

BUSINESS PROBLEM

The objective is to address a hypothetical business problem for a Flipkart Authorized Seller. According to the problem the individual is looking to sell mobile phones on Flipkart. For this, the individual is looking for the best product, brand, specification and deals that can generate the most revenue with the least amount of investment and budget constraints. QUESTIONS TO BE ANSWERED:

- Should he simply sell products for one brand, or should he try to sell models from various brands?
- Using EDA and Data Visualization find out insights and relation between different features.
- Perform detailed analysis of each brand.

ASSUMPTION

We don't have a direct sales record that shows how many units of a certain mobile model were sold.

In most cases, the number of people who rate a product is proportional to the number of units sold. As a result, we are considering the number of persons rating the product as the equivalent units sold in the solution.

DESCRIPTION OF ATTRIBUTES

- I. brand : Brand Name (Categorical)
- II. model : Model Name (Categorical)
- III. base_color : Phone Color (Categorical)
- IV. processor : Processor brand used (Categorical)
- V. screen_size : Categorical screen size (Categorical)
- VI. ROM : ROM in gigabyte (Numeric – Discrete)
- VII. RAM : RAM in gigabyte (Numeric – Discrete)
- VIII. display_size : Actual display size in inches (Numeric – Continuous)
- IX. num_rear_camera : No. of cameras on back (Numeric – Discrete)
- X. num_front_camera: No. of cameras on front (Numeric – Discrete)
- XI. battery_size : Battery in mAh (Numeric – Continuous)
- XII. ratings : Customer rating for the product (Numeric – Continuous)
- XIII. num_of_ratings : No. of people rating the product, also the equivalent no. of unit sold for our problem (Numeric – Continuous)
- XIV. sales_price : Selling price of the unit after discount (Numeric –Continuous)
- XV. discount_percent: Discount in percentage offered (Numeric –Continuous)
- XVI. sales : Sales of product in crore rupees (Numeric – Continuous)

EXPLORATORY DATA ANALYSIS

EDA is one of the most important phases in data science since it helps us to obtain critical insights and statistical metrics. In general, EDA can be categorised in two ways.

The first distinction is that each method is either non-graphical or graphical. Second, each method is univariate or multivariate in nature (usually just bivariate).

Non-graphical approaches typically include the computation of summary statistics, but graphical methods clearly summarize the data in a diagrammatic or pictorial manner.

Let's look at each type individually.

```
In [13]: # IMPORTING LIBRARIES
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib inline
import warnings
warnings.filterwarnings('ignore')

In [14]: #DATE RETRIEVAL
df = pd.read_csv('Flipkart_Mobile.csv')

In [15]: df.head()

Out[15]:
  brand  model  base_color  processor  screen_size  ROM  display_size  num_rear_camera  num_front_camera  battery_capacity  ratings  num_of_ratings  sales_price  discount_percent  sales
0  Apple  iPhone SE  Black  Water  Very Small  64  2  4.7  1  1  1800  4.5  38645  32999  0.17  127.52
1  Apple  iPhone 12 Mini  Red  Ceramic  Small  64  4  5.4  2  1  2815  4.5  2844  57149  0.04  1.39
2  Apple  iPhone SE  Red  Water  Very Small  64  2  4.7  1  1  1800  4.5  38645  32999  0.17  127.52
3  Apple  iPhone XR  Others  iOS  Medium  64  3  6.1  1  1  2942  4.6  5386  42999  0.10  23.07
4  Apple  iPhone 12  Red  Ceramic  Medium  128  4  6.1  2  1  2815  4.6  745  69149  0.02  5.15

In [16]: #QUICK INFORMATION ABOUT DATA
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 430 entries, 0 to 429
Data columns (total 16 columns):
 #   Column              Non-Null Count  Dtype
---  --
 0  brand              430 non-null     object
 1  model              430 non-null     object
 2  base_color         430 non-null     object
 3  processor          430 non-null     object
 4  screen_size        430 non-null     object
 5  ROM                430 non-null     int64
 6  RAM                430 non-null     int64
 7  display_size       430 non-null     float64
 8  num_rear_camera    430 non-null     int64
 9  num_front_camera   430 non-null     int64
10  battery_capacity    430 non-null     float64
11  ratings            430 non-null     float64
12  num_of_ratings      430 non-null     int64
13  sales_price        430 non-null     int64
14  discount_percent    430 non-null     float64
15  sales              430 non-null     float64
dtypes: float64(4), int64(7), object(5)
memory usage: 53.9+ KB

In [17]: # DATA DISTRIBUTION INFORMATION
df.describe()

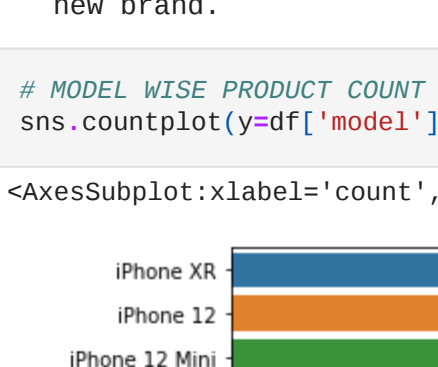
Out[17]:
   ROM  RAM  display_size  num_rear_camera  num_front_camera  battery_capacity  ratings  num_of_ratings  sales_price  discount_percent  sales
count  430.000000  430.000000  430.000000  430.000000  430.000000  430.000000  430.000000  430.000000  430.000000  430.000000  430.000000
mean    105.148837    5.320930    6.369767    2.904651    1.044186    4529.397674    4.339302    23567.944186    25433.234884    0.108000    29.752326
std     63.164064    2.182835    0.369767    0.952350    0.227280    986.807252    0.151484    56096.277784    22471.926588    0.073432    58.396988
min      8.000000    1.000000    4.700000    1.000000    1.000000    1800.000000    3.000000    4.000000    5742.000000    0.010000    0.000000
25%     64.000000    4.000000    6.300000    2.000000    1.000000    4000.000000    4.300000    745.000000    11999.000000    0.060000    1.640000
50%    128.000000    4.000000    6.500000    3.000000    1.000000    4500.000000    4.300000    5197.500000    16989.500000    0.090000    9.655000
75%    128.000000    6.000000    6.500000    4.000000    1.000000    5000.000000    4.400000    21089.250000    28999.000000    0.160000    29.717500
max    512.000000   12.000000    7.600000    4.000000    3.000000    7600.000000    4.600000    642373.000000   157999.000000    0.440000   550.190000

In [18]: # COLUMN NAMES FOR GETTING IDEA ABOUT DATA
df.columns

Out[18]:
Index(['brand', 'model', 'base_color', 'processor', 'screen_size', 'ROM', 'RAM', 'display_size', 'num_rear_camera', 'num_front_camera', 'battery_capacity', 'ratings', 'num_of_ratings', 'sales_price', 'discount_percent', 'sales'],
      dtype='object')
```

```
In [19]: ##BRAND-WISE DISTINCT PRODUCT
df['brand'].value_counts().plot(kind='pie', autopct='%2f')

Out[19]: <AxesSubplot: ylabel='brand'>
```

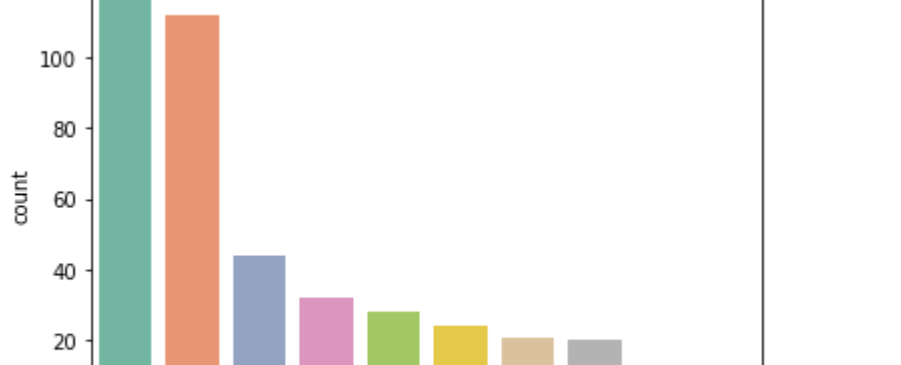


OBSERVATION :

Realme offers the most options, as seen by pie charts. In terms of specs, they have a broad range of phones to select from Poco, on the other hand, has the fewest alternatives. This might be since Poco is a relatively new brand.

```
In [20]: # MODEL WISE PRODUCT COUNT
sns.countplot(y=df['model'], order=df.model.value_counts().iloc[:10].index)

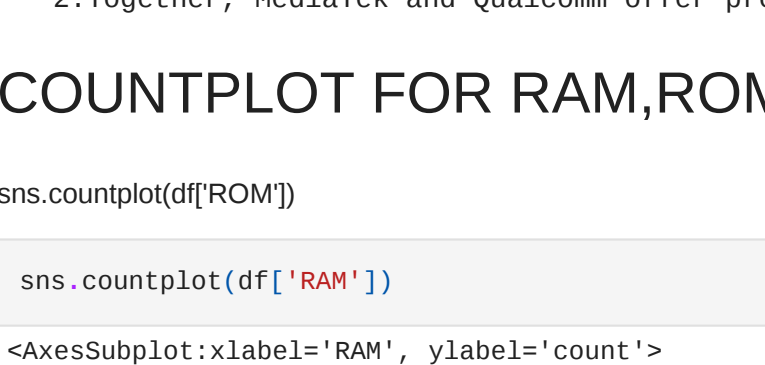
Out[20]: <AxesSubplot: xlabel='count', ylabel='model'>
```



- OBSERVATION :
- The model iPhone XR, iPhone 12, and iPhone 12 mini have the largest variance within the model, as shown in the above bar chart.
 - There might be differences in the mobile's specifications, pricing, and colour.
 - This also explains why Apple and Poco have a lower model count because they have the most versions with different specifications.

```
In [21]: #COLOUR WISE PRODUCT COUNT:
sns.countplot(df['base_color'], order=df.base_color.value_counts().iloc[:10].index,palette='Set2')

Out[21]: <AxesSubplot: xlabel='base_color', ylabel='count'>
```

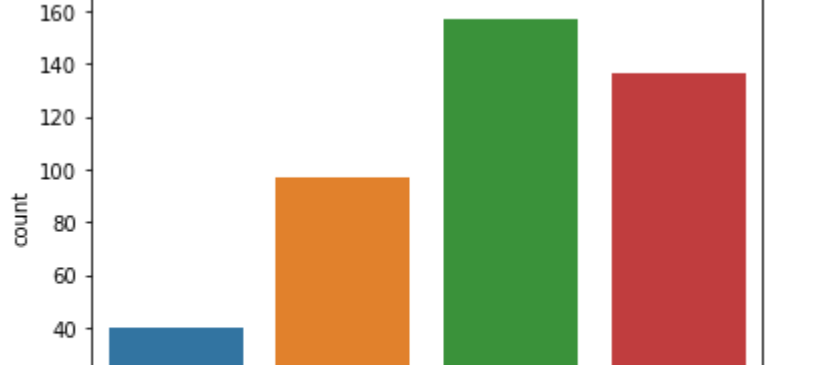


OBSERVATION :

Blue is the most common colour, followed by black and white.

```
In [22]: # PROCESSOR WISE PRODUCT COUNT
sns.countplot(y=df['processor'], order=df.processor.value_counts().iloc[:10].index)

Out[22]: <AxesSubplot: xlabel='count', ylabel='processor'>
```



- OBSERVATION :
- Qualcomm is the most prevalent CPU brand, accounting for 168 of the 430 mobile phones.
 - Together, MediaTek and Qualcomm offer processors for more than half of all mobile phones.

COUNTPLOT FOR RAM,ROM,REAR_CAMERA AND FRONT_CAMERA

```
sns.countplot(df['ROM'])

In [23]: sns.countplot(df['RAM'])

Out[23]: <AxesSubplot: xlabel='RAM', ylabel='count'>
```

```
In [24]: sns.countplot(df['num_rear_camera'])

Out[24]: <AxesSubplot: xlabel='num_rear_camera', ylabel='count'>
```

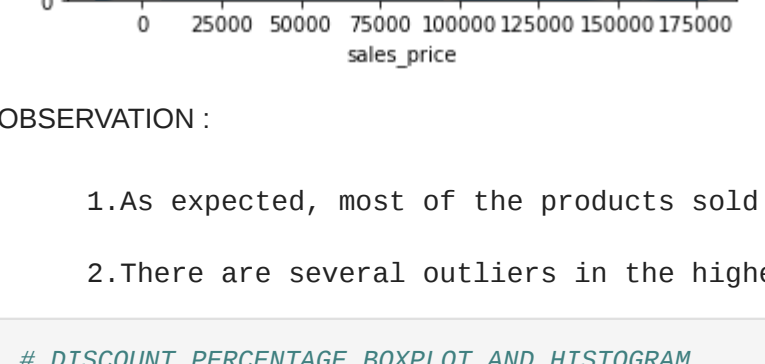
```
In [25]: sns.countplot(df['num_front_camera'])

Out[25]: <AxesSubplot: xlabel='num_front_camera', ylabel='count'>
```

- OBSERVATION :
- The bar graphs above indicate the number of different products based on the features - RAM, ROM, number of front cameras, and number of back cameras.
 - There are around 192 mobile phones with 128 GB of ROM and 133 mobile phones with 4 GB of RAM.
 - Mobile phones with multiple front cameras are rare, but phones with multiple rear cameras are widespread.

```
In [26]: # RATINGS BOXPLOT AND HISTOGRAM
sns.boxplot(df['ratings'],bandwidth=0.1,kde = True)

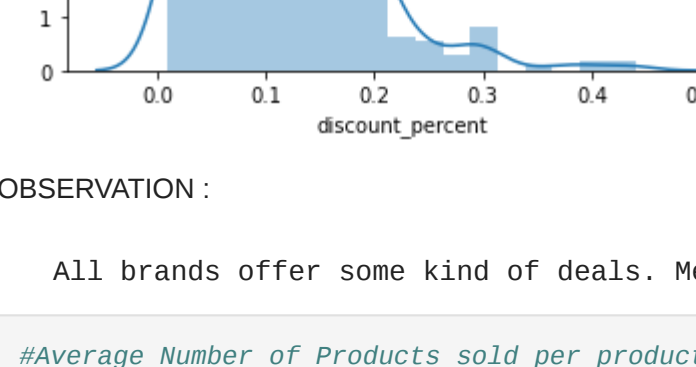
Out[26]: <AxesSubplot: xlabel='ratings', ylabel='Count'>
```



- OBSERVATION :
- Most of the products get excellent reviews.
 - With a mean of 4.4 and a median of 4.3.
 - Due to some mobiles with lower ratings, the distribution is skewed

```
In [28]: # SELLING PRICE BOXPLOT AND HISTOGRAM
sns.boxplot(df['sales_price'])

Out[28]: <AxesSubplot: xlabel='sales_price'>
```



```
In [29]: sns.distplot(df['sales_price'])

Out[29]: <AxesSubplot: xlabel='sales_price', ylabel='Density'>
```

- OBSERVATION :
- As expected, most of the products sold are under Rs 20000.
 - There are several outliers in the higher range, therefore the distribution is right skewed.

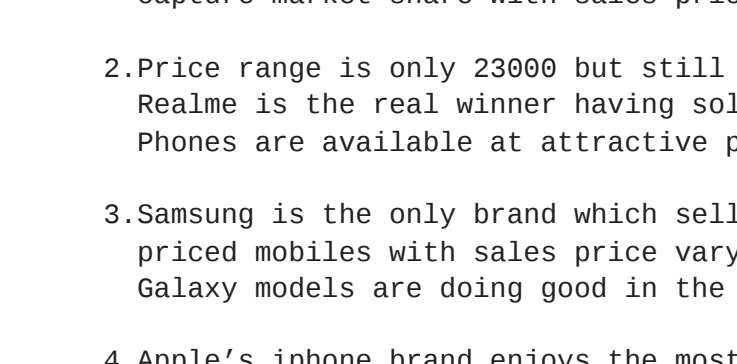
```
In [30]: # DISCOUNT PERCENTAGE BOXPLOT AND HISTOGRAM
sns.boxplot(df['discount_percent'])

Out[30]: <AxesSubplot: xlabel='discount_percent'>
```



```
In [31]: sns.distplot(df['discount_percent'])

Out[31]: <AxesSubplot: xlabel='discount_percent', ylabel='Density'>
```

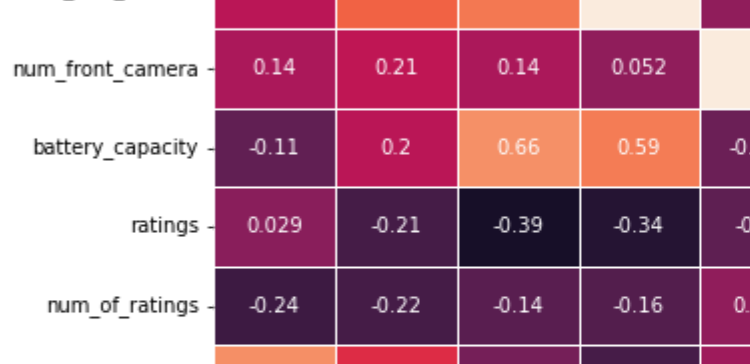


OBSERVATION :

All brands offer some kind of deals. Mean discount offered by brand is 10 percent.

```
In [32]: #Average Number of Products sold per product
sns.barplot(x='sales',x='brand',data=df)

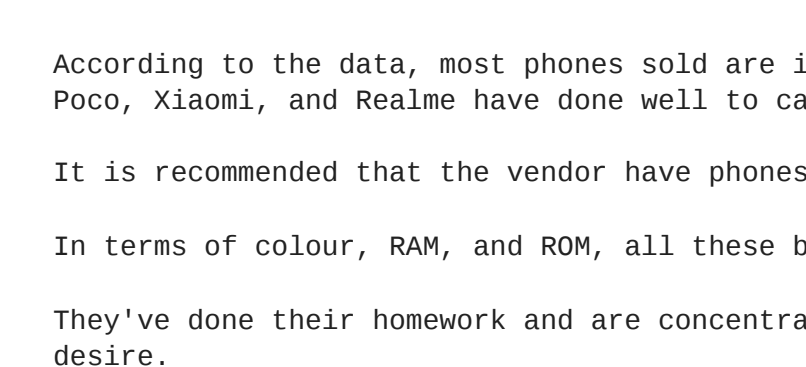
Out[32]: <AxesSubplot: xlabel='brand', ylabel='sales'>
```



- OBSERVATION
- We can see that Xiaomi and Poco are the most popular brands in India, with Realme following closely after. According to the bar graph, each Xiaomi device with any specification and colour will sell at least 28000 units, with an average of 48800 units.

```
In [33]: # BRAND VS SALES PRICE USING CATPLOT
sns.catplot(x='brand',y='sales_price',kind='strip',data=df)

Out[33]: <seaborn.axisgrid.FacetGrid at 0x1b378aa7a3b>
```



- OBSERVATION :
- Poco is a relatively new entrant in the market but it has managed to capture market share with sales price varying from 7999 to 30999.
 - Price range is only 23000 but still they have managed to sell 56 units. Realme is the real winner having sold the maximum number of phones. Phones are available at attractive prices with price ranging from 6499-41999.
 - Samsung is the only brand which sells low-priced, medium priced and high priced mobiles with sales price varying from 7999 to 157999. Some of the Galaxy models are doing good in the higher range.
 - Apple is the iPhone brand enjoys the most loyal customers and is the market leader in the 30-75K price segment with 92% of the existing iPhone users plan to stick to the brand when they upgrade to a new phone. For Apple, only some selected models are doing good.
 - For Xiaomi, 75% of the phones sold are below Rs. 21550.

```
In [35]: # CORRELATION HEATMAP
plt.figure(figsize = (15,9))
sns.heatmap(df.corr(),annot=True,linewidths=.5)

Out[35]: <AxesSubplot: >
```


- OBSERVATION :
- Correlation between different features:
- ROM - ROM is moderately correlated with ROM and sales_price, As RAM size increase sales price and RAM size also increases
 - RAM - RAM is positively related with display size and number of cameras. It is relatively less correlated with sales price.
 - Display Size - As Size of phone increase number of rear camera and battery capacity also increases.
 - Battery Capacity - Battery capacity opposite to what one may think, it is negatively correlated with sales price and ratings.
 - Number of ratings - As sales is a derived quantity of number of ratings that's why it has very high correlation with it.

CONCLUSION

According to the data, most phones sold are in the price range of RS.15,000 to RS.20,000. Poco, Xiaomi, and Realme have done well to capture the market in this segment.

It is recommended that the vendor have phones from all these companies in this price range.

In terms of colour, RAM, and ROM, all these brands provide a wide range of models.

They've done their homework and are concentrating on the processor, display size, and colour that Indian consumers desire.

However, if getting a franchise of multiple brands is difficult and costly, then the seller should choose Poco. Because Poco is a new brand, the seller may be able to negotiate better terms with the vendor, resulting in more revenue and profit. People in India are changing their buying patterns. The risk will also be substantial because the profit margin is unknown and the investment to be made is high.

```
In [ ]:
```