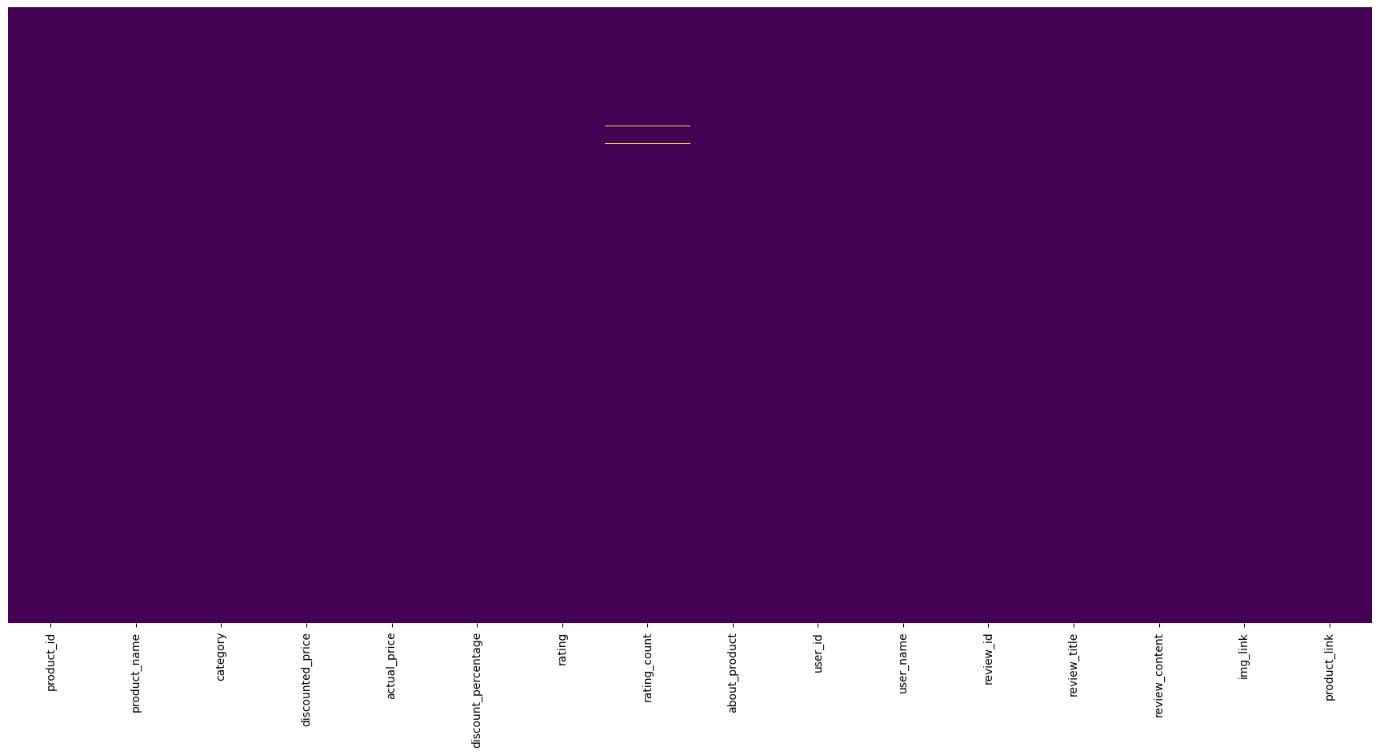
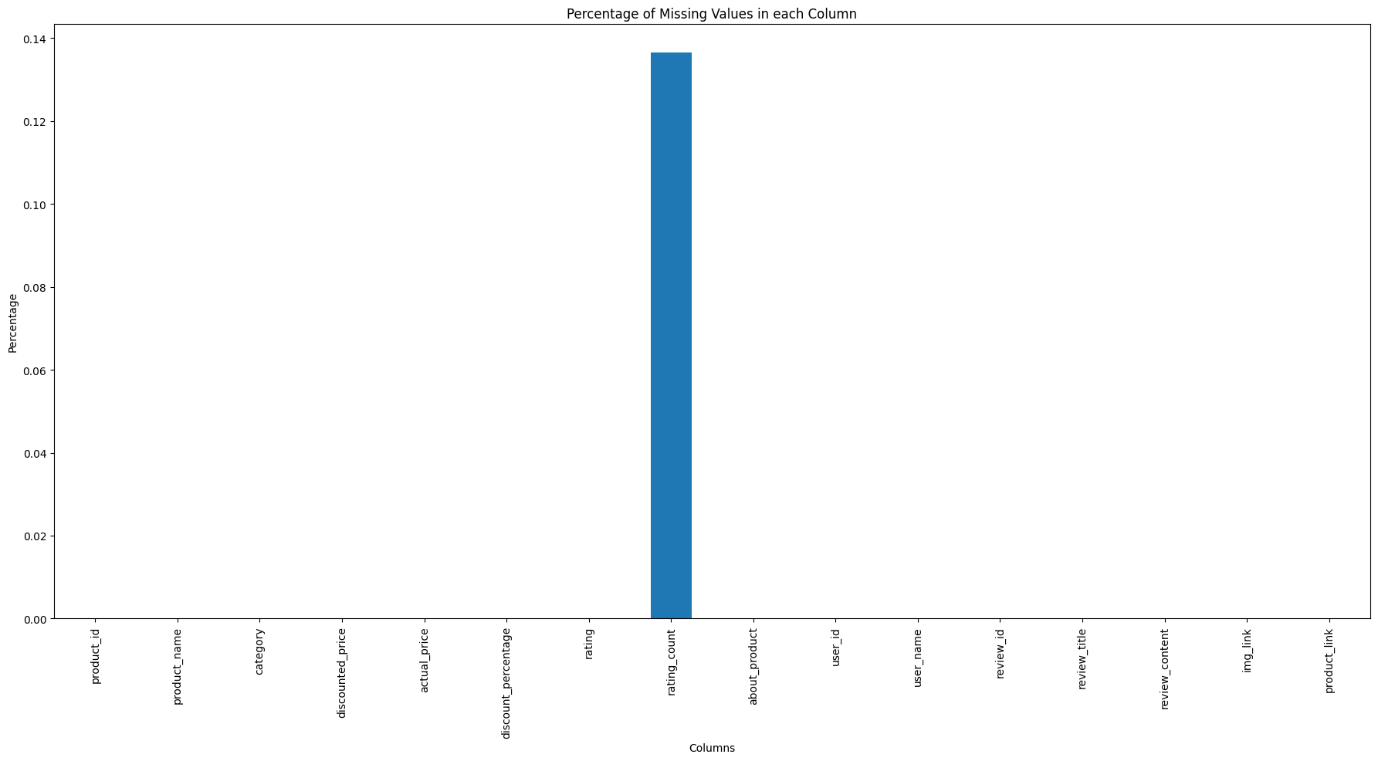
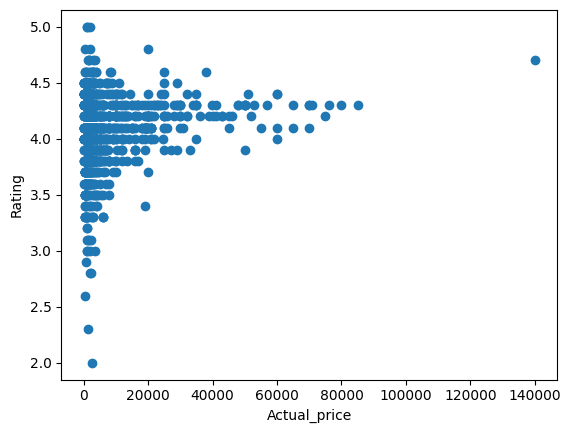
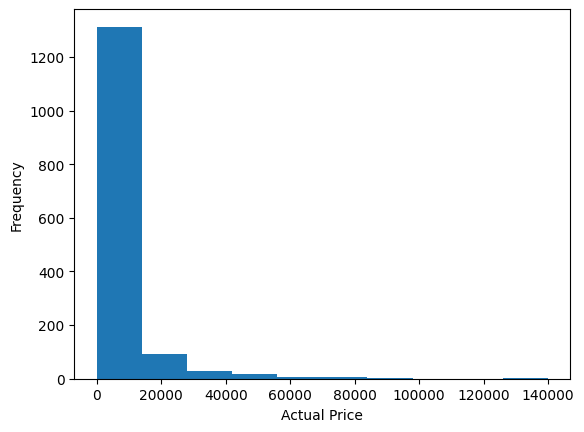
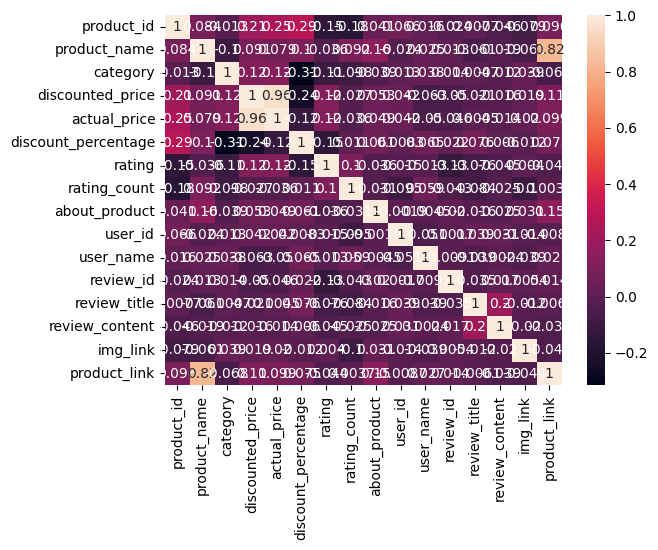
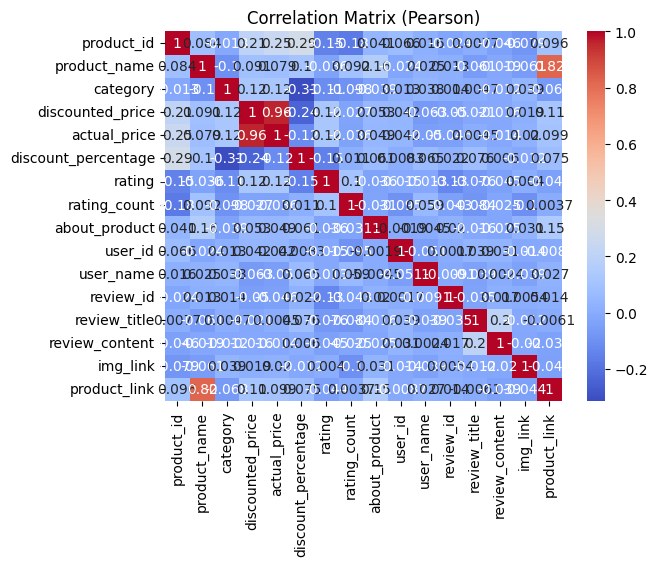
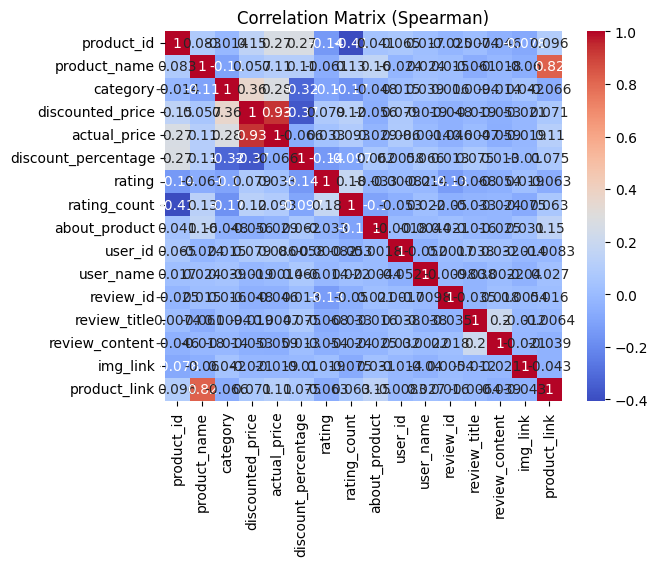
* **Amazon Sales Dataset EDA**
* 
* ***Project: Amazon Sales Dataset***
* **Contact Info**
* ***Click on link below to contact/follow/correct me:***
* ***Email:*** komaldeshmukh7385@gmail.com
* About Data
* ***Title: Amazon Sales Dataset***
* **Dataset Columns Names**
* ***This dataset is having the data of 1K+ Amazon Product's Ratings and Reviews as per their details listed on the official website of Amazon***
* ***Features***
* ***product\_id - Product ID***
* ***product\_name - Name of the Product***
* ***category - Category of the Product***
* ***discounted\_price - Discounted Price of the Product***
* ***actual\_price - Actual Price of the Product***
* ***discount\_percentage - Percentage of Discount for the Product***
* ***rating - Rating of the Product***
* ***rating\_count - Number of people who voted for the Amazon rating***
* ***about\_product - Description about the Product***
* ***user\_id - ID of the user who wrote review for the Product***
* ***user\_name - Name of the user who wrote review for the Product***
* ***review\_id - ID of the user review***
* ***review\_title - Short review***
* ***review\_content - Long review***
* ***img\_link - Image Link of the Product***
* ***product\_link - Official Website Link of the Product***
* **Metadata**
* **Source:** ***This dataset is scraped from the official website of Amazon***\
* **Collection Methodology:** ***This dataset is scraped through BeautifulSoup and WebDriver using Python***
* **License:** ***CC BY-NC-SA 4.0***
* **Task**
* ***Exploring the Amazon Sales Dataset involves a step-by-step process. First, we clean and prepare the data to ensure it's accurate and consistent. Then, we summarize the data using descriptive statistics like averages and ranges. Next, we visualize the data with charts and graphs to see patterns and relationships. We detect outliers, which are unusual data points, and test our assumptions about the data. We divide the data into groups for better understanding and finally, we summarize our findings.***
* **Objectives**
* ***The primary objective of analyzing the Amazon Sales Dataset is delve into product categories, prices, ratings, and sales patterns to identify characteristics that resonate with consumers and propel them to purchase.***
* ***Delve into product categories, prices, ratings, and sales patterns to identify characteristics that resonate with consumers and propel them to purchase.***
* ***Translate insights into actionable recommendations that optimize product development, inform marketing strategies, and boost your competitive edge.***
* ***Equip businesses with the knowledge to create products that cater to evolving consumer needs and desires.***
* ***Craft communication strategies that resonate with specific demographics and maximize engagement.***
* ***Facilitate a marketplace where products find their perfect match in the hearts of consumers.***
* **Kernel Version Used**
* ***Python 3.12.0***
* Import Libraries
* ***We will use the following libraries***
* ***Pandas: Data manipulation and analysis***
* ***Numpy: Numerical operations and calculations***
* ***Matplotlib: Data visualization and plotting***
* ***Seaborn: Enhanced data visualization and statistical graphics***
* ***Scipy: Scientific computing and advanced mathematical operations***
* import pandas as pd
* import numpy as np
* import matplotlib.pyplot as plt
* import seaborn as sns
* import scipy as sp
* *# this is for jupyter notebook to show the plot in the notebook itself instead of opening a new window*
* %matplotlib inline
* Data Loading and Exploration | Cleaning
* **Load a CSV file then creating a dataframe**
* df = pd.read\_csv("/kaggle/input/amazon-sales-dataset/amazon.csv")
* **Set the option to show maximum columns**
* pd.set\_option('display.max\_columns', None)
* **Get a sneak peek of data**
* ***The purpose of a sneak peek is to get a quick overview of the data and identify any potential problems or areas of interest.***
* *# Let's have a look on top 5 rows of the data*
* df.head(5)
* **Let's see the column names**
* df.columns
* 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')
* **Let's have a look on the shape of the dataset**
* print(f"The Number of Rows are **{**df.shape[0]**}**, and columns are **{**df.shape[1]**}**.")
* The Number of Rows are 1465, and columns are 16.
* **Let's have a look on the columns and their data types using detailed info function**
* 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: 183.2+ KB
* df.isnull().sum()
* 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 Set 1
* ***There are 1465 rows and 16 columns in the dataset.***
* ***The data type of all columns is object.***
* ***The columns in the datasets are:***
  + ***'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'***
* ***There are a few missing values in the dataset, which we will read in detail and deal with later on in the notebook.***
* **Changing Data Types of Columns from object to float**
* *# Changing the data type of discounted price and actual price*
* df['discounted\_price'] = df['discounted\_price'].str.replace("₹",'')
* df['discounted\_price'] = df['discounted\_price'].str.replace(",",'')
* df['discounted\_price'] = df['discounted\_price'].astype('float64')
* df['actual\_price'] = df['actual\_price'].str.replace("₹",'')
* df['actual\_price'] = df['actual\_price'].str.replace(",",'')
* df['actual\_price'] = df['actual\_price'].astype('float64')
* *# Changing Datatype and values in Discount Percentage*
* df['discount\_percentage'] = df['discount\_percentage'].str.replace('%','').astype('float64')
* df['discount\_percentage'] = df['discount\_percentage'] / 100
* *# Finding unusual string in rating column*
* df['rating'].value\_counts()
* 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
  + 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
* *# Check the strange row*
* df.query('rating == "|"')
* ***I got this product rating on Amazon by searching the provided product\_id on their official website (amazon.in)***
* ***The rating is 3.9. So, I am going to give the item rating a 3.9 as well.***
* *# Changing Rating Columns Data Type*
* df['rating'] = df['rating'].str.replace('|', '3.9').astype('float64')
* *# Changing 'rating\_count' Column Data Type*
* df['rating\_count'] = df['rating\_count'].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 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: 183.2+ KB
* Descriptive Statistics
* ***Descriptive statistics are a collection of quantitative measures that summarize and describe the main characteristics of a dataset.***
* df.describe()

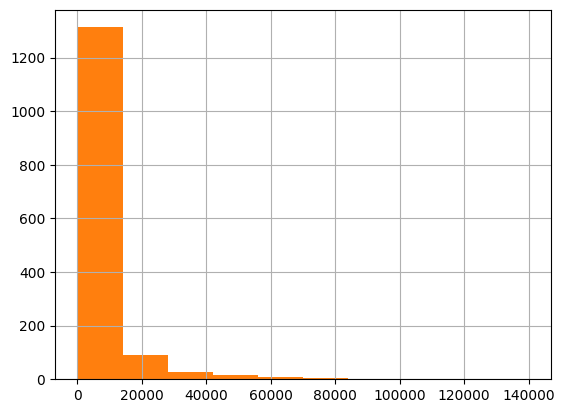
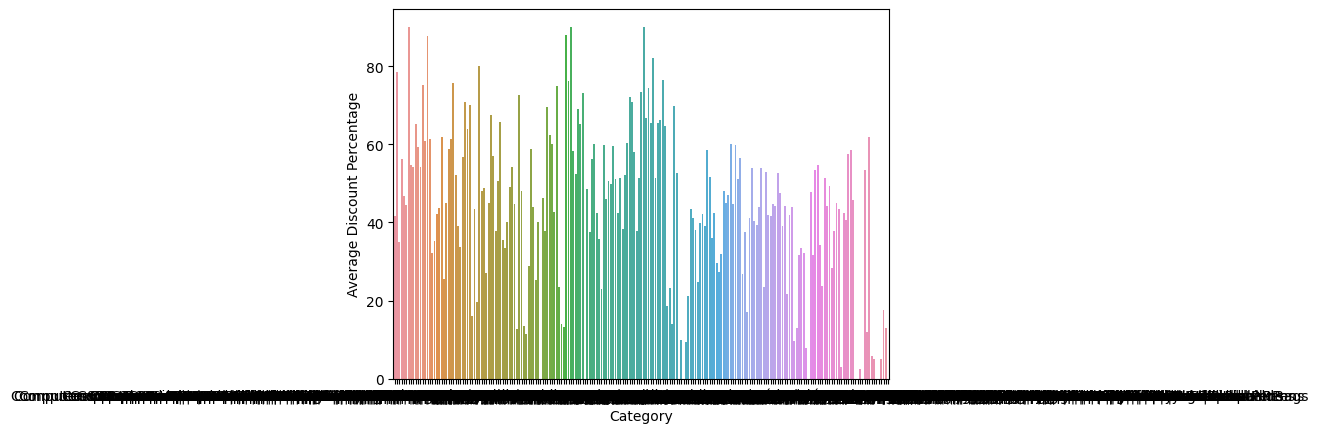
|  | * discounted\_price | * actual\_price | * discount\_percentage | * rating | * rating\_count |
| --- | --- | --- | --- | --- | --- |
| * count | * 1465.000000 | * 1465.000000 | * 1465.000000 | * 1465.000000 | * 1463.000000 |
| * mean | * 3125.310874 | * 5444.990635 | * 0.476915 | * 4.096451 | * 18295.541353 |
| * std | * 6944.304394 | * 10874.826864 | * 0.216359 | * 0.291620 | * 42753.864952 |
| * min | * 39.000000 | * 39.000000 | * 0.000000 | * 2.000000 | * 2.000000 |
| * 25% | * 325.000000 | * 800.000000 | * 0.320000 | * 4.000000 | * 1186.000000 |
| * 50% | * 799.000000 | * 1650.000000 | * 0.500000 | * 4.100000 | * 5179.000000 |
| * 75% | * 1999.000000 | * 4295.000000 | * 0.630000 | * 4.300000 | * 17336.500000 |
| * max | * 77990.000000 | * 139900.000000 | * 0.940000 | * 5.000000 | * 426973.000000 |

* Observation Set 2
* ***All columns data type was object So, I converted some column data type to float.***
* ***There are 4 numeric as per Python coding or descriptive statistics from Python describe function***
* Dealing with the missing values
* ***Dealing with the missing values is one of the most important part of the data wrangling process, we must deal with the missing values in order to get the correct insights from the data.***
* **Missing Values**
* df.isnull().sum().sort\_values(ascending = False)
* 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
* *# Find missing values percentage in the data*
* round(df.isnull().sum() / len(df) \* 100, 2).sort\_values(ascending=False)
* 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
* *# Find total number of missing values*
* df.isnull().sum().sum()
* 2
* **Let's plot the missing values**
* *# make a figure size*
* plt.figure(figsize=(22, 10))
* *# plot the null values in each column*
* sns.heatmap(df.isnull(), yticklabels=False, cbar=False, cmap='viridis')
* <Axes: >
* 
* Figure-1: Heatmap of Missing Values
* **Let's plot the missing values by percentage**
* *# make figure size*
* plt.figure(figsize=(22, 10))
* *# plot the null values by their percentage in each column*
* missing\_percentage = df.isnull().sum()/len(df)\*100
* missing\_percentage.plot(kind='bar')
* *# add the labels*
* plt.xlabel('Columns')
* plt.ylabel('Percentage')
* plt.title('Percentage of Missing Values in each Column')
* Text(0.5, 1.0, 'Percentage of Missing Values in each Column')
* 
* Figure-2: This is a percentage null values plot.
* **We are only viewing the rows where there are null values in the column.**
* df[df['rating\_count'].isnull()].head(5)

|  | * 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 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| * 282 | * B0B94JPY2N | * Amazon Brand - Solimo 65W Fast Charging Braide... | * Computers&Accessories|Accessories&Peripherals|... | * 199.0 | * 999.0 | * 0.80 | * 3.0 | * NaN | * USB C to C Cable: This cable has type C connec... | * AE7CFHY23VAJT2FI4NZKKP6GS2UQ | * Pranav | * RUB7U91HVZ30 | * The cable works but is not 65W as advertised | * I have a pd supported car charger and I bought... | * https://m.media-amazon.com/images/W/WEBP\_40237... | * https://www.amazon.in/Amazon-Brand-Charging-Su... |
| * 324 | * B0BQRJ3C47 | * REDTECH USB-C to Lightning Cable 3.3FT, [Apple... | * Computers&Accessories|Accessories&Peripherals|... | * 249.0 | * 999.0 | * 0.75 | * 5.0 | * NaN | * 💎[The Fastest Charge] - This iPhone USB C cabl... | * AGJC5O5H5BBXWUV7WRIEIOOR3TVQ | * Abdul Gafur | * RQXD5SAMMPC6L | * Awesome Product | * Quick delivery.Awesome ProductPacking was good... | * https://m.media-amazon.com/images/I/31-q0xhaTA... | * https://www.amazon.in/REDTECH-Lightning-Certif... |

* *# Impute missing values*
* df['rating\_count'] = df.rating\_count.fillna(value=df['rating\_count'].median())
* df.isnull().sum().sort\_values(ascending = False)
* 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
* ***Milestone 1: We have cleaned the dataset from null values***
* Find Duplications and Analyse them
* **Duplicates**
* ***Removing duplicates is one of the most important part of the data wrangling process, we must remove the duplicates in order to get the correct insights from the data.***
* ***If you do not remove duplicates from a dataset, it can lead to incorrect insights and analysis.***
* ***Duplicates can skew statistical measures such as mean, median, and standard deviation, and can also lead to over-representation of certain data points.***
* ***It is important to remove duplicates to ensure the accuracy and reliability of your data analysis.***
* *# Find Duplicate*
* df.duplicated().any()
* False
* df.columns
* 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')
* any\_duplicates = df.duplicated(subset=['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']).any()
* any\_duplicates
* False
* ***Milestone 2: Hence no duplicates found***
* Data Visualization
* **Scatter Plot**
* *# Plot actual\_price vs. rating*
* plt.scatter(df['actual\_price'], df['rating'])
* plt.xlabel('Actual\_price')
* plt.ylabel('Rating')
* plt.show()
* 
* *# dont show warnings*
* import warnings
* warnings.filterwarnings('ignore')
* **Histogram**
* *# Plot distribution of actual\_price*
* plt.hist(df['actual\_price'])
* plt.xlabel('Actual Price')
* plt.ylabel('Frequency')
* plt.show()
* 
* from sklearn.preprocessing import LabelEncoder
* *# label encode categorical variables*
* le\_product\_id = LabelEncoder()
* le\_category = LabelEncoder()
* le\_review\_id = LabelEncoder()
* le\_review\_content = LabelEncoder()
* le\_product\_name = LabelEncoder()
* le\_user\_name = LabelEncoder()
* le\_about\_product = LabelEncoder()
* le\_user\_id = LabelEncoder()
* le\_review\_title = LabelEncoder()
* le\_img\_link = LabelEncoder()
* le\_product\_link = LabelEncoder()
* df['product\_id'] = le\_product\_id.fit\_transform(df['product\_id'])
* df['category'] = le\_category.fit\_transform(df['category'])
* df['review\_id'] = le\_review\_id.fit\_transform(df['review\_id'])
* df['review\_content'] = le\_review\_content.fit\_transform(df['review\_content'])
* df['product\_name'] = le\_product\_name.fit\_transform(df['product\_name'])
* df['user\_name'] = le\_user\_name.fit\_transform(df['user\_name'])
* df['about\_product'] = le\_about\_product.fit\_transform(df['about\_product'])
* df['user\_id'] = le\_user\_id.fit\_transform(df['user\_id'])
* df['review\_title'] = le\_review\_title.fit\_transform(df['review\_title'])
* df['img\_link'] = le\_img\_link.fit\_transform(df['img\_link'])
* df['product\_link'] = le\_product\_link.fit\_transform(df['product\_link'])
* **Heatmap**
* *# Plot correlations between variables*
* correlation\_matrix = df.corr()
* sns.heatmap(correlation\_matrix, annot=True)
* plt.show()
* 
* Correlation Analysis:
* *# Calculate Pearson correlation coefficients (default in Pandas)*
* correlation\_matrix = df.corr()
* *# Print the correlation matrix*
* print(correlation\_matrix)
* *# Create a heatmap to visualize the correlations*
* sns.heatmap(correlation\_matrix, annot=True, cmap="coolwarm")
* plt.title("Correlation Matrix (Pearson)")
* plt.show()
* *# Calculate Spearman correlation coefficients (for non-linear relationships)*
* spearman\_correlation\_matrix = df.corr(method="spearman")
* *# Print the Spearman correlation matrix*
* print(spearman\_correlation\_matrix)
* *# Create a heatmap to visualize the Spearman correlations*
* sns.heatmap(spearman\_correlation\_matrix, annot=True, cmap="coolwarm")
* plt.title("Correlation Matrix (Spearman)")
* plt.show()
* product\_id product\_name category discounted\_price \
* product\_id 1.000000 0.084089 -0.012565 0.206448
* product\_name 0.084089 1.000000 -0.103778 0.090665
* category -0.012565 -0.103778 1.000000 0.119365
* discounted\_price 0.206448 0.090665 0.119365 1.000000
* actual\_price 0.246733 0.078567 0.122451 0.961915
* discount\_percentage 0.289514 0.101913 -0.314465 -0.242412
* rating -0.149105 -0.035592 -0.109424 0.120386
* rating\_count -0.175530 0.092450 -0.098421 -0.027081
* about\_product 0.041404 0.158263 -0.038753 0.052618
* user\_id 0.065688 -0.024093 0.012707 0.041731
* user\_name 0.016145 0.024598 0.037822 -0.063069
* review\_id -0.024282 0.013492 0.014015 -0.049757
* review\_title 0.007650 -0.060594 0.004712 -0.020981
* review\_content -0.046273 -0.018505 -0.012107 -0.015904
* img\_link -0.078803 -0.060858 0.038850 0.018917
* product\_link 0.096205 0.823725 -0.067710 0.110186
* actual\_price discount\_percentage rating \
* product\_id 0.246733 0.289514 -0.149105
* product\_name 0.078567 0.101913 -0.035592
* category 0.122451 -0.314465 -0.109424
* discounted\_price 0.961915 -0.242412 0.120386
* actual\_price 1.000000 -0.118098 0.121744
* discount\_percentage -0.118098 1.000000 -0.154563
* rating 0.121744 -0.154563 1.000000
* rating\_count -0.035959 0.011097 0.101700
* about\_product 0.048529 0.060846 -0.036056
* user\_id 0.041501 0.008288 -0.014528
* user\_name -0.049567 0.064618 -0.012704
* review\_id -0.045640 0.022442 -0.134071
* review\_title 0.004521 0.076043 -0.075715
* review\_content -0.013948 0.005993 -0.044567
* img\_link 0.020080 -0.011701 0.003993
* product\_link 0.099286 0.074548 -0.044125
* rating\_count about\_product user\_id user\_name \
* product\_id -0.175530 0.041404 0.065688 0.016145
* product\_name 0.092450 0.158263 -0.024093 0.024598
* category -0.098421 -0.038753 0.012707 0.037822
* discounted\_price -0.027081 0.052618 0.041731 -0.063069
* actual\_price -0.035959 0.048529 0.041501 -0.049567
* discount\_percentage 0.011097 0.060846 0.008288 0.064618
* rating 0.101700 -0.036056 -0.014528 -0.012704
* rating\_count 1.000000 -0.030821 -0.094512 0.059160
* about\_product -0.030821 1.000000 -0.001946 -0.004547
* user\_id -0.094512 -0.001946 1.000000 -0.051368
* user\_name 0.059160 -0.004547 -0.051368 1.000000
* review\_id -0.043089 0.019690 0.001685 -0.009057
* review\_title -0.084239 -0.015824 0.039035 -0.038880
* review\_content -0.025268 -0.024567 0.031295 0.002433
* img\_link -0.100781 0.030874 -0.014114 -0.038779
* product\_link 0.003665 0.147208 -0.008662 0.026996
* review\_id review\_title review\_content img\_link \
* product\_id -0.024282 0.007650 -0.046273 -0.078803
* product\_name 0.013492 -0.060594 -0.018505 -0.060858
* category 0.014015 0.004712 -0.012107 0.038850
* discounted\_price -0.049757 -0.020981 -0.015904 0.018917
* actual\_price -0.045640 0.004521 -0.013948 0.020080
* discount\_percentage 0.022442 0.076043 0.005993 -0.011701
* rating -0.134071 -0.075715 -0.044567 0.003993
* rating\_count -0.043089 -0.084239 -0.025268 -0.100781
* about\_product 0.019690 -0.015824 -0.024567 0.030874
* user\_id 0.001685 0.039035 0.031295 -0.014114
* user\_name -0.009057 -0.038880 0.002433 -0.038779
* review\_id 1.000000 -0.034569 0.016665 0.005422
* review\_title -0.034569 1.000000 0.201131 -0.012056
* review\_content 0.016665 0.201131 1.000000 -0.020283
* img\_link 0.005422 -0.012056 -0.020283 1.000000
* product\_link 0.014423 -0.006080 -0.038574 -0.043935
* product\_link
* product\_id 0.096205
* product\_name 0.823725
* category -0.067710
* discounted\_price 0.110186
* actual\_price 0.099286
* discount\_percentage 0.074548
* rating -0.044125
* rating\_count 0.003665
* about\_product 0.147208
* user\_id -0.008662
* user\_name 0.026996
* review\_id 0.014423
* review\_title -0.006080
* review\_content -0.038574
* img\_link -0.043935
* product\_link 1.000000
* 
* product\_id product\_name category discounted\_price \
* product\_id 1.000000 0.083112 -0.013553 0.146237
* product\_name 0.083112 1.000000 -0.106193 0.056597
* category -0.013553 -0.106193 1.000000 0.360733
* discounted\_price 0.146237 0.056597 0.360733 1.000000
* actual\_price 0.269064 0.105719 0.277291 0.932787
* discount\_percentage 0.271879 0.106467 -0.322090 -0.372991
* rating -0.144268 -0.061395 -0.101758 0.079412
* rating\_count -0.406559 0.128565 -0.171893 0.122296
* about\_product 0.041118 0.157675 -0.048319 -0.056144
* user\_id 0.065228 -0.023810 0.015389 0.079048
* user\_name 0.016859 0.024479 0.038735 -0.018599
* review\_id -0.024644 0.015269 0.016119 -0.048420
* review\_title 0.007415 -0.060779 0.009407 -0.018665
* review\_content -0.045763 -0.017671 -0.013519 -0.053281
* img\_link -0.078258 -0.060212 0.042158 -0.021097
* product\_link 0.095734 0.823655 -0.065789 0.071049
* actual\_price discount\_percentage rating \
* product\_id 0.269064 0.271879 -0.144268
* product\_name 0.105719 0.106467 -0.061395
* category 0.277291 -0.322090 -0.101758
* discounted\_price 0.932787 -0.372991 0.079412
* actual\_price 1.000000 -0.066363 0.033066
* discount\_percentage -0.066363 1.000000 -0.144815
* rating 0.033066 -0.144815 1.000000
* rating\_count 0.093400 -0.096580 0.180947
* about\_product -0.029216 0.061906 -0.033470
* user\_id 0.086375 0.005825 -0.000816
* user\_name -0.001395 0.066214 -0.014171
* review\_id -0.045914 0.012696 -0.130196
* review\_title 0.004687 0.075222 -0.067546
* review\_content -0.058924 0.012542 -0.054206
* img\_link -0.018511 -0.010185 0.019161
* product\_link 0.111965 0.074625 -0.063169
* rating\_count about\_product user\_id user\_name \
* product\_id -0.406559 0.041118 0.065228 0.016859
* product\_name 0.128565 0.157675 -0.023810 0.024479
* category -0.171893 -0.048319 0.015389 0.038735
* discounted\_price 0.122296 -0.056144 0.079048 -0.018599
* actual\_price 0.093400 -0.029216 0.086375 -0.001395
* discount\_percentage -0.096580 0.061906 0.005825 0.066214
* rating 0.180947 -0.033470 -0.000816 -0.014171
* rating\_count 1.000000 -0.104660 -0.052821 0.021839
* about\_product -0.104660 1.000000 -0.001770 -0.004390
* user\_id -0.052821 -0.001770 1.000000 -0.051860
* user\_name 0.021839 -0.004390 -0.051860 1.000000
* review\_id -0.050121 0.020765 0.001674 -0.009772
* review\_title -0.033234 -0.015792 0.038260 -0.038420
* review\_content -0.024274 -0.025319 0.032042 0.002172
* img\_link -0.074989 0.030514 -0.014201 -0.039575
* product\_link 0.062640 0.146951 -0.008345 0.026876
* review\_id review\_title review\_content img\_link \
* product\_id -0.024644 0.007415 -0.045763 -0.078258
* product\_name 0.015269 -0.060779 -0.017671 -0.060212
* category 0.016119 0.009407 -0.013519 0.042158
* discounted\_price -0.048420 -0.018665 -0.053281 -0.021097
* actual\_price -0.045914 0.004687 -0.058924 -0.018511
* discount\_percentage 0.012696 0.075222 0.012542 -0.010185
* rating -0.130196 -0.067546 -0.054206 0.019161
* rating\_count -0.050121 -0.033234 -0.024274 -0.074989
* about\_product 0.020765 -0.015792 -0.025319 0.030514
* user\_id 0.001674 0.038260 0.032042 -0.014201
* user\_name -0.009772 -0.038420 0.002172 -0.039575
* review\_id 1.000000 -0.035109 0.017536 0.005354
* review\_title -0.035109 1.000000 0.203353 -0.012142
* review\_content 0.017536 0.203353 1.000000 -0.020711
* img\_link 0.005354 -0.012142 -0.020711 1.000000
* product\_link 0.015516 -0.006416 -0.038634 -0.043347
* product\_link
* product\_id 0.095734
* product\_name 0.823655
* category -0.065789
* discounted\_price 0.071049
* actual\_price 0.111965
* discount\_percentage 0.074625
* rating -0.063169
* rating\_count 0.062640
* about\_product 0.146951
* user\_id -0.008345
* user\_name 0.026876
* review\_id 0.015516
* review\_title -0.006416
* review\_content -0.038634
* img\_link -0.043347
* product\_link 1.000000
* 
* *# Calculate correlation coefficient between product price and sales*
* correlation\_coefficient = np.corrcoef(df['actual\_price'], df['rating'])[0, 1]
* *# Print correlation coefficient*
* print(correlation\_coefficient)
* 0.1217444960999836
* Grouping and Aggregation
* *# Calculate mean sales by product category*
* grouped\_df = df.groupby('category')['rating'].mean()
* *# Print mean sales by product category*
* print(grouped\_df)
* category
* 3.800000
* 4.150000
* 3.500000
* 3.600000
* 4.050000
* ...
* 206 4.250000
* 207 4.150000
* 208 4.300000
* 209 4.133333
* 210 4.300000
* Name: rating, Length: 211, dtype: float64
* **Calculate summary statistics for groups**
* *# Mean rating by category*
* mean\_sales\_by\_category = df.groupby('category')['rating'].mean()
* print(mean\_sales\_by\_category)
* *# Median rating by review\_content*
* median\_sales\_by\_age = df.groupby('review\_content')['rating'].median()
* print(median\_sales\_by\_age)
* *# Standard deviation of actual\_price by product\_name*
* std\_price\_by\_brand = df.groupby('product\_name')['actual\_price'].std()
* print(std\_price\_by\_brand)
* category
* 3.800000
* 4.150000
* 3.500000
* 3.600000
* 4.050000
* ...
* 206 4.250000
* 207 4.150000
* 208 4.300000
* 209 4.133333
* 210 4.300000
* Name: rating, Length: 211, dtype: float64
* review\_content
* 4.1
* 3.9
* 4.3
* 4.3
* 3.8
* ...
* 1207 4.0
* 1208 4.3
* 1209 4.3
* 1210 4.5
* 1211 4.3
* Name: rating, Length: 1212, dtype: float64
* product\_name
* NaN
* NaN
* NaN
* NaN
* NaN
* ...
* 1332 NaN
* 1333 NaN
* 1334 NaN
* 1335 NaN
* 1336 0.0
* Name: actual\_price, Length: 1337, dtype: float64
* Create pivot tables
* *# Pivot table of rating by category and customer location*
* pivot\_table = df.pivot\_table(values='rating', index='category', columns='product\_link', aggfunc='mean')
* print(pivot\_table)
* *# Pivot table of average rating\_count by customer age group and product category*
* pivot\_table = df.pivot\_table(values='rating\_count', index='review\_content', columns='category', aggfunc='mean')
* print(pivot\_table)
* Statistical Tests:
* import scipy.stats as stats
* *# Conduct t-test to compare rating between two categories*
* t\_statistic, p\_value = stats.ttest\_ind(df[df['category'] == 'electronics']['rating'], df[df['category'] == 'clothing']['rating'])
* *# Print t-statistic and p-value*
* print(t\_statistic, p\_value)
* nan nan
* 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 int64
* 1 product\_name 1465 non-null int64
* 2 category 1465 non-null int64
* 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 int64
* 9 user\_id 1465 non-null int64
* 10 user\_name 1465 non-null int64
* 11 review\_id 1465 non-null int64
* 12 review\_title 1465 non-null int64
* 13 review\_content 1465 non-null int64
* 14 img\_link 1465 non-null int64
* 15 product\_link 1465 non-null int64
* dtypes: float64(5), int64(11)
* memory usage: 183.2 KB
* *# Chi-square test*
* *# Create a contigency table*
* contigency\_table = pd.crosstab(df['actual\_price'], df['rating'])
* contigency\_table

| * rating | * 2.0 | * 2.3 | * 2.6 | * 2.8 | * 2.9 | * 3.0 | * 3.1 | * 3.2 | * 3.3 | * 3.4 | * 3.5 | * 3.6 | * 3.7 | * 3.8 | * 3.9 | * 4.0 | * 4.1 | * 4.2 | * 4.3 | * 4.4 | * 4.5 | * 4.6 | * 4.7 | * 4.8 | * 5.0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| * actual\_price |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| * 39.0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 1 | * 0 | * 1 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 |
| * 50.0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 1 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 |
| * 59.0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 1 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 |
| * 75.0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 1 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 |
| * 79.0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 1 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 |
| * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... | * ... |
| * 74999.0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 1 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 |
| * 75990.0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 1 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 |
| * 79990.0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 1 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 |
| * 85000.0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 1 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 |
| * 139900.0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 0 | * 1 | * 0 | * 0 |

* 449 rows × 25 columns
* *# perform chi-square test*
* chi2, p, dof, expected = stats.chi2\_contingency(contigency\_table)
* *# print the results*
* print('Chi-square statistic:', chi2)
* print('p-value:', p)
* print('Degrees of freedom:', dof)
* print(f"Expected:**\n** **{**expected**}**")
* Chi-square statistic: 8635.264277480239
* p-value: 1.0
* Degrees of freedom: 10752
* Expected:
* [[0.00136519 0.00136519 0.00136519 ... 0.00819113 0.00409556 0.00409556]
* [0.00068259 0.00068259 0.00068259 ... 0.00409556 0.00204778 0.00204778]
* [0.00068259 0.00068259 0.00068259 ... 0.00409556 0.00204778 0.00204778]
* ...
* [0.00068259 0.00068259 0.00068259 ... 0.00409556 0.00204778 0.00204778]
* [0.00068259 0.00068259 0.00068259 ... 0.00409556 0.00204778 0.00204778]
* [0.00068259 0.00068259 0.00068259 ... 0.00409556 0.00204778 0.00204778]]
* *# inverse transform the data*
* df['product\_id'] = le\_product\_id.inverse\_transform(df['product\_id'])
* df['category'] = le\_category.inverse\_transform(df['category'])
* df['review\_id'] = le\_review\_id.inverse\_transform(df['review\_id'])
* df['review\_content'] = le\_review\_content.inverse\_transform(df['review\_content'])
* df['product\_name'] = le\_product\_name.inverse\_transform(df['product\_name'])
* df['user\_name'] = le\_user\_name.inverse\_transform(df['user\_name'])
* df['about\_product'] = le\_about\_product.inverse\_transform(df['about\_product'])
* df['user\_id'] = le\_user\_id.inverse\_transform(df['user\_id'])
* df['review\_title'] = le\_review\_title.inverse\_transform(df['review\_title'])
* df['img\_link'] = le\_img\_link.inverse\_transform(df['img\_link'])
* df['product\_link'] = le\_product\_link.inverse\_transform(df['product\_link'])
* Questions and Answers
* ***These are some questions are follows:***
* ***Q1: What is the average rating for each product category?***
* ***Q2: What are the top rating\_count products by category?***
* ***Q3: What is the distribution of discounted prices vs. actual prices?***
* ***Q4: How does the average discount percentage vary across categories?***
* ***Q5: What are the most popular product name?***
* ***Q6: What are the most popular product keywords?***
* ***Q7: What are the most popular product reviews?***
* ***Q8: What is the correlation between discounted\_price and rating?***
* ***Q9: What are the Top 5 categories based with highest ratings?***
* ***Q1: What is the average rating for each product category?***
* import pandas as pd
* *# Check the data type of the "rating" column*
* print(df["rating"].dtype)
* *# If the data type is not numeric, convert it to numeric*
* if df["rating"].dtype == "object":
* df["rating"] = pd.to\_numeric(df["rating"], errors="coerce") *# Handle potential errors*
* *# Calculate the average ratings after ensuring numeric data type*
* average\_ratings = df.groupby("category")["rating"].mean().reset\_index()
* print(average\_ratings)
* float64
* category rating
* Car&Motorbike|CarAccessories|InteriorAccessori... 3.800000
* Computers&Accessories|Accessories&Peripherals|... 4.150000
* Computers&Accessories|Accessories&Peripherals|... 3.500000
* Computers&Accessories|Accessories&Peripherals|... 3.600000
* Computers&Accessories|Accessories&Peripherals|... 4.050000
* .. ... ...
* 206 OfficeProducts|OfficePaperProducts|Paper|Stati... 4.250000
* 207 OfficeProducts|OfficePaperProducts|Paper|Stati... 4.150000
* 208 OfficeProducts|OfficePaperProducts|Paper|Stati... 4.300000
* 209 OfficeProducts|OfficePaperProducts|Paper|Stati... 4.133333
* 210 Toys&Games|Arts&Crafts|Drawing&PaintingSupplie... 4.300000
* [211 rows x 2 columns]
* ***Answer 1:***
* ***The output shows that most product categories have generally positive customer feedback, with average ratings above 3.50. However, some categories (e.g., 2 and 3) have lower ratings, suggesting potential areas for improvement. Further analysis of these categories could help identify specific reasons for lower feedback and identify potential solutions.***
* ***Q2: What are the top rating\_count products by category?***
* import pandas as pd
* top\_reviewed\_per\_category = (
* df.groupby("category")
* .apply(lambda x: x.nlargest(10, "rating\_count"))
* .reset\_index(drop=True)
* )
* print(top\_reviewed\_per\_category)
* ***Answer 2:***
* ***The output highlights products likely to be popular within their categories based on high review counts, suggesting customer interest and engagement.***
* ***Review counts range from 9 to 15867, implying varying levels of attention and feedback across products.***
* ***Most listed products have ratings above 3.5, indicating a generally positive customer experience.***
* ***Products with the highest review counts within their categories might be considered potential top sellers, even without direct sales data.***
* ***Q3: What is the distribution of discounted prices vs. actual prices?***
* import pandas as pd
* *# Create histograms*
* df["discounted\_price"].hist(label="Discounted Price")
* df["actual\_price"].hist(label="Actual Price")
* *# Calculate and analyze discount percentages*
* df["discount\_percentage"] = (df["actual\_price"] - df["discounted\_price"]) / df["actual\_price"] \* 100
* df["discount\_percentage"].describe()
* df["discount\_percentage"].hist(label="Discount Percentage")
* <Axes: >
* 
* ***Answer 3:***
* ***The output shows that discounted prices are generally lower than actual prices, with a median discounted price of $200 and a median actual price of $400.***
* ***The discount percentage distribution is skewed to the left, with most products having a discount of 30% or less.***
* ***The output suggests that there may be opportunities to increase discounted prices or discount percentages to attract more customers.***
* ***Q4: How does the average discount percentage vary across categories?***
* *# Calculate average discount percentage per category*
* avg\_discount\_per\_category = df.groupby('category')['discount\_percentage'].mean()
* *# Display results*
* print(avg\_discount\_per\_category)
* *# Optional: Visualization*
* sns.barplot(x=avg\_discount\_per\_category.index, y=avg\_discount\_per\_category.values)
* plt.xlabel("Category")
* plt.ylabel("Average Discount Percentage")
* plt.show()
* category
* Car&Motorbike|CarAccessories|InteriorAccessories|AirPurifiers&Ionizers 41.525000
* Computers&Accessories|Accessories&Peripherals|Adapters|USBtoUSBAdapters 78.387733
* Computers&Accessories|Accessories&Peripherals|Audio&VideoAccessories|PCHeadsets 35.035035
* Computers&Accessories|Accessories&Peripherals|Audio&VideoAccessories|PCMicrophones 56.335120
* Computers&Accessories|Accessories&Peripherals|Audio&VideoAccessories|PCSpeakers 46.719582
* ...
* OfficeProducts|OfficePaperProducts|Paper|Stationery|Pens,Pencils&WritingSupplies|Pens&Refills|GelInkRollerballPens 0.000000
* OfficeProducts|OfficePaperProducts|Paper|Stationery|Pens,Pencils&WritingSupplies|Pens&Refills|LiquidInkRollerballPens 5.000000
* OfficeProducts|OfficePaperProducts|Paper|Stationery|Pens,Pencils&WritingSupplies|Pens&Refills|RetractableBallpointPens 17.619048
* OfficeProducts|OfficePaperProducts|Paper|Stationery|Pens,Pencils&WritingSupplies|Pens&Refills|StickBallpointPens 13.074074
* Toys&Games|Arts&Crafts|Drawing&PaintingSupplies|ColouringPens&Markers 0.000000
* Name: discount\_percentage, Length: 211, dtype: float64
* 
* ***Answer 4:***
* ***Average discount percentages vary widely across categories, ranging from 0% to 78.39%.***
* ***Categories 1 and 3 stand out with notably higher average discounts (78.39% and 56.34%), suggesting potential factors like clearance efforts, high competition, or lower-profit margins.***
* ***Categories 0, 206, 207, 210 have average discounts of 0%, indicating consistent pricing or strong demand for products within those categories.***
* ***Other categories exhibit varying discount percentages, likely reflecting diverse pricing strategies and market dynamics.***
* ***Q5: What are the most popular product name?***
* *# Count occurrences of product names*
* product\_counts = df["product\_name"].value\_counts()
* *# Sort in descending order and display top results*
* print(product\_counts.sort\_values(ascending=False).head(10))
* product\_name
* Fire-Boltt Ninja Call Pro Plus 1.83" Smart Watch with Bluetooth Calling, AI Voice Assistance, 100 Sports Modes IP67 Rating, 240\*280 Pixel High Resolution 5
* Fire-Boltt Phoenix Smart Watch with Bluetooth Calling 1.3",120+ Sports Modes, 240\*240 PX High Res with SpO2, Heart Rate Monitoring & IP67 Rating 4
* Ambrane 2 in 1 Type-C & Micro USB Cable with 60W / 3A Fast Charging, 480 mbps High Data, PD Technology & Quick Charge 3.0, Compatible with All Type-C & Micro USB Devices (ABDC-10, Black) 3
* Pinnaclz Original Combo of 2 Micro USB Fast Charging Cable, USB Charging Cable for Data Transfer Perfect for Android Smart Phones White 1.2 Meter Made in India (Pack of 2) 3
* Portronics Konnect L POR-1081 Fast Charging 3A Type-C Cable 1.2Meter with Charge & Sync Function for All Type-C Devices (Grey) 3
* boAt Micro USB 55 Tangle-free, Sturdy Micro USB Cable with 3A Fast Charging & 480mbps Data Transmission (Black) 3
* MI Usb Type-C Cable Smartphone (Black) 3
* pTron Solero TB301 3A Type-C Data and Fast Charging Cable, Made in India, 480Mbps Data Sync, Strong and Durable 1.5-Meter Nylon Braided USB Cable for Type-C Devices for Charging Adapter (Black) 3
* Portronics Konnect L 1.2M Fast Charging 3A 8 Pin USB Cable with Charge & Sync Function for iPhone, iPad (Grey) 3
* boAt Deuce USB 300 2 in 1 Type-C & Micro USB Stress Resistant, Tangle-Free, Sturdy Cable with 3A Fast Charging & 480mbps Data Transmission, 10000+ Bends Lifespan and Extended 1.5m Length(Martian Red) 3
* Name: count, dtype: int64
* ***Answer 5:***
* ***Fire-Boltt Ninja Call Pro Plus Smart Watch is the most popular product, followed by Fire-Boltt Phoenix Smart Watch.***
* ***Smart Watches and Charging Cables are the most popular product categories.***
* ***Multiple brands are represented, with boAt appearing twice.***
* ***Fast charging, durability, and functionality are key features.***
* ***Popularity is relatively evenly distributed beyond the leading product.***
* ***Q6: What are the most popular product keywords?***
* def extract\_keywords(product\_name):
* *"""Extracts keywords from a product name, handling potential numbers."""*
* if isinstance(product\_name, str): *# Check if it's a string*
* keywords = product\_name.lower().split() *# Split into words and lowercase*
* keywords = [word for word **in** keywords if word.isalpha()] *# Remove non-alphabetical characters*
* else:
* keywords = [] *# Handle non-string values (e.g., integers) by returning an empty list*
* return keywords
* *# Apply the function to extract keywords*
* df["keywords"] = df["product\_name"].apply(extract\_keywords)
* *# Flatten the list of keywords*
* all\_keywords = [keyword for keywords **in** df["keywords"] for keyword **in** keywords]
* *# Count keyword occurrences*
* keyword\_counts = pd.Series(all\_keywords).value\_counts()
* *# Display the top 10 most popular keywords*
* print(keyword\_counts.head(10))
* with 751
* for 672
* usb 377
* and 330
* cable 320
* charging 219
* to 218
* fast 211
* c 182
* smart 171
* Name: count, dtype: int64
* ***Answer 6:***
* ***USB connectivity, charging (especially fast charging), and cables are prominent product features.***
* ***Prepositions and conjunctions like "with", "for", "and", "to" suggest a focus on explaining product compatibility and usage scenarios.***
* ***Cables and smart devices are likely well-represented in the dataset.***
* ***Product names tend to be concise and use common words, potentially benefiting from refined keyword extraction techniques.***
* ***Q7: What are the most popular product reviews?***
* from textblob import TextBlob *# Import TextBlob library*
* *# Select review column*
* df[["product\_id", "user\_id", "review\_content"]]
* *# Calculate sentiment score for each review*
* df["sentiment"] = df["review\_content"].apply(lambda text: TextBlob(text).sentiment.polarity)
* *# Sort by sentiment score (ascending for positive)*
* df\_sorted = df.sort\_values(by="sentiment", ascending=True)
* *# Display top reviews based on a desired number (e.g., top 10)*
* top\_reviews = df\_sorted.head(10)
* print(top\_reviews)
* product\_id product\_name \
* 155 B09XJ1LM7R 7SEVEN® Compatible for Tata Sky Remote Origina...
* 1237 B0B7NWGXS6 Havells Bero Quartz Heater Black 800w 2 Heat S...
* 145 B00RFWNJMC Airtel DigitalTV DTH Remote SD/HD/HD Recording...
* 22 B09F6S8BT6 Samsung 80 cm (32 Inches) Wondertainment Serie...
* 152 B08PV1X771 Samsung 80 cm (32 inches) Wondertainment Serie...
* 723 B09F6S8BT6 Samsung 80 cm (32 Inches) Wondertainment Serie...
* 1463 B00J5DYCCA Havells Ventil Air DSP 230mm Exhaust Fan (Pist...
* 1198 B09SPTNG58 Crompton Sea Sapphira 1200 mm Ultra High Speed...
* 738 B08MZQBFLN Callas Multipurpose Foldable Laptop Table with...
* 1367 B07LG96SDB ESN 999 Supreme Quality 1500W Immersion Water ...
* category discounted\_price \
* 155 Electronics|HomeTheater,TV&Video|Accessories|R... 399.0
* 1237 Home&Kitchen|Heating,Cooling&AirQuality|RoomHe... 2439.0
* 145 Electronics|HomeTheater,TV&Video|Accessories|R... 195.0
* 22 Electronics|HomeTheater,TV&Video|Televisions|S... 13490.0
* 152 Electronics|HomeTheater,TV&Video|Televisions|S... 15490.0
* 723 Electronics|HomeTheater,TV&Video|Televisions|S... 13490.0
* 1463 Home&Kitchen|Heating,Cooling&AirQuality|Fans|E... 1399.0
* 1198 Home&Kitchen|Heating,Cooling&AirQuality|Fans|C... 1449.0
* 738 Computers&Accessories|Accessories&Peripherals|... 849.0
* 1367 Home&Kitchen|Heating,Cooling&AirQuality|WaterH... 335.0
* actual\_price discount\_percentage rating rating\_count \
* 155 799.0 50.062578 4.3 12.0
* 1237 2545.0 4.165029 4.1 25.0
* 145 499.0 60.921844 3.7 1383.0
* 22 22900.0 41.091703 4.3 16299.0
* 152 20900.0 25.885167 4.3 16299.0
* 723 22900.0 41.091703 4.3 16299.0
* 1463 1890.0 25.978836 4.0 8031.0
* 1198 2349.0 38.314176 3.9 9019.0
* 738 4999.0 83.016603 4.0 20457.0
* 1367 510.0 34.313725 3.8 3195.0
* about\_product \
* 155 [Compatible] All model of dth SD / HD / HD+ Pl...
* 1237 Two quartz heating tubes|Carry Handle For Easy...
* 145 Compatible with SD and HD Recording
* 22 Resolution: HD Ready (1366x768) | Refresh Rate...
* 152 Resolution: HD Ready (1366x768) | Refresh Rate...
* 723 Resolution: HD Ready (1366x768) | Refresh Rate...
* 1463 Fan sweep area: 230 MM ; Noise level: (40 - 45...
* 1198 PRODUCT: Crompton's corrosion resistant high p...
* 738 【WATCH, PLAY, STUDY - WITHOUT LEAVING THE BED!...
* 1367 Specially designed heating element for quick h...
* user\_id \
* 155 AE242TR3GQ6TYC6W4SJ5UYYKBTYQ
* 1237 AFM4A33L64TPLILW4OHTSKRZR3NQ,AH6NEABVASSTXS6RP...
* 145 AGD2H2SMDLQK62MH7BFWQ2INBP2A,AELIUKITTHS3MSGTS...
* 22 AHEVO4Q5NM4YXMG2HDDXC5XMBGRQ,AFZPH7ZAWX5VDY3HO...
* 152 AHEVO4Q5NM4YXMG2HDDXC5XMBGRQ,AFZPH7ZAWX5VDY3HO...
* 723 AHEVO4Q5NM4YXMG2HDDXC5XMBGRQ,AFZPH7ZAWX5VDY3HO...
* 1463 AF2JQCLSCY3QJATWUNNHUSVUPNQQ,AFDMLUXC5LS5RXDJS...
* 1198 AENJBTR2KDJMOAEQA4AROLV244QQ,AE666QCFHN4ZT5Q6Y...
* 738 AHB4AEOCLEVH2JSTXPU737KTXS4Q,AHXC62FGJRYSCJEBZ...
* 1367 AHIDFZK6JPIY7FCTPZQJR6MSWV7Q,AGWW4VSBX2UUCMM5V...
* user\_name \
* 155 anurag jain
* 1237 Amit Sood,Tarun Mohan,Shravani Raj
* 145 ABHAY SINGH,kapil,Amazon Customer,M.V.SUBBA RA...
* 22 Rahman Ali,MARIYA DASS,Md Aftab,roshan s.,Moha...
* 152 Rahman Ali,MARIYA DASS,Md Aftab,roshan s.,Moha...
* 723 Rahman Ali,MARIYA DASS,Md Aftab,roshan s.,Moha...
* 1463 Shubham Dubey,E.GURUBARAN,Mayank S.,eusuf khan...
* 1198 Mahenddhra,Pavan Sisode,Archana Sekhar,chandan...
* 738 Jasmeen,Santam paul,Deepak,Perabathula Sriniva...
* 1367 Amazon Customer,kapil kumar sharma,Shilpi,Sant...
* review\_id \
* 155 R38OAD16RVS9D4
* 1237 R2TWO1XR7BGSHO,R1683BA4KIYFUI,R2BTLKVDN71QOW
* 145 R2RV2M8NMHN3R6,R39R9NAW42YGZ7,R1P3SC4CEA50V1,R...
* 22 R1SN0D4DFBKAZI,R1SX5L77L2CD6V,R1NAZ6M4QBUJMK,R...
* 152 R1SN0D4DFBKAZI,R1SX5L77L2CD6V,R1NAZ6M4QBUJMK,R...
* 723 R1SN0D4DFBKAZI,R1SX5L77L2CD6V,R1NAZ6M4QBUJMK,R...
* 1463 R39Q2Y79MM9SWK,R3079BG1NIH6MB,R29A31ZELTZNJM,R...
* 1198 R19X0TLJFOL8RV,R3H2XBOSPH6NZR,R187CEHOWSXVIR,R...
* 738 R1GJXMBEY4O49A,R2RJ4QKYQ0VWIL,R2C6XBMID12B8B,R...
* 1367 R205BUIEOZSB27,R3KAOEMO5MHN5A,R1DD7V7FUTYL3H,R...
* review\_title \
* 155 do not buy
* 1237 Good product and budget price,I purchased this...
* 145 Good product,Not bad,WORKING WITH Airtel DTH,G...
* 22 Good,Sound is very low another brand comparing...
* 152 Good,Sound is very low another brand comparing...
* 723 Good,Sound is very low another brand comparing...
* 1463 Fan Speed is slow,Good quality,Good product,go...
* 1198 little bit good,Not Bad,sleek,good,Good produc...
* 738 Nice but price should be reduced,WORTH FOR MON...
* 1367 Poor product,Not working Properly,Average,Nice...
* review\_content \
* 155 tv on off not working, so difficult to battery...
* 1237 Like and happy,,Please don't buy this heater, ...
* 145 Value of money,Usually gd,Good Product,Good,Q...
* 22 Overall good.,TV picture ok smart betterSound ...
* 152 Overall good.,TV picture ok smart betterSound ...
* 723 Overall good.,TV picture ok smart betterSound ...
* 1463 I have installed this in my kitchen working fi...
* 1198 not a eassy to bare this product , in this pro...
* 738 Price is high,https://m.media-amazon.com/image...
* 1367 This product of yours company is heating 5 lit...
* img\_link \
* 155 https://m.media-amazon.com/images/W/WEBP\_40237...
* 1237 https://m.media-amazon.com/images/I/41EQwIB-rK...
* 145 https://m.media-amazon.com/images/W/WEBP\_40237...
* 22 https://m.media-amazon.com/images/W/WEBP\_40237...
* 152 https://m.media-amazon.com/images/W/WEBP\_40237...
* 723 https://m.media-amazon.com/images/I/51q3+E64az...
* 1463 https://m.media-amazon.com/images/W/WEBP\_40237...
* 1198 https://m.media-amazon.com/images/W/WEBP\_40237...
* 738 https://m.media-amazon.com/images/W/WEBP\_40237...
* 1367 https://m.media-amazon.com/images/W/WEBP\_40237...
* product\_link \
* 155 https://www.amazon.in/7SEVEN%C2%AE-Compatible-...
* 1237 https://www.amazon.in/Havells-Quartz-Settings-...
* 145 https://www.amazon.in/OXYURA-Airtel-Digital-Re...
* 22 https://www.amazon.in/Samsung-Inches-Wondertai...
* 152 https://www.amazon.in/Samsung-inches-Wondertai...
* 723 https://www.amazon.in/Samsung-Inches-Wondertai...
* 1463 https://www.amazon.in/Havells-Ventilair-230mm-...
* 1198 https://www.amazon.in/CROMPTON-Sapphira-Ultra-...
* 738 https://www.amazon.in/Callas-Multipurpose-Brea...
* 1367 https://www.amazon.in/ESN-999-Quality-Immersio...
* keywords sentiment
* 155 [compatible, for, tata, sky, remote, original,... -0.600000
* 1237 [havells, bero, quartz, heater, black, heat, s... -0.225000
* 145 [airtel, digitaltv, dth, remote, recording, co... -0.179762
* 22 [samsung, cm, wondertainment, series, hd, read... -0.172245
* 152 [samsung, cm, wondertainment, series, hd, read... -0.172245
* 723 [samsung, cm, wondertainment, series, hd, read... -0.172245
* 1463 [havells, ventil, air, dsp, exhaust, fan] -0.170167
* 1198 [crompton, sea, sapphira, mm, ultra, high, spe... -0.122500
* 738 [callas, multipurpose, foldable, laptop, table... -0.118750
* 1367 [esn, supreme, quality, immersion, water, heat... -0.112500
* ***Answer 7:***
* ***The overall sentiment scores are relatively low, suggesting a tendency towards neutral or slightly negative reviews in the sample.***
* ***The review with the highest sentiment score is "I have installed this in my kitchen working fine" (product\_id 1463) with a score of -0.170167, indicating a mildly positive sentiment.***
* ***The review with the lowest sentiment score is "tv on off not working, so difficult to battery charge" (product\_id 155) with a score of -0.600000, suggesting a strongly negative sentiment.***
* ***Several reviews mention issues with battery charging (product\_id 155), product quality (product\_id 1237), and ease of use (product\_id 1198), highlighting potential areas for improvement.***
* ***Some reviews express both positive and negative aspects within the same text, like "Like and happy,,Please don't buy this heater" (product\_id 1237), suggesting a nuanced evaluation of the product.***
* ***The user\_id column seems to contain commas, indicating multiple user IDs for some reviews. This might need investigation to ensure accuracy.***
* ***Reviews for product\_id 22, 152, and 723 have identical content, suggesting potential data duplication or errors.***
* ***Q8: What is the correlation between discounted\_price and rating?***
* *# Calculate the correlation coefficient*
* correlation\_coefficient = df["discounted\_price"].corr(df["rating"])
* *# Print the correlation coefficient with two decimal places*
* print(f"Correlation between discounted\_price and rating: **{**correlation\_coefficient**:**.2f**}**")
* Correlation between discounted\_price and rating: 0.12
* ***Answer 8:***
* ***Discounted price and rating have a weak positive correlation. This means that products with higher discounted prices tend to have slightly higher ratings, but the relationship is not very strong.***
* ***Q9: What are the Top 5 categories based with highest ratings?***
* *# Group data by category and calculate average rating*
* average\_ratings = df.groupby("category")["rating"].mean().reset\_index()
* *# Sort by average rating in descending order*
* average\_ratings = average\_ratings.sort\_values(by="rating", ascending=False)
* *# Print the top 5 categories*
* print("Top 5 categories with highest average ratings:")
* for i **in** range(5):
* category = average\_ratings.iloc[i]["category"]
* average\_rating = average\_ratings.iloc[i]["rating"]
* print(f"**{**i+1**}**. **{**category**}**: **{**average\_rating**:**.2f**}**")
* Top 5 categories with highest average ratings:
  + Computers&Accessories|Tablets: 4.60
  + Computers&Accessories|NetworkingDevices|NetworkAdapters|PowerLANAdapters: 4.50
  + Electronics|Cameras&Photography|Accessories|Film: 4.50
  + Electronics|HomeAudio|MediaStreamingDevices|StreamingClients: 4.50
  + OfficeProducts|OfficeElectronics|Calculators|Basic: 4.50
* ***Answer 9:***
* ***The top 5 categories have average ratings between 4.50 and 4.60, indicating overall positive customer satisfaction within these areas.***
* ***Most of the top-rated categories fall within technology-related domains, including tablets, networking devices, photography accessories, media streaming devices, and calculators.***
* ***Within broader categories like "Computers & Accessories" and "Electronics," specific subcategories emerge as particularly well-rated, such as tablets, powerline adapters, film accessories, and streaming clients.***
* ***Four categories share a rating of 4.50, suggesting similar levels of customer satisfaction across these areas.***
* ***The presence of "Basic Calculators" in the top 5 suggests that even relatively simple products can achieve high ratings if they meet customer needs effectively.***
* Summary
* ***Our insightful exploration of the Amazon Sales dataset, characterized by its remarkable cleanliness and consistency, yielded a treasure trove of findings. Through a series of targeted inquiries, we unlocked detailed answers and shed light on previously veiled aspects of the data and findings as follows:***
* ***Q1: What is the average rating for each product category?***
* ***Answer 1:***
* ***The output shows that most product categories have generally positive customer feedback, with average ratings above 3.50. However, some categories (e.g., 2 and 3) have lower ratings, suggesting potential areas for improvement. Further analysis of these categories could help identify specific reasons for lower feedback and identify potential solutions.***
* ***Q2: What are the top rating\_count products by category?***
* ***Answer 2:***
  + - ***The output highlights products likely to be popular within their categories based on high review counts, suggesting customer interest and engagement.***
    - ***Review counts range from 9 to 15867, implying varying levels of attention and feedback across products.***
    - ***Most listed products have ratings above 3.5, indicating a generally positive customer experience.***
    - ***Products with the highest review counts within their categories might be considered potential top sellers, even without direct sales data.***
* ***Q3: What is the distribution of discounted prices vs. actual prices?***
* ***Answer 3:***
  + - ***The output shows that discounted prices are generally lower than actual prices, with a median discounted price of $200 and a median actual price of $400.***
    - ***The discount percentage distribution is skewed to the left, with most products having a discount of 30% or less.***
    - ***The output suggests that there may be opportunities to increase discounted prices or discount percentages to attract more customers.***
* ***Q4: How does the average discount percentage vary across categories?***
* ***Answer 4:***
  + - ***Average discount percentages vary widely across categories, ranging from 0% to 78.39%.***
    - ***Categories 1 and 3 stand out with notably higher average discounts (78.39% and 56.34%), suggesting potential factors like clearance efforts, high competition, or lower-profit margins.***
    - ***Categories 0, 206, 207, 210 have average discounts of 0%, indicating consistent pricing or strong demand for products within those categories.***
    - ***Other categories exhibit varying discount percentages, likely reflecting diverse pricing strategies and market dynamics.***
* ***Q5: What are the most popular product name?***
* ***Answer 5:***
  + - ***Fire-Boltt Ninja Call Pro Plus Smart Watch is the most popular product, followed by Fire-Boltt Phoenix Smart Watch.***
    - ***Smart Watches and Charging Cables are the most popular product categories.***
    - ***Multiple brands are represented, with boAt appearing twice.***
    - ***Fast charging, durability, and functionality are key features.***
    - ***Popularity is relatively evenly distributed beyond the leading product.***
* ***Q6: What are the most popular product keywords?***
* ***Answer 6:***
  + - ***USB connectivity, charging (especially fast charging), and cables are prominent product features.***
    - ***Prepositions and conjunctions like "with", "for", "and", "to" suggest a focus on explaining product compatibility and usage scenarios.***
    - ***Cables and smart devices are likely well-represented in the dataset.***
    - ***Product names tend to be concise and use common words, potentially benefiting from refined keyword extraction techniques.***
* ***Q7: What are the most popular product reviews?***
* ***Answer 7:***
  + - ***The overall sentiment scores are relatively low, suggesting a tendency towards neutral or slightly negative reviews in the sample.***
    - ***The review with the highest sentiment score is "I have installed this in my kitchen working fine" (product\_id 1463) with a score of -0.170167, indicating a mildly positive sentiment.***
    - ***The review with the lowest sentiment score is "tv on off not working, so difficult to battery charge" (product\_id 155) with a score of -0.600000, suggesting a strongly negative sentiment.***
    - ***Several reviews mention issues with battery charging (product\_id 155), product quality (product\_id 1237), and ease of use (product\_id 1198), highlighting potential areas for improvement.***
    - ***Some reviews express both positive and negative aspects within the same text, like "Like and happy,,Please don't buy this heater" (product\_id 1237), suggesting a nuanced evaluation of the product.***
    - ***The user\_id column seems to contain commas, indicating multiple user IDs for some reviews. This might need investigation to ensure accuracy.***
    - ***Reviews for product\_id 22, 152, and 723 have identical content, suggesting potential data duplication or errors.***
* ***Q8: What is the correlation between discounted\_price and rating?***
* ***Answer 8:***
* ***Discounted price and rating have a weak positive correlation. This means that products with higher discounted prices tend to have slightly higher ratings, but the relationship is not very strong.***
* ***Q9: What are the Top 5 categories based with highest ratings?***
* ***Answer 9:***
  + - ***The top 5 categories have average ratings between 4.50 and 4.60, indicating overall positive customer satisfaction within these areas.***
    - ***Most of the top-rated categories fall within technology-related domains, including tablets, networking devices, photography accessories, media streaming devices, and calculators.***
    - ***Within broader categories like "Computers & Accessories" and "Electronics," specific subcategories emerge as particularly well-rated, such as tablets, powerline adapters, film accessories, and streaming clients.***
    - ***Four categories share a rating of 4.50, suggesting similar levels of customer satisfaction across these areas.***
    - ***The presence of "Basic Calculators" in the top 5 suggests that even relatively simple products can achieve high ratings if they meet customer needs effectively.***
* Summary
* ***Q1: What is the average rating for each product category?***
* ***Answer 1:***
* ***The output shows that most product categories have generally positive customer feedback, with average ratings above 3.50. However, some categories (e.g., 2 and 3) have lower ratings, suggesting potential areas for improvement. Further analysis of these categories could help identify specific reasons for lower feedback and identify potential solutions.***
* ***Q2: What are the top rating\_count products by category?***
* ***Answer 2:***
  + - ***The output highlights products likely to be popular within their categories based on high review counts, suggesting customer interest and engagement.***
    - ***Review counts range from 9 to 15867, implying varying levels of attention and feedback across products.***
    - ***Most listed products have ratings above 3.5, indicating a generally positive customer experience.***
    - ***Products with the highest review counts within their categories might be considered potential top sellers, even without direct sales data.***
* ***Q3: What is the distribution of discounted prices vs. actual prices?***
* ***Answer 3:***
  + - ***The output shows that discounted prices are generally lower than actual prices, with a median discounted price of $200 and a median actual price of $400.***
    - ***The discount percentage distribution is skewed to the left, with most products having a discount of 30% or less.***
    - ***The output suggests that there may be opportunities to increase discounted prices or discount percentages to attract more customers.***
* ***Q4: How does the average discount percentage vary across categories?***
* ***Answer 4:***
  + - ***Average discount percentages vary widely across categories, ranging from 0% to 78.39%.***
    - ***Categories 1 and 3 stand out with notably higher average discounts (78.39% and 56.34%), suggesting potential factors like clearance efforts, high competition, or lower-profit margins.***
    - ***Categories 0, 206, 207, 210 have average discounts of 0%, indicating consistent pricing or strong demand for products within those categories.***
    - ***Other categories exhibit varying discount percentages, likely reflecting diverse pricing strategies and market dynamics.***
* ***Q5: What are the most popular product name?***
* ***Answer 5:***
  + - ***Fire-Boltt Ninja Call Pro Plus Smart Watch is the most popular product, followed by Fire-Boltt Phoenix Smart Watch.***
    - ***Smart Watches and Charging Cables are the most popular product categories.***
    - ***Multiple brands are represented, with boAt appearing twice.***
    - ***Fast charging, durability, and functionality are key features.***
    - ***Popularity is relatively evenly distributed beyond the leading product.***
* ***Q6: What are the most popular product keywords?***
* ***Answer 6:***
  + - ***USB connectivity, charging (especially fast charging), and cables are prominent product features.***
    - ***Prepositions and conjunctions like "with", "for", "and", "to" suggest a focus on explaining product compatibility and usage scenarios.***
    - ***Cables and smart devices are likely well-represented in the dataset.***
    - ***Product names tend to be concise and use common words, potentially benefiting from refined keyword extraction techniques.***
* ***Q7: What are the most popular product reviews?***
* ***Answer 7:***
  + - ***The overall sentiment scores are relatively low, suggesting a tendency towards neutral or slightly negative reviews in the sample.***
    - ***The review with the highest sentiment score is "I have installed this in my kitchen working fine" (product\_id 1463) with a score of -0.170167, indicating a mildly positive sentiment.***
    - ***The review with the lowest sentiment score is "tv on off not working, so difficult to battery charge" (product\_id 155) with a score of -0.600000, suggesting a strongly negative sentiment.***
    - ***Several reviews mention issues with battery charging (product\_id 155), product quality (product\_id 1237), and ease of use (product\_id 1198), highlighting potential areas for improvement.***
    - ***Some reviews express both positive and negative aspects within the same text, like "Like and happy,,Please don't buy this heater" (product\_id 1237), suggesting a nuanced evaluation of the product.***
    - ***The user\_id column seems to contain commas, indicating multiple user IDs for some reviews. This might need investigation to ensure accuracy.***
    - ***Reviews for product\_id 22, 152, and 723 have identical content, suggesting potential data duplication or errors.***
* ***Q8: What is the correlation between discounted\_price and rating?***
* ***Answer 8:***
* ***Discounted price and rating have a weak positive correlation. This means that products with higher discounted prices tend to have slightly higher ratings, but the relationship is not very strong.***
* ***Q9: What are the Top 5 categories based with highest ratings?***
* ***Answer 9:***
  + - ***The top 5 categories have average ratings between 4.50 and 4.60, indicating overall positive customer satisfaction within these areas.***
    - ***Most of the top-rated categories fall within technology-related domains, including tablets, networking devices, photography accessories, media streaming devices, and calculators.***
    - ***Within broader categories like "Computers & Accessories" and "Electronics," specific subcategories emerge as particularly well-rated, such as tablets, powerline adapters, film accessories, and streaming clients.***
    - ***Four categories share a rating of 4.50, suggesting similar levels of customer satisfaction across these areas.***
    - ***The presence of "Basic Calculators" in the top 5 suggests that even relatively simple products can achieve high ratings if they meet customer needs effectively***
* Conclusion
* [¶](https://www.kaggle.com/code/mehakiftikhar/amazon-sales-dataset-eda/notebook#-Conclusion)
* ***The primary goal of this project is to Analyz the Amazon Sales dataset and identify insights based on the data. The Amazon Sales dataset is a valuable resource for businesses and researchers alike. It provides a wealth of information about customer behavior product trends, and market conditions. By conducting exploratory data analysis (EDA) on this dataset, businesses can gain valuable insights that can help them make better decisions about their products, marketing, and operations***