Smartphone Market Insights Analysis

Table Contents

In [87]:
```
# importing libraries

import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt
import seaborn as sns
```

Using SQL Connector in the Project

In this project, we use an SQL connector to interact with our database. The SQL connector allows us to establish a connection between our Python scripts and the database, execute SQL queries, and fetch results for analysis. This helps us efficiently retrieve and manipulate the data needed for our analysis of the smartphone market.

In [88]:
```
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="Bhushan@148",
  database="project_02"
)
```

In [89]:
```
mydb
```

Out[89]:
```
<mysql.connector.connection_cext.CMySQLConnection at 0x19a4027ad50>
```

In [90]:
```
# Show All Tables in the Database

df = pd.read_sql_query(
"""
show tables
"""
, mydb)
df
```

Out[90]:

	Tables_in_project_02
0	smartphone

In [91]:
```
# Execute SQL query to describe the 'smartphone' table

df = pd.read_sql_query(
"""
describe smartphone
"""
, mydb)
df
```

Out[91]:

	Field	Type	Null	Key	Default	Extra
0	Model	text	YES		None	
1	Sim	text	YES		None	
2	OS	text	YES		None	
3	Price	int	YES		None	
4	Ratings	double	YES		None	
5	Display	text	YES		None	
6	Card	text	YES		None	
7	Processor_Type	text	YES		None	
8	Processor_Speed	text	YES		None	
9	RAM(GB)	int	YES		None	
10	ROM(GB)	int	YES		None	
11	Battery(mAh)	int	YES		None	
12	Charging_Type	text	YES		None	
13	Front_Camera	text	YES		None	
14	Rear_Camera	text	YES		None	

Project Title: Smartphone Market Insights Analysis

This project aims to provide a clear and detailed understanding of the smartphone market. We will analyze various aspects such as prices, features, user ratings, and technical specifications of different smartphone models using SQL queries. The dataset includes detailed information about each smartphone model, such as its price, battery life, camera quality, processor type, and more. By running these SQL queries, we will uncover important trends and patterns. This information can help consumers make better choices when buying a smartphone and assist manufacturers in developing better products.

1. Categorizes smartphones into distinct price ranges

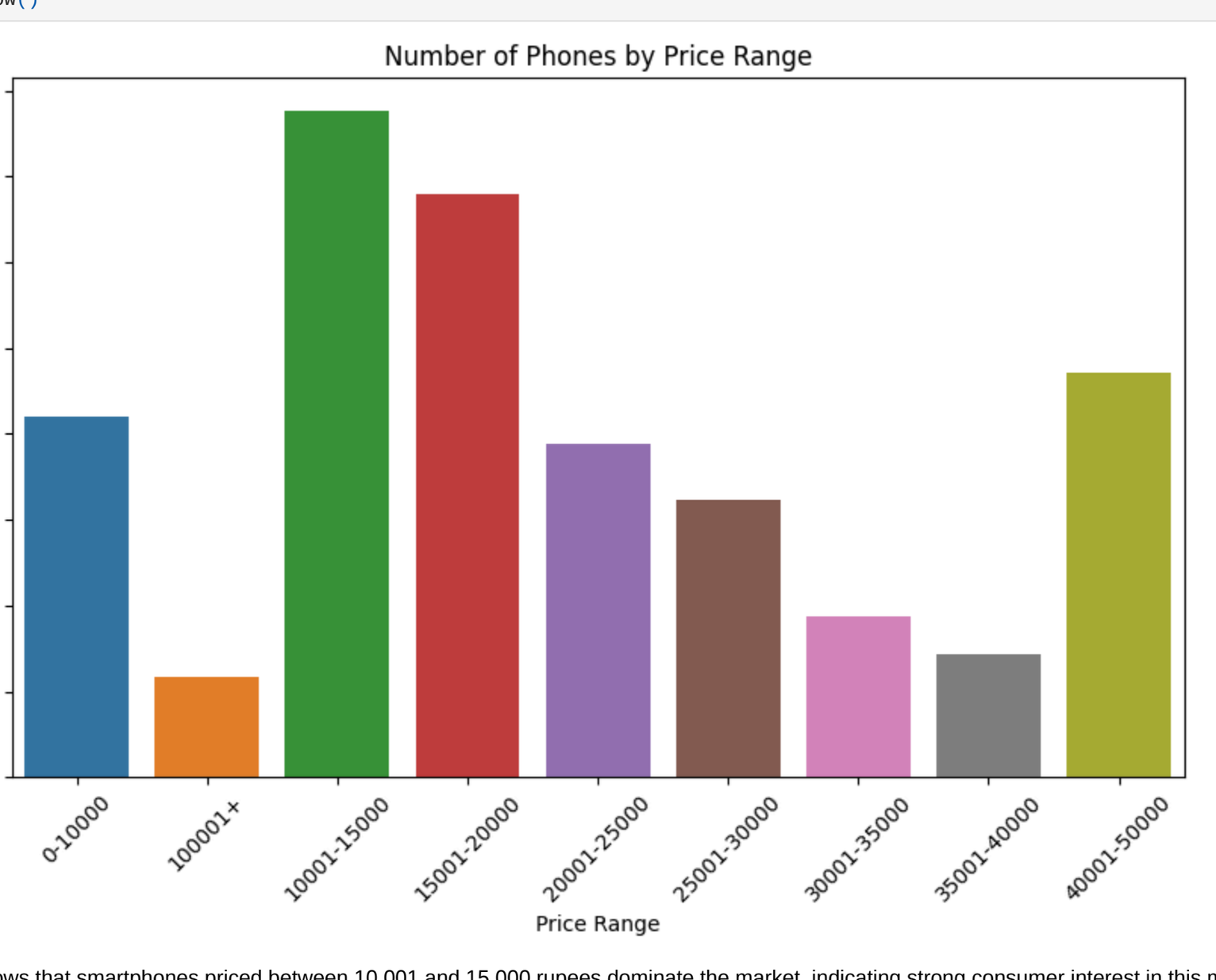
In [92]:
```
df = pd.read_sql_query(
"""
SELECT
    CASE
        WHEN price BETWEEN 0 AND 10000 THEN '0-10000'
        WHEN price BETWEEN 10001 AND 15000 THEN '10001-15000'
        WHEN price BETWEEN 15001 AND 20000 THEN '15001-20000'
        WHEN price BETWEEN 20001 AND 25000 THEN '20001-25000'
        WHEN price BETWEEN 25001 AND 30000 THEN '25001-30000'
        WHEN price BETWEEN 30001 AND 35000 THEN '30001-35000'
        WHEN price BETWEEN 35001 AND 40000 THEN '35001-40000'
        WHEN price BETWEEN 40001 AND 50000 THEN '40001-50000'
        WHEN price BETWEEN 50001 AND 100000 THEN '40001-50000'
        ELSE '100001+'
    END AS price_range,
    COUNT(*) AS no_of_phones
FROM
    smartphone
GROUP BY
    price_range
order by price_range

"""
, mydb)
df
```

Out[92]:

	price_range	no_of_phones
0	0-10000	105
1	100001+	29
2	10001-15000	194
3	15001-20000	170
4	20001-25000	97
5	25001-30000	81
6	30001-35000	47
7	35001-40000	36
8	40001-50000	118

In [93]:
```
plt.figure(figsize=(10, 6))
sns.barplot(x=df.price_range, y=df.no_of_phones, data=df)
plt.xlabel('Price Range')
plt.ylabel('Number of Phones')
plt.title('Number of Phones by Price Range')
plt.xticks(rotation=45)
plt.show()
```



Insights : The data shows that smartphones priced between 10,001 and 15,000 rupees dominate the market, indicating strong consumer interest in this mid-range price segment. Higher-priced models above 40,000 rupees and lower-priced options below 10,000 rupees also attract distinct but smaller segments of the market.

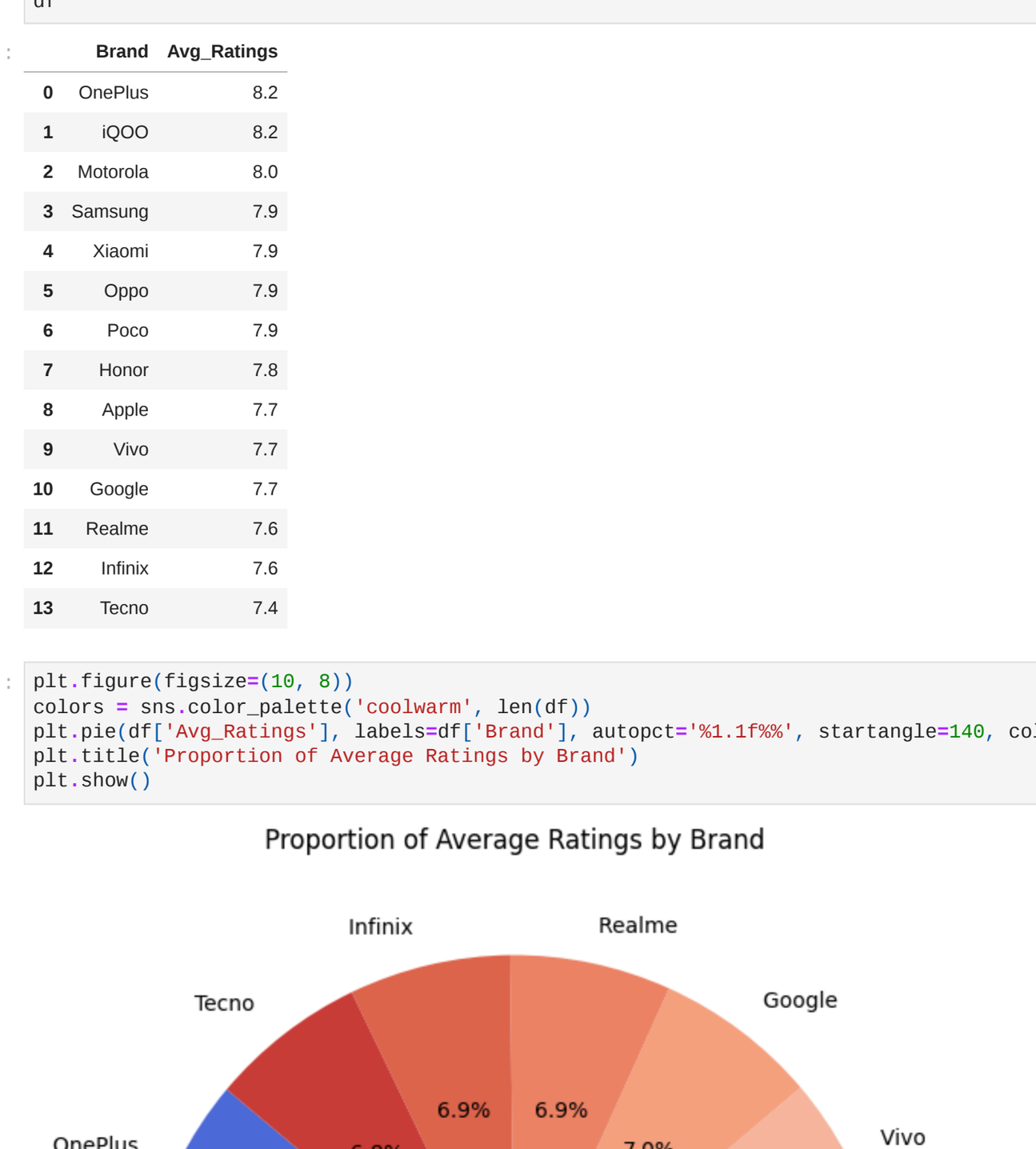
2. Calculates the average user ratings for each smartphone brand

In [94]:
```
df = pd.read_sql_query(
"""
with cte as (
        select *, left(Model, LOCATE(' ', model)) as Brand
        from smartphone)
select Brand, round(avg(Ratings),1) as Avg_Ratings
from cte
group by Brand
having count(*) > 10
order by Avg_Ratings desc;
"""
, mydb)
df
```

Out[94]:

	Brand	Avg_Ratings
0	OnePlus	8.2
1	iQOO	8.2
2	Motorola	8.0
3	Samsung	7.9
4	Xiaomi	7.9
5	Oppo	7.9
6	Poco	7.9
7	Honor	7.8
8	Apple	7.7
9	Vivo	7.7
10	Google	7.7
11	Realme	7.6
12	Infinix	7.6
13	Tecno	7.4

In [95]:
```
plt.figure(figsize=(10, 8))
colors = sns.color_palette('coolwarm', len(df))
plt.pie(df['Avg_Ratings'], labels=df['Brand'], autopct='%1.1f%%', startangle=140, colors=colors)
plt.title('Proportion of Average Ratings by Brand')
plt.show()
```



Insights: OnePlus and iQOO lead with the highest average ratings of 8.2, suggesting strong customer satisfaction with their products. Brands like Motorola, Samsung, and Xiaomi closely follow with average ratings around 8.0, reflecting their competitive positions in the market.

3. Evaluating Processor Types Based on User Ratings

In [96]:
```
df = pd.read_sql_query(
"""
SELECT substring_index(Processor_Type," ",1) as Processor_Type_, round(AVG(Ratings), 2) as Avg_Ratings
FROM smartphone
GROUP BY Processor_Type_
having count(*)>10
order by Avg_Ratings desc;
"""
, mydb)
df
```

Out[96]:

	Processor_Type_	Avg_Ratings
0	Snapdragon	8.16
1	Dimensity	8.11
2	Exynos	8.06
3	Bionic	7.86
4	Helio	7.25
5	Octa	7.06
6	Tiger	6.72
7	Unisoc	6.55

Insights: Processor types with more than 10 models are ranked by their average user ratings, highlighting which processors are generally preferred by users.

4. Analysis of Smartphone Price Segmentation by Memory Card Support

In [97]:
```
df = pd.read_sql_query(
"""
SELECT
    CASE
        WHEN Card = 'Memory Card Supported' THEN 'Memory Card Supported'
        ELSE 'No Memory Card Support'
    END AS Memory_Card_Support,
    COUNT(*) AS Model_Count,
    ROUND(AVG(Price), 2) AS Avg_Price,
    ROUND(AVG(Ratings), 2) AS Avg_Ratings
FROM smartphone
GROUP BY Memory_Card_Support
ORDER BY Avg_Price DESC;
"""
, mydb)
df
```

Out[97]:

	Memory_Card_Support	Model_Count	Avg_Price	Avg_Ratings
0	No Memory Card Support	800	30632.26	7.86
1	Memory Card Supported	77	17168.66	7.52

Insights: Models without memory card support are more expensive on average (average price of 30,632.26 rupees) and tend to have higher ratings (average rating of 7.86). In contrast, models with memory card support are more affordable (average price of 17,168.66 rupees) but have slightly lower ratings (average rating of 7.52).

5. Average Ratings and Battery Capacity by Processor Speed Range

In [98]:
```
df = pd.read_sql_query(
"""
SELECT
    CASE
        WHEN Processor_Speed < 2.0 THEN '< 2.0 GHz'
        WHEN Processor_Speed BETWEEN 2.0 AND 2.5 THEN '2.0-2.5 GHz'
        WHEN Processor_Speed BETWEEN 2.5 AND 3.0 THEN '2.5-3.0 GHz'
        ELSE '> 3.0 GHz'
    END as Processor_Speed_Range,
    round(AVG(Ratings),1) as Avg_Ratings,
    concat("Rs."," ",round(AVG(price))) as avg_price
FROM smartphone
GROUP BY Processor_Speed_Range
ORDER BY Processor_Speed_Range;
"""
, mydb)
df
```

Out[98]:

	Processor_Speed_Range	Avg_Ratings	avg_price
0	< 2.0 GHz	7.6	Rs. 58156
1	> 3.0 GHz	7.1	Rs. 20486
2	2.0-2.5 GHz	8.1	Rs. 31946

Insights: Devices with slower processors (< 2.0 GHz) are priced higher on average (Rs. 58,156) and have a rating of 7.6. Faster processors (> 3.0 GHz) are more affordable (Rs. 20,486) but have a slightly lower rating of 7.1. Processors in the 2.0-2.5 GHz range are the highest-rated (8.1) and moderately priced (Rs. 31,946), offering good performance for their cost.

6. Top 5 Models with Above Average Ratings and Low Price

In [99]:
```
df = pd.read_sql_query(
"""
WITH Avg_Ratings AS (
    SELECT AVG(Ratings) as Overall_Avg_Ratings
    FROM smartphone
),
Above_Avg_Ratings AS (
    SELECT *
    FROM smartphone
    WHERE Ratings > (SELECT Overall_Avg_Ratings FROM Avg_Ratings)
)
SELECT Model, Price, Ratings
FROM Above_Avg_Ratings
ORDER BY Price ASC
LIMIT 5
"""
, mydb)
df
```

Out[99]:

	Model	Price	Ratings
0	Xiaomi Redmi Note 10 Lite (6GB RAM + 128GB)	11999	8.0
1	Xiaomi Redmi Note 11SE	11999	8.0
2	Xiaomi Redmi Note 10S (6GB RAM + 128GB)	11999	7.9
3	Xiaomi Redmi Note 11 (6GB RAM + 64GB)	12188	7.9
4	Realme 10s	12999	8.0

Conclusion: Top 5 budget-friendly smartphone models with ratings above 7.9 include Xiaomi Redmi Note 10 Lite, Xiaomi Redmi Note 11SE, Xiaomi Redmi Note 10S, Xiaomi Redmi Note 11, and Realme 10s.

7. Market Share Analysis by Brand

In [100…
```
df = pd.read_sql_query(
"""
SELECT LEFT(Model, LOCATE(' ', Model)) AS Brand,
       COUNT(*) AS Model_Count,
       ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM smartphone), 2) AS Market_Share_Percentage
FROM smartphone
GROUP BY Brand
ORDER BY Model_Count DESC;
"""
, mydb)
df
```

Out[100]:

	Brand	Model_Count	Market_Share_Percentage
0	Xiaomi	125	14.25
1	Samsung	116	13.23
2	Vivo	96	10.95
3	Realme	92	10.49
4	Oppo	82	9.35
5	Motorola	48	5.47
6	Apple	43	4.90
7	Poco	40	4.56
8	OnePlus	38	4.33
9	iQOO	32	3.65
10	Tecno	31	3.53
11	Infinix	28	3.19
12	Google	12	1.37
13	Honor	11	1.25
14	Huawei	10	1.14
15	Nokia	10	1.14
16	Sony	7	0.80
17	Asus	6	0.68
18	Nubia	6	0.68
19	Nothing	5	0.57
20	iKall	3	0.34
21	Lava	3	0.34
22	Redmi	3	0.34
23	Oukitel	3	0.34
24	Micromax	3	0.34
25	ZTE	2	0.23
26	Lenovo	2	0.23
27	LG	2	0.23
28	Royole	2	0.23
29	Doogee	2	0.23
30	LeEco	1	0.11
31	Cola	1	0.11
32	Vertu	1	0.11
33	BLU	1	0.11
34	Tesla	1	0.11
35	Lyf	1	0.11
36	Jio	1	0.11
37	Letv	1	0.11
38	itel	1	0.11
39	Gionee	1	0.11
40	Leitz	1	0.11
41	TCL	1	0.11
42	Sharp	1	0.11
43	Blackview	1	0.11

Insights: Xiaomi leads the market with 14.25% share, followed closely by Samsung at 13.23%. Together, the top five brands (Xiaomi, Samsung, Vivo, Realme, Oppo) hold over half of the total market share, indicating strong competition among these players.

8. Distribution of Display Resolutions

In [101…
```
df = pd.read_sql_query(
"""
SELECT CASE
           WHEN Display LIKE '%1080%' THEN '1080p'
           WHEN Display LIKE '%1440%' THEN '1440p'
           ELSE 'Other'
       END AS Display_Resolution,
       COUNT(*) AS Model_Count
FROM smartphone
GROUP BY Display_Resolution;
"""
, mydb)
df
```

Out[101]:

	Display_Resolution	Model_Count
0	1440p	36
1	1080p	594
2	Other	247

Insights : The majority of smartphones in the dataset feature a 1080p display resolution, with 594 models, while 36 models have a higher resolution of 1440p. Other resolutions collectively account for 247 models, indicating a significant preference for Full HD displays in the market.

9. Evolution of Camera Megapixel Trends Over Time

In [102…
```
df = pd.read_sql_query(
"""
SELECT SUBSTRING_INDEX(Model, ' ', 1) AS Model_Name,
       MAX(Rear_Camera) AS Max_Megapixels,
       MIN(Rear_Camera) AS Min_Megapixels
FROM smartphone
GROUP BY Model_Name
ORDER BY Model_Name;
"""
, mydb)
df
```

Out[102]:

	Model_Name	Max_Megapixels	Min_Megapixels
0	Apple	Memory Card Not Supported	12MP
1	Asus	64MP + 16MP + 8MP Triple	50MP + 12MP Dual
2	Blackview	13MP Dual	13MP Dual
3	BLU	48MP Quad	48MP Quad
4	Cola	50MP + 2MP Dual	50MP + 2MP Dual
5	Doogee	108MP + 64MP + 16MP Triple	108MP + 20MP + 16MP Triple
6	Gionee	16MP	16MP
7	Google	50MP + 8MP Dual	12.2MP
8	Honor	64MP Quad	108MP Quad
9	Huawei	Foldable Display	108MP + 8MP + 2MP Triple
10	iKall	20MP	13MP
11	Infinix	8MP Dual	108MP + 13MP + 2MP Triple
12	iQOO	64MP + 8MP + 2MP Triple	48MP + 13MP + 13MP Triple
13	itel	8MP + 0.08MP Dual	8MP + 0.08MP Dual
14	Jio	13MP	13MP
15	Lava	64MP Quad	13MP + 2MP Triple
16	LeEco	13MP + Depth Sensor Dual	13MP + Depth Sensor Dual
17	Leitz	47.2MP + 1.9MP Dual	47.2MP + 1.9MP Dual
18	Lenovo	64MP + 16MP Dual	64MP + 13MP Dual
19	Letv	8MP + Depth Sensor Dual	8MP + Depth Sensor Dual
20	LG	Dual Display	48MP + 8MP + 5MP Triple
21	Lyf	13MP + 2MP Dual	13MP + 2MP Dual
22	Micromax	8MP	13MP + 2MP Dual
23	Motorola	64MP Quad	108MP + 13MP + 2MP Triple
24	Nokia	50MP + 5MP + 2MP Triple	108MP Quad
25	Nothing	50MP + 50MP Dual	50MP + 12MP Dual
26	Nubia	64MP + 8MP + 2MP Triple	50MP + 8MP + 2MP Triple
27	OnePlus	64MP Quad	108MP + 2MP + 2MP Triple
28	Oppo	Foldable Display, Dual Display	108MP + 13MP + 2MP Triple
29	Oukitel	Dual Display	16MP + 13MP + 13MP Triple
30	Poco	8MP Dual	108MP + 13MP + 2MP Triple
31	Realme	8MP	108MP + 2MP Dual
32	Redmi	50MP + Depth Sensor Dual	108MP + 8MP + 2MP Triple
33	Royole	Foldable Display, Dual Display	Foldable Display, Dual Display
34	Samsung	Foldable Display, Dual Display	108MP Quad
35	Sharp	48MP + 12.2MP + 12.2MP Triple	48MP + 12.2MP + 12.2MP Triple
36	Sony	50MP + 2MP Dual	12MP + 12MP + 12MP Triple
37	TCL	13MP	13MP
38	Tecno	64MP + 50MP + 2MP Triple	13MP + 2MP Dual
39	Tesla	50MP + 50MP + 50MP Triple	50MP + 50MP + 50MP Triple
40	Vertu	13MP	13MP
41	Vivo	8MP	108MP + 13MP + 2MP Triple
42	Xiaomi	8MP + 0.3MP Dual	108MP + 13MP + 5MP Triple
43	ZTE	64MP + 64MP + 64MP Triple	50MP Quad

Insights : Camera resolutions among smartphone models have shown a significant evolution over time, with newer models featuring higher megapixel counts compared to older ones. The trend indicates a shift towards higher resolutions, such as 108MP and beyond, among leading brands and newer entries in the market, while older models typically range from 8MP to 64MP.

10. Comparison of Memory Card Support Across Brands

In [103…
```
df = pd.read_sql_query(
"""
SELECT LEFT(Model, LOCATE(' ', Model)) AS Brand,
       CASE
           WHEN Card = 'Memory Card Supported' THEN 'Supported'
           ELSE 'Not Supported'
       END AS Memory_Card_Support,
       COUNT(*) AS Model_Count
FROM smartphone
GROUP BY Brand, Memory_Card_Support
ORDER BY Brand, Memory_Card_Support;

"""
, mydb)
df
```

Out[103]:

	Brand	Memory_Card_Support	Model_Count
0	Apple	Not Supported	43
1	Asus	Not Supported	6
2	Blackview	Supported	1
3	BLU	Supported	1
4	Cola	Not Supported	1
5	Doogee	Not Supported	2
6	Gionee	Not Supported	1
7	Google	Not Supported	12
8	Honor	Not Supported	10
9	Honor	Supported	1
10	Huawei	Not Supported	7
11	Huawei	Supported	3
12	iKall	Not Supported	3
13	Infinix	Not Supported	23
14	Infinix	Supported	5
15	iQOO	Not Supported	32
16	itel	Not Supported	1
17	Jio	Not Supported	1
18	Lava	Not Supported	3
19	LeEco	Not Supported	1
20	Leitz	Not Supported	1
21	Lenovo	Not Supported	2
22	Letv	Not Supported	1
23	LG	Not Supported	2
24	Lyf	Not Supported	1
25	Micromax	Not Supported	3
26	Motorola	Not Supported	45
27	Motorola	Supported	3
28	Nokia	Not Supported	7
29	Nokia	Supported	3
30	Nothing	Not Supported	5
31	Nubia	Not Supported	6
32	OnePlus	Not Supported	33
33	OnePlus	Supported	5
34	OPPO	Not Supported	63
35	Oppo	Supported	19
36	Oukitel	Not Supported	1
37	Oukitel	Supported	2
38	Poco	Not Supported	40
39	Realme	Not Supported	88
40	Realme	Supported	4
41	Redmi	Not Supported	3
42	Royole	Not Supported	2
43	Samsung	Not Supported	113
44	Samsung	Supported	3
45	Sharp	Not Supported	1
46	Sony	Not Supported	6
47	Sony	Supported	1
48	TCL	Not Supported	1
49	Tecno	Not Supported	27
50	Tecno	Supported	4
51	Tesla	Not Supported	1
52	Vertu	Not Supported	1
53	Vivo	Not Supported	78
54	Vivo	Supported	18
55	Xiaomi	Not Supported	121
56	Xiaomi	Supported	4
57	ZTE	Not Supported	2

Insight: Among smartphone brands, the majority do not support external memory cards, with Xiaomi and Vivo leading in models that do support them, highlighting a preference for integrated storage solutions over expandable storage options across the market.

11. Analysis of Price and Ratings by Camera Setup

In [104…
```
df = pd.read_sql_query(
"""
SELECT
    CASE
        WHEN Rear_Camera LIKE '%Triple%' THEN 'Triple Camera'
        WHEN Rear_Camera LIKE '%Quad%' THEN 'Quad Camera'
        ELSE 'Other'
    END AS Camera_Setup,
    ROUND(AVG(Price), 2) AS Avg_Price,
    ROUND(AVG(Ratings), 2) AS Avg_Ratings,
    COUNT(*) AS Model_Count
FROM smartphone
GROUP BY Camera_Setup
ORDER BY Avg_Price DESC;

"""
, mydb)
df
```

Out[104]:

	Camera_Setup	Avg_Price	Avg_Ratings	Model_Count
0	Triple Camera	31473.18	8.05	506
1	Other	28323.55	7.23	242
2	Quad Camera	23628.43	8.07	129

Insight: The analysis shows that smartphones with triple camera setups tend to have higher average prices and ratings compared to those with quad camera setups, despite the latter having a higher model count.

12. Analysis of Smartphone Models with High Ratings and Low Price

In [105…
```
df = pd.read_sql_query(
"""
WITH Avg_Ratings AS (
    SELECT AVG(Ratings) AS Overall_Avg_Ratings
    FROM smartphone
),
Above_Avg_Ratings AS (
    SELECT Model, Price, Ratings
    FROM smartphone
    WHERE Ratings > (SELECT Overall_Avg_Ratings FROM Avg_Ratings)
)
SELECT Model, Price, Ratings
FROM Above_Avg_Ratings
WHERE Price < (SELECT AVG(Price) FROM smartphone)
ORDER BY Ratings DESC, Price ASC
LIMIT 10;


"""
, mydb)
df
```

Out[105]:

	Model	Price	Ratings
0	Xiaomi Redmi Note 12 Explorer	24999	8.9
1	Infinix Zero 20	17999	8.7
2	Motorola Edge 20 Fusion 5G	18999	8.7
3	Motorola Edge S	22490	8.7
4	Xiaomi Redmi Note 11 Pro Plus 5G (8GB RAM + 25...	22999	8.7
5	Samsung Galaxy M53 5G (8GB RAM + 128GB)	25289	8.7
6	Realme X50 Pro 5G (12GB RAM + 256GB)	27999	8.7
7	Motorola Moto G82 (8GB RAM + 128GB)	20499	8.6
8	Samsung Galaxy A52 (8GB RAM + 128GB)	22494	8.6
9	Samsung Galaxy A71	22494	8.6

Insight: The listed smartphone models combine high ratings with relatively low prices, making them attractive options for consumers seeking value and performance.

See All about Project Material :

Smartphone Dataset in csv format

See More Projects

- SQL

-

-

Connect With Me On

- Linkdin Profile
- GitHub Profile
- Email id
- Contact With Whatsapp
- Instagram id
- FaceBook id
- Address Google
- Resume

In [ ]:

In [ ]: