UNIVERSITY INSTITUTE *of*
COMPUTING
Asia's Fastest Growing University

Case Study of Data Mining

Code-24CAT-611

Data Cleaning in E-Commerce Customer Analytics

Submitted to:
Dr.Arun kumar
Professor
Employee ID: E17568

Submitted by:
Pragati Gupta
UID:24MCI10255

Problem statement

An e-commerce company, XYZ, aims to enhance its customer experience and improve sales through data-driven insights. The company collects vast amounts of data from customer transactions, website interactions, and feedback surveys. However, as the data grows, inconsistencies, duplicates, and errors have emerged, leading to unreliable analytics.

Take dataset from any source, clean it by applying different data cleaning techniques whatever required into that along with the explanation.

GitHub Link for this Case study:

Rubrics for evaluation:

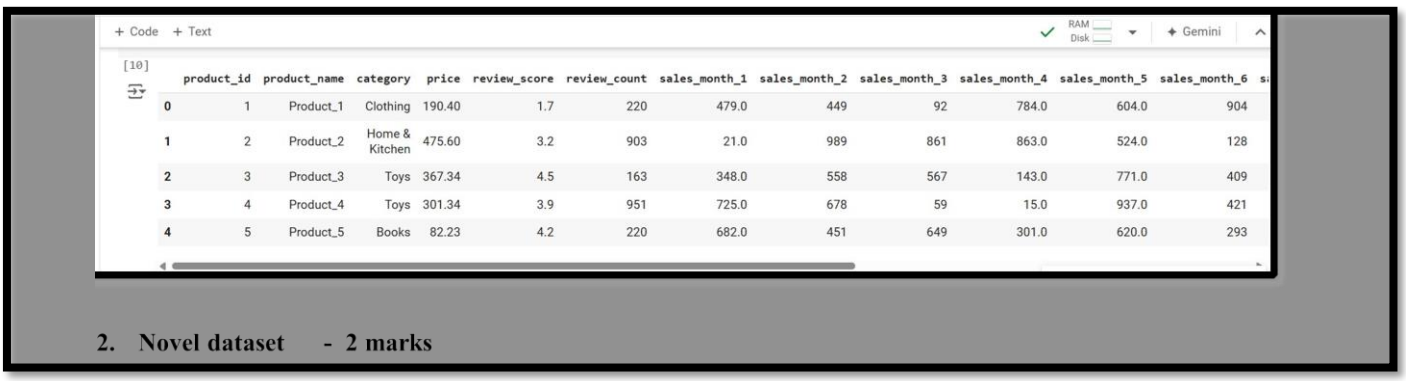
1. **Relevant dataset _ 2 marks**

```
+ Code  + Text

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# 1. Relevant dataset:
df=pd.read_csv('/content/ecommerce_analysis.csv')
display(df.head())

product_id = df['product_id']
product_name = df['product_name']
category = df['category']
price = df['price']
review_score = df['review_score']
review_count = df['review_count']
sales_month_1 = df['sales_month_1']
sales_month_2 = df['sales_month_2']
sales_month_3 = df['sales_month_3']
sales_month_4 = df['sales_month_4']
sales_month_5 = df['sales_month_5']
sales_month_6 = df['sales_month_6']
sales_month_7 = df['sales_month_7']
sales_month_8 = df['sales_month_8']
sales_month_9 = df['sales_month_9']
sales_month_10 = df['sales_month_10']
sales_month_11 = df['sales_month_11']
sales_month_12 = df['sales_month_12']
```



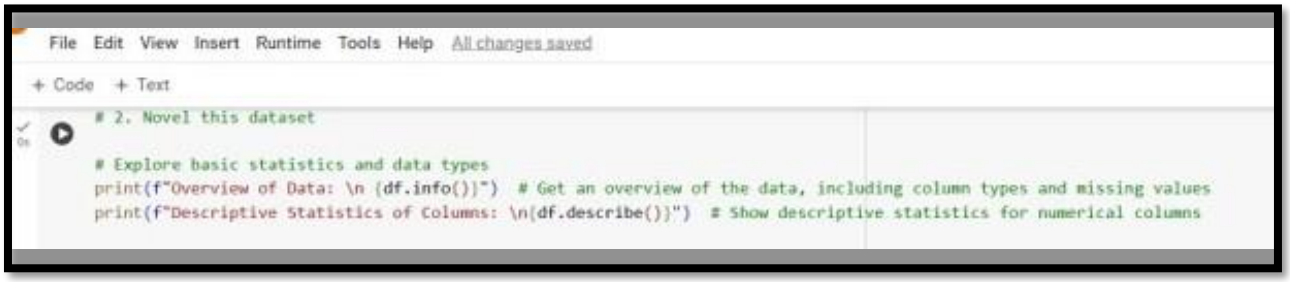
+ Code + Text — RAM Disk — ✦ Gemini

[10]

	product_id	product_name	category	price	review_score	review_count	sales_month_1	sales_month_2	sales_month_3	sales_month_4	sales_month_5	sales_month_6	s
0	1	Product_1	Clothing	190.40	1.7	220	479.0	449	92	784.0	604.0	904	
1	2	Product_2	Home & Kitchen	475.60	3.2	903	21.0	989	861	863.0	524.0	128	
2	3	Product_3	Toys	367.34	4.5	163	348.0	558	567	143.0	771.0	409	
3	4	Product_4	Toys	301.34	3.9	951	725.0	678	59	15.0	937.0	421	
4	5	Product_5	Books	82.23	4.2	220	682.0	451	649	301.0	620.0	293	

2. **Novel dataset - 2 marks**

2. **Novel dataset - 2 marks**


File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
# 2. Novel this dataset
# Explore basic statistics and data types
print(f"Overview of Data: \n {df.info()}") # Get an overview of the data, including column types and missing values
print(f"Descriptive Statistics of Columns: \n{df.describe()}") # Show descriptive statistics for numerical columns


File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 18 columns):
Column Non-Null Count Dtype
0 product_id 1000 non-null int64
1 product_name 1000 non-null object
2 category 1000 non-null object
3 price 1000 non-null float64
4 review_score 1000 non-null float64
5 review_count 1000 non-null int64
6 sales_month_1 1000 non-null float64
7 sales_month_2 1000 non-null int64
8 sales_month_3 1000 non-null int64
9 sales_month_4 999 non-null float64
10 sales_month_5 999 non-null float64
11 sales_month_6 1000 non-null int64
12 sales_month_7 1000 non-null int64
13 sales_month_8 999 non-null float64
14 sales_month_9 1000 non-null int64
15 sales_month_10 1000 non-null int64
16 sales_month_11 1000 non-null int64
17 sales_month_12 1000 non-null int64
dtypes: float64(6), int64(10), object(2)
memory usage: 140.8+ KB
Overview of Data:
None

File Edit View Insert Runtime Tools Help All changes saved

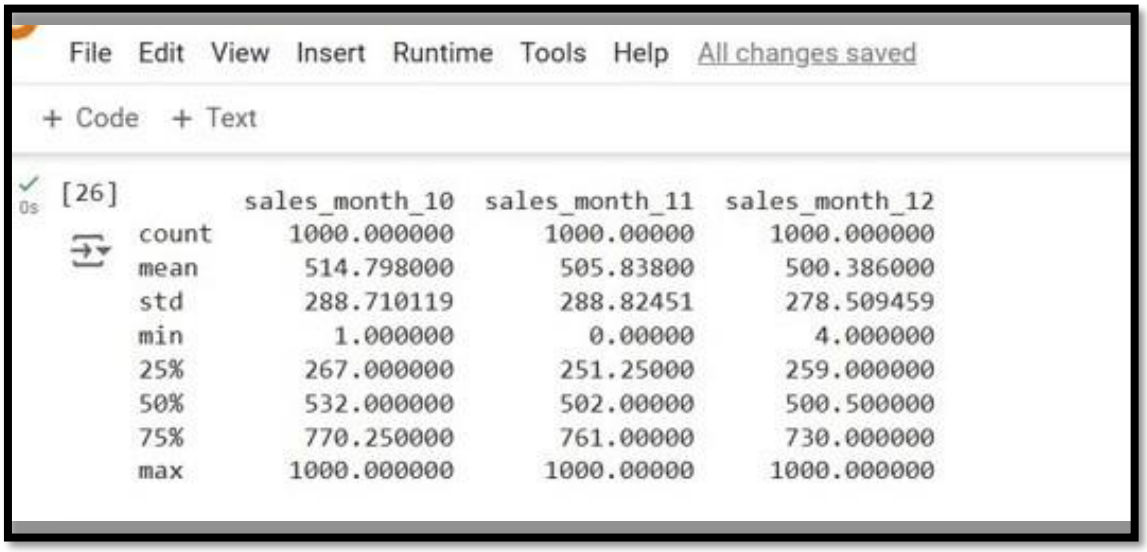
+ Code + Text

Descriptive Statistics of Columns:

	product_id	price	review_score	review_count	sales_month_1
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	500.500000	247.677130	3.027600	526.506000	498.814000
std	288.819436	144.607983	1.171243	282.269932	289.512243
min	1.000000	7.290000	1.000000	1.000000	0.000000
25%	250.750000	121.810000	2.000000	283.750000	246.000000
50%	500.500000	250.920000	3.100000	543.000000	508.000000
75%	750.250000	373.435000	4.000000	772.000000	740.750000
max	1000.000000	499.860000	5.000000	999.000000	1000.000000

	sales_month_2	sales_month_3	sales_month_4	sales_month_5
count	1000.000000	1000.000000	999.000000	999.000000
mean	507.661000	506.739000	504.324324	487.674675
std	285.992689	294.010873	286.350194	287.586675
min	2.000000	0.000000	0.000000	0.000000
25%	262.500000	243.750000	263.000000	222.000000
50%	508.000000	493.000000	502.000000	497.000000
75%	756.250000	777.250000	750.000000	727.000000
max	1000.000000	999.000000	1000.000000	1000.000000

	sales_month_6	sales_month_7	sales_month_8	sales_month_9
count	1000.000000	1000.000000	999.000000	1000.000000
mean	491.653000	507.011000	505.022022	491.934000
std	289.234018	291.047287	289.736615	287.514731
min	0.000000	0.000000	5.000000	0.000000
25%	236.000000	254.000000	241.500000	247.250000
50%	479.500000	522.500000	500.000000	495.500000
75%	740.500000	757.250000	762.500000	735.250000
max	1000.000000	1000.000000	1000.000000	1000.000000



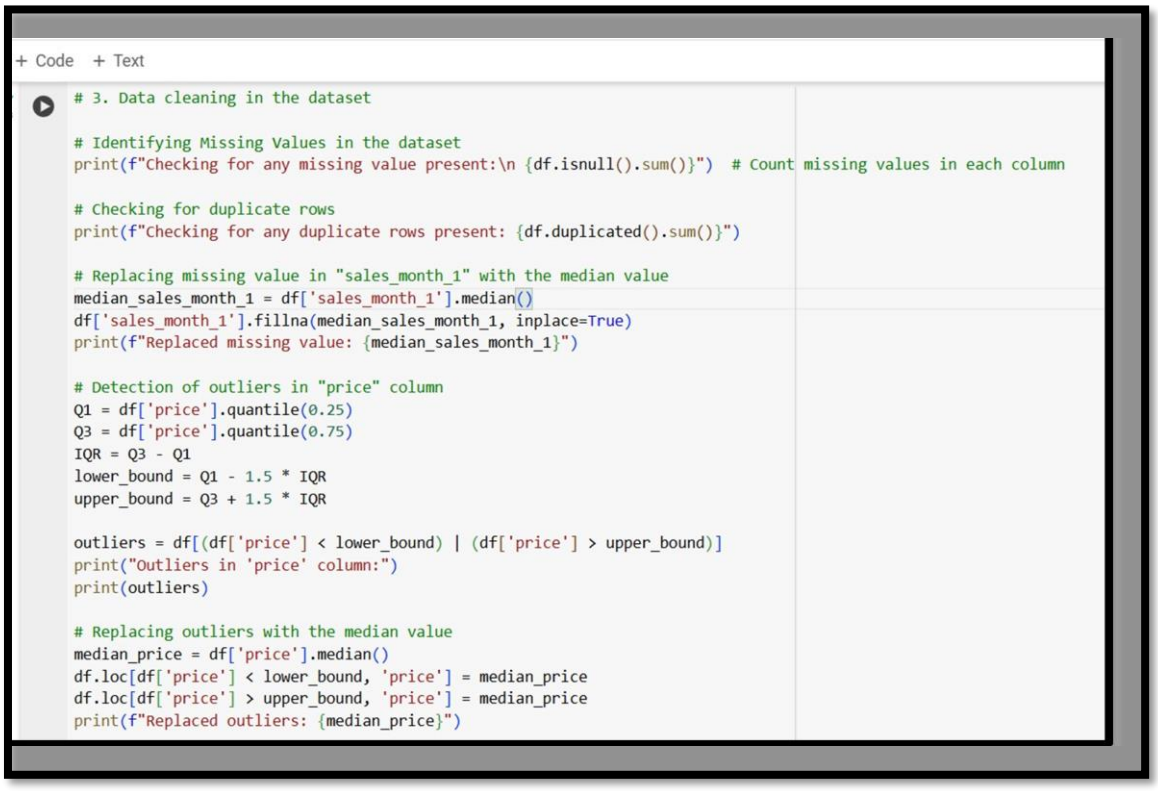
	sales_month_10	sales_month_11	sales_month_12
count	1000.000000	1000.00000	1000.000000
mean	514.798000	505.83800	500.386000
std	288.710119	288.82451	278.509459
min	1.000000	0.00000	4.000000
25%	267.000000	251.25000	259.000000
50%	532.000000	502.00000	500.500000
75%	770.250000	761.00000	730.000000
max	1000.000000	1000.00000	1000.000000

3. **Data cleaning techniques along with the relevant explanation that why these techniques are being applied. -- 3 marks**
   a) **Identifying missing values:** The complete Dataset is checked if there is the presence of any null value or not.
   b) **Checking for the duplicate rows:** The dataset is checked for any kind of duplicate row present in the dataset.

c) **Replaced Missing values with a central tendency, i.e., Median:** One of the suitable method to handle Missing value is to replace it with the central tendencies like mean, median, mode, or standard deviation.

d) **Detecting the outliers:** Outliers refers to value out of range with respect to the dataset available. We have detected outliers based on their IQR (Inter quartile Range).

e) **Replacing the outliers with Median:** Outliers detected by the IQR are replaced with the central tendency in order to clean the dataset.



4. Relevant graphs and tables depicting clean data –3 marks

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
# 4.Relevant graphs and tables depicting clean data

# Table showing basic statistics of the cleaned data
print(f"Table of cleaned data:\n {df.describe()}")

# a) Distribution of Product Prices
plt.figure(figsize=(8, 6))
sns.histplot(df['price'], bins=20, kde=True, color='y')
plt.title('Distribution of Product Prices')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.show()


# b) Relationship between Price and Review Score
plt.figure(figsize=(8, 6))
sns.scatterplot(x='price', y='review_score', data=df,c=price, cmap='viridis')
plt.title('Relationship between Price and Review Score')
plt.xlabel('Price')
plt.ylabel('Review Score')
plt.show()


# c) Top 10 Categories by Sales in Month 1
top_10_categories_month_1 = df.groupby('category')['sales_month_1'].sum().nlargest(10)
plt.figure(figsize=(10, 6))
sns.barplot(x=top_10_categories_month_1.values, y=top_10_categories_month_1.index, color='red')
plt.title('Top 10 Categories by Sales in Month 1')
plt.xlabel('Total Sales')
plt.ylabel('Category')
plt.show()
```

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[25] # d) Average Review Score by Category
     average_review_score_by_category = df.groupby('category')['review_score'].mean()
     plt.figure(figsize=(10, 6))
     sns.barplot(x=average_review_score_by_category.values, y=average_review_score_by_category.index, color='k')
     plt.title('Average Review Score by Category')
     plt.xlabel('Average Review Score')
     plt.ylabel('Category')
     plt.show()


     # e) Correlation Matrix
     correlation_matrix = df[['price', 'review_score', 'review_count', 'sales_month_1']].corr()
     plt.figure(figsize=(8, 6))
     sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
     plt.title('Correlation Matrix')
     plt.show()


     # f) Boxplot of Sales in Month 1 by Category
     plt.figure(figsize=(10, 6))
     sns.boxplot(x='category', y='sales_month_1', data=df, color='cyan')
     plt.title('Boxplot of Sales in Month 1 by Category')
     plt.xlabel('Category')
     plt.ylabel('Sales in Month 1')
     plt.xticks(rotation=90)
     plt.show()
```

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Table of cleaned data:

	product_id	price	review_score	review_count	sales_month_1	\
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	
mean	500.500000	247.677130	3.027600	526.506000	498.814000	
std	288.819436	144.607983	1.171243	282.269932	289.512243	
min	1.000000	7.290000	1.000000	1.000000	0.000000	
25%	250.750000	121.810000	2.000000	283.750000	246.000000	
50%	500.500000	250.920000	3.100000	543.000000	508.000000	
75%	750.250000	373.435000	4.000000	772.000000	740.750000	
max	1000.000000	499.860000	5.000000	999.000000	1000.000000	

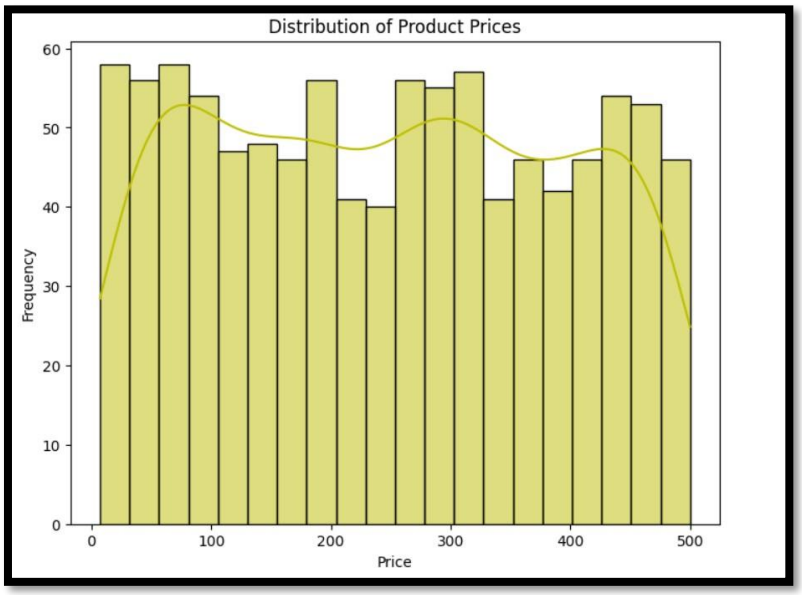
	sales_month_2	sales_month_3	sales_month_4	sales_month_5	\
count	1000.000000	1000.000000	999.000000	999.000000	
mean	507.661000	506.739000	504.324324	487.674675	
std	285.992689	294.010873	286.350194	287.586675	
min	2.000000	0.000000	0.000000	0.000000	
25%	262.500000	243.750000	263.000000	222.000000	
50%	508.000000	493.000000	502.000000	497.000000	
75%	756.250000	777.250000	750.000000	727.000000	
max	1000.000000	999.000000	1000.000000	1000.000000	

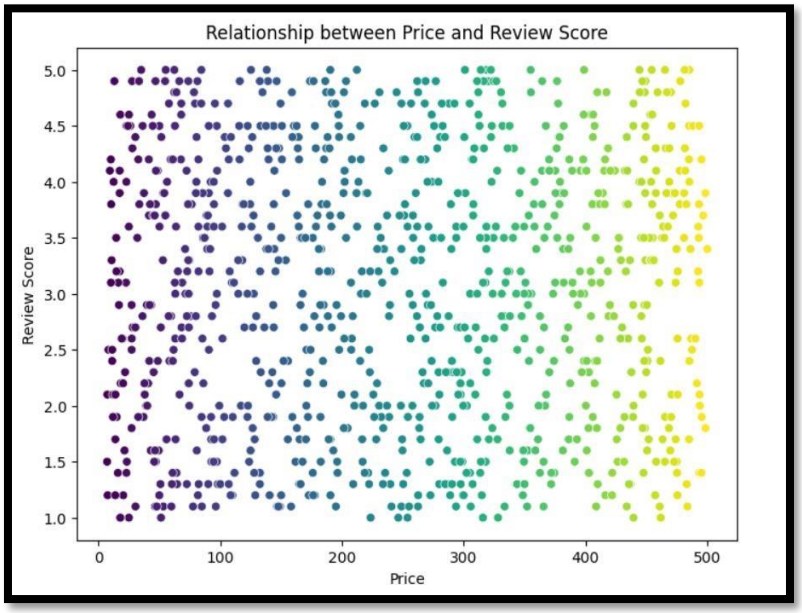
	sales_month_6	sales_month_7	sales_month_8	sales_month_9	\
count	1000.000000	1000.000000	999.000000	1000.000000	
mean	491.653000	507.011000	505.022022	491.934000	
std	289.234018	291.047287	289.736615	287.514731	
min	0.000000	0.000000	5.000000	0.000000	
25%	236.000000	254.000000	241.500000	247.250000	
50%	479.500000	522.500000	500.000000	495.500000	
75%	740.500000	757.250000	762.500000	735.250000	
max	1000.000000	1000.000000	1000.000000	1000.000000	

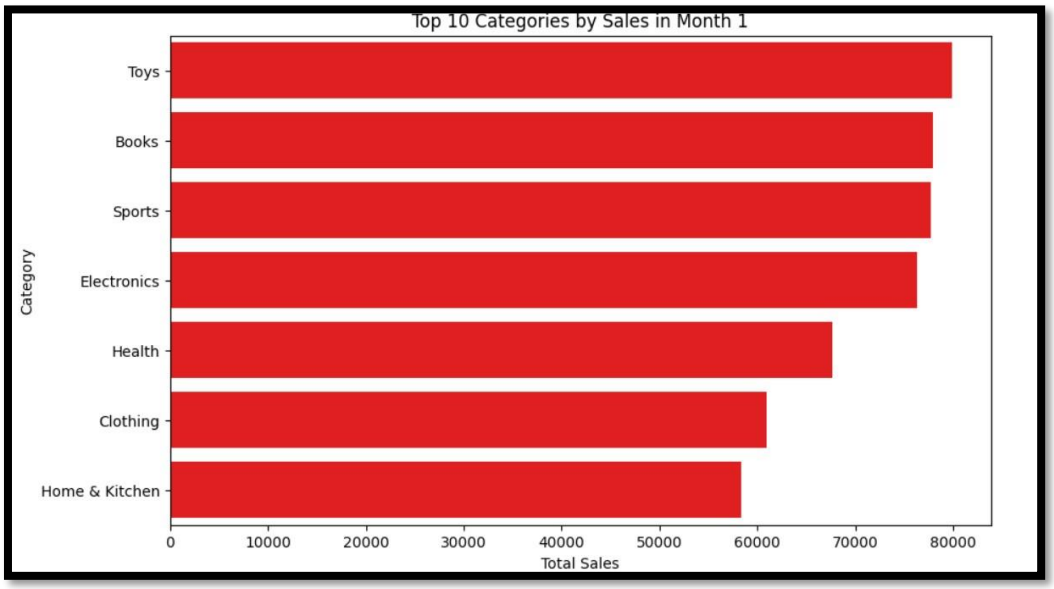
File Edit View Insert Runtime Tools Help All changes saved

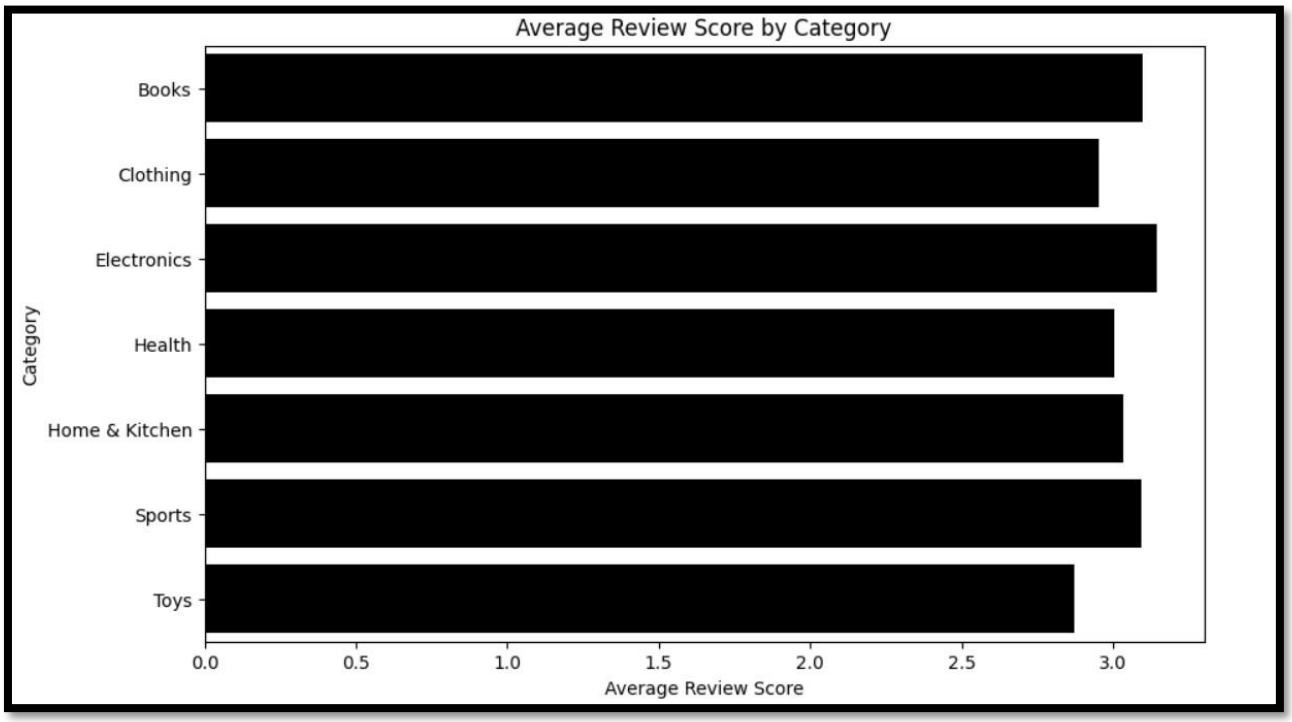
+ Code + Text

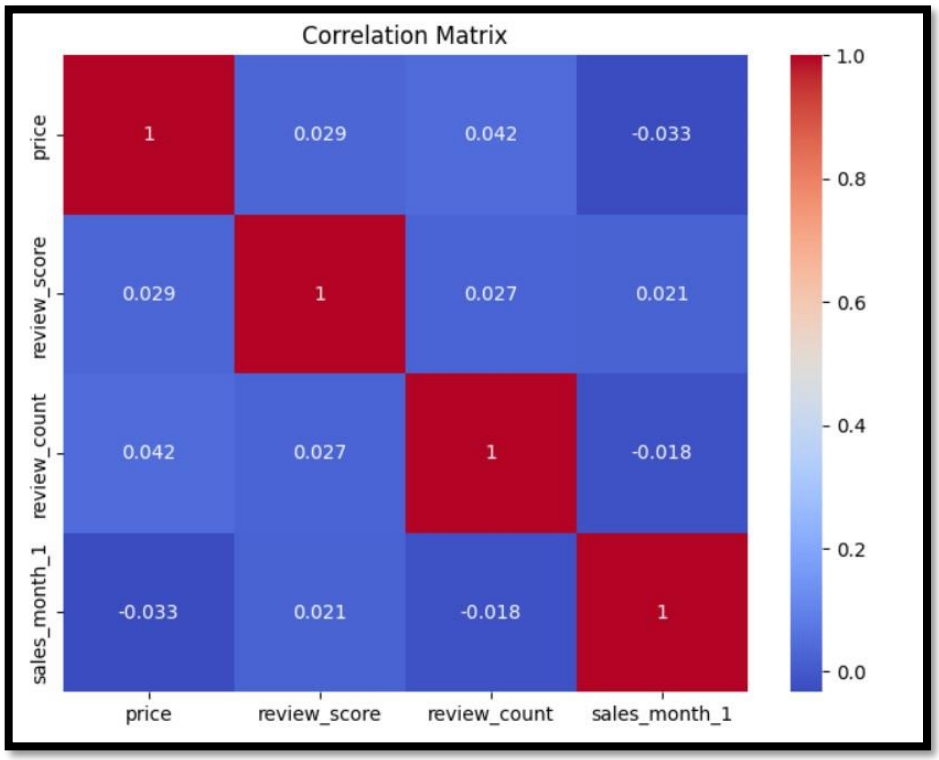
	sales_month_10	sales_month_11	sales_month_12
count	1000.000000	1000.00000	1000.000000
mean	514.798000	505.83800	500.386000
std	288.710119	288.82451	278.509459
min	1.000000	0.00000	4.000000
25%	267.000000	251.25000	259.000000
50%	532.000000	502.00000	500.500000
75%	770.250000	761.00000	730.000000
max	1000.000000	1000.00000	1000.000000

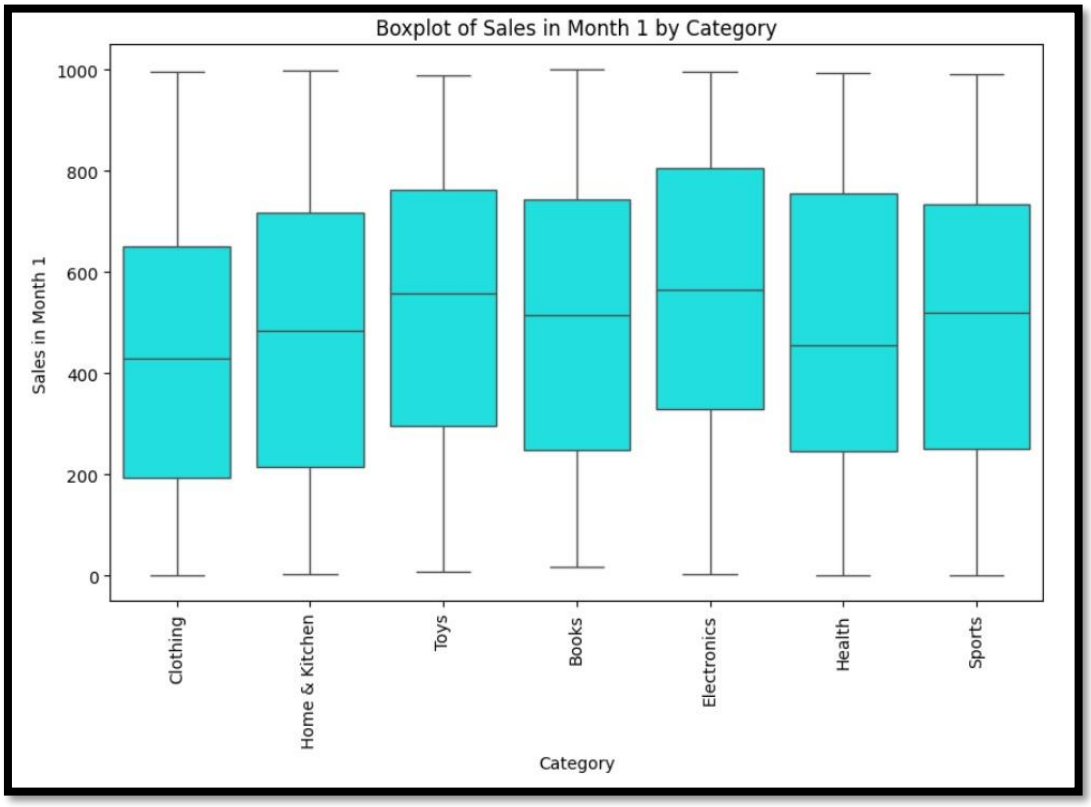





Top 10 Categories by Sales in Month 1
Toys
Books
Sports
Electronics
Health
Clothing
Home & Kitchen
Category
0
10000
20000
30000
40000
50000
60000
70000
80000
Total Sales


Average Review Score by Category
Books
Clothing
Electronics
Health
Home & Kitchen
Sports
Toys
Category
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Average Review Score


Correlation Matrix
price
review_score
review_count
sales_month_1
1
0.029
0.042
-0.033
0.029
1
0.027
0.021
0.042
0.027
1
-0.018
-0.033
0.021
-0.018
1
price
review_score
review_count
sales_month_1
1.0
0.8
0.6
0.4
0.2
0.0


Boxplot of Sales in Month 1 by Category
Sales in Month 1
1000
800
600
400
200
0
Clothing
Home & Kitchen
Toys
Books
Electronics
Health
Sports
Category