis representing the number of ratings and the Y-axis representing the frequency of occurrences.

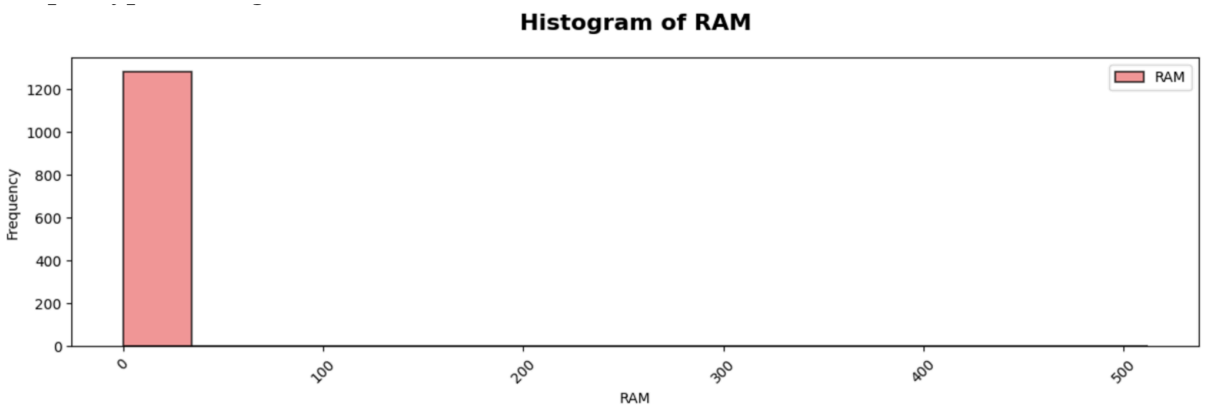
Observation:

- A long tail extends towards the right of the histogram, indicating that the data is right/positively skewed
- Most of the products have a very low number of ratings.

Insight:

- This suggests that the majority of the products are either new, less popular, or have not been widely rated by customers.
- Only a handful of products received a high number of ratings

- **Feature:** Ram
- **Graph Type:** Histogram



Description: The histogram visualizes the distribution of the 'RAM' feature, with the X-axis representing the RAM values and the Y-axis representing the frequency of occurrences.

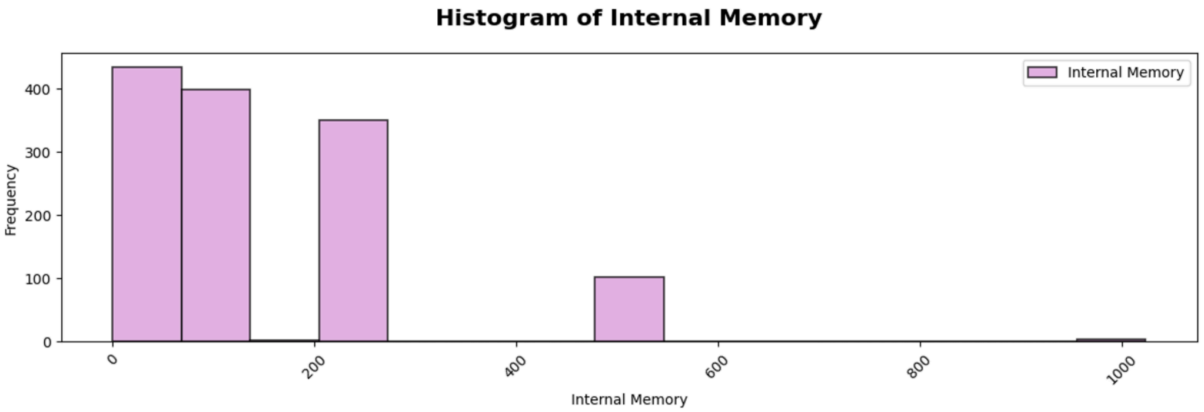
Observation:

- Almost all the values are concentrated near zero, indicating minimal variation in the RAM values
- A lack of diversity in the RAM capacities is evident.

Insight:

- The limited variation in RAM suggests that most products have similar RAM capacity, which could indicate a standardized offering in the market

- **Feature:** Internal Memory
- **Graph Type:** Histogram



Description: The histogram visualizes the distribution of the 'Internal Memory' feature, with the X-axis representing the internal memory values and the Y-axis representing the frequency of occurrences.

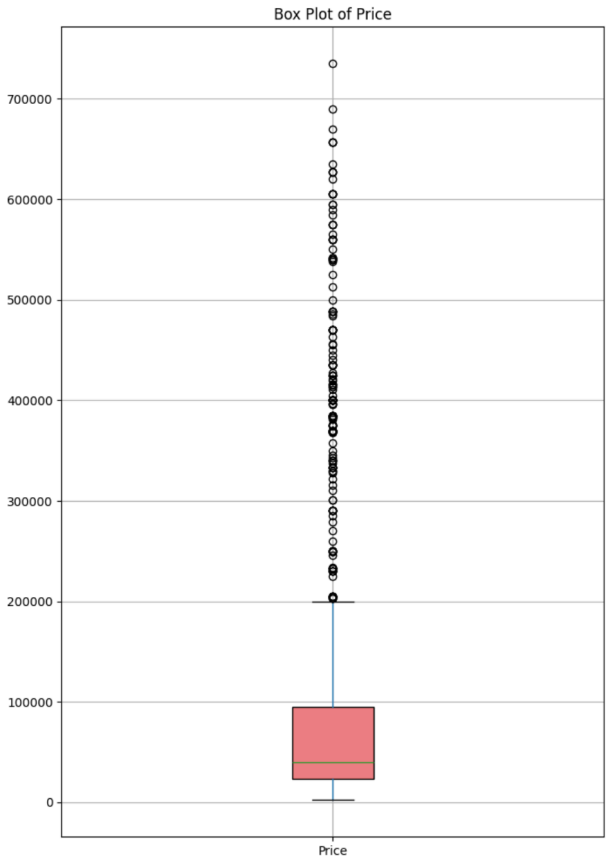
Observation:

- The histogram shows a left-skewed distribution, where most of the frequency is concentrated near the lower range of the normalized internal memory values.
- As the internal memory size increases (toward 1.0), the frequency drops significantly.
- Higher internal memory capacities are significantly less common, as seen in the smaller bars toward the right.
- Internal memory is typically a discrete variable (e.g., 64GB, 128GB, 256GB, etc.), and the histogram reflects this, with clear gaps.

Insight:

- There is a higher demand for products with lower internal memory capacities, which could be driven by cost considerations.
- The smaller number of higher capacity products may indicate a premium segment targeting users with greater storage needs.

- **Feature:** Price
- **Graph Type:** BoxPlot



Description: The box plot visualizes the distribution of the 'Price' feature, highlighting the median, quartiles, and any potential outliers.

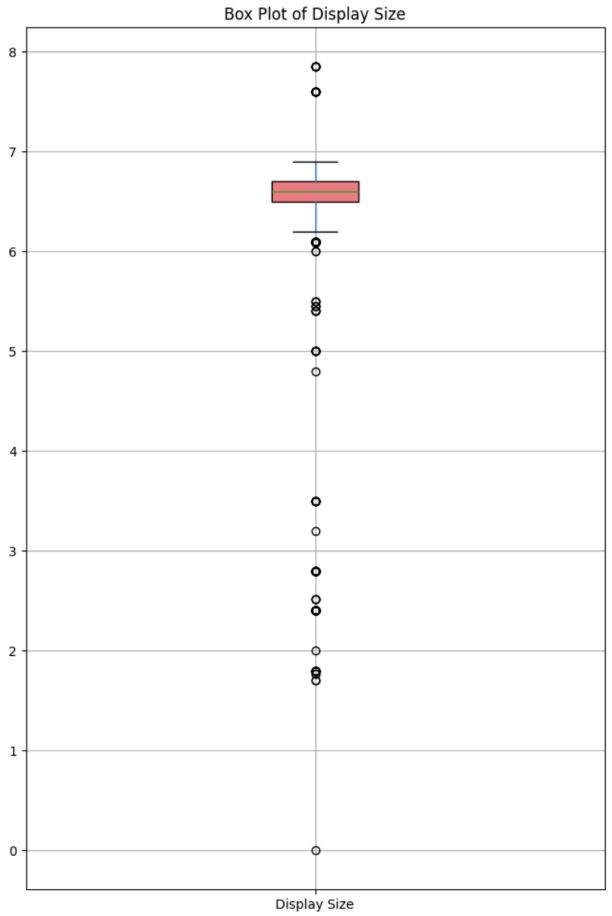
Observation:

- The median price is towards the lower end, indicating a concentration of products with lower prices.
- There are numerous outliers present, which suggests a significant number of products are priced much higher than the rest.
- The interquartile range (IQR) indicates that most products fall within a relatively low price range, with a smaller segment priced substantially higher.

Insight:

- The presence of many outliers implies a significant disparity in product pricing, with a few premium products driving up the price range.
- The overall pricing pattern may indicate a competitive budget market with a limited premium segment.

- **Feature:** Display Size
- **Graph Type:** BoxPlot



Description: The box plot visualizes the distribution of the Display feature, highlighting the median, quartiles, and any potential outliers.

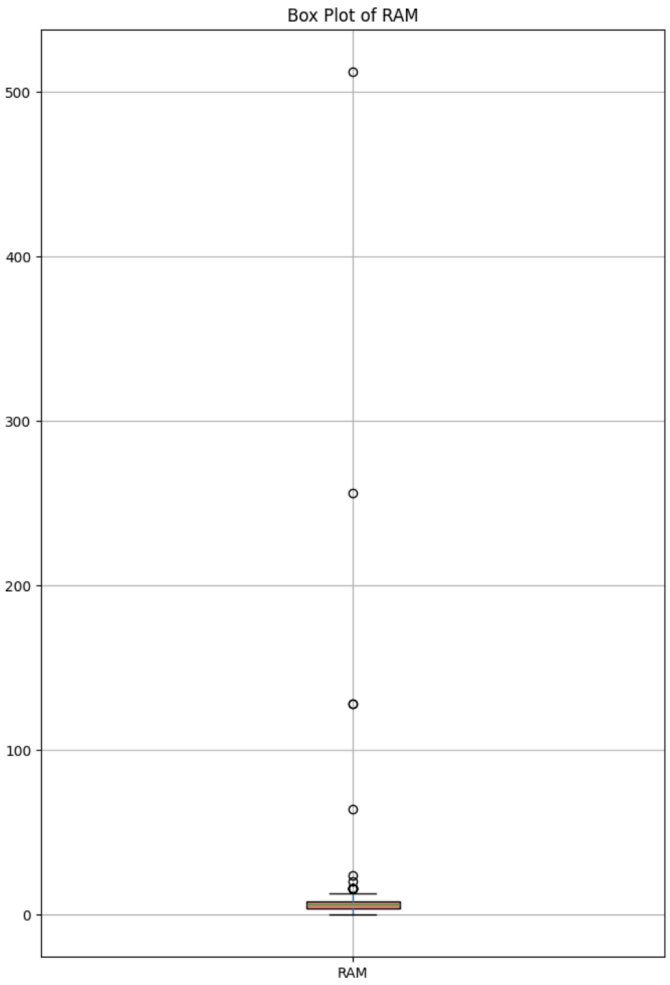
Observation:

- The median display size is centered around 6 to 7 inches, indicating a preference for mid-to-large screens.
- There are several outliers on both ends, suggesting that while most products have a display size in the mid-range, there are a few with significantly smaller or larger screens.
- The interquartile range (IQR) indicates that the majority of products fall within a common size range, with some variation.

Insight:

- There is a higher demand for products with lower internal memory capacities, which could be driven by cost considerations.
- The smaller number of higher capacity products may indicate a premium segment targeting users with greater storage needs.

- **Feature:** Ram
- **Graph Type:** BoxPlot



Description: The box plot visualizes the distribution of the 'RAM' feature, highlighting the median, quartiles, and any potential outliers.

Observation:

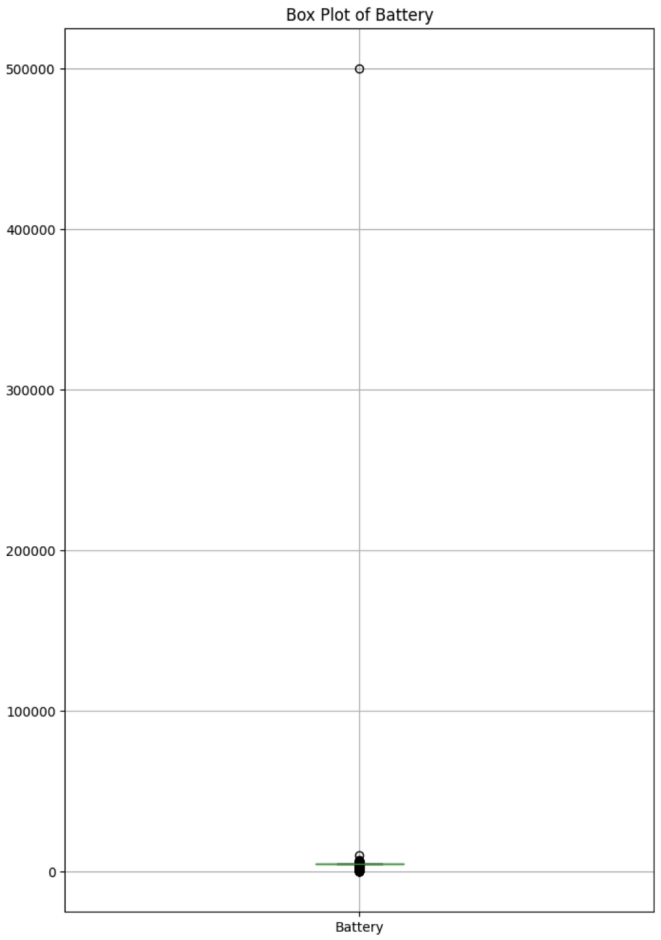
- The median RAM value is quite low, indicating that most products offer minimal memory capacity.
- There are several outliers, with a few products offering significantly higher RAM values compared to the rest.

- The interquartile range (IQR) is narrow, suggesting that the majority of products have similar, limited RAM capacity.

Insight:

- The limited RAM capacity for most products may reflect a budget market focus, where high RAM is less common and more expensive.
- The presence of outliers suggests that there is a small segment of products targeting power users or those who need enhanced performance.

- **Feature:** Battery
- **Graph Type:** BoxPlot



Description: The box plot visualizes the distribution of the 'Battery' feature, highlighting the median, quartiles, and any potential outliers.

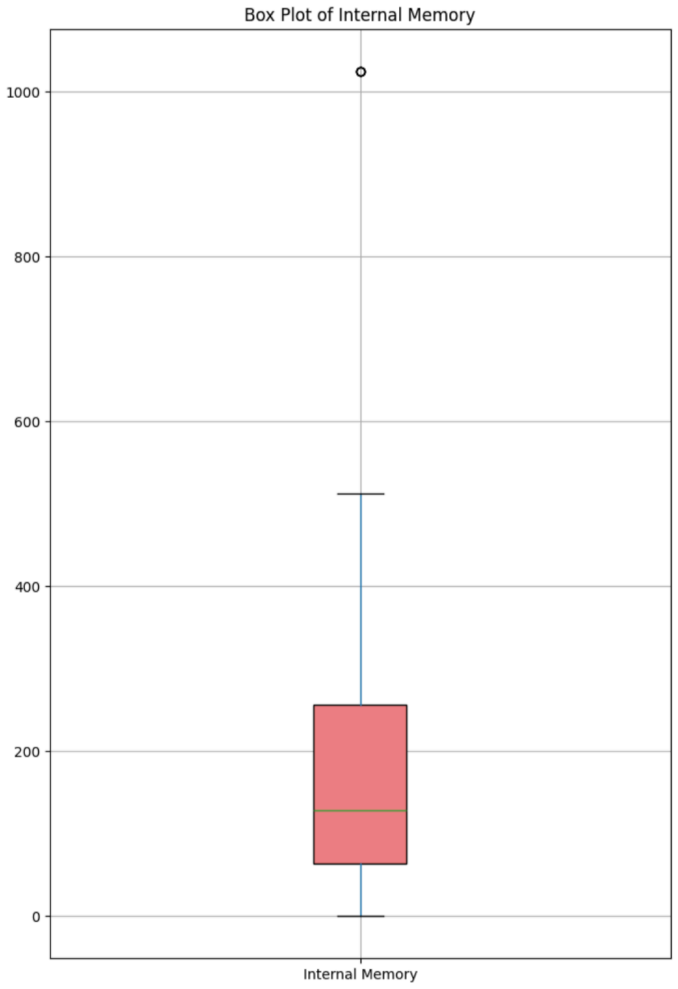
Observation:

- The median battery capacity is very low, indicating minimal variation in battery performance across products.
- There is a significant outlier at the top, representing a product with a much higher battery capacity compared to the rest.
- The interquartile range (IQR) is extremely narrow, showing that most products have almost identical battery capacities.

Insight:

- The limited battery capacity indicates a standardized offering, possibly to meet budget constraints.
- The presence of a high outlier suggests that there is a very small segment offering higher battery capacity, potentially targeting power users.

- **Feature:** Internal Memory
- **Graph Type:** BoxPlot



Description: The box plot visualizes the distribution of the 'Internal Memory' feature, highlighting the median, quartiles, and any potential outliers.

Observation:

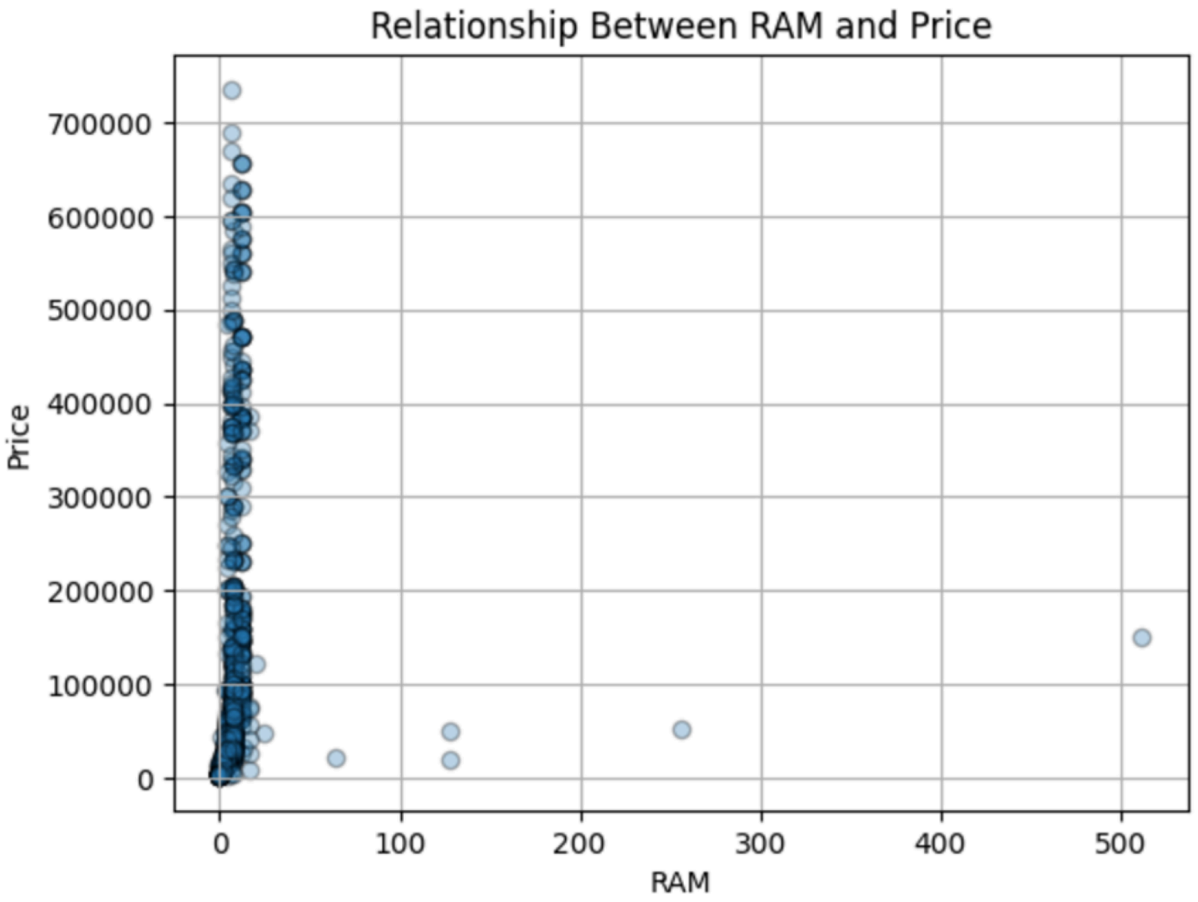
- The median internal memory is around 200 GB, suggesting that most products offer a mid-range storage capacity.
- There is a significant outlier at the top, representing a product with a much larger internal memory compared to the rest.
- The interquartile range (IQR) is relatively wide, indicating a varied range of storage capacities available in the market.

Insight:

- The presence of a high outlier suggests that there is a niche segment for high-capacity internal memory products.
- The wide IQR indicates that manufacturers offer a variety of internal memory options, catering to different consumer needs and price points.

2. Bivariate and Multivariate Analysis

- **Scatter Plot:** Relation between Ram & Price
- **Graph Type:** Scatter Plot



Description: The scatter plot visualizes the relationship between 'RAM' and 'Price', with the X-axis representing the RAM values and the Y-axis representing the price values.

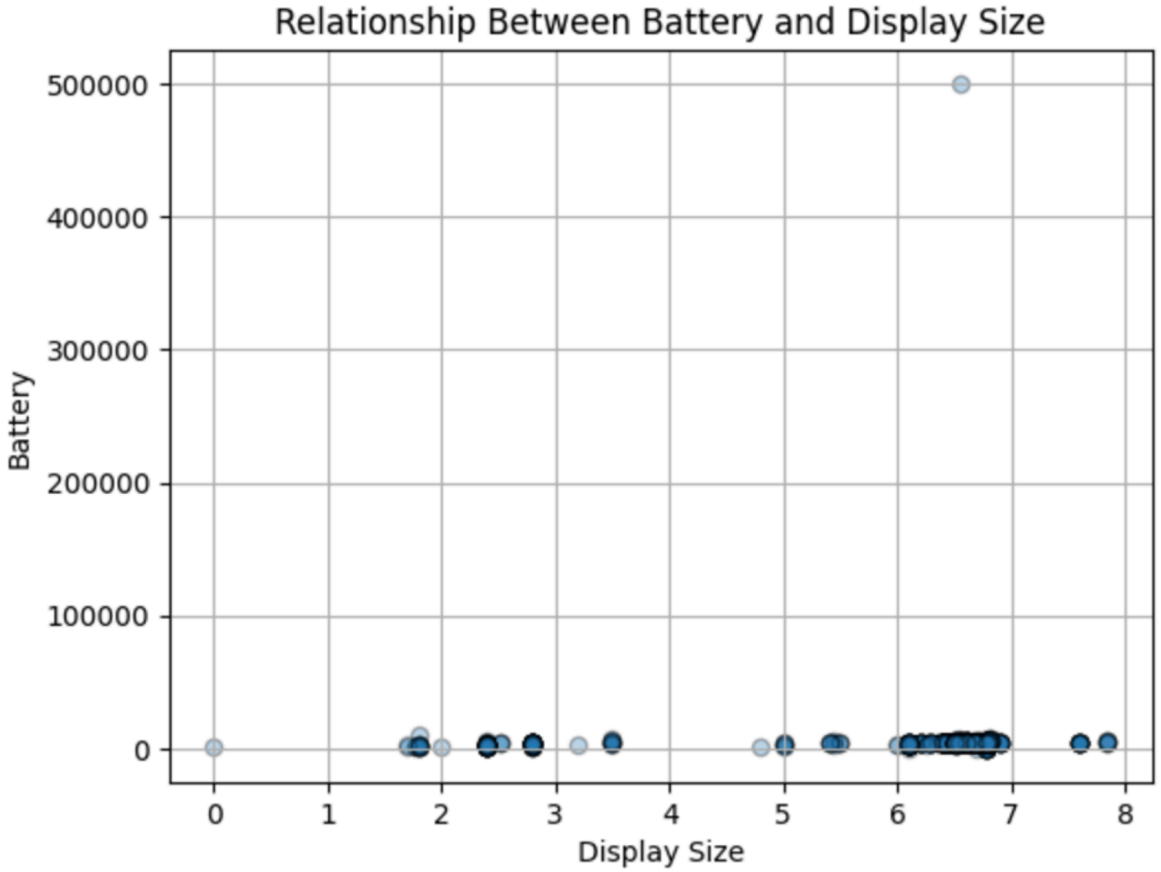
Observation:

- Most data points are clustered around lower RAM values (below 20-30 GB).
- This indicates that the majority of devices have smaller RAM sizes, and there is little variation in the price for these smaller RAM sizes.
- A few data points with significantly higher RAM values (above 100 GB) are present, but they do not follow a consistent trend in terms of price.
- There does not seem to be a clear linear relationship between RAM size and price.
- There are some very high-priced devices regardless of their RAM.

Insight:

- The lack of a strong trend suggests that RAM might not be the sole or dominant factor determining price.
- Additional variables may need to be analyzed for a more comprehensive understanding of price determinants.

- **Scatter Plot:** Relation between Display & Battery
- **Graph Type:** Scatter Plot



Description: The scatter plot visualizes the relationship between 'Display Size' and 'Battery', with the X-axis representing the display size values and the Y-axis representing the battery capacity values.

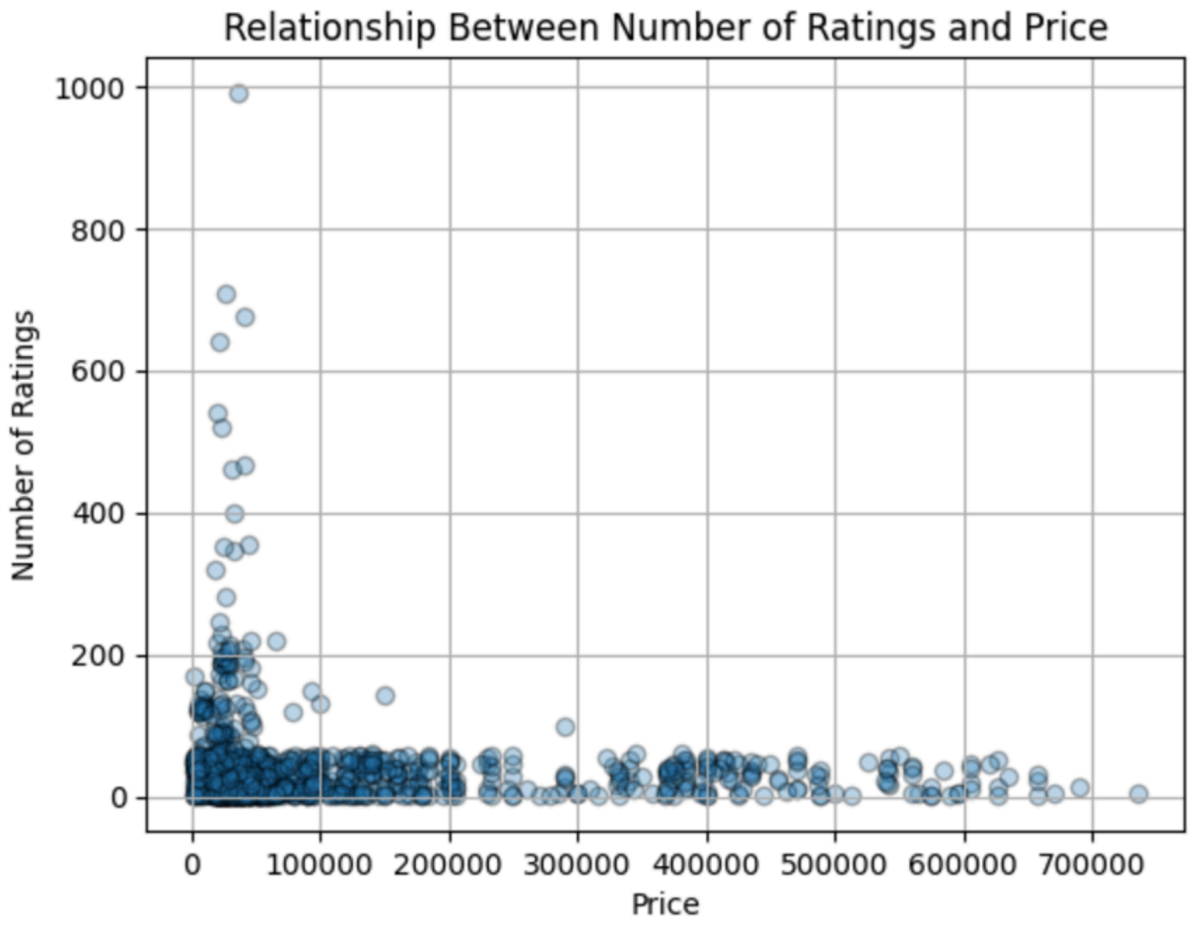
Observation:

- Most data points are concentrated in the range of display sizes between 5 and 7 inches
- The battery values for these display sizes do not show a clear upward trend, indicating that larger screens do not necessarily have significantly larger batteries
- The plot does not exhibit a strong or consistent relationship between display size and battery capacity.
- For smaller display sizes (less than 5 inches), there are fewer data points.

Insight:

- The lack of a strong trend suggests that display size alone may not be the primary factor determining battery capacity.

- **Scatter Plot:** Relation between Number of Rating & Price
- **Graph Type:** Scatter Plot



Description: The scatter plot visualizes the relationship between 'Number of Ratings' and 'Price', with the X-axis representing the price values and the Y-axis representing the number of ratings.

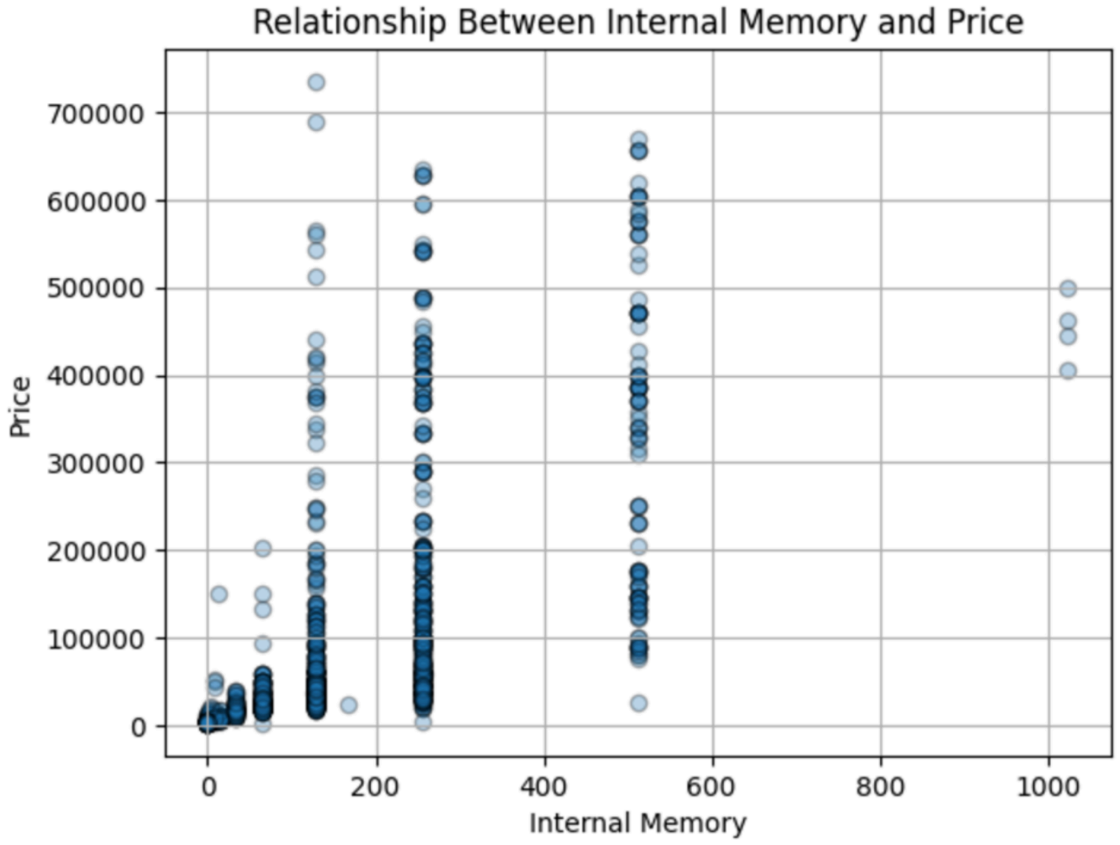
Observation:

- Most data points with a high number of ratings are concentrated at lower price ranges (below 100,000). This suggests that more affordable products tend to be more popular or accessible, receiving more ratings.
- High-priced products (above 300,000) have fewer ratings, which could indicate limited affordability or a niche market for these products.
- A few outliers exist with exceptionally high ratings, but these are almost exclusively in the lower price range.

Insight:

- Lower-priced products tend to attract a higher number of ratings, suggesting they are more popular or widely purchased.
- Higher-priced products, while likely targeting a more specialized audience, tend to receive fewer ratings, reflecting their limited reach..

- **Scatter Plot:** Relation between Internal Memory & Price
- **Graph Type:** Scatter Plot



Description: The scatter plot visualizes the relationship between 'Internal Memory' and 'Price', with the X-axis representing the internal memory values and the Y-axis representing the price values.

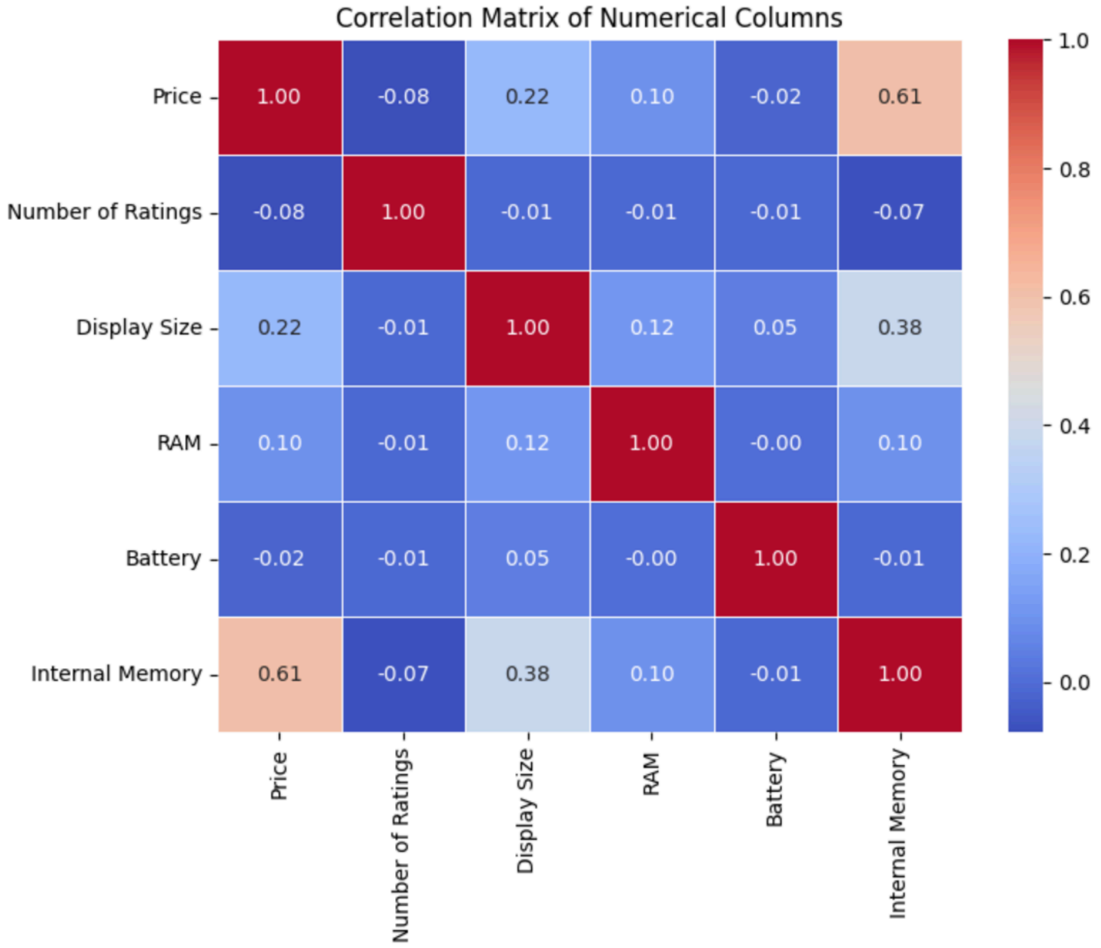
Observation:

- The data points are grouped into distinct clusters corresponding to common internal memory sizes, such as 64GB, 128GB, 256GB, 512GB, and 1TB. This reflects standard memory configurations for most products..
- Within each cluster, there is an upward trend in price as the internal memory size increases. This suggests that more storage tends to increase the cost.

Insight:

- The relationship appears somewhat non-linear, with diminishing returns in price for very high memory sizes.

- **Correlation Matrix:** of Numerical Column
- **Graph Type:** HeatMap



Description: The correlation matrix heatmap visualizes the pairwise correlation between numerical features in the dataset, with each cell color representing the strength and direction of the correlation.

- **Strongest Correlation:** There is a strong positive correlation between internal memory and price. This suggests that products with more storage tend to have higher prices, aligning with the earlier scatter plot observations.

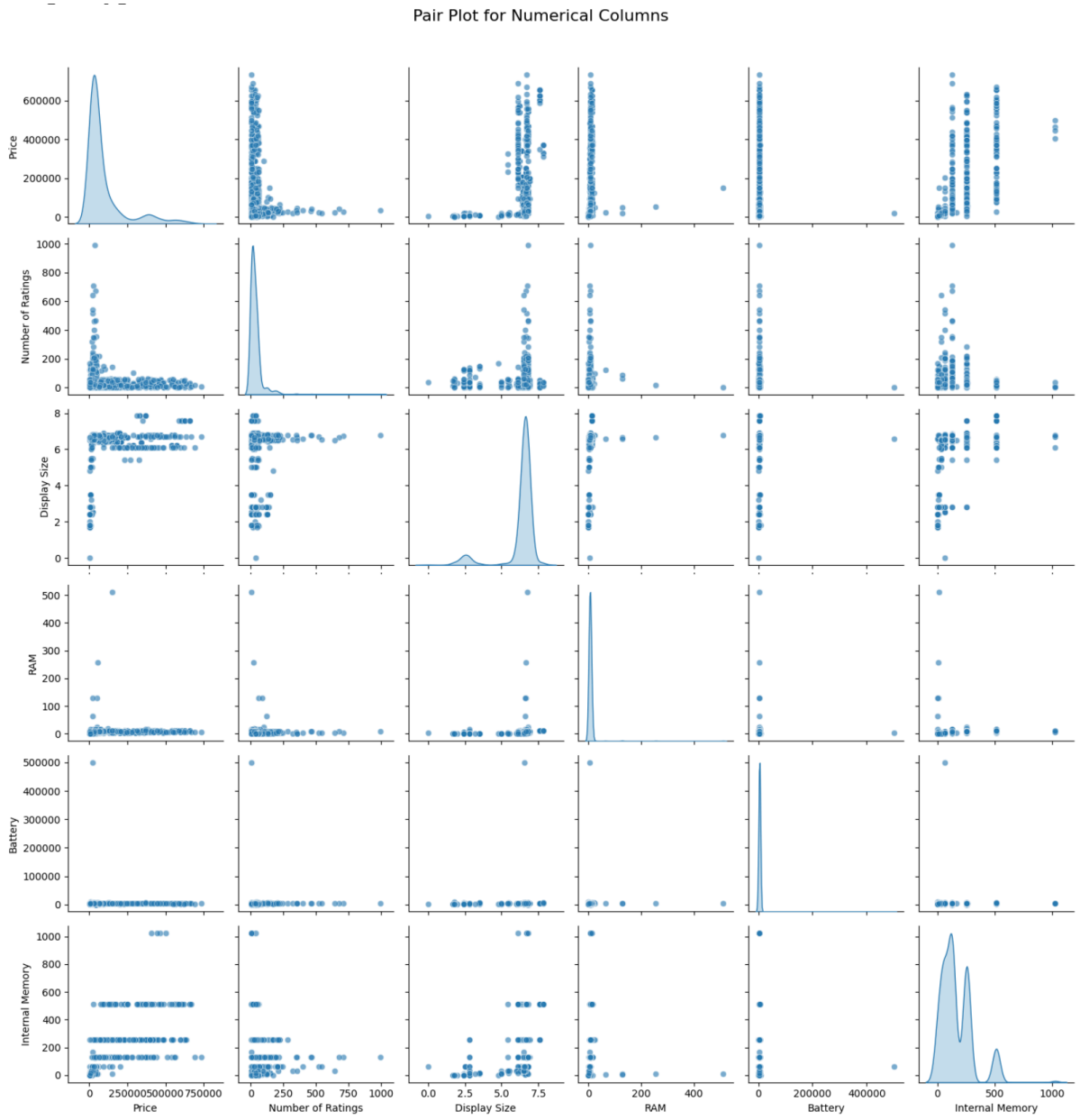
- **Moderate Correlation:** There is a weak to moderate positive correlation between display size and price. Larger screens may slightly contribute to higher prices but are not a dominant factor.

- **Weak Correlation:** The correlation between RAM and price is very weak, indicating that RAM alone does not strongly determine the price. There is a moderate correlation between display size and internal memory, suggesting that devices with larger screens might also tend to have more internal memory.

- **Negligible or Negative Correlation:** A weak negative correlation suggests that lower-priced products are more likely to receive higher numbers of ratings, which aligns with affordability and accessibility for a wider audience. Virtually no correlation between battery size and price, indicating battery capacity does not significantly influence product cost. A slight negative correlation indicates that products with higher internal memory are not necessarily the most frequently rated.

- **Correlation Matrix:** of Numerical Column
- **Graph Type:** Pait Plot

Product Data Analysis



Description: The pair plot visualizes the relationships between multiple numerical features in the dataset by plotting scatter plots and kernel density estimates for each pair of features.

Observation:

- The pair plot reveals potential correlations and distributions among numericalfeatures such as price, RAM, internal memory, display size, battery, and the number of ratings
- Most scatter plots show clusters of data points, with a few outliers present.
- Kernel density plots on the diagonal provide an insight into the distribution of each feature, indicating skewness or the presence of multiple modes.
- The plot reveals that price has a positive relationship with internal memory, as seen in the cluster trend.

Insight:

- The clustering of data points suggests the presence of standardized configurations, which may indicate a competitive product market.
- The density plots reveal skewness in price and memory, which might indicate a predominant focus on budget products with fewer premium options.

3. Feature Analysis

- **Graph Type:** Heat Map

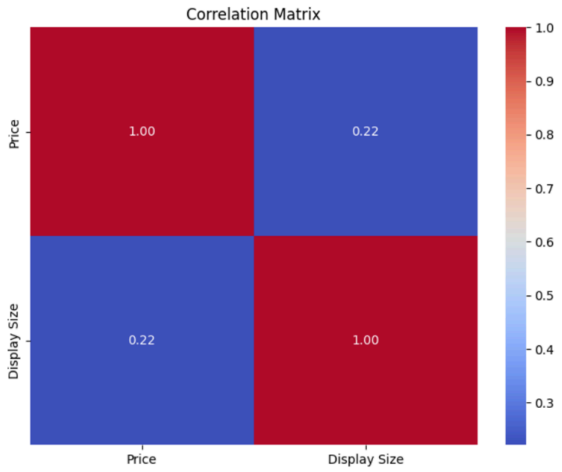


Figure 1: Price & Display

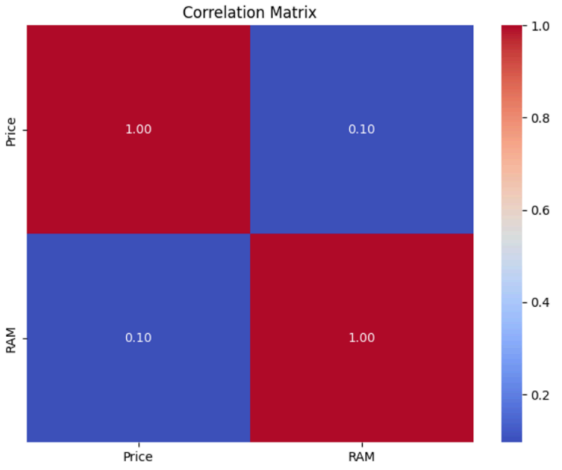


Figure 2: Price & Ram

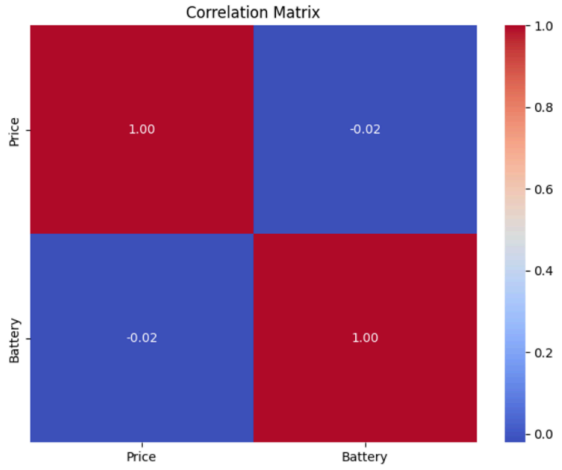


Figure 3: Price & Battery

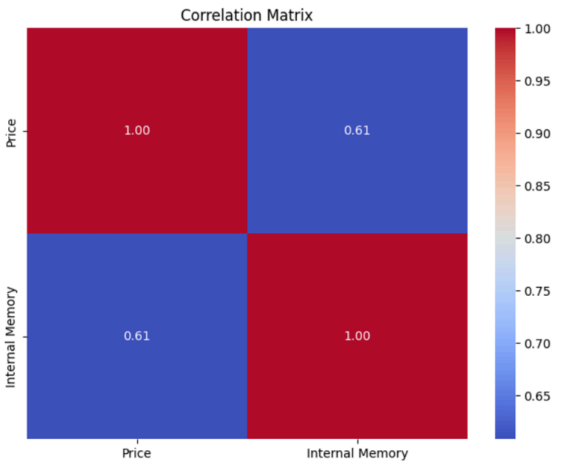


Figure 4: Price & Internal Memory

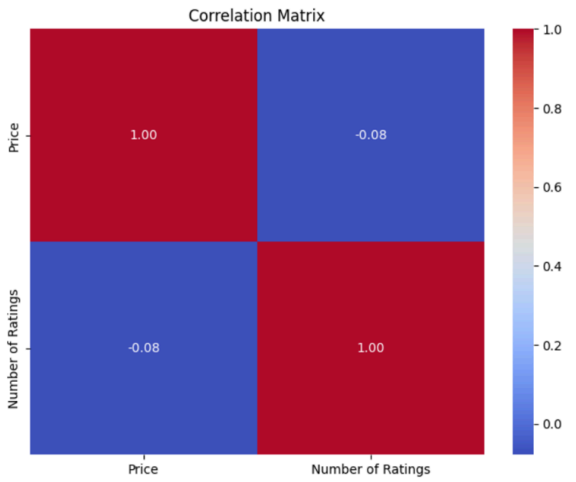


Figure 5: Price & Number of Rating

Observation:

- Among the features analyzed, Internal Memory shows the strongest correlation with the target variable (Price). It should be prioritized for predictive modeling

Analyzing Correlation Between Categorical Features and Target Variable Using Group-By Summaries:

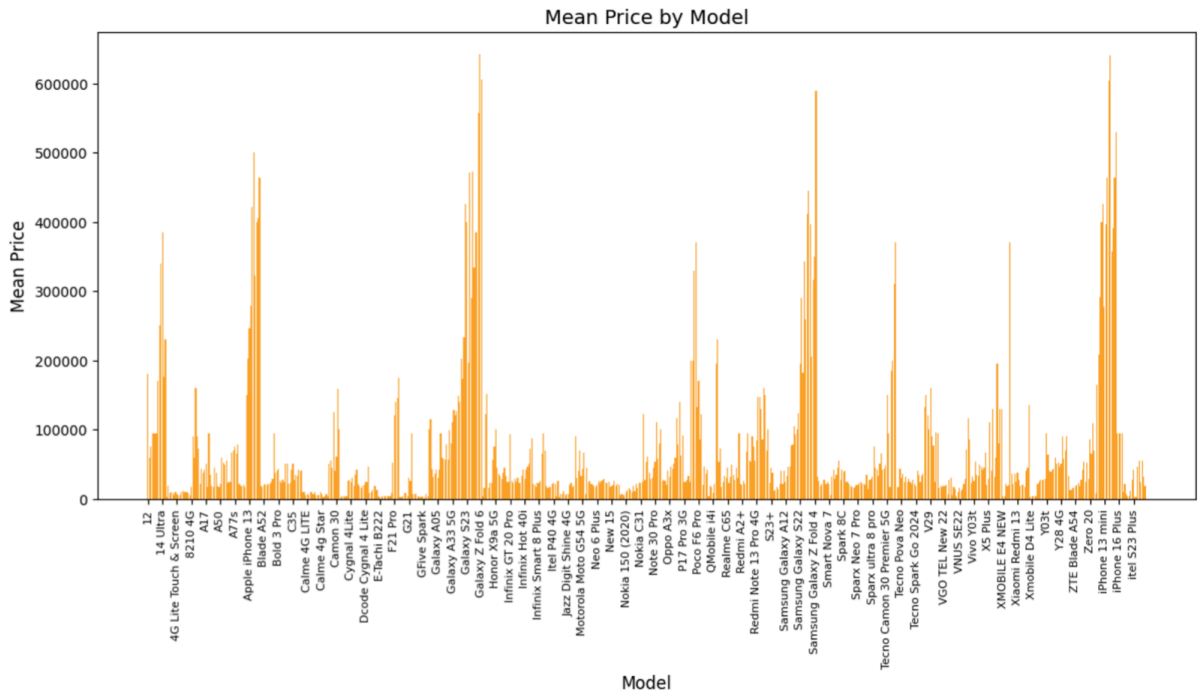


Figure 6: Mean Price & Model

Observation:

- **By Model:**
  - The mean prices by model display high variability, with certain models having extremely high average prices compared to others.
  - The variation across different models within the same brand highlights the availability of both budget and premium products under the same brand umbrella.
  - The scatter in mean prices indicates a broad pricing strategy aimed at various customer segments, from cost-conscious buyers to those looking for premium features.
  - Median values are generally aligned with the mean prices, but a few models show discrepancies, indicating possible outliers or skewed pricing distributions.

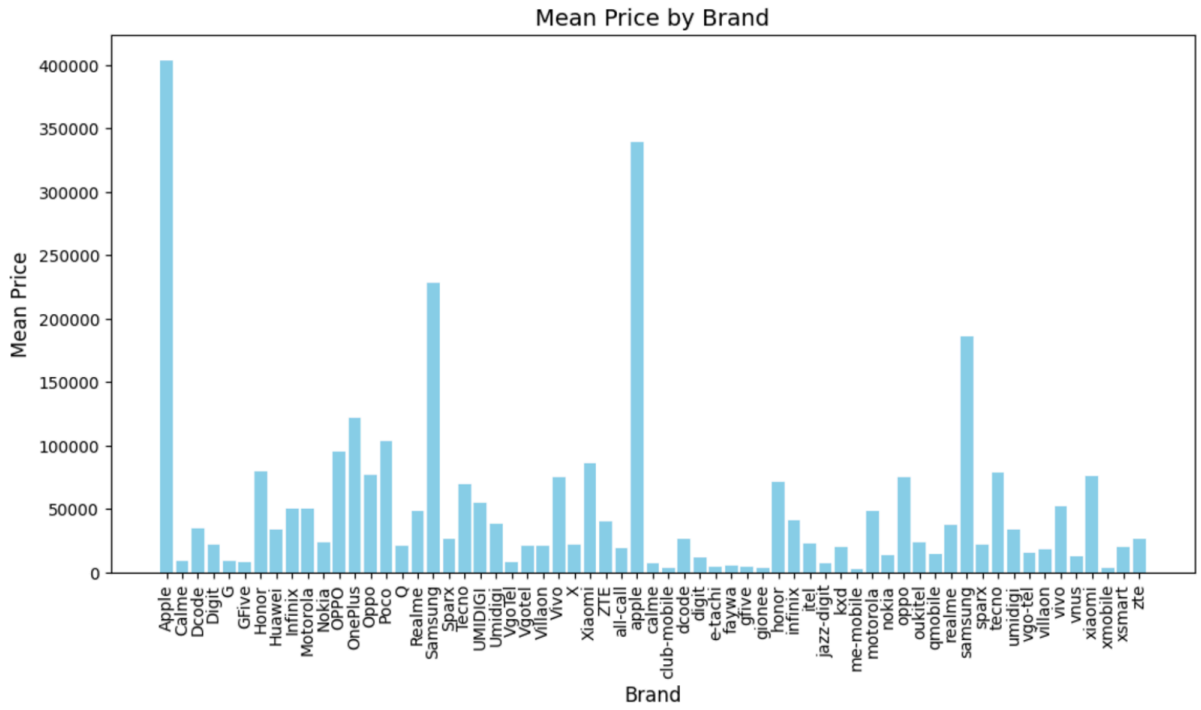


Figure 7: Mean Price & Brand

Observation:

- **By Brand:**
  - The analysis of mean prices by brand reveals significant variation across brands.
  - Apple has the highest average price, indicating its positioning in the premium segment.
  - Brands like Xiaomi, Samsung, and others show relatively moderate mean prices, suggesting they cater to both budget and mid-range segments.
  - Some brands have a noticeably low mean price, indicating a focus on budget-friendly products.

- The median values are generally close to the mean for most brands, suggesting that the price distributions are not highly skewed.

4. Model Training

1. Feature Selection

- **Selected Feature**
  - **Internal memory:** strong positive correlation wiith price because devices with larger internal memory typically have higher prices due to better storage capacity.
  - Scatter plots reveal some upward trend between Display Size and Price.
  - RAM could interact with other features (like Internal Memory) in influencing price, so it's worth exploring its contribution further.
  - Some brands may generally produce higher-priced devices than others (e.g., premium brands like Apple or Samsung).
  - Model is categorical and represents the specific device type, which can heavily influence Price.
- **Excluded Feature**
  - Battery has very weak correlation with Price, Histogram and scatter plots show almost no variance in battery values after normalization, which suggests it carries minimal information.
  - The number of ratings reflects a product's popularity rather than its price. For example, a cheap device may have many ratings, but that doesn't mean it's expensive.

```python
#since the model column is one hot encoded, we have individual columns for each category so we are gathering all the columns that are encoded into a single variable
model_columns = [col for col in data.columns if col.startswith('Model_')]
selected_features = ['RAM_Scaled',  'Display_Size_Scaled', 'Internal_Memory_Scaled','Brand_Encoded']
selected_features.extend(model_columns)


X = data[selected_features]
y = data['Price_Scaled'] #storing target variable into y

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Training data shape:", X_train.shape)
print("Testing data shape:", X_test.shape)
```

Python

```
Training data shape: (1029, 692)
Testing data shape: (258, 692)
```

Observation:

- Relevant features (RAM_Scaled, Internal_Memory_Scaled, etc.) and the target column (Price_Scaled) are selected.
- The data is split into training (80%) and testing (20%) sets.

Insight:

- Selecting the right features ensures the model focuses on relevant predictors.
- Splitting the dataset prevents overfitting and helps evaluate model performance on unseen data.

2. Model Selection

Two predictive models were chosen to evaluate the dataset:

1. **Random Forest Regressor (Machine Learning):**
   - A robust ensemble learning algorithm known for handling non-linear relationships.
   - Suitable for small-to-medium-sized datasets and provides feature importance insights.
2. **Neural Network (Deep Learning):**
   - A fully connected neural network designed to model complex relationships in the data.
   - Chosen to test the scalability and generalization ability of deep learning for price prediction.

3. Model Training

- **Random Forest**:
  - The model was trained using the selected features with 100 estimators (`n_estimators=100`) and a maximum depth of 10 (`max_depth=10`) to balance performance and computational efficiency.
  - The training dataset (`X_train`, `y_train`) was split using an 80-20 train-test split to ensure reliable evaluation.
- **Neural Network**:
  - A neural network with two hidden layers (64 neurons each) was designed using the ReLU activation function.
  - The input features were normalized, and the model was compiled with the Adam optimizer and Mean Squared Error (MSE) as the loss function.
  - The model was trained for 50 epochs with a batch size of 32, using the same 80-20 split for train-test data.

```python
#Machine Learning Model
rf = RandomForestRegressor(n_estimators=100, max_depth=10, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
mae_rf = mean_absolute_error(y_test, y_pred_rf)
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)

print(f"Random Forest - MAE: {mae_rf}, MSE: {mse_rf}, R2: {r2_rf}")
```

Python

```
Random Forest - MAE: 0.030210937384977 96, MSE: 0.0030223016953574 55, R2: 0.9027580822949589
```

```python
#Deep Learning Model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

model = Sequential([
    Dense(64, input_dim=X_train.shape[1], activation='relu'),
    Dense(64, activation='relu'),
    Dense(1)
])

model.compile(optimizer='adam', loss='mse', metrics=['mae'])

history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=50, batch_size=32, verbose=1)

y_pred_nn = model.predict(X_test)

mse_nn = mean_squared_error(y_test, y_pred_nn)
r2_nn = r2_score(y_test, y_pred_nn)

print(f"Neural Network - MSE: {mse_nn}, R2: {r2_nn}")
```

Python

4. Model Evaluation

Random Forest Regressor

- **MAE (Mean Absolute Error):** 0.0302
  - The model's average prediction error is approximately 0.0302 in scaled units. This indicates that the model's predictions are close to the actual prices.
- **MSE (Mean Squared Error):** 0.00302
  - The low MSE suggests that the model performs well, penalizing larger errors less frequently.
- **R² Score:** 0.9028
  - The Random Forest Regressor explains approximately 90.28% of the variance in the price data, showing excellent predictive performance.

```
Random Forest - MAE: 0.030210937384977796, MSE: 0.003022301695357455, R2: 0.9027580822949589
```

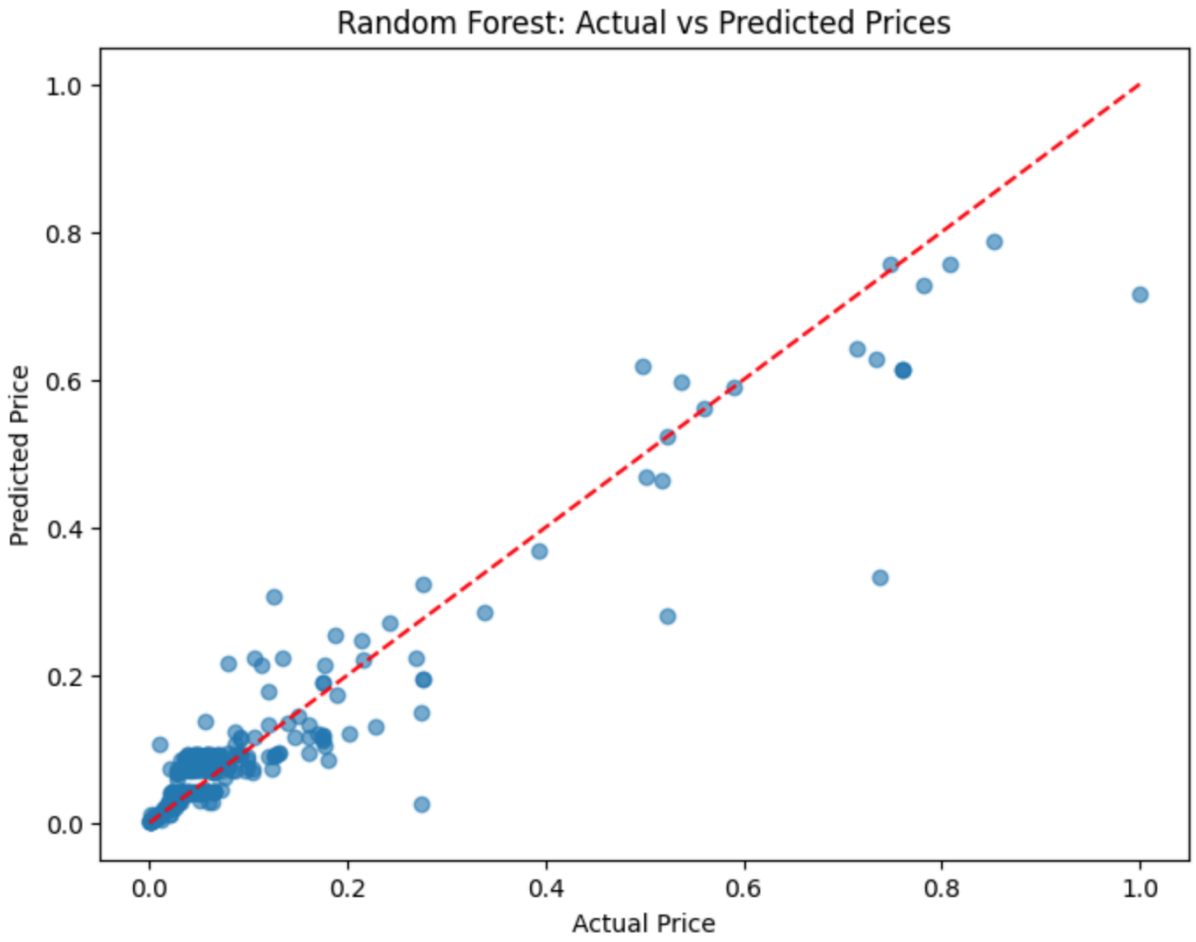
Neural Network

- **MSE (Mean Squared Error):** 0.00549
  - Higher than the Random Forest's MSE, indicating slightly less accurate predictions.
- **R² Score:** 0.8234
  - The model explains 82.34% of the variance, showing reasonable predictive power but weaker than Random Forest.

```
Epoch 50/50
33/33 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 3.9489e-04 - mae: 0.0121 - val_loss: 0.0055 - val_mae: 0.0488
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
Neural Network - MSE: 0.005490082834636752, R2: 0.8233577461774662
```

5. Comparison

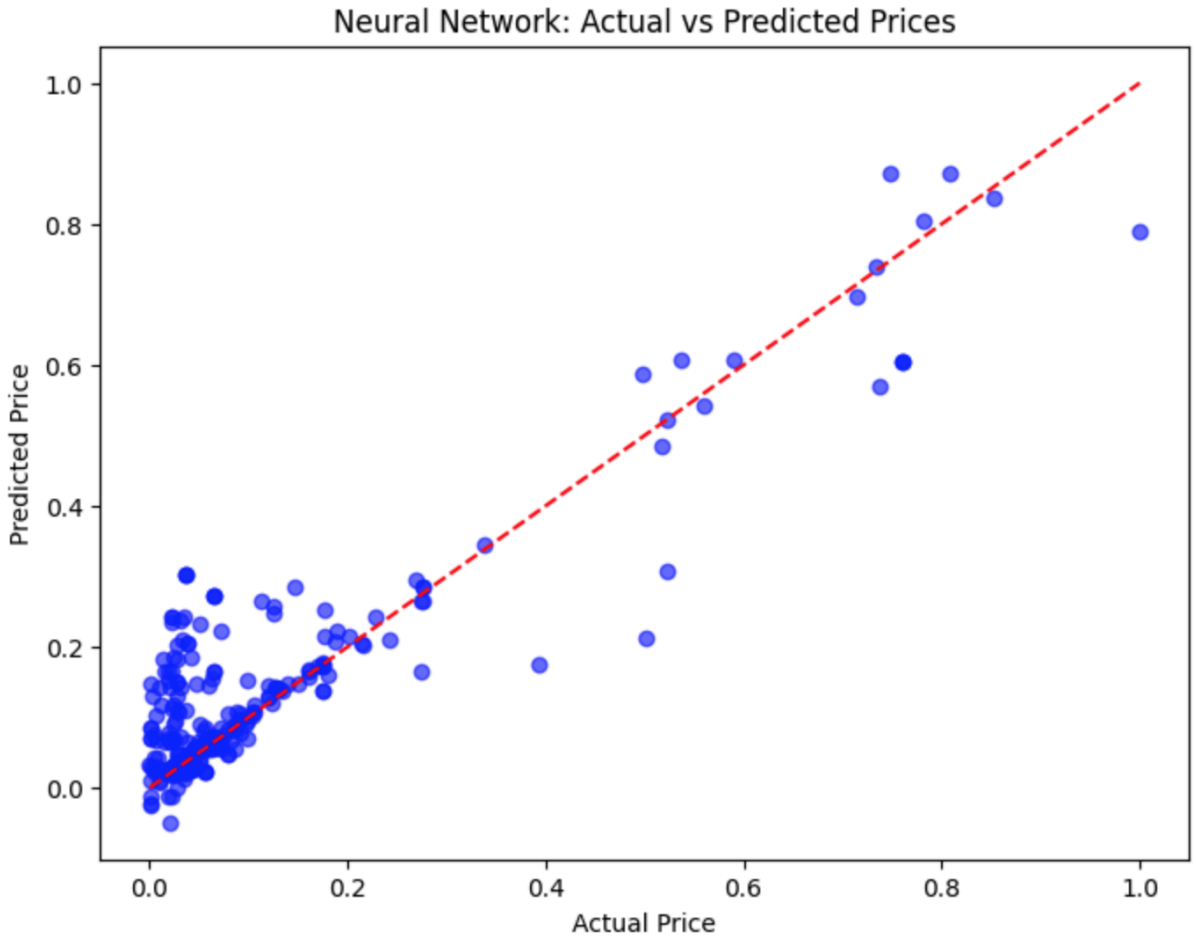
- **Random Forest:** Actual VS Predicted Price



Observation:

- Many data points are clustered near the lower price range (bottom left corner). This suggests that the dataset has more examples with lower prices.
- As the price increases, the predictions tend to deviate more from the diagonal line. This may indicate that the model struggles to predict higher-priced examples accurately which may be due to Insufficient training examples for higher-priced items or Model limitations in capturing relationships for extreme values.

- **Neural Network:** Actual VS Predicted Price



Observation:

- The plot shows that the Neural Network captures the trend reasonably well for many data points, but there are deviations, especially for extreme values.

5. Conclusion

This project successfully explored and predicted mobile device prices based on key specifications, leveraging machine learning and deep learning techniques. After comprehensive data preparation, feature selection, and model evaluation, the following key findings and insights were identified:

1. **Key Findings**:
    - The **Random Forest Regressor** outperformed the Neural Network, achieving a superior **R^2 score of 0.9028** and the lowest error metrics (**MAE: 0.0302, MSE: 0.00302**).
    - Features such as **Internal Memory**, **RAM**, and **Display Size** were identified as the most influential attributes in determining device prices.

    - The **Neural Network**, while capable of modeling non-linear relationships, performed slightly worse due to the dataset size and potential underfitting, achieving an **R^2 score of 0.8234** and **MSE of 0.00549**.
2. **Insights**:
    - Random Forest Regressor demonstrated its strength in handling small-to-medium-sized datasets with non-linear relationships, making it the preferred model for this application.
    - Effective preprocessing and feature scaling played a significant role in improving the predictive performance of both models.

This study highlights the importance of combining domain knowledge with robust predictive modeling to derive actionable insights into pricing strategies.

Future Work

Building upon the foundation of this project, several potential improvements and extensions are suggested to enhance the model's applicability and performance:

1. **Dataset Improvements**:
    - Expand the dataset by including more product records and additional features, such as **processor type**, **camera quality**, and **network compatibility**.
    - Incorporate real-time market data, such as demand trends and competitor pricing, to improve prediction accuracy.
2. **Feature Engineering**:
    - Create new features, such as **price-to-performance ratios**, **battery life scores**, or **customer ratings**, to add depth to the analysis.