# AmaAmazon sales Data Analysis


In [3]:
```
import numpy as np # linear algebra
import pandas as pd # data preprocessing
```

In [6]:
```
df = pd.read_csv(r"C:\Users\chitt\Downloads\amazon.csv\amazon.csv")
```

In [8]:
```
df
```

Out[8]:

| | product_id | product_name | category | d |
|---|---|---|---|---|
| **0** | B07JW9H4J1 | Wayona Nylon Braided USB to Lightning Fast Cha... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **1** | B098NS6PVG | Ambrane Unbreakable 60W / 3A Fast Charging 1.5... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **2** | B096MSW6CT | Sounce Fast Phone Charging Cable & Data Sync U... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **3** | B08HDJ86NZ | boAt Deuce USB 300 2 in 1 Type-C & Micro USB S... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **4** | B08CF3B7N1 | Portronics Konnect L 1.2M Fast Charging 3A 8 P... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **...** | ... | ... | ... | |
| **1460** | B08L7J3T31 | Noir Aqua - 5pcs PP Spun Filter + 1 Spanner \| ... | Home&Kitchen\|Kitchen&HomeAppliances\|WaterPurif... | |
| **1461** | B01M6453MB | Prestige Delight PRWO Electric Rice Cooker (1 ... | Home&Kitchen\|Kitchen&HomeAppliances\|SmallKitch... | |
| **1462** | B009P2LIL4 | Bajaj Majesty RX10 2000 Watts Heat Convector R... | Home&Kitchen\|Heating,Cooling&AirQuality\|RoomHe... | |
| **1463** | B00J5DYCCA | Havells Ventil Air DSP 230mm Exhaust Fan (Pist... | Home&Kitchen\|Heating,Cooling&AirQuality\|Fans\|E... | |
| **1464** | B01486F4G6 | Borosil Jumbo 1000-Watt Grill Sandwich Maker (... | Home&Kitchen\|Kitchen&HomeAppliances\|SmallKitch... | |

1465 rows × 16 columns

In [10]:
```
df.head()
```

Out[10]:

| | product_id | product_name | category | discoun |
|---|---|---|---|---|
| **0** | B07JW9H4J1 | Wayona Nylon Braided USB to Lightning Fast Cha... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **1** | B098NS6PVG | Ambrane Unbreakable 60W / 3A Fast Charging 1.5... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **2** | B096MSW6CT | Sounce Fast Phone Charging Cable & Data Sync U... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **3** | B08HDJ86NZ | boAt Deuce USB 300 2 in 1 Type-C & Micro USB S... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **4** | B08CF3B7N1 | Portronics Konnect L 1.2M Fast Charging 3A 8 P... | Computers&Accessories\|Accessories&Peripherals\|... | |

In [12]:
```
df.tail()
```

Out[12]:

| | product_id | product_name | category | d |
|---|---|---|---|---|
| **1460** | B08L7J3T31 | Noir Aqua - 5pcs PP Spun Filter + 1 Spanner \| ... | Home&Kitchen\|Kitchen&HomeAppliances\|WaterPurif... | |
| **1461** | B01M6453MB | Prestige Delight PRWO Electric Rice Cooker (1 ... | Home&Kitchen\|Kitchen&HomeAppliances\|SmallKitch... | |
| **1462** | B009P2LIL4 | Bajaj Majesty RX10 2000 Watts Heat Convector R... | Home&Kitchen\|Heating,Cooling&AirQuality\|RoomHe... | |
| **1463** | B00J5DYCCA | Havells Ventil Air DSP 230mm Exhaust Fan (Pist... | Home&Kitchen\|Heating,Cooling&AirQuality\|Fans\|E... | |
| **1464** | B01486F4G6 | Borosil Jumbo 1000-Watt Grill Sandwich Maker (... | Home&Kitchen\|Kitchen&HomeAppliances\|SmallKitch... | |

In [14]:
```
df.columns
```

Out[14]:
```
Index(['product_id', 'product_name', 'category', 'discounted_price',
       'actual_price', 'discount_percentage', 'rating', 'rating_count',
       'about_product', 'user_id', 'user_name', 'review_id', 'review_title',
       'review_content', 'img_link', 'product_link'],
      dtype='object')
```

In [16]:
```
df.shape
```

Out[16]:
```
(1465, 16)
```

In [18]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1465 entries, 0 to 1464
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   product_id           1465 non-null   object
 1   product_name         1465 non-null   object
 2   category             1465 non-null   object
 3   discounted_price     1465 non-null   object
 4   actual_price         1465 non-null   object
 5   discount_percentage  1465 non-null   object
 6   rating               1465 non-null   object
 7   rating_count         1463 non-null   object
 8   about_product        1465 non-null   object
 9   user_id              1465 non-null   object
 10  user_name            1465 non-null   object
 11  review_id            1465 non-null   object
 12  review_title         1465 non-null   object
 13  review_content       1465 non-null   object
 14  img_link             1465 non-null   object
 15  product_link         1465 non-null   object
dtypes: object(16)
memory usage: 183.3+ KB
```

In [20]:
```
df.isnull().sum()
```

Out[20]:
```
product_id            0
product_name          0
category              0
discounted_price      0
actual_price          0
discount_percentage   0
rating                0
rating_count          2
about_product         0
user_id               0
user_name             0
review_id             0
review_title          0
review_content        0
img_link              0
product_link          0
dtype: int64
```

# Observation

1. all feature are object type
2. there are 2 missing value in rating_count
3. 1465 rows and 16 columns

# Data Cleaning

In [24]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1465 entries, 0 to 1464
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   product_id           1465 non-null   object
 1   product_name         1465 non-null   object
 2   category             1465 non-null   object
 3   discounted_price     1465 non-null   object
 4   actual_price         1465 non-null   object
 5   discount_percentage  1465 non-null   object
 6   rating               1465 non-null   object
 7   rating_count         1463 non-null   object
 8   about_product        1465 non-null   object
 9   user_id              1465 non-null   object
 10  user_name            1465 non-null   object
 11  review_id            1465 non-null   object
 12  review_title         1465 non-null   object
 13  review_content       1465 non-null   object
 14  img_link             1465 non-null   object
 15  product_link         1465 non-null   object
dtypes: object(16)
memory usage: 183.3+ KB
```

## 1.1 : Change Data type

In [27]:
```
#DISCOUNTED PRICE
# df['discounted_price'].astype('float64')

df['discounted_price']=df['discounted_price'].str.replace('₹', '')
```

In [29]:
```
df['discounted_price']=df['discounted_price'].str.replace(',','')
```

In [31]:
```
df['discounted_price']=df['discounted_price'].astype('float64')
```

In [33]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1465 entries, 0 to 1464
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   product_id           1465 non-null   object
 1   product_name         1465 non-null   object
 2   category             1465 non-null   object
 3   discounted_price     1465 non-null   float64
 4   actual_price         1465 non-null   object
 5   discount_percentage  1465 non-null   object
 6   rating               1465 non-null   object
 7   rating_count         1463 non-null   object
 8   about_product        1465 non-null   object
 9   user_id              1465 non-null   object
 10  user_name            1465 non-null   object
 11  review_id            1465 non-null   object
 12  review_title         1465 non-null   object
 13  review_content       1465 non-null   object
 14  img_link             1465 non-null   object
 15  product_link         1465 non-null   object
dtypes: float64(1), object(15)
memory usage: 183.3+ KB
```

In [37]:
```
#actual_price
df['actual_price'] = df['actual_price'].str.replace(',','')
df['actual_price'] = df['actual_price'].str.replace('₹','').astype('float64')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1465 entries, 0 to 1464
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   product_id           1465 non-null   object
 1   product_name         1465 non-null   object
 2   category             1465 non-null   object
 3   discounted_price     1465 non-null   float64
 4   actual_price         1465 non-null   float64
 5   discount_percentage  1465 non-null   object
 6   rating               1465 non-null   object
 7   rating_count         1463 non-null   object
 8   about_product        1465 non-null   object
 9   user_id              1465 non-null   object
 10  user_name            1465 non-null   object
 11  review_id            1465 non-null   object
 12  review_title         1465 non-null   object
 13  review_content       1465 non-null   object
 14  img_link             1465 non-null   object
 15  product_link         1465 non-null   object
dtypes: float64(2), object(14)
memory usage: 183.3+ KB
```

In [39]:
```
df.describe()
```

Out[39]:

| | discounted_price | actual_price |
|---|---|---|
| **count** | 1465.000000 | 1465.000000 |
| **mean** | 3125.310874 | 5444.990635 |
| **std** | 6944.304394 | 10874.826864 |
| **min** | 39.000000 | 39.000000 |
| **25%** | 325.000000 | 800.000000 |
| **50%** | 799.000000 | 1650.000000 |
| **75%** | 1999.000000 | 4295.000000 |
| **max** | 77990.000000 | 139900.000000 |

In [41]:
```
# discount_percentage

df['discount_percentage']= df['discount_percentage'].str.replace('%','').astype(
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1465 entries, 0 to 1464
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   product_id           1465 non-null   object 
 1   product_name         1465 non-null   object 
 2   category             1465 non-null   object 
 3   discounted_price     1465 non-null   float64
 4   actual_price         1465 non-null   float64
 5   discount_percentage  1465 non-null   float64
 6   rating               1465 non-null   object 
 7   rating_count         1463 non-null   object 
 8   about_product        1465 non-null   object 
 9   user_id              1465 non-null   object 
 10  user_name            1465 non-null   object 
 11  review_id            1465 non-null   object 
 12  review_title         1465 non-null   object 
 13  review_content       1465 non-null   object 
 14  img_link             1465 non-null   object 
 15  product_link         1465 non-null   object 
dtypes: float64(3), object(13)
memory usage: 183.3+ KB
```

In [43]:
```
#rating
df['rating'].unique()
```

Out[43]:
```
array(['4.2', '4.0', '3.9', '4.1', '4.3', '4.4', '4.5', '3.7', '3.3',
       '3.6', '3.4', '3.8', '3.5', '4.6', '3.2', '5.0', '4.7', '3.0',
       '2.8', '4', '3.1', '4.8', '2.3', '|', '2', '3', '2.6', '2.9'],
      dtype=object)
```

In [45]:
```
df['rating'].value_counts()
```

Out[45]:
```
rating
4.1    244
4.3    230
4.2    228
4.0    129
3.9    123
4.4    123
3.8     86
4.5     75
4       52
3.7     42
3.6     35
3.5     26
4.6     17
3.3     16
3.4     10
4.7      6
3.1      4
5.0      3
3.0      3
4.8      3
3.2      2
2.8      2
2.3      1
|        1
2        1
3        1
2.6      1
2.9      1
Name: count, dtype: int64
```

In [47]:
```
# df['rating'].str.replace('|', '').astype('float64')
# df.query('rating=="|"')
df['rating'] = df['rating'].str.replace('|', '3.5').astype('float64')
```

In [49]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1465 entries, 0 to 1464
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   product_id           1465 non-null   object 
 1   product_name         1465 non-null   object 
 2   category             1465 non-null   object 
 3   discounted_price     1465 non-null   float64
 4   actual_price         1465 non-null   float64
 5   discount_percentage  1465 non-null   float64
 6   rating               1465 non-null   float64
 7   rating_count         1463 non-null   object 
 8   about_product        1465 non-null   object 
 9   user_id              1465 non-null   object 
 10  user_name            1465 non-null   object 
 11  review_id            1465 non-null   object 
 12  review_title         1465 non-null   object 
 13  review_content       1465 non-null   object 
 14  img_link             1465 non-null   object 
 15  product_link         1465 non-null   object 
dtypes: float64(4), object(12)
memory usage: 183.3+ KB
```

In [51]:
```
df['rating_count']= df['rating_count'].str.replace(',','').astype('float64')
```

In [53]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1465 entries, 0 to 1464
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   product_id           1465 non-null   object 
 1   product_name         1465 non-null   object 
 2   category             1465 non-null   object 
 3   discounted_price     1465 non-null   float64
 4   actual_price         1465 non-null   float64
 5   discount_percentage  1465 non-null   float64
 6   rating               1465 non-null   float64
 7   rating_count         1463 non-null   float64
 8   about_product        1465 non-null   object 
 9   user_id              1465 non-null   object 
 10  user_name            1465 non-null   object 
 11  review_id            1465 non-null   object 
 12  review_title         1465 non-null   object 
 13  review_content       1465 non-null   object 
 14  img_link             1465 non-null   object 
 15  product_link         1465 non-null   object 
dtypes: float64(5), object(11)
memory usage: 183.3+ KB
```

In [55]:
```
#descriptive stats

df.describe()
```

Out[55]:

| | discounted_price | actual_price | discount_percentage | rating | rating_coun |
|---|---|---|---|---|---|
| **count** | 1465.000000 | 1465.000000 | 1465.000000 | 1465.000000 | 1463.00000 |
| **mean** | 3125.310874 | 5444.990635 | 47.691468 | 4.096177 | 18295.54135 |
| **std** | 6944.304394 | 10874.826864 | 21.635905 | 0.291991 | 42753.86495 |
| **min** | 39.000000 | 39.000000 | 0.000000 | 2.000000 | 2.00000 |
| **25%** | 325.000000 | 800.000000 | 32.000000 | 4.000000 | 1186.00000 |
| **50%** | 799.000000 | 1650.000000 | 50.000000 | 4.100000 | 5179.00000 |
| **75%** | 1999.000000 | 4295.000000 | 63.000000 | 4.300000 | 17336.50000 |
| **max** | 77990.000000 | 139900.000000 | 94.000000 | 5.000000 | 426973.00000 |

# 1.2 Handle missing values

In [58]:
```
df.isnull().sum().sort_values(ascending=False)
```

Out[58]:
```
rating_count           2
product_id             0
product_name           0
category               0
discounted_price       0
actual_price           0
discount_percentage    0
rating                 0
about_product          0
user_id                0
user_name              0
review_id              0
review_title           0
review_content         0
img_link               0
product_link           0
dtype: int64
```

In [60]:
```
df.shape[0]
```

Out[60]:
```
1465
```

In [62]:
```
round(df.isnull().sum()/df.shape[0]*100,2).sort_values(ascending=False)
```

Out[62]:
```
rating_count           0.14
product_id             0.00
product_name           0.00
category               0.00
discounted_price       0.00
actual_price           0.00
discount_percentage    0.00
rating                 0.00
about_product          0.00
user_id                0.00
user_name              0.00
review_id              0.00
review_title           0.00
review_content         0.00
img_link               0.00
product_link           0.00
dtype: float64
```

In [64]:
```
df[df['rating_count'].isnull()]
```

Out[64]:

| | product_id | product_name | category | discou |
|---|---|---|---|---|
| **282** | B0B94JPY2N | Amazon Brand - Solimo 65W Fast Charging Braide... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **324** | B0BQRJ3C47 | REDTECH USB-C to Lightning Cable 3.3FT, [Apple... | Computers&Accessories\|Accessories&Peripherals\|... | |

In [66]:
```
print("mean: ",df['rating_count'].mean())
print("median: ", df['rating_count'].median())
```

```
mean:  18295.541353383458
median:  5179.0
```

In [68]:
```
# import statistics
# x = [1,2,1,2,3,1,100,100,100,100]

# statistics.mode(x)
```

In [70]:
```
#replace ratingcount null values with median
df['rating_count']=df.rating_count.fillna(value=df['rating_count'].median())
```

In [72]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1465 entries, 0 to 1464
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   product_id           1465 non-null   object 
 1   product_name         1465 non-null   object 
 2   category             1465 non-null   object 
 3   discounted_price     1465 non-null   float64
 4   actual_price         1465 non-null   float64
 5   discount_percentage  1465 non-null   float64
 6   rating               1465 non-null   float64
 7   rating_count         1465 non-null   float64
 8   about_product        1465 non-null   object 
 9   user_id              1465 non-null   object 
 10  user_name            1465 non-null   object 
 11  review_id            1465 non-null   object 
 12  review_title         1465 non-null   object 
 13  review_content       1465 non-null   object 
 14  img_link             1465 non-null   object 
 15  product_link         1465 non-null   object 
dtypes: float64(5), object(11)
memory usage: 183.3+ KB
```

In [74]:
```
df.isnull().sum()
```

Out[74]:
```
product_id            0
product_name          0
category              0
discounted_price      0
actual_price          0
discount_percentage   0
rating                0
rating_count          0
about_product         0
user_id               0
user_name             0
review_id             0
review_title          0
review_content        0
img_link              0
product_link          0
dtype: int64
```

## 1.3 Check for duplicates

In [77]:
```
df.duplicated().sum()
```

Out[77]:
```
0
```

In [79]:
```
df.columns
```

Out[79]:
```
Index(['product_id', 'product_name', 'category', 'discounted_price',
       'actual_price', 'discount_percentage', 'rating', 'rating_count',
       'about_product', 'user_id', 'user_name', 'review_id', 'review_title',
       'review_content', 'img_link', 'product_link'],
      dtype='object')
```

In [81]:
```
duplicateValue = df.duplicated(subset=['product_id', 'product_name', 'category',
       'actual_price', 'discount_percentage', 'rating', 'rating_count',
       'about_product', 'user_id', 'user_name', 'review_id', 'review_title',
       'review_content', 'img_link', 'product_link'])

duplicateValue
```

Out[81]:
```
0       False
1       False
2       False
3       False
4       False
        ...
1460    False
1461    False
1462    False
1463    False
1464    False
Length: 1465, dtype: bool
```

# EDA & Visualization

In [84]:
```
df.head()
```

Out[84]:

| | product_id | product_name | category | discoun |
|---|---|---|---|---|
| **0** | B07JW9H4J1 | Wayona Nylon Braided USB to Lightning Fast Cha... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **1** | B098NS6PVG | Ambrane Unbreakable 60W / 3A Fast Charging 1.5... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **2** | B096MSW6CT | Sounce Fast Phone Charging Cable & Data Sync U... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **3** | B08HDJ86NZ | boAt Deuce USB 300 2 in 1 Type-C & Micro USB S... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **4** | B08CF3B7N1 | Portronics Konnect L 1.2M Fast Charging 3A 8 P... | Computers&Accessories\|Accessories&Peripherals\|... | |

In [86]:
```
df['category'].value_counts()
```

Out[86]:
```
category
Computers&Accessories|Accessories&Peripherals|Cables&Accessories|Cables|USBCabl
es                                                  233
Electronics|WearableTechnology|SmartWatches
76
Electronics|Mobiles&Accessories|Smartphones&BasicMobiles|Smartphones
68
Electronics|HomeTheater,TV&Video|Televisions|SmartTelevisions
63
Electronics|Headphones,Earbuds&Accessories|Headphones|In-Ear
52

...
Electronics|Cameras&Photography|Accessories|Batteries&Chargers|BatteryChargers
1
Computers&Accessories|NetworkingDevices|DataCards&Dongles
1
Electronics|HomeAudio|Speakers|MultimediaSpeakerSystems
1
OfficeProducts|OfficePaperProducts|Paper|Copy&PrintingPaper|ColouredPaper
1
Home&Kitchen|Kitchen&HomeAppliances|Vacuum,Cleaning&Ironing|Vacuums&FloorCare|V
acuumAccessories|VacuumBags|HandheldBags      1
Name: count, Length: 211, dtype: int64
```

In [90]:
```
df['main_category']=df['category'].astype(str).str.split('|').str[0]
```

In [92]:
```
df['sub_category']=df['category'].astype(str).str.split('|').str[-1]
```

In [94]:
```
df['sub_category'].value_counts()
```

Out[94]:
```
sub_category
USBCables             233
SmartWatches           76
Smartphones            68
SmartTelevisions       63
In-Ear                 52
                     ...
InternalHardDrives      1
WoodenPencils           1
BatteryChargers         1
DataCards&Dongles       1
HandheldBags            1
Name: count, Length: 207, dtype: int64
```

In [96]:
```
df.columns
```

Out[96]:
```
Index(['product_id', 'product_name', 'category', 'discounted_price',
       'actual_price', 'discount_percentage', 'rating', 'rating_count',
       'about_product', 'user_id', 'user_name', 'review_id', 'review_title',
       'review_content', 'img_link', 'product_link', 'main_category',
       'sub_category'],
      dtype='object')
```

In [98]:
```
df[['rating_count','rating']]
```

Out[98]:

| | rating_count | rating |
|---|---|---|
| **0** | 24269.0 | 4.2 |
| **1** | 43994.0 | 4.0 |
| **2** | 7928.0 | 3.9 |
| **3** | 94363.0 | 4.2 |
| **4** | 16905.0 | 4.2 |
| **...** | ... | ... |
| **1460** | 1090.0 | 4.0 |
| **1461** | 4118.0 | 4.1 |
| **1462** | 468.0 | 3.6 |
| **1463** | 8031.0 | 4.0 |
| **1464** | 6987.0 | 4.3 |

1465 rows × 2 columns

In [100…
```
df['rating_weight'] = df['rating']*df['rating_count']
```

In [102…
```
df.columns
```

Out[102…
```
Index(['product_id', 'product_name', 'category', 'discounted_price',
       'actual_price', 'discount_percentage', 'rating', 'rating_count',
       'about_product', 'user_id', 'user_name', 'review_id', 'review_title',
       'review_content', 'img_link', 'product_link', 'main_category',
       'sub_category', 'rating_weight'],
      dtype='object')
```

In [104…
```
df.isnull().sum()
```

Out[104…
```
product_id            0
product_name          0
category              0
discounted_price      0
actual_price          0
discount_percentage   0
rating                0
rating_count          0
about_product         0
user_id               0
user_name             0
review_id             0
review_title          0
review_content        0
img_link              0
product_link          0
main_category         0
sub_category          0
rating_weight         0
dtype: int64
```

In [106…
```
df.duplicated().sum()
```

Out[106…
```
0
```

# Data Visualization

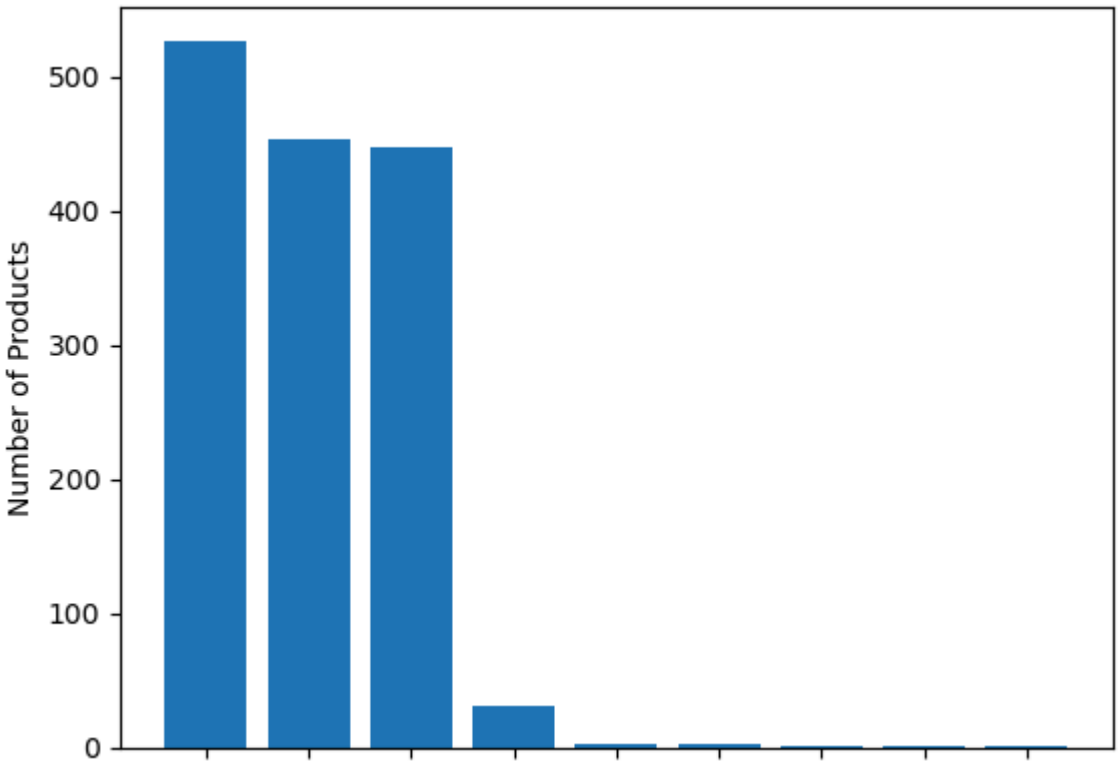
In [109…
```
import matplotlib.pyplot as plt
```

1.Data Distribution of products by main category

In [112…
```
mainCategoryCount= df['main_category'].value_counts()
mainCategoryCount
```

Out[112…
```
main_category
Electronics              526
Computers&Accessories    453
Home&Kitchen             448
OfficeProducts            31
MusicalInstruments         2
HomeImprovement            2
Toys&Games                 1
Car&Motorbike              1
Health&PersonalCare        1
Name: count, dtype: int64
```

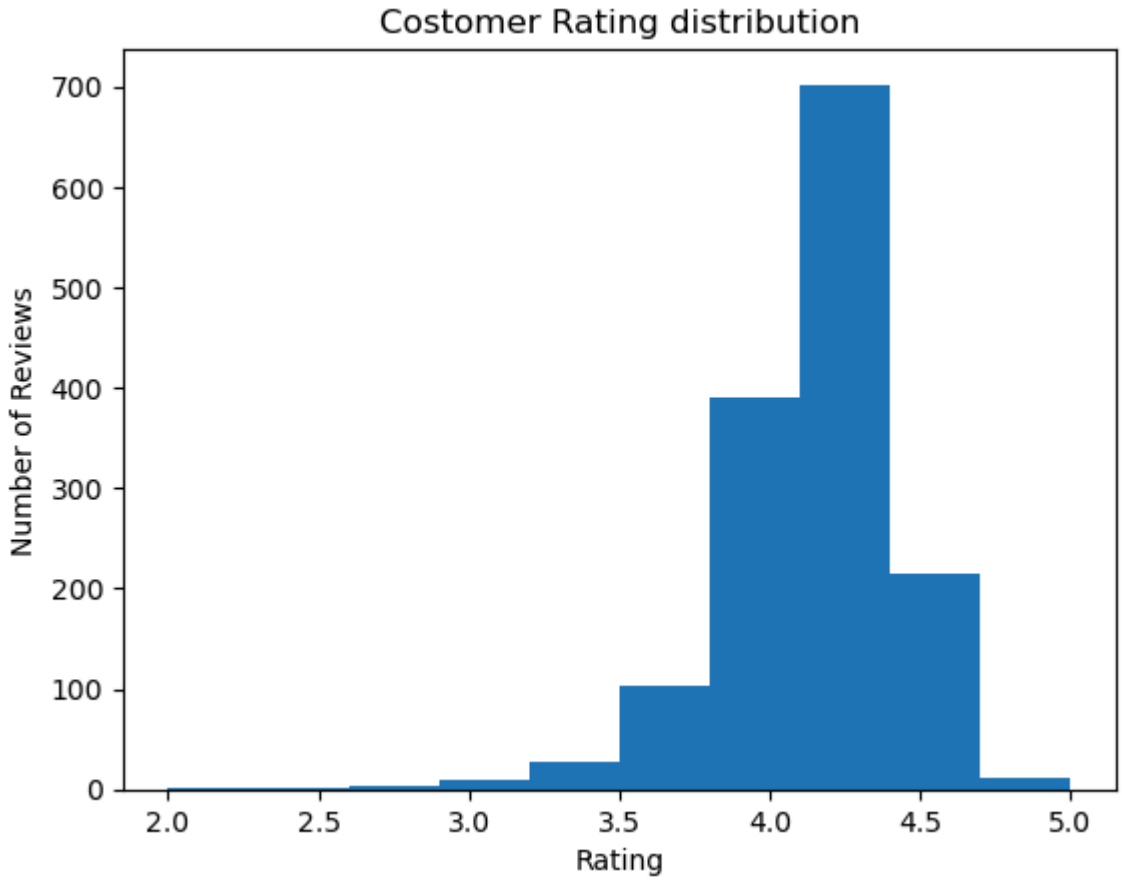
In [114…
```
plt.bar(range(len(mainCategoryCount)),mainCategoryCount.values)
plt.ylabel('Number of Products')
plt.xticks(range(len(mainCategoryCount)),'')
plt.show()
```




-Observation

1. Top 3 main categories are: Electronics, Computers&Accessories,Home&Kitchen

In [117…
```
# analyze distribution of customers rating

plt.hist(df['rating'])
plt.xlabel('Rating')
plt.ylabel('Number of Reviews')
plt.title('Costomer Rating distribution')
plt.show()
```




In [119…
```
bins = [0,1,2,3,4,5]
df['cluster'] = pd.cut(df['rating'], bins=bins, labels = ['0-1','1-2','2-3','3-4
```

In [121…
```
df['cluster'].value_counts().sort_values()
```

Out[121…
```
cluster
0-1      0
1-2      1
2-3      9
3-4    526
4-5    929
Name: count, dtype: int64
```

-Observation:

1. Majority Ratings fall withing 3-4 and 4-5 range.
2. There is a slight increase in ratings 2-3 compered to 1-2 and 0-1 ranges.
3. Overall customers are satisfied with the products

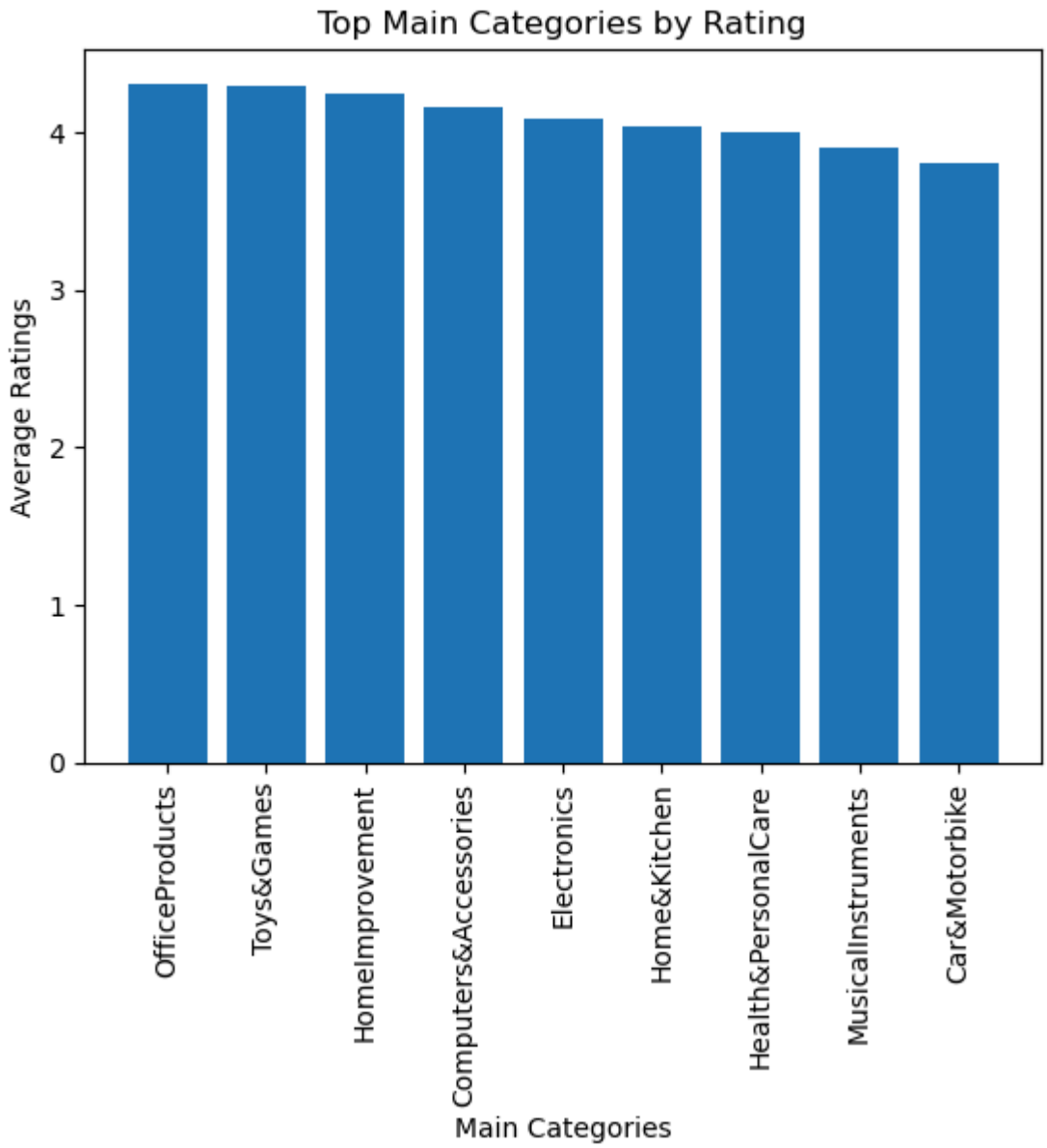
In [124…
```
topCategories_byRating = df.groupby(['main_category'])['rating'].mean().sort_val
topCategories_byRating
```

Out[124…

| | main_category | rating |
|---|---|---|
| **0** | OfficeProducts | 4.309677 |
| **1** | Toys&Games | 4.300000 |
| **2** | HomeImprovement | 4.250000 |
| **3** | Computers&Accessories | 4.154967 |
| **4** | Electronics | 4.081749 |
| **5** | Home&Kitchen | 4.039509 |
| **6** | Health&PersonalCare | 4.000000 |
| **7** | MusicalInstruments | 3.900000 |
| **8** | Car&Motorbike | 3.800000 |

In [126…

```
plt.bar(topCategories_byRating['main_category'],topCategories_byRating['rating']
plt.xticks(rotation=90)
plt.xlabel('Main Categories')
plt.ylabel('Average Ratings')
plt.title("Top Main Categories by Rating")
plt.show()
```




In [128…
```
#show the table
ranking = df.groupby(['main_category'])['rating'].mean().sort_values(ascending=F
ranking[:5]
```

Out[128…

| | main_category | rating |
|---|---|---|
| **0** | OfficeProducts | 4.309677 |
| **1** | Toys&Games | 4.300000 |
| **2** | HomeImprovement | 4.250000 |
| **3** | Computers&Accessories | 4.154967 |
| **4** | Electronics | 4.081749 |

-Observation

1. top reted category is Office Products
2. top 3 Rated Categories: Office Products, Toys& Games, HomeImprovement
3. Top 3 Highest Rated Categories with mean ratings is above 4

In [131…
```
# Top Sub Categories
topRated_subCategories = df.groupby(['sub_category'])['rating'].mean().sort_valu
```

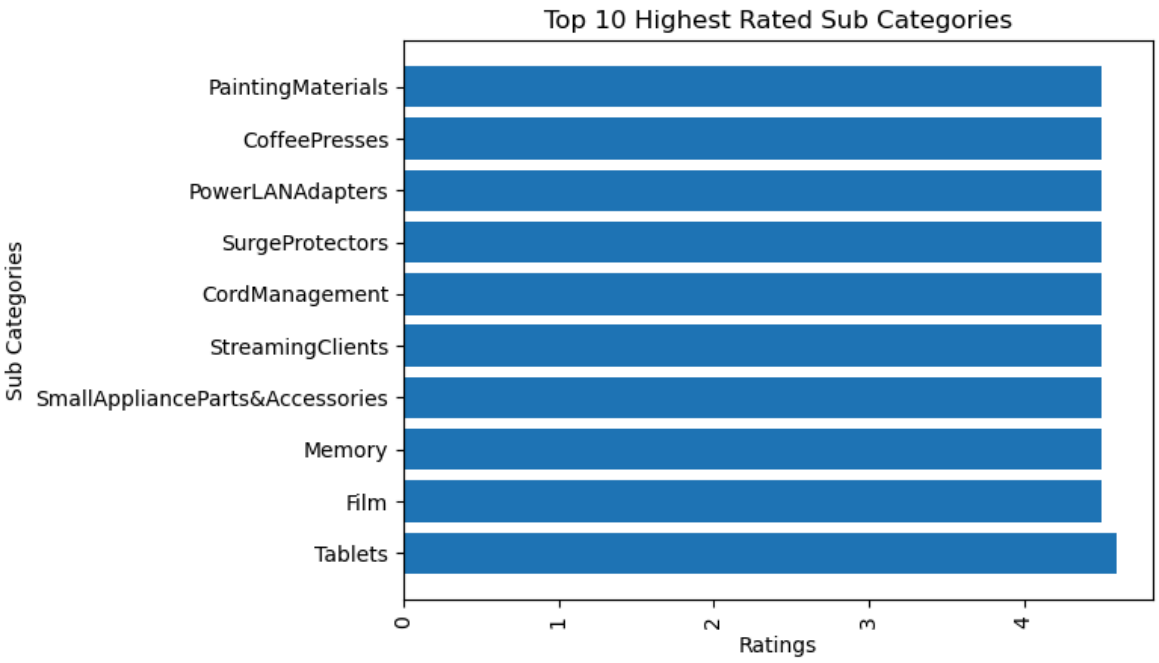
In [133…
```
topRated_10subCategories= topRated_subCategories.head(10)
```

In [135…
```
topRated_10subCategories
```

Out[135…

| | sub_category | rating |
|---|---|---|
| **0** | Tablets | 4.6 |
| **1** | Film | 4.5 |
| **2** | Memory | 4.5 |
| **3** | SmallApplianceParts&Accessories | 4.5 |
| **4** | StreamingClients | 4.5 |
| **5** | CordManagement | 4.5 |
| **6** | SurgeProtectors | 4.5 |
| **7** | PowerLANAdapters | 4.5 |
| **8** | CoffeePresses | 4.5 |
| **9** | PaintingMaterials | 4.5 |

In [137…
```
#plot sub categories
plt.barh(topRated_10subCategories['sub_category'], topRated_10subCategories['rat
plt.xticks(rotation=90)
plt.ylabel('Sub Categories')
plt.xlabel('Ratings')
plt.title('Top 10 Highest Rated Sub Categories')
plt.show()
```




In [139…
```
topRated_subCategories.tail()
```

Out[139...

| | sub_category | rating |
|---|---|---|
| **202** | InkjetPrinters | 3.6 |
| **203** | PCHeadsets | 3.5 |
| **204** | 3DGlasses | 3.5 |
| **205** | DustCovers | 3.4 |
| **206** | ElectricGrinders | 3.3 |

Observation:

1. Highest Rated Sub Categories are : Tablets, Film & Memory
2. Lowest are Electric Grinders, Dust Covers, 3dGlasses

In [142...
```
df.columns
```

Out[142...
```
Index(['product_id', 'product_name', 'category', 'discounted_price',
       'actual_price', 'discount_percentage', 'rating', 'rating_count',
       'about_product', 'user_id', 'user_name', 'review_id', 'review_title',
       'review_content', 'img_link', 'product_link', 'main_category',
       'sub_category', 'rating_weight', 'cluster'],
      dtype='object')
```

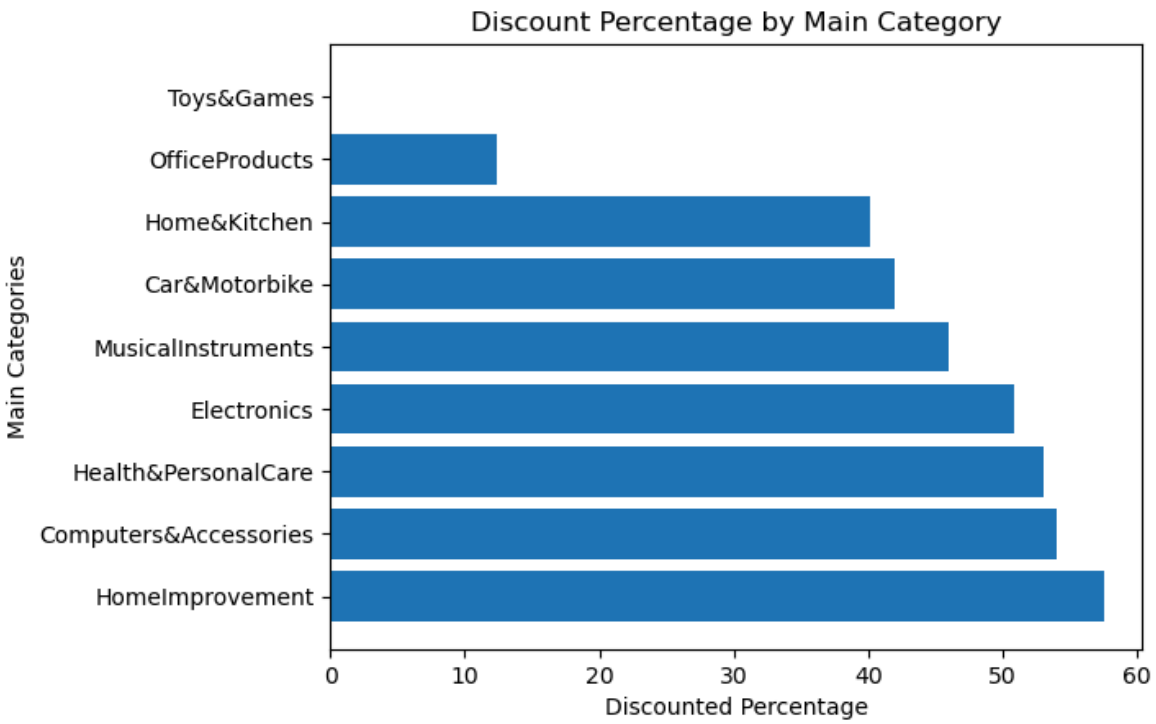
In [144...
```
avg_discount_percentage_MCategory = df.groupby('main_category')['discount_percen

avg_discount_percentage_MCategory.reset_index()
```

Out[144...

| | main_category | discount_percentage |
|---|---|---|
| **0** | HomeImprovement | 57.500000 |
| **1** | Computers&Accessories | 54.024283 |
| **2** | Health&PersonalCare | 53.000000 |
| **3** | Electronics | 50.828897 |
| **4** | MusicalInstruments | 46.000000 |
| **5** | Car&Motorbike | 42.000000 |
| **6** | Home&Kitchen | 40.120536 |
| **7** | OfficeProducts | 12.354839 |
| **8** | Toys&Games | 0.000000 |

In [146...
```
plt.barh(avg_discount_percentage_MCategory.index, avg_discount_percentage_MCateg
plt.xlabel('Discounted Percentage')
plt.ylabel('Main Categories')
plt.title("Discount Percentage by Main Category")
```

Out[146...
```
Text(0.5, 1.0, 'Discount Percentage by Main Category')
```




Observations:

1. Toys&Games has 0 discount, it means it has high demand
2. Home Improvement has maximum discount of 57%
3. Computer & Accessories, Health & Personal Care, & Electronics has almost same discount percentage

In [150…

```
# SUb category
top20_avg_discount_percentage_SCategory=df.groupby('sub_category')['discount_per
top20_avg_discount_percentage_SCategory.reset_index()
```

Out[150…

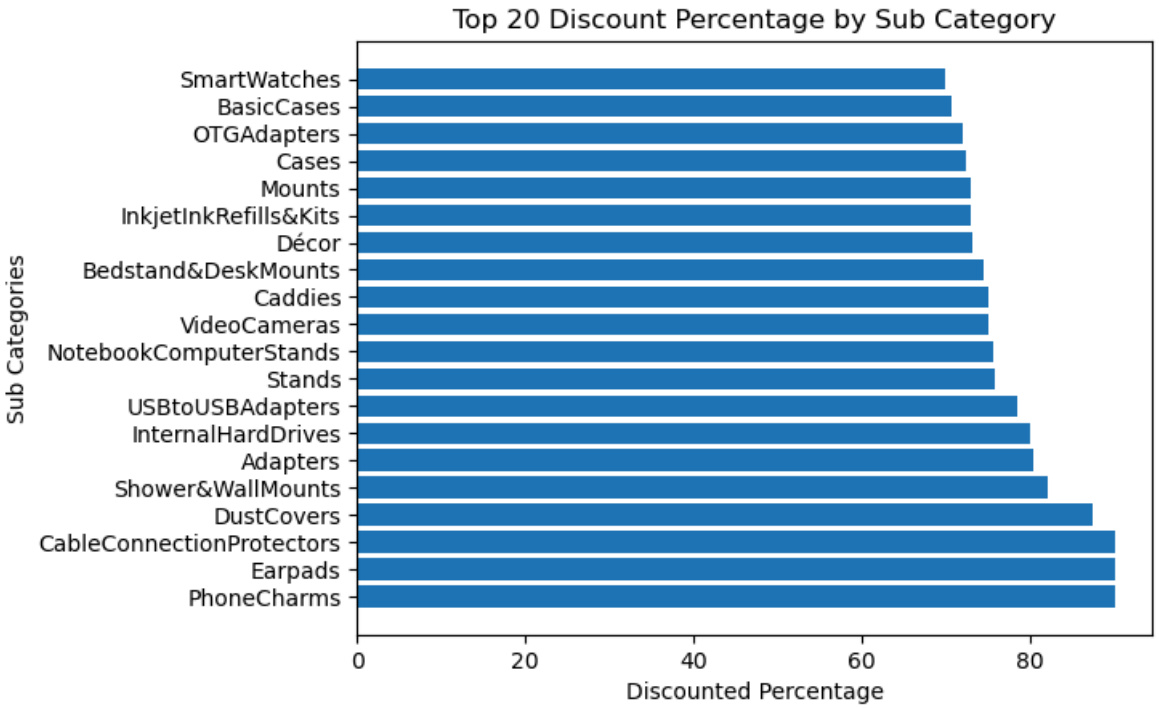
| | sub_category | discount_percentage |
|---|---|---|
| 0 | PhoneCharms | 90.000000 |
| 1 | Earpads | 90.000000 |
| 2 | CableConnectionProtectors | 90.000000 |
| 3 | DustCovers | 87.500000 |
| 4 | Shower&WallMounts | 82.000000 |
| 5 | Adapters | 80.333333 |
| 6 | InternalHardDrives | 80.000000 |
| 7 | USBtoUSBAdapters | 78.500000 |
| 8 | Stands | 75.818182 |
| 9 | NotebookComputerStands | 75.666667 |
| 10 | VideoCameras | 75.000000 |
| 11 | Caddies | 75.000000 |
| 12 | Bedstand&DeskMounts | 74.500000 |
| 13 | Décor | 73.200000 |
| 14 | InkjetInkRefills&Kits | 73.000000 |
| 15 | Mounts | 73.000000 |
| 16 | Cases | 72.333333 |
| 17 | OTGAdapters | 72.000000 |
| 18 | BasicCases | 70.750000 |
| 19 | SmartWatches | 69.815789 |

In [152…

```
#plot
plt.barh(top20_avg_discount_percentage_SCategory.index, top20_avg_discount_perce
plt.xlabel('Discounted Percentage')
plt.ylabel('Sub Categories')
plt.title("Top 20 Discount Percentage by Sub Category")
```

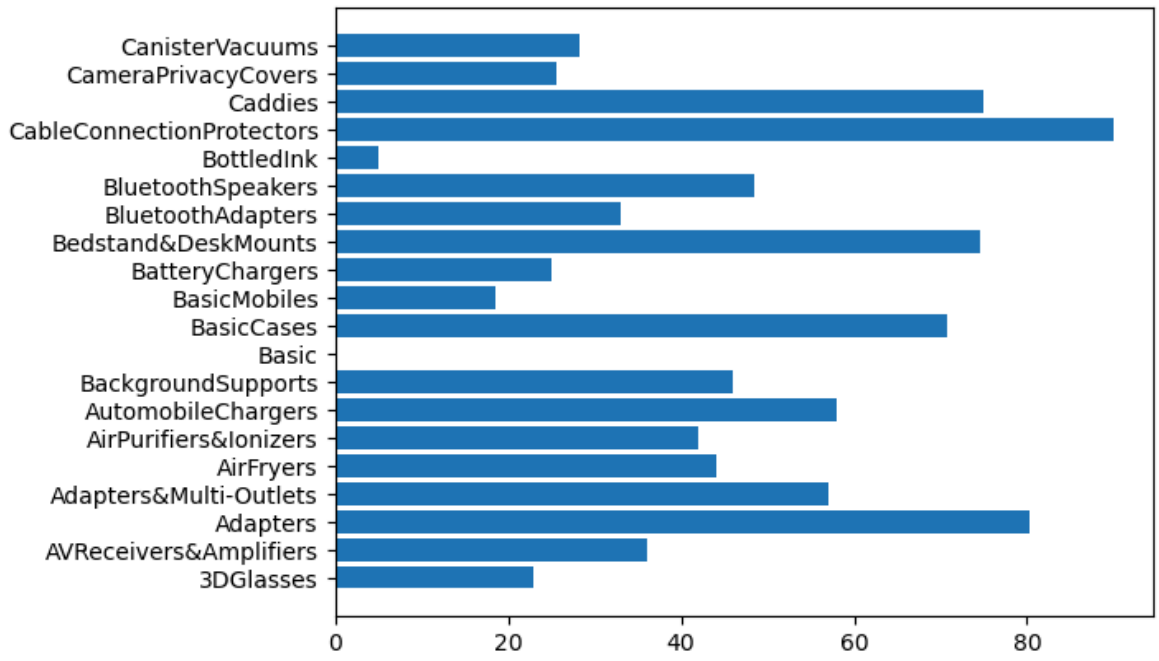
Out[152…

```
Text(0.5, 1.0, 'Top 20 Discount Percentage by Sub Category')
```

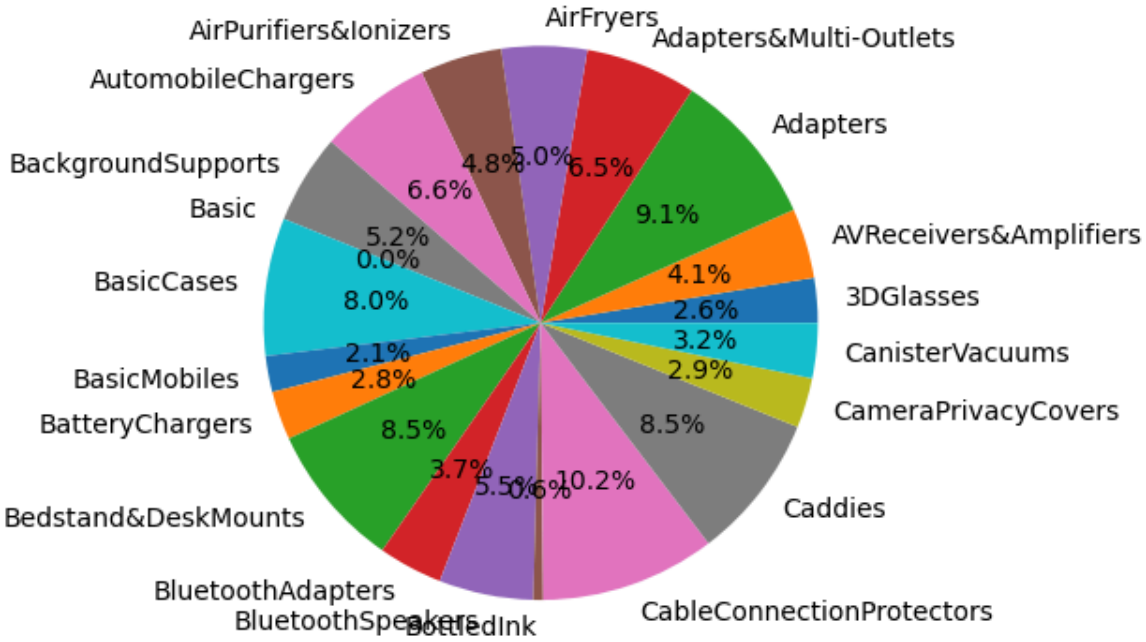



In [154…
```
top20SubCategories= df.groupby('sub_category')['discount_percentage'].mean().hea
plt.barh(top20SubCategories.index, top20SubCategories.values)
```

Out[154… <BarContainer object of 20 artists>

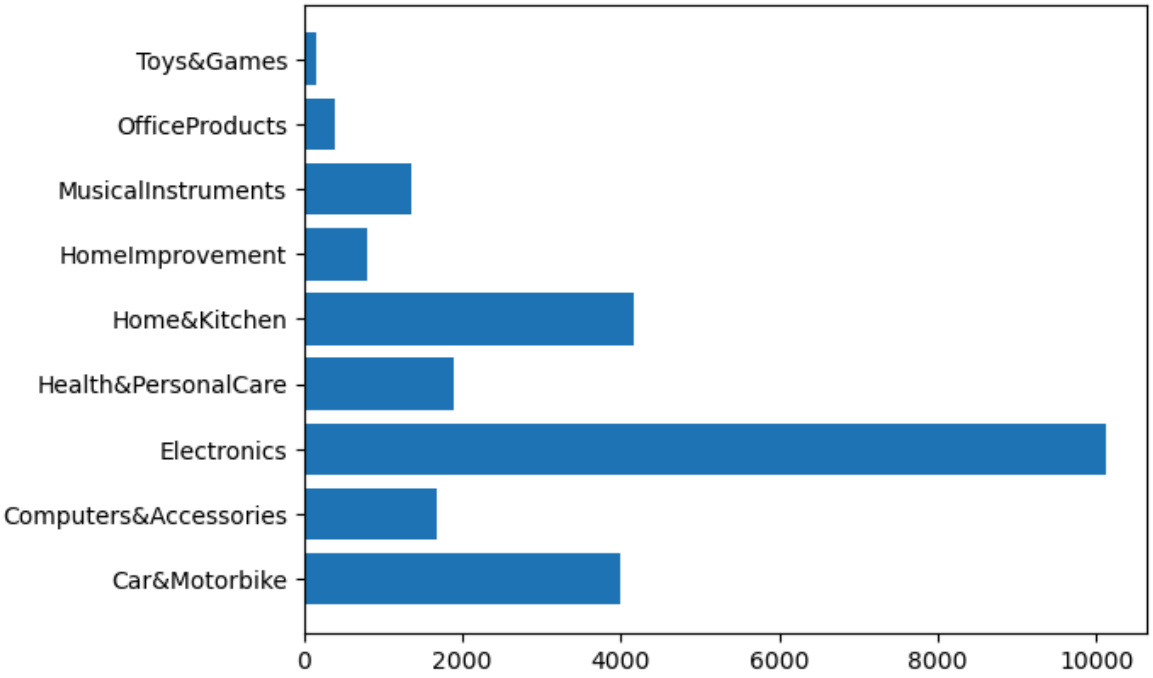



In [156…
```
plt.pie( top20SubCategories.values, labels = top20SubCategories.index,autopct='%
plt.show()
```

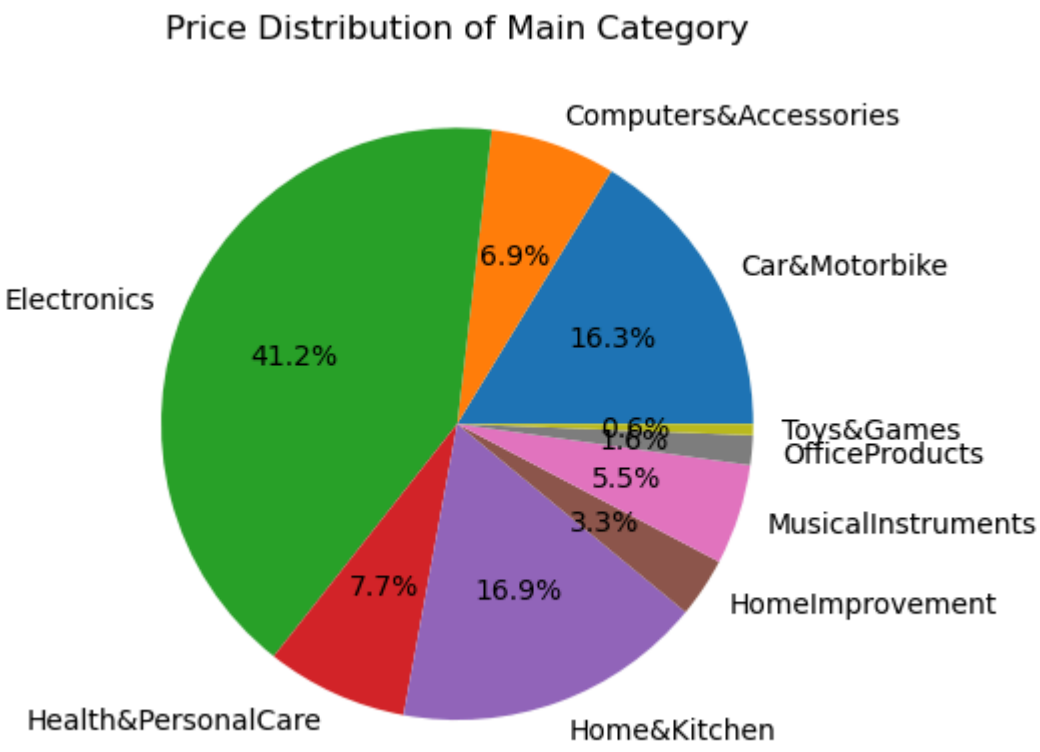



In [158…
```
mainCategory_actualPrice = df.groupby('main_category')['actual_price'].mean()
plt.barh(mainCategory_actualPrice.index, mainCategory_actualPrice.values)
```

Out[158…
```
<BarContainer object of 9 artists>
```




In [160…
```
plt.pie(mainCategory_actualPrice.values, labels=mainCategory_actualPrice.index,
plt.title('Price Distribution of Main Category')
plt.show()
```




# Analysis Reviews

In [163…
```
df['review_content'][0]
```

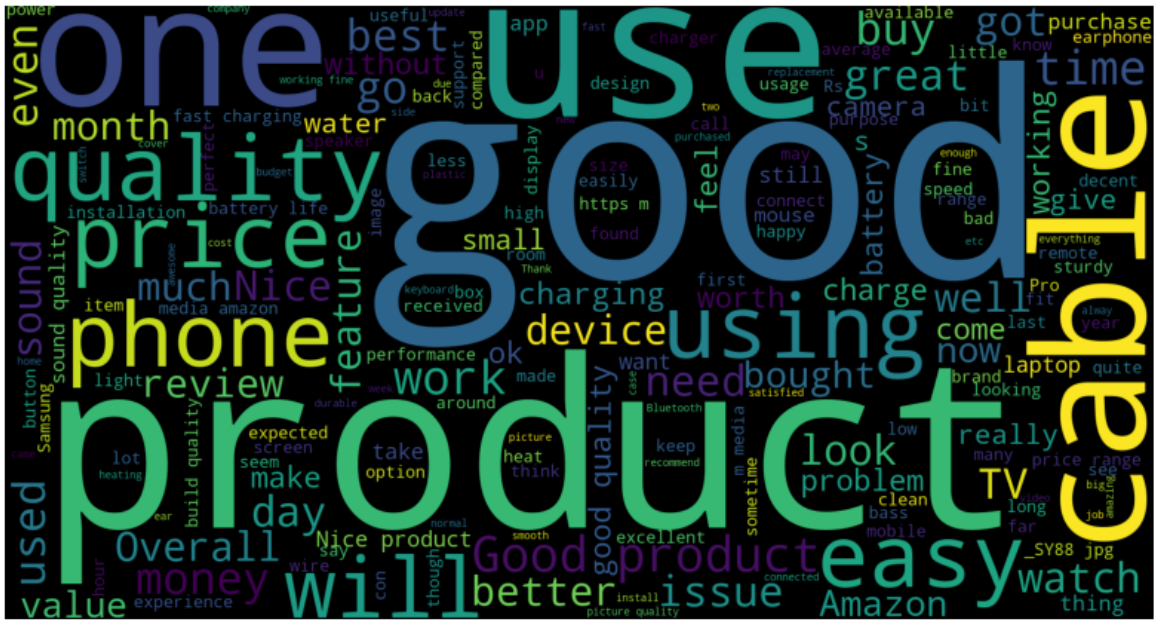
Out[163…
```
'Looks durable Charging is fine tooNo complains,Charging is really fast, good p
roduct.,Till now satisfied with the quality.,This is a good product . The charg
ing speed is slower than the original iPhone cable,Good quality, would recommen
d,https://m.media-amazon.com/images/W/WEBP_402378-T1/images/I/81---F1ZgHL._SY8
8.jpg,Product had worked well till date and was having no issue.Cable is also s
turdy enough...Have asked for replacement and company is doing the same...,Valu
e for money'
```

In [165…
```
reviewContent = ' '.join(df['review_content'].dropna().values)
# reviewContent
```

In [167…
```
# pip install wordcloud
from wordcloud import WordCloud

review_wordCloud = WordCloud(width=1500, height=800).generate(reviewContent)
```

In [169…
```
plt.figure(figsize=(10,10))
plt.imshow(review_wordCloud)
plt.axis("off")
```

Out[169…
```
(-0.5, 1499.5, 799.5, -0.5)
```




In [171…
```
rating4_df = df[df['rating']>4]
rating4_df.head()
```

Out[171…

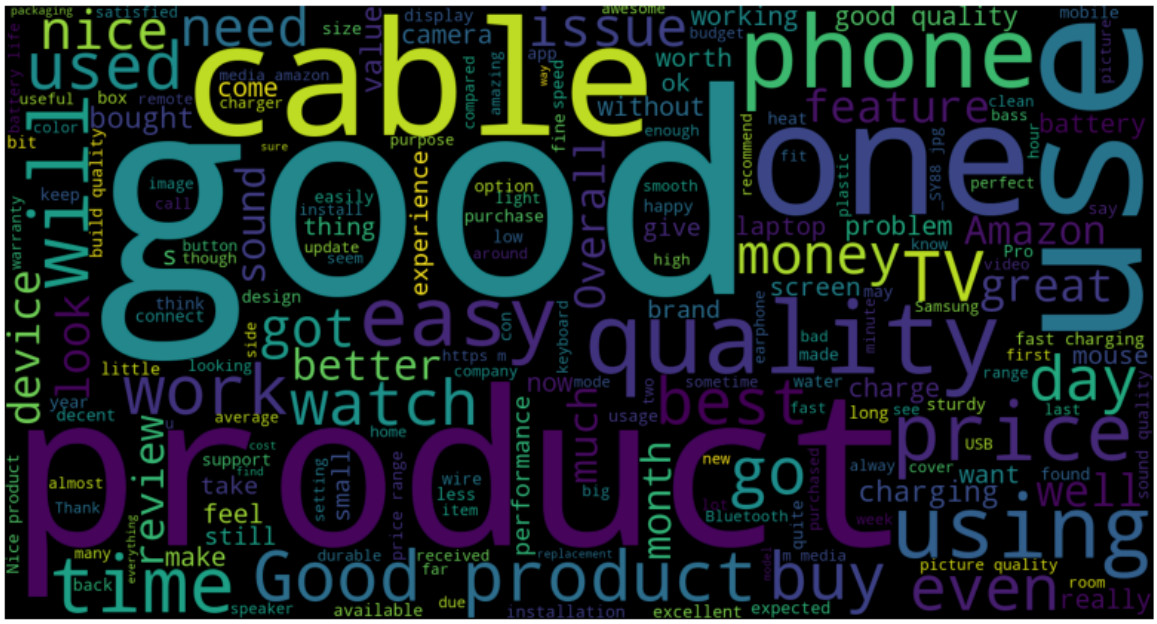
| | product_id | product_name | category | discour |
|---|---|---|---|---|
| **0** | B07JW9H4J1 | Wayona Nylon Braided USB to Lightning Fast Cha... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **3** | B08HDJ86NZ | boAt Deuce USB 300 2 in 1 Type-C & Micro USB S... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **4** | B08CF3B7N1 | Portronics Konnect L 1.2M Fast Charging 3A 8 P... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **6** | B08WRWPM22 | boAt Micro USB 55 Tangle-free, Sturdy Micro US... | Computers&Accessories\|Accessories&Peripherals\|... | |
| **7** | B08DDRGWTJ | MI Usb Type-C Cable Smartphone (Black) | Computers&Accessories\|Accessories&Peripherals\|... | |

In [173…
```
rating4_review =' '.join(rating4_df['review_content'].dropna().values)
```

In [175…
```
rating4_review_wordCloud = WordCloud(width=1500, height=800).generate(rating4_re
```

In [177…
```
plt.figure(figsize=(10,10))
plt.imshow(rating4_review_wordCloud)
plt.axis("off")
```

Out[177… (-0.5, 1499.5, 799.5, -0.5)




In [179…
```
low_rating_df = df[df['rating']<3]
low_rating_review =' '.join(low_rating_df['review_content'].dropna().values)
low_rating_review_wordCloud = WordCloud(width=1500, height=800).generate(low_rat
plt.figure(figsize=(10,10))
plt.imshow(low_rating_review_wordCloud)
plt.axis("off")
```

Out[179… (-0.5, 1499.5, 799.5, -0.5)


# Correlation between numerical Feature

In [182…
```
numerical_df = df.select_dtypes(include='float64')
numerical_df.head()
```

Out[182...

| | discounted_price | actual_price | discount_percentage | rating | rating_count | rating_weig |
|---|---|---|---|---|---|---|
| **0** | 399.0 | 1099.0 | 64.0 | 4.2 | 24269.0 | 10192 |
| **1** | 199.0 | 349.0 | 43.0 | 4.0 | 43994.0 | 17597 |
| **2** | 199.0 | 1899.0 | 90.0 | 3.9 | 7928.0 | 3091 |
| **3** | 329.0 | 699.0 | 53.0 | 4.2 | 94363.0 | 39632 |
| **4** | 154.0 | 399.0 | 61.0 | 4.2 | 16905.0 | 7100 |

In [184...

```
correlation_matrix = numerical_df.corr()
correlation_matrix
```

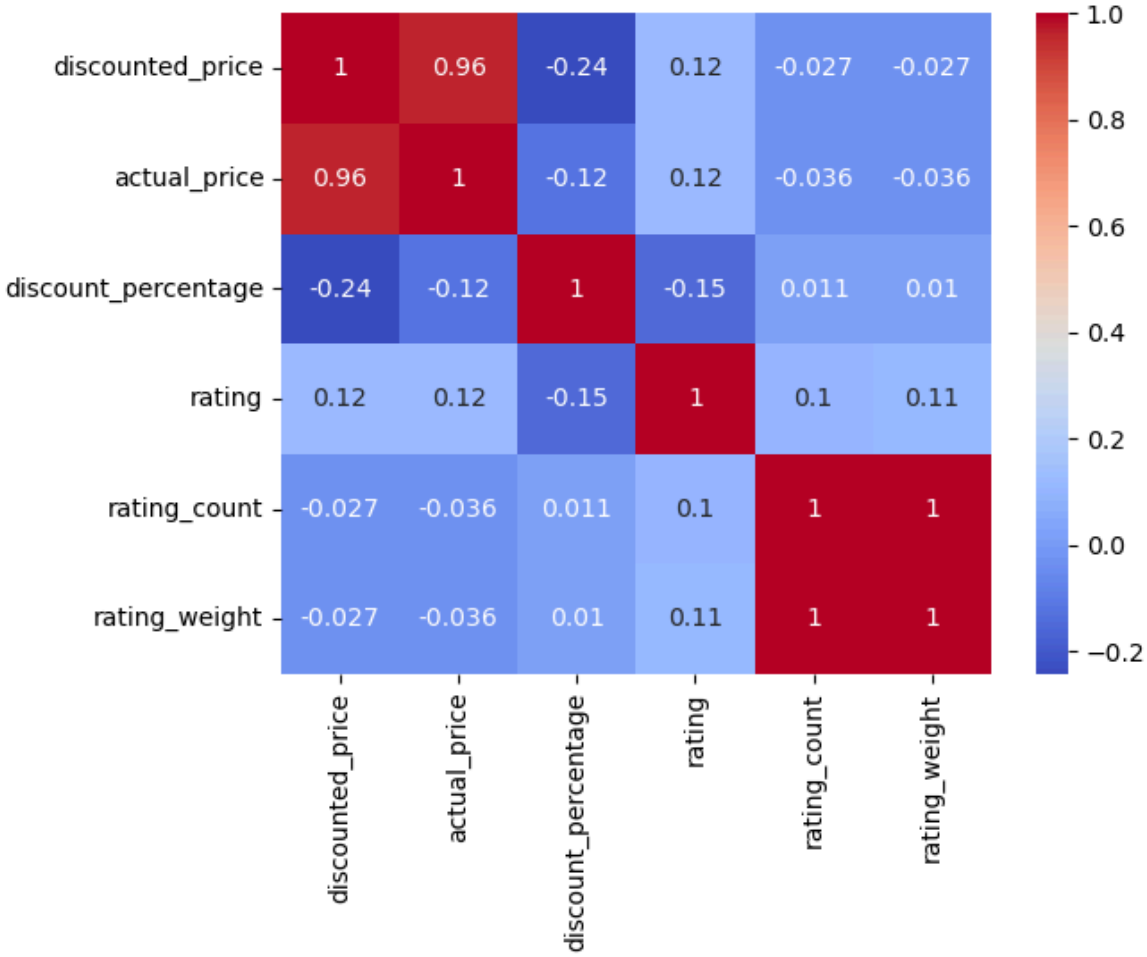
Out[184...

| | discounted_price | actual_price | discount_percentage | rating | rati |
|---|---|---|---|---|---|
| **discounted_price** | 1.000000 | 0.961915 | -0.242412 | 0.120371 | |
| **actual_price** | 0.961915 | 1.000000 | -0.118098 | 0.121843 | |
| **discount_percentage** | -0.242412 | -0.118098 | 1.000000 | -0.152996 | |
| **rating** | 0.120371 | 0.121843 | -0.152996 | 1.000000 | |
| **rating_count** | -0.027081 | -0.035959 | 0.011097 | 0.101949 | |
| **rating_weight** | -0.026815 | -0.035703 | 0.010497 | 0.114939 | |

In [186...

```
import seaborn as sns
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
```

Out[186... <Axes: >




# Recommendation System

In [189…

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1465 entries, 0 to 1464
Data columns (total 20 columns):
 #   Column               Non-Null Count  Dtype   
---  ------               --------------  -----   
 0   product_id           1465 non-null   object  
 1   product_name         1465 non-null   object  
 2   category             1465 non-null   object  
 3   discounted_price     1465 non-null   float64 
 4   actual_price         1465 non-null   float64 
 5   discount_percentage  1465 non-null   float64 
 6   rating               1465 non-null   float64 
 7   rating_count         1465 non-null   float64 
 8   about_product        1465 non-null   object  
 9   user_id              1465 non-null   object  
 10  user_name            1465 non-null   object  
 11  review_id            1465 non-null   object  
 12  review_title         1465 non-null   object  
 13  review_content       1465 non-null   object  
 14  img_link             1465 non-null   object  
 15  product_link         1465 non-null   object  
 16  main_category        1465 non-null   object  
 17  sub_category         1465 non-null   object  
 18  rating_weight        1465 non-null   float64 
 19  cluster              1465 non-null   category
dtypes: category(1), float64(6), object(13)
memory usage: 219.2+ KB
```

In [191…

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()


#fit transform training data
df['encoded_user_id'] = le.fit_transform(df['user_id'])
df.head(1)
```

Out[191…

| | product_id | product_name | category | discounte |
|---|---|---|---|---|
| **0** | B07JW9H4J1 | Wayona Nylon Braided USB to Lightning Fast Cha... | Computers&Accessories\|Accessories&Peripherals\|... | |

1 rows × 21 columns

In [193…

```
user_id_freq_table = pd.DataFrame({'UserID': df['encoded_user_id'].value_counts(
                                   'Frequency': df['encoded_user_id'].value_coun

user_id_freq_table
```

Out[193…

| | UserID | Frequency |
|---|---|---|
| **0** | 1048 | 10 |
| **1** | 623 | 8 |
| **2** | 674 | 8 |
| **3** | 254 | 7 |
| **4** | 88 | 7 |
| **…** | … | … |
| **1189** | 429 | 1 |
| **1190** | 506 | 1 |
| **1191** | 11 | 1 |
| **1192** | 900 | 1 |
| **1193** | 433 | 1 |

1194 rows × 2 columns

In [195…
```
le.inverse_transform([1048])
```

Out[195…
```
array(['AHIKJUDTVJ4T6DV6IUGFYZ5LXMPA,AE55KTFVNXYFD5FPYWP2OUPEYNPQ,AEBWA5I4QFCA3
P3OBEPMELBGN4GQ,AHMGAC6QM62UXNEOCZIHLHSXPP2Q,AFHROSCGIXUPV3FYQ7H5QOD46Q7Q,AEAMI
R3CMSA32IDEINSJKHRNANTA,AF355FTXYAKFH5NYPRTE7SL3WO3Q,AG5DWPD54QGSLWJ6QUFERLPNAX
4Q'],
      dtype=object)
```

TF IDF Explanation¶

1. "red pen, red cap"
2. "pink skirt, silk, party"
3. 'black shirt, casual, office'

TF Red = 2 (P1) = 2/4 =2 Silk = 1(P2) = 1/4 = 0.25 Black = 1

IDF silk = total products are 100 in which 5 products has *Silk in the description so IDF = log(100/5) = log(20) = 1.3

TFIDF(silk) = 0.25*1.3 = 0.325

In [198…
```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(stop_words='english')
```

In [200…
```
df['about_product'] = df['about_product'].fillna('')
df['about_product']
```

Out[200…
```
0       High Compatibility : Compatible With iPhone 12...
1       Compatible with all Type C enabled devices, be...
2       【 Fast Charger& Data Sync】-With built-in safet...
3       The boAt Deuce USB 300 2 in 1 cable is compati...
4       [CHARGE & SYNC FUNCTION]- This cable comes wit...
                              ...
1460    SUPREME QUALITY 90 GRAM 3 LAYER THIK PP SPUN F...
1461                         230 Volts, 400 watts, 1 Year
1462    International design and styling|Two heat sett...
1463    Fan sweep area: 230 MM ; Noise level: (40 - 45...
1464    Brand-Borosil, Specification â€“ 23V ~ 5Hz;1 W...
Name: about_product, Length: 1465, dtype: object
```

In [202…
```
tfidf_matrix = tfidf.fit_transform(df['about_product'])
tfidf_matrix
```

Out[202…
```
<Compressed Sparse Row sparse matrix of dtype 'float64'
        with 92662 stored elements and shape (1465, 8911)>
```

In [204…
```
user_product_details = df[df['encoded_user_id'] == 900]
user_product_details
```

Out[204…

| | product_id | product_name | category | discou |
|---|---|---|---|---|
| **716** | B08FB2LNSZ | JBL Tune 215BT, 16 Hrs Playtime with Quick Cha... | Electronics\|Headphones,Earbuds&Accessories\|Hea... | |

1 rows × 21 columns

Cosine Similiarty¶

1. P1 = "red pen, red cap"
2. p2 = "pink skirt, silk, party"
3. p3 = 'black shirt, casual, office'

TFIDF p1 = [0.23, 0.11, 0.45] p2 = [0.1, 0.4, 0.2]

1. 1: Means the vectors are pointing in the exact same direction. They are very similar.
2. 0: Means the vectors are perpendicular to each other. They are not similar at all.
3. -1: Means the vectors are pointing in opposite directions. They are completely dissimilar.

In [209…
```
indices = user_product_details.index.tolist()
indices
```

Out[209…
```
[716]
```

In [211…
```
from sklearn.metrics.pairwise import cosine_similarity

cosine_similairty_user = cosine_similarity(tfidf_matrix[indices], tfidf_matrix)
cosine_similairty_user
```

Out[211…
```
array([[0.03457362, 0.02342515, 0.04488096, ..., 0.        , 0.03494306,
        0.        ]])
```

In [213…
```
products = df.iloc[indices]['product_name']

indices = pd.Series(products.index, index=products)
indices
```

Out[213…
```
product_name
JBL Tune 215BT, 16 Hrs Playtime with Quick Charge, in Ear Bluetooth Wireless Ea
rphones with Mic, 12.5mm Premium Earbuds with Pure Bass, BT 5.0, Dual Pairing,
Type C & Voice Assistant Support (Black)    716
dtype: int64
```

In [215…
```
similiarity_scores = list(enumerate(cosine_similairty_user[-1]))
similiarity_scores=[(i, score) for (i,score) in similiarity_scores if i not in i
similiarity_scores= sorted(similiarity_scores, key=lambda x:x[1], reverse=True)
```

In [217…
```
#top 5 recommended producst
top_5_products = [i[0] for i in similiarity_scores[1:6]]
top_5_products
```

Out[217…
```
[883, 977, 785, 939, 664]
```

In [219…
```
#name of top 5 products
recommended_products = df.iloc[top_5_products]['product_name'].tolist()
```

In [221…
```
top5scores = [similiarity_scores[i][1] for i in range(5)]
top5scores
```

Out[221…
```
[0.9999999999999996,
 0.3299631371523861,
 0.31592300978836335,
 0.20870111277385178,
 0.2053676966074428]
```

In [223…
```
recommended_products
```

Out[223…
```
['Infinity (JBL Glide 510, 72 Hrs Playtime with Quick Charge, Wireless On Ear H
eadphone with Mic, Deep Bass, Dual Equalizer, Bluetooth 5.0 with Voice Assistan
t Support (Black)',
 'Sennheiser CX 80S in-Ear Wired Headphones with in-line One-Button Smart Remot
e with Microphone Black',
 'Noise Buds VS201 V2 in-Ear Truly Wireless Earbuds with Dual Equalizer | with
Mic | Total 14-Hour Playtime | Full Touch Control | IPX5 Water Resistance and B
luetooth v5.1 (Olive Green)',
 'Wecool Moonwalk M1 ENC True Wireless in Ear Earbuds with Mic, Titanium Driver
s for Rich Bass Experience, 40+ Hours Play Time, Type C Fast Charging, Low Late
ncy, BT 5.3, IPX5, Deep Bass (Black)',
 'boAt Rockerz 550 Over Ear Bluetooth Headphones with Upto 20 Hours Playback, 5
0MM Drivers, Soft Padded Ear Cushions and Physical Noise Isolation, Without Mic
(Black)']
```

In [225…
```
df.columns
```

Out[225…
```
Index(['product_id', 'product_name', 'category', 'discounted_price',
       'actual_price', 'discount_percentage', 'rating', 'rating_count',
       'about_product', 'user_id', 'user_name', 'review_id', 'review_title',
       'review_content', 'img_link', 'product_link', 'main_category',
       'sub_category', 'rating_weight', 'cluster', 'encoded_user_id'],
      dtype='object')
```

In [227…
```
#lets convert all in df
result_df = pd.DataFrame({'Encoded_ID': 900,
'Recommended Products': recommended_products,
'Recommendation Score': top5scores})
result_df
```

Out[227…

| | Encoded_ID | Recommended Products | Recommendation Score |
|---|---|---|---|
| **0** | 900 | Infinity (JBL Glide 510, 72 Hrs Playtime with ... | 1.000000 |
| **1** | 900 | Sennheiser CX 80S in-Ear Wired Headphones with... | 0.329963 |
| **2** | 900 | Noise Buds VS201 V2 in-Ear Truly Wireless Earb... | 0.315923 |
| **3** | 900 | Wecool Moonwalk M1 ENC True Wireless in Ear Ea... | 0.208701 |
| **4** | 900 | boAt Rockerz 550 Over Ear Bluetooth Headphones... | 0.205368 |

In [229…
```
# lets build a fucntion for recommedation system
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity


tfidf = TfidfVectorizer(stop_words='english')

def recommendation_system(df, encoded_user_id):
    tfidf = TfidfVectorizer(stop_words='english')
    df['about_product'] = df['about_product'].fillna('')
    tfidf_matrix = tfidf.fit_transform(df['about_product'])
    user_product_details = df[df['encoded_user_id'] == encoded_user_id]

    indices = user_product_details.index.tolist()
    if indices:
        cosine_similairty_user = cosine_similarity(tfidf_matrix[indices], tfidf_
        products = df.iloc[indices]['product_name']

        indices = pd.Series(products.index, index=products)
        similiarity_scores = list(enumerate(cosine_similairty_user[-1]))
        similiarity_scores=[(i, score) for (i,score) in similiarity_scores if i
        similiarity_scores= sorted(similiarity_scores, key=lambda x:x[1], revers
        top_5_products = [i[0] for i in similiarity_scores[1:6]]
        #name of top 5 products
        recommended_products = df.iloc[top_5_products]['product_name'].tolist()

        top5scores = [similiarity_scores[i][1] for i in range(5)]
        result_df = pd.DataFrame({'Encoded_ID': encoded_user_id,
        'Recommended Products': recommended_products,
        'Recommendation Score': top5scores})
```

```
        return result_df
    return "Id Incorrect"
```

In [231…

```
recommendation_system(df, 100)
```

Out[231…

| | Encoded_ID | Recommended Products | Recommendation Score |
|---|---|---|---|
| **0** | 100 | Classmate Soft Cover 6 Subject Spiral Binding ... | 1.000000 |
| **1** | 100 | Classmate Pulse Spiral Notebook - 240 mm x 180... | 0.793460 |
| **2** | 100 | Classmate Drawing Book - Unruled, 40 Pages, 21... | 0.786062 |
| **3** | 100 | Classmate Soft Cover 6 Subject Spiral Binding ... | 0.768637 |
| **4** | 100 | Classmate 2100117 Soft Cover 6 Subject Spiral ... | 0.762039 |

# Completed