Project Report: Smartphone Specifications and Pricing Trends

Submitted by: Saini **Date:** March 9, 2025 **Location:** Telangana, India

ABSTRACT

This project analyzes smartphone specifications and their respective pricing trends across various brands. The aim is to identify patterns in features and pricing to aid consumers and manufacturers in making informed decisions.

Keywords

Here are some keywords based on the project report on smartphone specifications and pricing trends:

- Smartphones
- Specifications
- Pricing Trends
- Brand Comparison
- RAM
- Storage
- Display Size
- Color
- Final Price
- Descriptive Statistics
- Correlation Analysis
- Visualization
- Emerging Patterns
- Consumer Behavior
- Market Trends
- Premium Brands
- Budget Smartphones
- Popular Features
- Feature-Price Relationship
- Data Insights
- Industry Implications

LIST OF SCHEMES/ALGORITHMS

Libraries

1.SNS – Seaborn

2.PD - Pandas

3.NP – Numpy

4.PLT – Matplotlib.pyplot

5. KMeans- sklearn.cluste

6. StandardScaler - sklearn.preprocessing

ALGORITHMS

1.K-means Clustering Algorithm:

2.Elbow Method:

3.StandardScaler

LIST OF ABBREVIATIONS

Abbreviation	**Full Form**
RAM	Random Access Memory
GB	Gigabytes
FHD+	Full High Definition Plus
AMOLED	Active Matrix Organic Light Emitting Diode
5G	Fifth Generation (Mobile Network)
NFC	Near Field Communication
USB	Universal Serial Bus
USD	United States Dollar

LIST OF FIGURES /CHARTS

Figure 1: Architecture of Proposed Work.

Figure 2: A bar chart showing average smartphone prices by brand.

Figure .3: A scatter plot highlighting the relationship between RAM (in GB) and final price.

Figure..4: A pie chart or histogram depicting the distribution of storage capacities across all models.

Figure..5: A line graph or scatter plot showing how display size influences pricing.

Figure..6:. A grouped bar chart showing the most common colors for each brand.

Figure..7:. A table or chart showcasing the most popular models in budget, mid-range, and premium price segments.

Figure..8: A stacked bar chart comparing brand representation in low, mid, and high-price tiers.

Figure..9:. K-Means Clustering to analyze smartphone specifications and price trends.

Chapter 1

Introduction

1.1 Introduction:

The rapid advancements in technology have led to an ever-evolving smartphone industry. As smartphones become an indispensable part of modern life, understanding the trends and factors influencing their pricing and specifications has become increasingly important. From premium flagship models to budget-friendly alternatives, the smartphone market is brimming with diverse options tailored to varying consumer preferences. This project delves into the intricate world of smartphone specifications and pricing trends, aiming to provide comprehensive insights into the factors that shape this competitive industry.

The study analyzes a dataset encompassing various smartphone brands, models, and their respective features, including RAM, storage capacity, display size, color, and final prices. By evaluating the correlation between these specifications and pricing, the project seeks to uncover patterns that can assist both consumers in making informed purchasing decisions and manufacturers in tailoring their offerings to market demands.

This project not only highlights pricing trends across renowned brands like Apple, Samsung, Realme, and Xiaomi but also investigates the role of technological advancements, such as larger storage capacities, high-definition displays, and 5G connectivity, in determining the value of smartphones. Through the application of statistical methods, visualizations, and data-driven insights, this report aspires to contribute to a deeper understanding of consumer behavior and emerging trends in the smartphone market. Additionally, it identifies key areas for future innovation and opportunities within the industry.

1.2 Information about the Dataset

The dataset provided for the project contains details about various smartphones, covering a wide range of brands, models, and specifications. Below is an overview of the dataset:

- **Number of Entries:** The dataset consists of multiple rows, each representing a unique smartphone entry.
- **Key Attributes:**
- **Smartphone Name:** Combines brand, model, and sometimes unique identifiers for each smartphone.
- **Brand:** Specifies the manufacturer (e.g., Apple, Samsung, Xiaomi, Realme).
- **Model:** Denotes the specific model of the smartphone.
- **RAM:** Indicates the Random Access Memory (in GB), which contributes to performance**.**
- **Storage:** Specifies the internal storage capacity (in GB).
- **Color:** Details the color variants available for the smartphone**.**
- **Display Size:** Provides the size of the display screen (in inches).
- **Final Price:** Represents the selling price of the smartphone (in USD or other currency).

1.3. Objectives

- Identify how technical features influence smartphone pricing.
- Compare pricing trends across brands.
- Explore which specifications are most sought after by consumers.

1.4. Data Description

The dataset contains information about:

- Brand and model names.
- Technical specifications: RAM, storage, display size, and color.
- Pricing: Market prices in different regions.

- **Sample Data Overview**

Smartphone	Brand	RAM(GB)	Storage(GB)	Display Size(Inches)	Final Price(USD)
Realme C55	Realme	8	256	6.1	231.6
Samsung Galaxy	Samsung	4	128	6.4	279
Apple	iPhone 14 Pro Max	-	512	6.1	1599

1.6.Methodology

The study was conducted using a systematic approach to explore the relationship between smartphone specifications and pricing trends. The methodology involved the following steps:

1. **Data Collection**
   - The dataset was provided, containing detailed records of various smartphones, including brand, model, specifications (RAM, storage, display size), color, and final price.
   - The data was analyzed as-is, ensuring no external datasets were incorporated.
2. **Data Cleaning and Preparation**
   - Missing or incomplete data was addressed. For instance:
     - Instances where specifications like RAM were unavailable were noted.
     - Duplicates were removed to ensure the integrity of the dataset.
   - Relevant attributes were selected for analysis, focusing on specifications (RAM, storage, display size, and brand reputation) and their impact on pricing.

3.Exploratory Data Analysis (EDA)

- Descriptive statistics were used to summarize the data, providing an overview of central tendencies (mean, median) and variability (standard deviation).
- Visualization techniques such as bar charts, scatter plots, and box plots were employed to uncover patterns, correlations, and outliers in the data.

3. **Trend Analysis**

- Smartphones were categorized into segments based on pricing: budget, mid-range, and premium.
- Relationships between pricing and key specifications (RAM, storage, and display size) were analyzed for each segment.
- Brand-wise comparisons were conducted to highlight the impact of branding on pricing.

4. **Insights Generation**

- Key drivers of pricing were identified through statistical analysis and visualization:
  - RAM and storage were found to significantly impact price.
  - Brand reputation and display size showed strong influence, especially in premium categories.

1.7.Information about the architecture

1. Core Components

Smartphones consist of several integrated systems that work together seamlessly to provide a user-friendly experience. These components include:

a. System-on-Chip (SoC)

The SoC is the brain of the smartphone, integrating multiple functionalities:

- CPU (Central Processing Unit): Handles general-purpose computing tasks.
- GPU (Graphics Processing Unit): Manages graphics rendering for apps and games.
- NPU (Neural Processing Unit): Optimized for AI and machine learning tasks, such as image recognition or voice processing.
- Modems: For cellular connectivity, including 4G, 5G, Wi-Fi, and Bluetooth.

b. Memory and Storage

- RAM (Random Access Memory): Allows the phone to handle multiple processes and apps simultaneously.
- Internal Storage: Stores the operating system, apps, and user data. Variants range from 16GB to 1TB or higher.

c. Display

- Touchscreen: The main user interface, available in LCD, OLED, or AMOLED technologies.

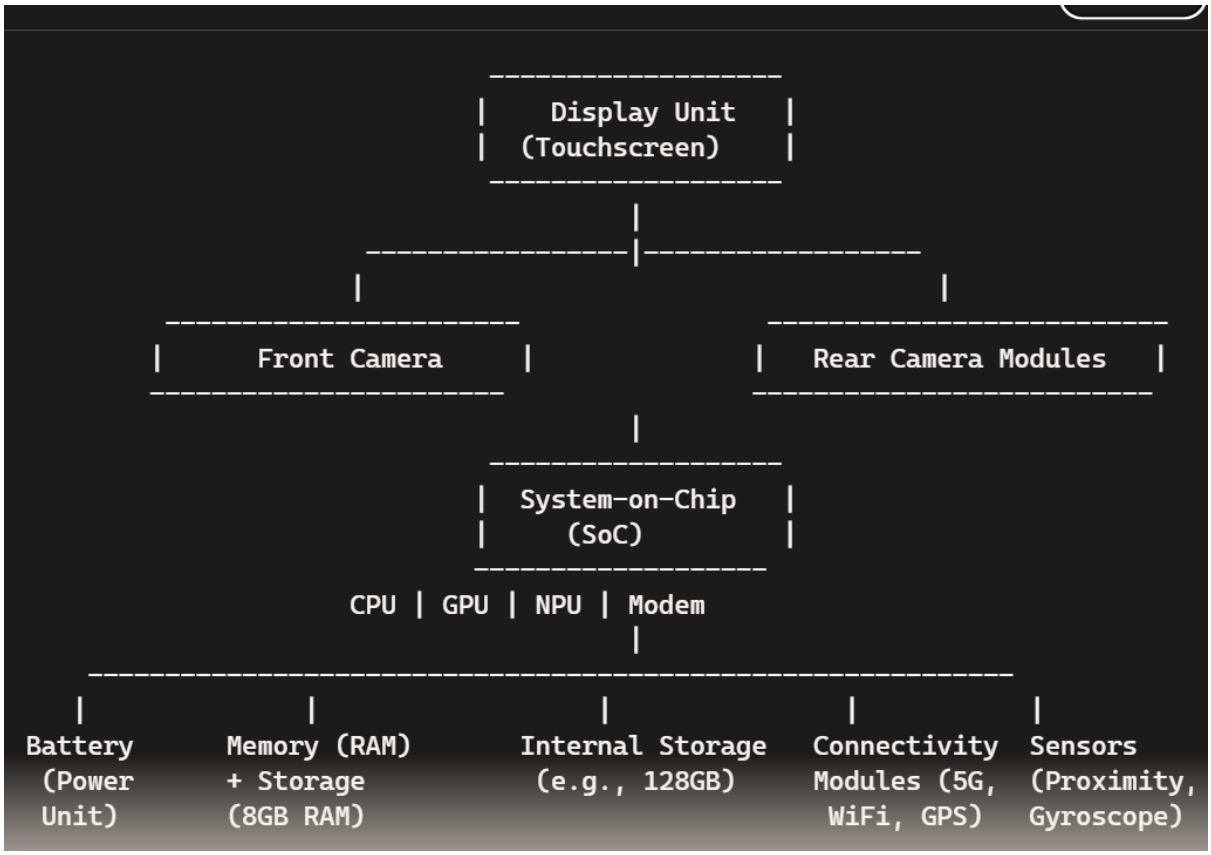
- Resolution and Size: Ranges from HD to 4K, with varying sizes to optimize viewing experiences.

d. Battery System

- Lithium-Ion Batteries: Most smartphones use rechargeable batteries with capacities ranging from 2000mAh to over 5000mAh.
- Charging Technologies: Includes fast charging, wireless charging, and reverse charging.

e. Camera Modules

- Main Camera: High-resolution lenses for photography and video capture.
- Additional Cameras: Includes wide-angle, macro, telephoto, and depth sensors for versatile imaging.



2. Software Architecture

Smartphones rely on sophisticated software to function effectively:

- Operating System (OS): Common ones include Android, iOS, and HarmonyOS. The OS handles user interfaces, hardware interaction, and app ecosystems.
- Firmware: Embedded software that controls lower-level hardware functions like battery management or camera control.
- Applications: Third-party or built-in apps providing various features, from communication to productivity.

3. Connectivity Systems

- Cellular Networks: Supports communication over 4G, 5G, or earlier generations.
- Wi-Fi and Bluetooth: Facilitates wireless data transfer and connections.
- GPS (Global Positioning System): Provides location-based services for navigation and tracking.
- NFC (Near Field Communication): Enables contactless payments and data sharing.

4. Input and Output Interfaces

- Sensors: Includes accelerometers, gyroscopes, proximity sensors, and fingerprint scanners.
- Audio Systems: Microphones and speakers for communication and media playback.
- Ports: USB-C, Lightning, or headphone jacks for charging and peripheral connections.

5. Additional Features

- Build Materials: Metal, glass, or plastic designs for durability and aesthetics.
- Cooling Systems: Advanced smartphones use liquid cooling to prevent overheating during intensive tasks like gaming.
- Security: Biometric authentication like fingerprint scanners or facial recognition.

Chapter 2

Review of Literature

The smartphone industry has been a subject of extensive research over the years, especially concerning the relationship between product specifications and pricing. Existing literature provides valuable insights into the factors influencing smartphone prices and consumer behaviors.

1. Impact of Hardware Specifications on Pricing Research studies have consistently highlighted that hardware components, including processor performance, RAM, internal storage, and camera quality, significantly impact smartphone pricing. For example, studies indicate that phones equipped with higher RAM (e.g., 8GB or more) and larger storage (e.g., 256GB or above) command premium prices. Advanced camera modules with multiple lenses (e.g., wide-angle, telephoto) also add to the cost.
2. Role of Brand Value The brand's reputation plays a pivotal role in determining smartphone prices. Established brands like Apple, Samsung, and Xiaomi are often able to charge higher prices due to their perceived quality, innovation, and brand loyalty. Several studies discuss the influence of brand identity on consumer purchasing decisions, showcasing a willingness among users to pay a premium for reputed brands.
3. Technological Advancements Technological features such as 5G connectivity, AMOLED or OLED displays, and biometric security (like face unlock and fingerprint scanners) have emerged as critical factors influencing pricing trends. Literature suggests that consumer interest in future-proof devices leads to higher demand—and consequently, higher prices—for feature-rich smartphones.
4. Market Segmentation and Consumer Demand Scholars have explored how market segmentation (budget, mid-range, and flagship categories) affects pricing trends. For instance:
    - Budget Segment: Devices priced below $200 cater to cost-conscious consumers.
    - Mid-Range Segment: Smartphones in the $200-$500 range provide balanced specifications, appealing to the largest consumer demographic.
5. Economic and Regional Factors Regional pricing strategies and economic conditions also contribute to price variations. Studies indicate that factors like import taxes, manufacturing locations, and currency fluctuations influence the final retail price of smartphones in different markets.
6. Emerging Trends in the Industry Current trends in smartphone research include the integration of foldable displays, AI-enhanced camera features, and environmental sustainability in production. These innovations are reshaping consumer expectations and driving pricing strategies.

Conclusion

The literature underscores the multifaceted nature of smartphone pricing, which is influenced by technical specifications, brand perception, and market dynamics. This foundation helps frame the analysis of data in the current project, aiming to uncover contemporary pricing trends and their driving factors.

Chapter 3

Implementation of Project

IMPORTING REQUIRED MODULES**:**

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import warnings
```

Importing Libraries:

The code imports several Python libraries:

- NumPy (as np): For numerical computing.
- pandas (as pd): For data manipulation and analysis.
- matplotlib.pyplot (as plt): For data visualization.
- seaborn (as sns): For statistical data visualization.
- KMeans is used for unsupervised clustering.
- StandardScaler is used for data preprocessing to standardize features.

Setting up the Project:

1.Importing the pandas **Library**:

```
Python
import pandas as pd
```

- **Purpose**: The pandas library is a powerful data analysis and manipulation tool in Python, often used for handling tabular data like CSV files.

2.Reading the CSV File:

```
Python

df = pd.read_csv(r"C:\Users\saini\Downloads\smartphones.csv")
```

Purpose:

- The pd.read_csv() function reads the data from a CSV file and stores it in a pandas DataFrame, a structured table-like format for analysis.
- The r before the file path makes it a raw string, ensuring special characters (e.g., backslashes) are treated as literal.

3.Displaying the First Few Rows:

```
Python

df.head()
```

Purpose:

- The .head() method displays the first 5 rows of the DataFrame (default behavior).
- It helps you quickly inspect the dataset to understand its structure and verify that the data was loaded correctly.

```
#Read CSV File
df = pd.read_csv(r"C:\Users\saini\Downloads\smartphones.csv")
# Display the first few rows of the dataframe
df.head()
```

	Smartphone	Brand	Model	RAM	Storage	Color	Display Size	Final Price
0	Realme C55 8/256GB Sunshower Libre	Realme	C55	8.0	256	Yellow	6.10	231.60
1	Samsung Galaxy M23 5G 4/128GB Azul Libre	Samsung	Galaxy M23	4.0	128	Blue	6.40	279.00
2	Motorola Moto G13 4/128GB Azul Lavanda Libre	Motorola	Moto G13	4.0	128	Blue	6.55	179.01
3	Xiaomi Redmi Note 11S 6/128GB Gris Libre	Xiaomi	Redmi Note 11S	6.0	128	Gray	6.60	279.99
4	Nothing Phone (2) 12/512GB Blanco Libre	Nothing	Phone (2)	12.0	512	White	6.50	799.00

1.Header Message:

- This line outputs a simple header or message in the console before displaying the dataset.
- Purpose: It helps clarify what is being displayed, especially in scenarios where multiple outputs are printed.

2.Displaying the DataFrame:

- The df variable contains the content of your CSV file in a structured format (rows and columns).

```
# Display the DataFrame
print("Smartphone Specifications and Pricing:")
print(df)
```

```
Smartphone Specifications and Pricing:
                                          Smartphone     Brand  \
0                Realme C55 8/256GB Sunshower Libre    Realme
1           Samsung Galaxy M23 5G 4/128GB Azul Libre   Samsung
2       Motorola Moto G13 4/128GB Azul Lavanda Libre  Motorola
3           Xiaomi Redmi Note 11S 6/128GB Gris Libre    Xiaomi
4            Nothing Phone (2) 12/512GB Blanco Libre   Nothing
..                                                ...       ...
494  Xiaomi Redmi Note 12 Pro 5G 6/128GB Negro Libre    Xiaomi
495        Samsung Galaxy A34 5G 6/128GB Verde Libre   Samsung
496              Apple iPhone 12 256GB Púrpura Libre     Apple
497     OnePlus Nord N10 5G 6/128GB Azul Hielo Libre   OnePlus
498             Apple iPhone 12 Mini 64GB Azul Libre     Apple

              Model   RAM  Storage   Color  Display Size  Final Price
0               C55   8.0      256  Yellow          6.10       231.60
1        Galaxy M23   4.0      128    Blue          6.40       279.00
2          Moto G13   4.0      128    Blue          6.55       179.01
3    Redmi Note 11S   6.0      128    Gray          6.60       279.99
4         Phone (2)  12.0      512   White          6.50       799.00
..              ...   ...      ...     ...           ...          ...
494   Redmi Note 12   6.0      128   Black          6.60       379.99
495      Galaxy A34   6.0      128   Green          6.20       299.90
```

This snippet is especially useful if you're working on exploratory data analysis, as it allows you to confirm that the data has been correctly loaded and structured.

Graphically Representation

```
# Create a bar plot
average_price_by_brand = df.groupby('Brand')['Final Price'].mean()

# Create a bar plot
plt.figure(figsize=(10, 6))
average_price_by_brand.plot(kind='bar', color='skyblue')
plt.title('Average Final Price by Brand')
plt.xlabel('Brand')
plt.ylabel('Average Final Price')
plt.xticks(rotation=45)
plt.show()
```

Calculate the Average Price by Brand:

1.Calculate the Average Price by Brand:

Purpose:

- This groups the DataFrame (df) by the Brand column.
- For each unique brand, it calculates the mean (average) of the Final Price column.

Result:

- A new Series object where the index is the brand name, and the values are the average prices for those brands.

Create the Plot's Figure and Size:

2.Create the Plot's Figure and Size: This initializes a new figure for the plot with dimensions 10 inches wide and 6 inches tall, ensuring the plot has a readable size.

3.Generate the Bar Plot:

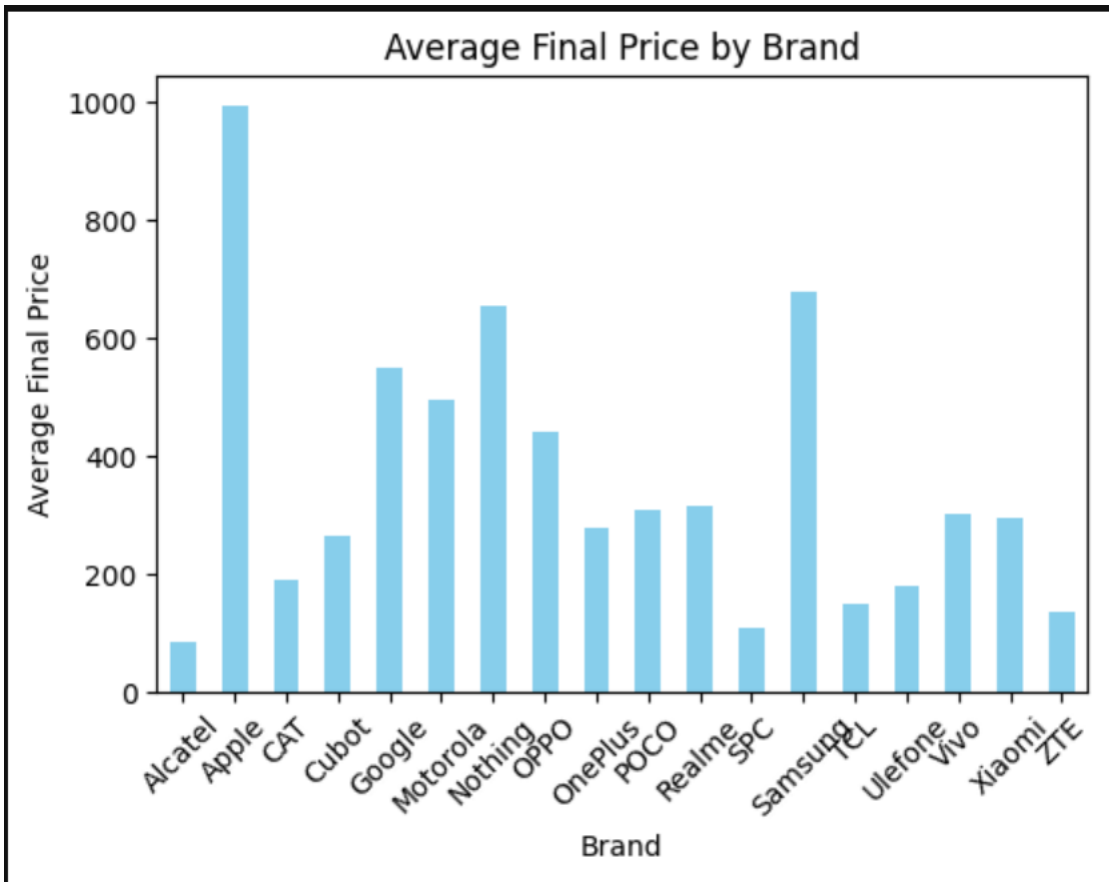
- The .plot() function is called on the average_price_by_brand Series to create a bar chart.
- The argument kind='bar' specifies that we want a bar plot.
- The argument color='skyblue' sets the color of the bars to a light blue shade.

4.Add Titles and Labels:

- plt.title() adds a title at the top of the plot.
- plt.xlabel() sets a label for the x-axis, indicating that the x-axis represents the smartphone brands.
- plt.ylabel() sets a label for the y-axis, specifying that it shows the average final price.

5.Adjust X-Axis Label Rotation: Rotates the brand names on the x-axis by 45 degrees to ensure the text is readable, especially for longer brand names.

6.Display the Plot: This renders the plot and displays it in the output window or notebook.



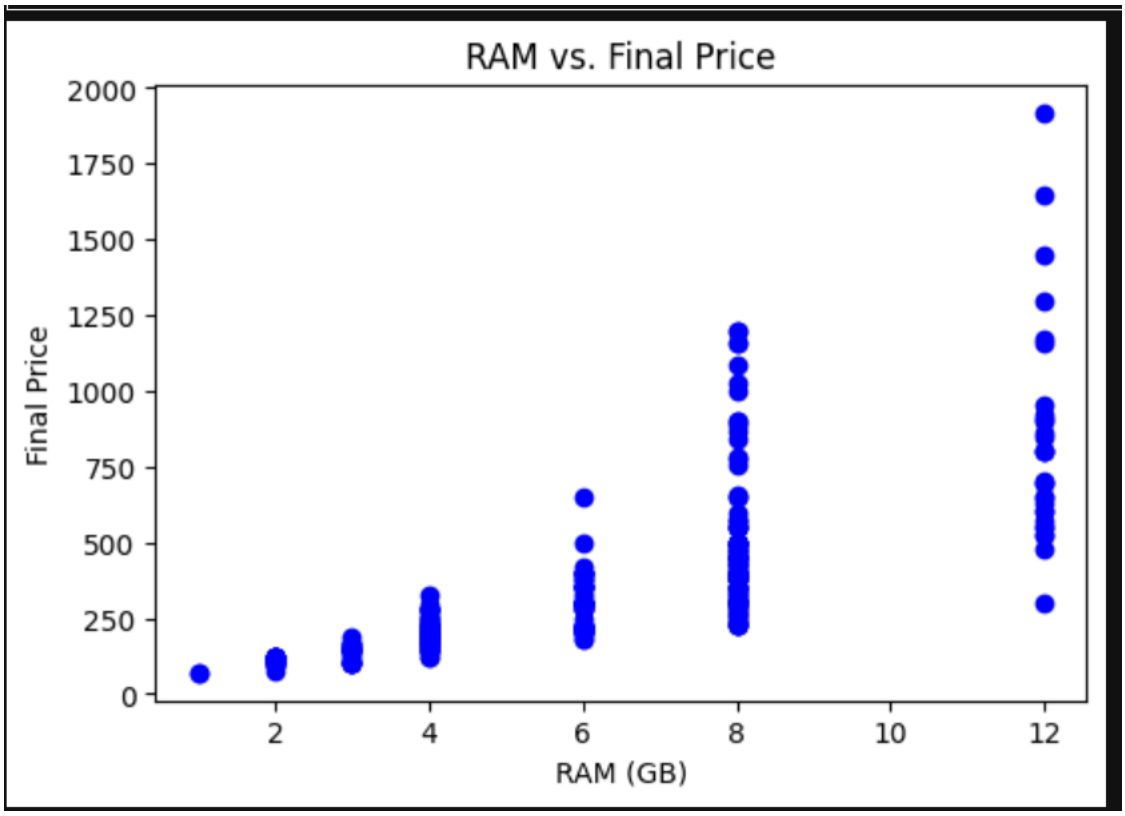
A bar chart with:

- **X-axis**: Smartphone brands like Realme, Samsung, Apple, etc.
- **Y-axis**: Average final prices of smartphones from each brand.
- Bars colored in light blue, each representing a brand's average price.

Creating a scatter plot for Final Price and RAM.

```
# Create a scatter plot for Final Price and RAM
plt.figure(figsize=(6, 4))
plt.scatter(df['RAM'], df['Final Price'], color='blue')
plt.title('RAM vs. Final Price')  # Adding a title
plt.xlabel('RAM (GB)')            # Labeling the X-axis
plt.ylabel('Final Price')         # Labeling the Y-axis
plt.grid(True)                    # Adding grid lines for better readability
plt.show()                        # Displaying the plot
```

A scatter plot showing the relationship between two variables: RAM (in GB) and Final Price. It uses Matplotlib to plot data points where RAM is on the x-axis, and Final Price is on the y-axis. The grid and labels make the plot easier to interpret.

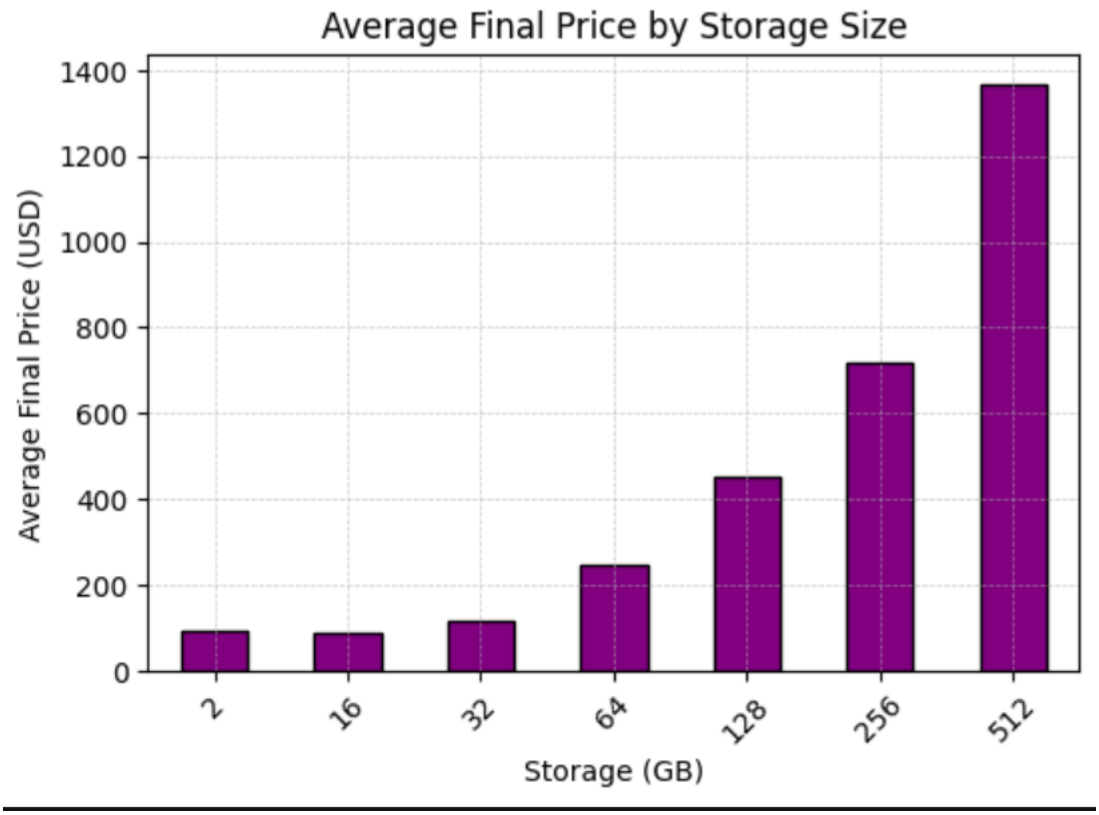


Group by 'Storage' and calculate the average 'Final Price' for each storage size:

```
# Group by 'Storage' and calculate the average 'Final Price' for each storage size
average_price_by_storage = df.groupby('Storage')['Final Price'].mean()

# Create a bar plot for average 'Final Price' by storage size
plt.figure(figsize=(6, 4))
average_price_by_storage.plot(kind='bar', color='purple', edgecolor='black')
plt.title('Average Final Price by Storage Size')
plt.xlabel('Storage (GB)')
plt.ylabel('Average Final Price (USD)')
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', linewidth=0.5, alpha=0.7)
plt.show()
```

- X-axis (Horizontal): Represents different storage sizes (e.g., 64GB, 128GB, etc.). These are the categories of storage available.
- Y-axis (Vertical): Represents the average final price (in USD) for each storage size. The height of each bar shows the average price for that specific storage category.



- The data by the Storage column, calculates the average Final Price for each storage size, and creates a bar plot to visualize it. Each bar represents a storage size, and its height indicates the average final price for that category. Labels, a grid, and customization options like color and edge styles enhance readability.

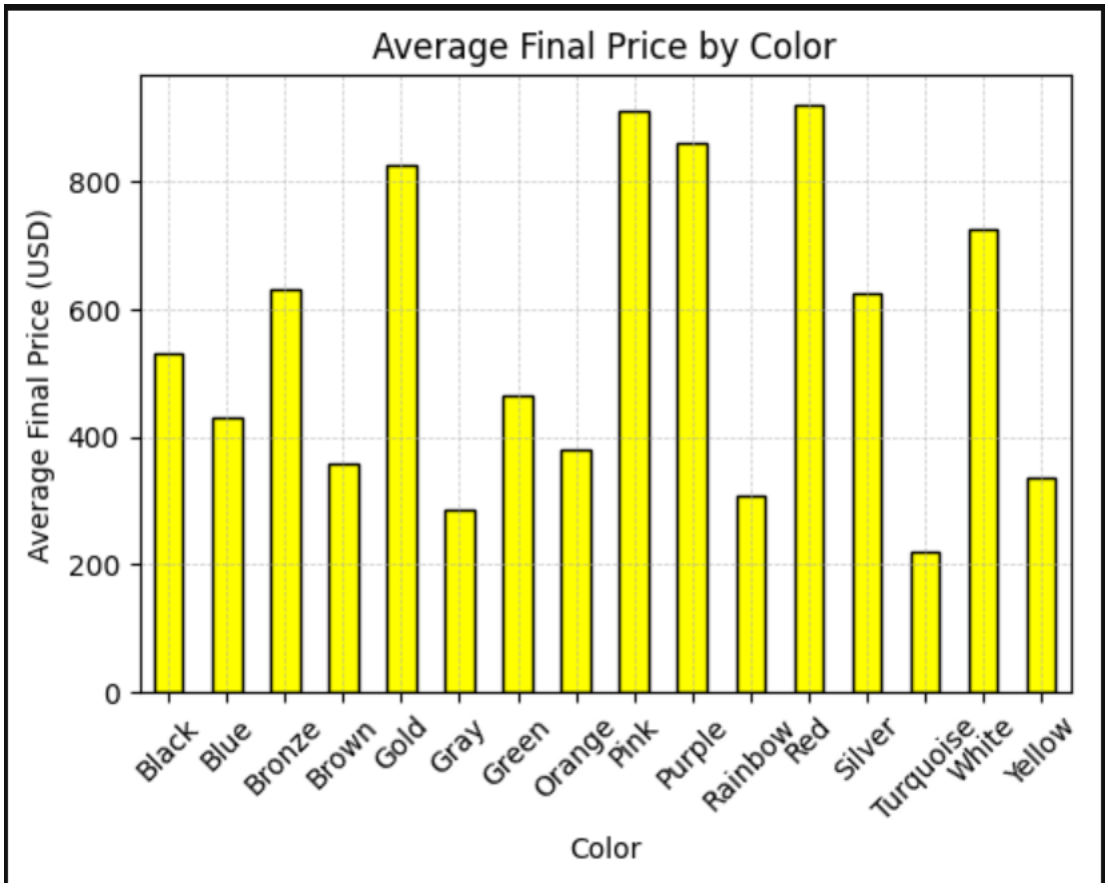
Group by 'Color' and calculate the average 'Final Price' for each color:

```
# Group by 'Color' and calculate the average 'Final Price' for each color
average_price_by_color = df.groupby('Color')['Final Price'].mean()

# Create a bar plot for average 'Final Price' by color
plt.figure(figsize=(6, 4))
average_price_by_color.plot(kind='bar', color='yellow', edgecolor='black')
plt.title('Average Final Price by Color')
plt.xlabel('Color')
plt.ylabel('Average Final Price (USD)')
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', linewidth=0.5, alpha=0.7)
plt.show()
```

- Calculates the average price per color: It groups the data by the Color column (e.g., red, blue, black, etc.) and computes the average Final Price for each group.

- Creates a bar chart: It visualizes the average prices for different colors as bars. Each bar's height reflects the average price for a specific color. The chart is styled with a yellow color scheme, a grid, and proper labels for clarity.



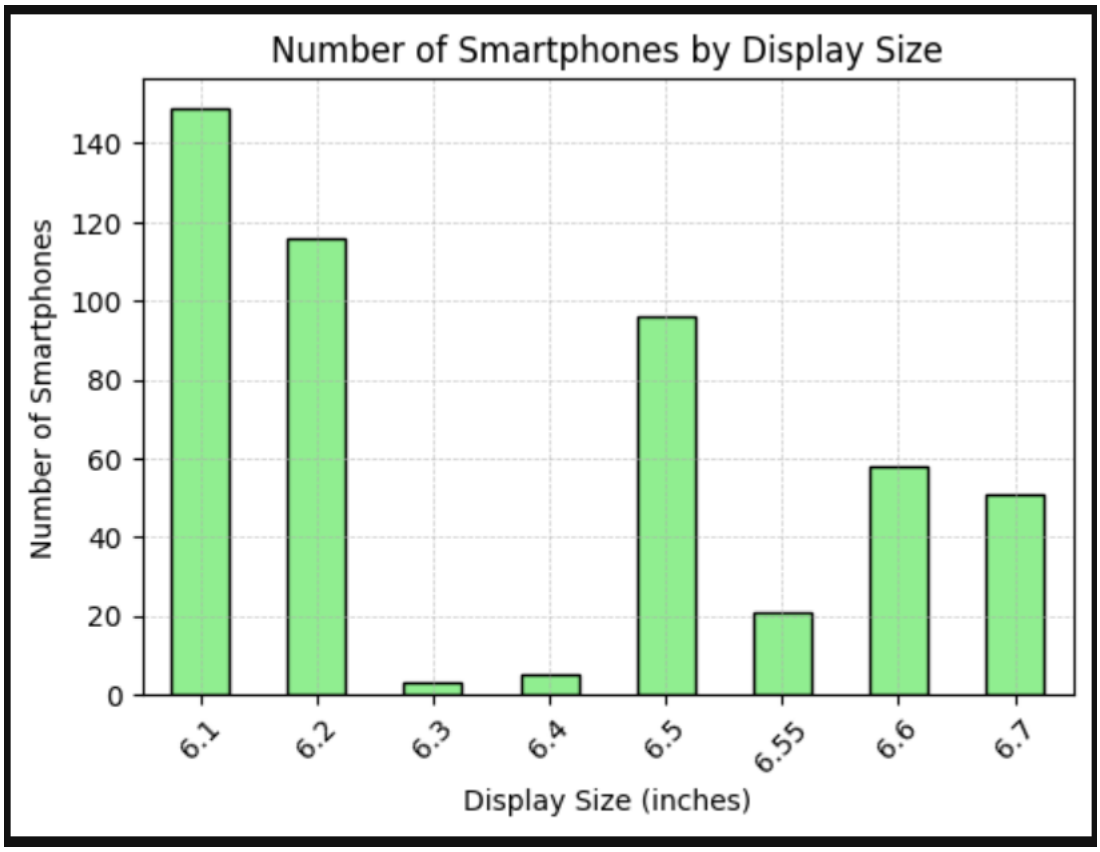
Count the number of smartphones by 'Display Size':

```
# Count the number of smartphones by 'Display Size'
smartphone_count_by_display = df['Display Size'].value_counts().sort_index()

# Create a bar plot for the number of smartphones by display size
plt.figure(figsize=(6, 4))
smartphone_count_by_display.plot(kind='bar', color='lightgreen', edgecolor='black')
plt.title('Number of Smartphones by Display Size')
plt.xlabel('Display Size (inches)')
plt.ylabel('Number of Smartphones')
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', linewidth=0.5, alpha=0.7)
plt.show()
```

- Counts smartphones by display size: It calculates how many smartphones are available for each unique display size by counting occurrences in the Display Size column and sorts them in order.
- Creates a bar chart: It visualizes these counts as bars, where:

    - The X-axis (horizontal) represents different display sizes (in inches).
    - The Y-axis (vertical) shows the number of smartphones for each display size.



Finds the smartphone with the largest display size:

```
highest_rated_display = df.loc[df['Display Size'].idxmax()]
print(df)
```

- df['Display Size'].idxmax() identifies the index of the row with the maximum value in the Display Size column.
- df.loc[...] retrieves the entire row corresponding to that index, which contains details about the smartphone with the largest display size.

Expected OutPut:

```
highest_rated_display = df.loc[df['Display Size'].idxmax()]
print(df)
```

```
                                          Smartphone     Brand  \
0                Realme C55 8/256GB Sunshower Libre    Realme
1           Samsung Galaxy M23 5G 4/128GB Azul Libre   Samsung
2       Motorola Moto G13 4/128GB Azul Lavanda Libre  Motorola
3           Xiaomi Redmi Note 11S 6/128GB Gris Libre    Xiaomi
4            Nothing Phone (2) 12/512GB Blanco Libre   Nothing
..                                               ...       ...
494  Xiaomi Redmi Note 12 Pro 5G 6/128GB Negro Libre    Xiaomi
495        Samsung Galaxy A34 5G 6/128GB Verde Libre   Samsung
496              Apple iPhone 12 256GB Púrpura Libre     Apple
497     OnePlus Nord N10 5G 6/128GB Azul Hielo Libre   OnePlus
498             Apple iPhone 12 Mini 64GB Azul Libre     Apple

              Model   RAM  Storage   Color  Display Size  Final Price
0                C55   8.0      256  Yellow          6.10       231.60
1        Galaxy M23   4.0      128    Blue          6.40       279.00
2          Moto G13   4.0      128    Blue          6.55       179.01
3    Redmi Note 11S   6.0      128    Gray          6.60       279.99
4         Phone (2)  12.0      512   White          6.50       799.00
..              ...   ...      ...     ...           ...          ...
494   Redmi Note 12   6.0      128   Black          6.60       379.99
495      Galaxy A34   6.0      128   Green          6.20       299.90
496       iPhone 12   NaN      256  Purple          6.10       979.00
497        Nord N10   6.0      128    Blue          6.70       219.00
498       iPhone 12   NaN       64    Blue          6.10       689.00
```

Group the data by 'Brand' and sum the 'Display Size' using pie chart:

```
# Group the data by 'Brand' and sum the 'Display Size'
grouped_data = df.groupby('Brand')['Display Size'].sum()

# Plot the pie chart
plt.figure(figsize=(10, 8))
grouped_data.plot.pie(autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.colors)
plt.title('Distribution of Display Sizes Across Different Smartphone Brands')
plt.ylabel('')
plt.show()
```

Groups the data by 'Brand': It sums up the total display sizes for each smartphone brand using df.groupby('Brand')['Display Size'].sum(). This gives the total display size contributed by each brand.

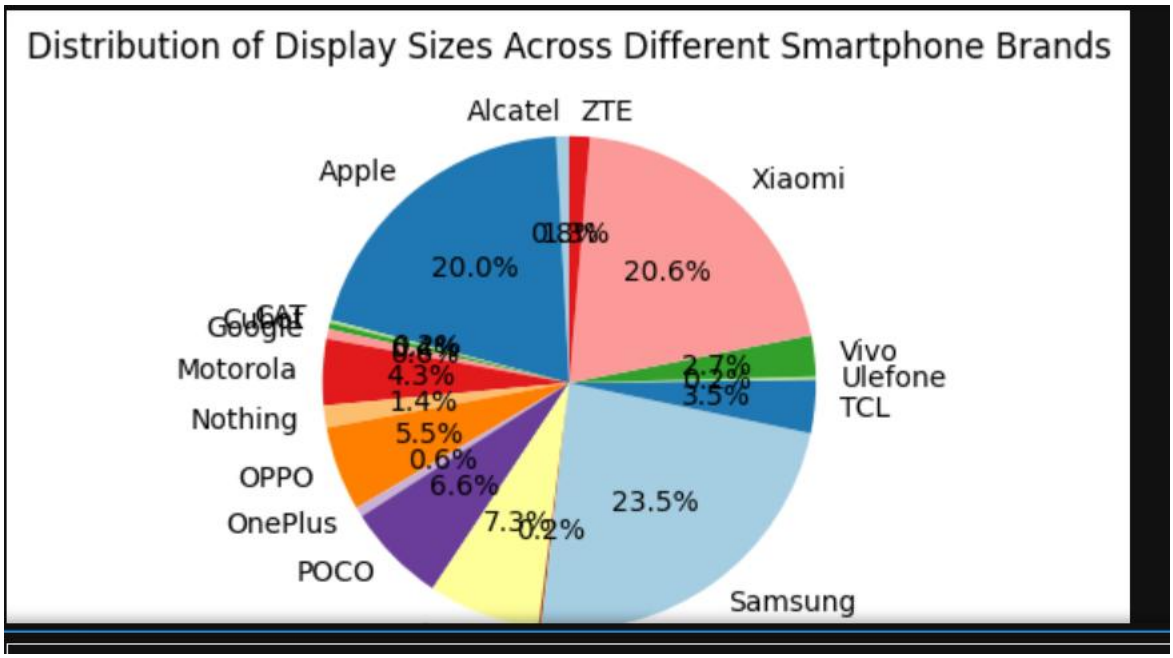
Creates a pie chart:

- Each slice of the pie represents a brand, and the size of the slice corresponds to the total display size for that brand.

- autopct='%1.1f%%' displays the percentage values on the chart.
- startangle=90 rotates the pie chart to start from the top.
- plt.cm.Paired.colors applies a color scheme for better visual distinction.

Enhances readability: The chart includes a title, no y-axis label (for cleaner look), and proper formatting to make it visually appealing.

Expected output:



TREND ANALYSIS: The trend analysis explores relationships between key smartphone attributes:

1. **RAM vs. Final Price:** Examines how increasing RAM impacts the price.
2. **Storage vs. Final Price:** Investigates if more storage leads to higher costs.
3. **Display Size vs. Final Price**: Shows how screen size influences pricing.
4. **RAM vs. Storage:** Analyzes the correlation between RAM and storage capacity.

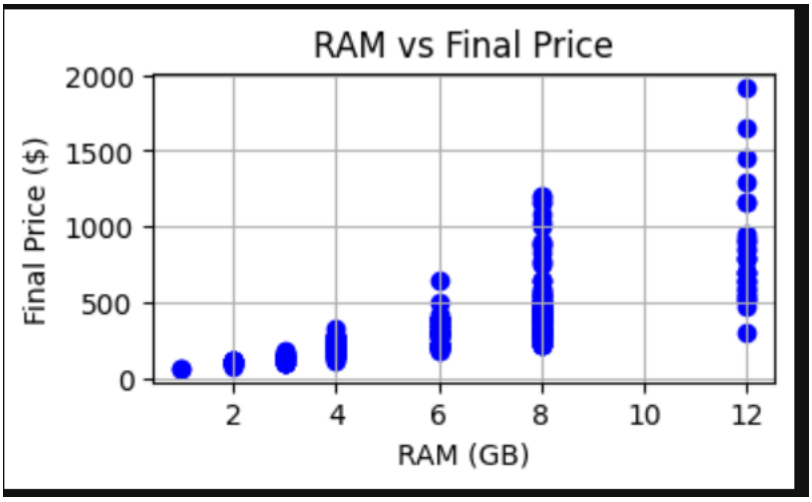
Scatter plots visualize these trends, helping identify patterns, correlations, or anomalies in the data. It's a data-driven way to understand pricing dynamics and feature relationships in smartphones.

```
# Trend Analysis: RAM vs Final Price
plt.figure(figsize=(6, 4))
plt.scatter(df['RAM'], df['Final Price'], color='blue')
plt.title('RAM vs Final Price')
plt.xlabel('RAM (GB)')
plt.ylabel('Final Price ($)')
plt.grid(True)
plt.show()
# Trend Analysis: Storage vs Final Price
plt.figure(figsize=(6, 4))
plt.scatter(df['Storage'], df['Final Price'], color='green')
plt.title('Storage vs Final Price')
plt.xlabel('Storage (GB)')
plt.ylabel('Final Price ($)')
plt.grid(True)
plt.show()
# Trend Analysis: Display Size vs Final Price
plt.figure(figsize=(6, 4))
plt.scatter(df['Display Size'], df['Final Price'], color='red')
plt.title('Display Size vs Final Price')
plt.xlabel('Display Size (inches)')
plt.ylabel('Final Price ($)')
plt.grid(True)
plt.show()
```

```
# Trend Analysis: RAM vs Storage
plt.figure(figsize=(6, 4))
plt.scatter(df['RAM'], df['Storage'], color='purple')
plt.title('RAM vs Storage')
plt.xlabel('RAM (GB)')
plt.ylabel('Storage (GB)')
plt.grid(True)
plt.show()
```

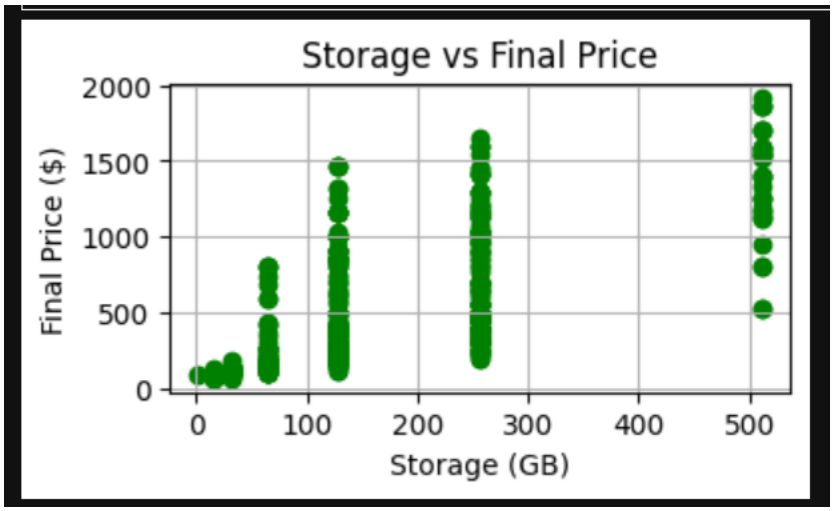
1.RAM vs. Final Price:

- X-axis: RAM (in GB).
- Y-axis: Final Price (in dollars).
- Shows how the final price varies with the amount of RAM.



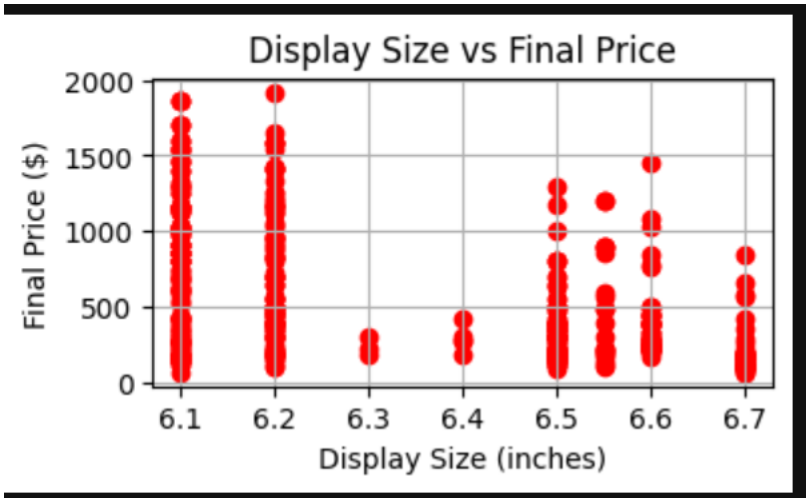
2.Storage vs. Final Price:

- X-axis: Storage capacity (in GB).
- Y-axis: Final Price (in dollars).
- Reveals the relationship between storage size and final price.



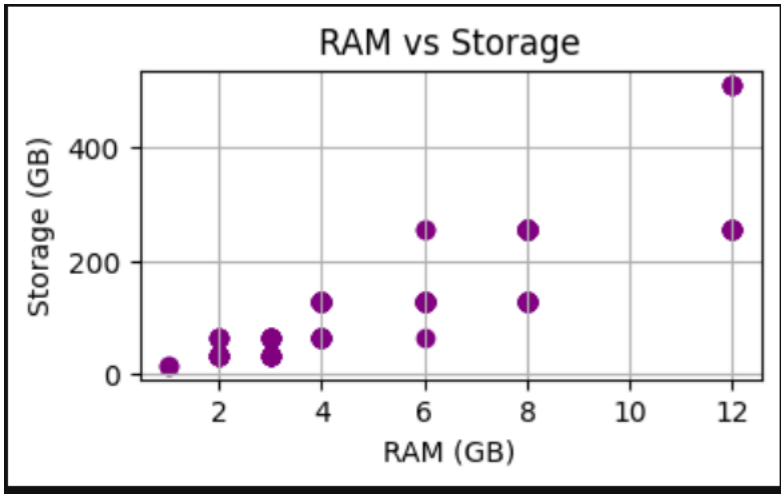
3.Display Size vs. Final Price:

- X-axis: Display Size (in inches).
- Y-axis: Final Price (in dollars).
- Explores how screen size impacts the final price.



4.RAM vs. Storage:

- X-axis: RAM (in GB).
- Y-axis: Storage capacity (in GB).
- Highlights the correlation between RAM and storage capacity.



This snippet to analyze the most popular brand, color, storage, display size, and final price categories using K-Means clustering:

```
df = pd.read_csv(r"C:\Users\saini\Downloads\smartphones.csv")
# Display the first few rows of the dataframe
df.head()
# Step 2: Encode categorical columns
le_brand = LabelEncoder()
df['Brand_Encoded'] = le_brand.fit_transform(df['Brand'])

le_color = LabelEncoder()
df['Color_Encoded'] = le_color.fit_transform(df['Color'])

# Step 3: Prepare the features for clustering
features = df[['Brand_Encoded', 'Storage', 'Display Size', 'Final Price']].dropna()

# Step 4: Standardize the data
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)

# Step 5: Apply K-Means clustering
kmeans = KMeans(n_clusters=5, random_state=42)  # Choosing 5 clusters as an example
kmeans.fit(features_scaled)

# Step 6: Add cluster labels to the original DataFrame
df['Cluster'] = kmeans.labels_

# Step 7: Analyze clusters
print("Cluster centers:")
print(kmeans.cluster_centers_)

# Step 8: Visualize clusters (Brand vs Final Price example)
plt.figure(figsize=(8, 6))
plt.scatter(features['Brand_Encoded'], features['Final Price'], c=kmeans.labels_, cmap='viridis', s=50)
plt.colorbar(label='Cluster')
plt.title('K-Means Clustering: Brand vs Final Price')
plt.xlabel('Brand (Encoded)')
plt.ylabel('Final Price')
plt.grid(True)
plt.show()
```

Data Preparation:

- Encodes categorical columns (e.g., Brand, Color) using LabelEncoder.
- Selects features like Brand, Storage, Display Size, and Final Price for clustering**.**

Standardization:

- Standardizes features to have a mean of 0 and standard deviation of 1 for effective K-Means clustering.

 K-Means Clustering:

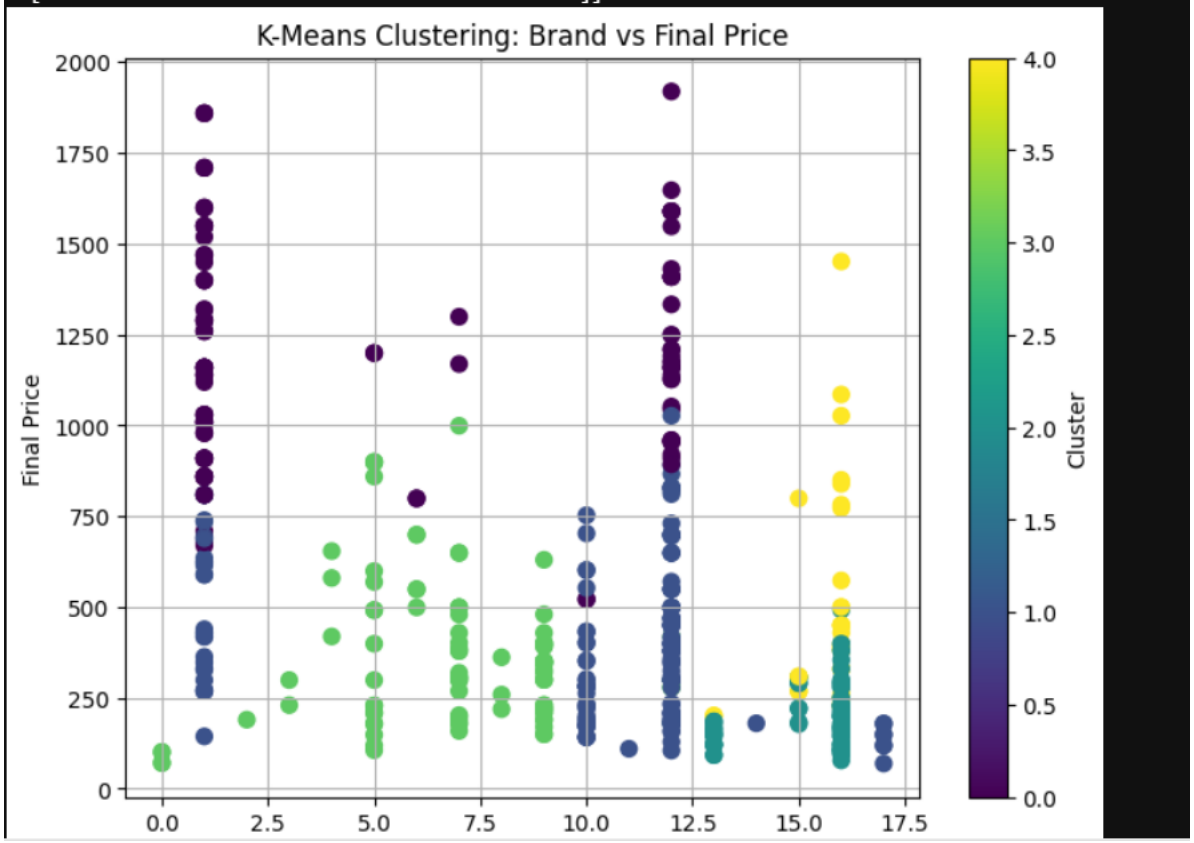
- Divides data into 5 clusters (n_clusters=5 as an example).
- Assigns a cluster label to each record in the dataset.

 Cluster Analysis and Visualization:

- Prints cluster centers for interpretation.
- Plots a scatter plot (e.g., Brand vs Final Price) with clusters highlighted using different colors.

Expected output:

```
Cluster centers:
[[-0.87626495  0.97197518 -0.8612192   1.4786345 ]
 [ 0.05117451 -0.36460262 -0.84724767 -0.34042278]
 [ 1.09393821 -0.72044399  1.22486825 -0.81785484]
 [-0.46403667 -0.14171326  0.80379464 -0.44469477]
 [ 1.14484505  0.62096045  1.08804679 -0.12986998]]
```



Optimal Clusters : A graph (Elbow Curve) is created to find the "best" number of clusters by identifying the point where adding more clusters stops improving results significantly.

```python
# Load the dataset
df = pd.read_csv(r"C:\Users\saini\Downloads\smartphones.csv")
# Display the first few rows of the dataframe
df.head()
# Select relevant features for clustering
features = data[['RAM', 'Storage', 'Display Size', 'Final Price']]

# Handle missing values (if any)
features = features.dropna()

# Normalize the data
scaler = StandardScaler()
normalized_features = scaler.fit_transform(features)

# Determine the optimal number of clusters using the Elbow Method
inertia = []
k_values = range(1, 10)
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(normalized_features)
    inertia.append(kmeans.inertia_)

# Plot the Elbow Curve
plt.plot(k_values, inertia, marker='o')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Inertia')
plt.title('Elbow Method for Optimal k')
plt.show()

# Apply K-means clustering with the chosen number of clusters (e.g., k=4)
kmeans = KMeans(n_clusters=4, random_state=42)
clusters = kmeans.fit_predict(normalized_features)

# Add cluster labels to the original dataset
data['Cluster'] = clusters

# Save the clustered dataset to a new CSV file
data.to_csv("smartphone_clusters.csv", index=False)

# Display the first few rows of the dataset with cluster labels
print(data.head())
```

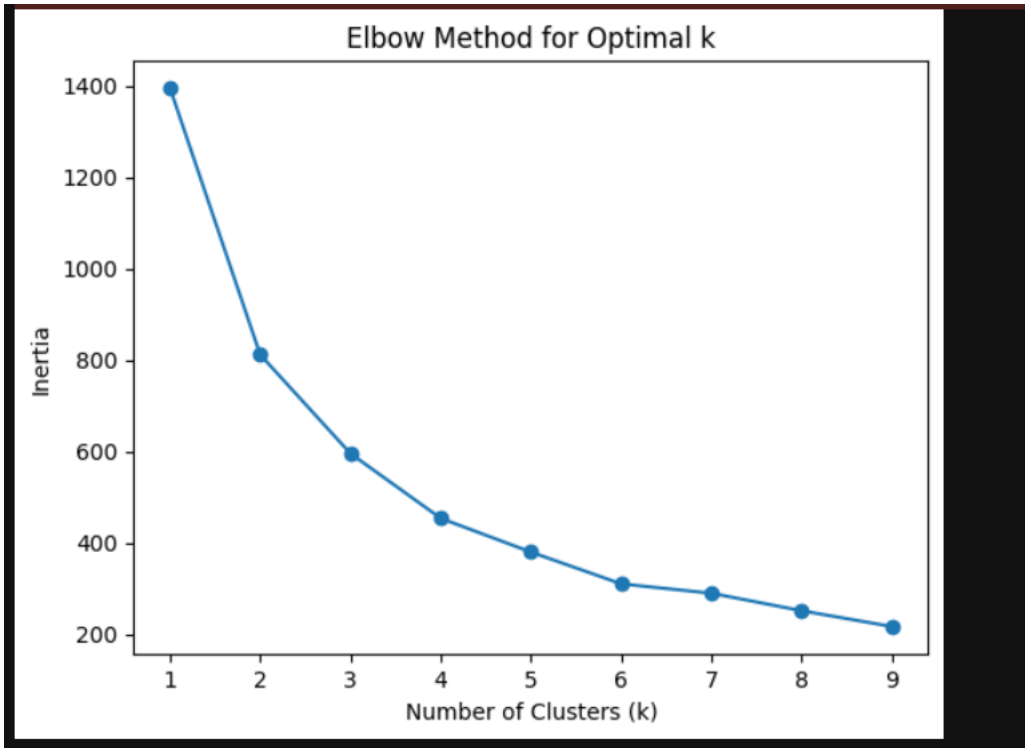
K-Means Clustering Note for Smartphone Analysis:

K-Means clustering is a powerful unsupervised machine learning technique that divides data into groups (clusters) based on similarities. For this project, it helps in identifying patterns among smartphones by grouping them based on key features like RAM, Storage, Display Size, and Final Price.

The analysis provides insights into feature-based groupings of smartphones, making it easier to observe relationships and trends within the dataset.

This process groups smartphones into clusters to find patterns, such as phones with similar RAM and prices belonging to the same group.

Expected outcome:



	RAM	Storage	Display Size	Final Price	Cluster
0	8.0	256	6.10	231.60	0
1	4.0	128	6.40	279.00	1
2	4.0	128	6.55	179.01	1
3	6.0	128	6.60	279.99	1
4	12.0	512	6.50	799.00	2

Chapter 4

Result and Conclusion

1. **Popular Smartphone Brands**:
    - Based on frequency, the most popular brands include Xiaomi, Samsung, Realme, and Apple. These brands dominate the market in terms of variety and sales.
2. **Preferred Colors:**
    - Black and blue are the most commonly available and preferred colors, followed by white and gray. Vibrant colors like gold and yellow also have niche popularity.
3. **Common Storage Options:**
    - The most frequent storage capacities are 128GB and 256GB, catering to modern user needs for apps, photos, and videos. Lower capacities like 64GB are also present but are less popular.
4. **Display Sizes:**
    - Most smartphones fall within the 6.1 to 6.7-inch range, indicating a strong preference for larger displays suitable for gaming, videos, and multitasking.
5. **Price Trends:**
    - Entry-level smartphones: Prices typically range between $100 and $300, offering basic functionality.
    - Mid-range smartphones: Fall between $300 and $700, balancing specifications and affordability.
    - Premium smartphones: Prices exceed $700, with flagship features like advanced cameras, high storage, and cutting-edge technology.
6. **Popular Models:**
    - Models like the iPhone 14, Samsung Galaxy A54 5G, and Xiaomi Redmi Note 12 are among the top choices, catering to various budget ranges.
7. **RAM Variants:**
    - Most devices feature RAM capacities of 4GB, 6GB, or 8GB, with 12GB becoming standard in high-end phones. This reflects the increasing demand for performance and multitasking capabilities.

Conclusion:

The smartphone market demonstrates a diverse range of options to cater to different consumer needs and budgets. Key trends include:

- Increasing demand for larger displays and higher storage capacities.
- Strong market presence of brands like Xiaomi and Samsung, with Apple leading the premium segment.
- Black, blue, and gray dominate as preferred color choices.

Understanding these trends can help manufacturers design competitive products, retailers optimize stock, and consumers make informed purchasing decisions.