E-commerce Sales Analysis

Team Member : (*Dipali Gantait , Sudipta Singha , Riyajul Saha*)

Instructor : *Sourav Goswami* (**Ardent Computech PVT LTD**)

Date : 07.06.2025

- **Project Overview :** This project focuses on analyzing product-level sales data from an ecommerce platform to uncover key trends, patterns, and customer preferences over a one-year period. The dataset includes detailed information about various products, including their category, price, review scores, review counts, and monthly sales for 12 months. The primary objective of this analysis is to gain insights into which product categories perform best, identify sales trends across the year, examine the relationship between customer reviews and sales, and understand pricing impact on performance. We also aim to identify high-performing products and those that may need attention or improvement. We used data analysis techniques and visualizations to explore the dataset, including bar charts, line graphs, and category-wise comparisons. This approach helps in making the sales data more understandable and actionable for ecommerce stakeholders. The insights derived can assist in better inventory planning, marketing strategies, and customer satisfaction improvements.

- **Tools & Technologies:** We performed the data analysis using Google Colab, utilizing the Python programming language. The following Python libraries were used in this project:
  1. Pandas – for data manipulation and analysis
  2. NumPy – for numerical operations
  3. Matplotlib – for creating static visualizations
  4. Seaborn – for advanced and attractive data visualizations

- **Problem Statement :** The ecommerce platform sells a wide variety of products across different categories, but lacks clear insight into which products perform best over time, how customer feedback affects sales, and whether product pricing impacts performance. This project aims to analyze a dataset containing 12 months of product-level sales data, along with product prices, categories, and customer review scores. The key objectives are:
  1. To identify the top-selling products and categories across the year.
  2. To examine month-wise sales trends to detect seasonal sales patterns.
  3. To explore the relationship between customer review scores and total product sales.
  4. To examine all categories products yearly sales difference .

  Through this analysis, we aim to generate useful business insights that can help the ecommerce platform improve product selection, pricing strategy, and customer engagement.

❖ Data Collection and summaries:

We collect ecommerce_sales_analysis.csv from kaggle.

As we have used python for data analysis , so now we import data using pandas library . Before we have kept our data in my drive , so we will load data from drive using pandas library pd.read_csv() .

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sn
import numpy as np
```

```
Ecom = pd.read_csv('/content/drive/My Drive/Dataset/ecommerce_sales_analysis.csv')
```

```
Ecom.head()
```

	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	sales_month_7	sales_month_8
0	1	Product_1	Clothing	190.40	1.7	220	479	449	92	784	604	904	446	603
1	2	Product_2	Home & Kitchen	475.60	3.2	903	21	989	861	863	524	128	610	436
2	3	Product_3	Toys	367.34	4.5	163	348	558	567	143	771	409	290	828
3	4	Product_4	Toys	301.34	3.9	951	725	678	59	15	937	421	670	933
4	5	Product_5	Books	82.23	4.2	220	682	451	649	301	620	293	411	258

Next steps: Generate code with Ecom | View recommended plots | New interactive sheet

Now we will see all information of data that help better understand to us .

```
Ecom.shape
```

```
(1000, 18)
```

```
list[Ecom.columns]
```

```
list[Index(['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', 'sales_month_7',
       'sales_month_8', 'sales_month_9', 'sales_month_10', 'sales_month_11',
       'sales_month_12'],
      dtype='object')]
```

```
Ecom.info()
```

```
<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   int64  
 7   sales_month_2   1000 non-null   int64  
 8   sales_month_3   1000 non-null   int64  
 9   sales_month_4   1000 non-null   int64  
 10  sales_month_5   1000 non-null   int64  
 11  sales_month_6   1000 non-null   int64  
 12  sales_month_7   1000 non-null   int64  
 13  sales_month_8   1000 non-null   int64  
 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(2), int64(14), object(2)
memory usage: 140.8+ KB
```

```
Ecom.describe()
```

	product_id	price	review_score	review_count	sales_month_1	sales_month_2	sales_month_3	sales_month_4	sales_month_5	sales_month_6	sales_month_7	sales_month_8	sales_month_9
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	500.500000	247.677130	3.027600	526.506000	498.306000	507.661000	506.739000	503.823000	487.194000	491.653000	507.011000	504.569000	491.934000
std	288.819436	144.607983	1.171243	282.269932	289.941478	285.992689	294.010873	286.645567	287.844324	289.234018	291.047287	289.945691	287.514731
min	1.000000	7.290000	1.000000	1.000000	0.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	5.000000	0.000000
25%	250.750000	121.810000	2.000000	283.750000	245.500000	262.500000	243.750000	261.500000	221.000000	236.000000	254.000000	240.500000	247.250000
50%	500.500000	250.920000	3.100000	543.000000	507.500000	508.000000	493.000000	501.500000	497.000000	479.500000	522.500000	499.500000	495.500000
75%	750.250000	373.435000	4.000000	772.000000	740.750000	756.250000	777.250000	749.500000	727.000000	740.500000	757.250000	762.250000	735.250000
max	1000.000000	499.860000	5.000000	999.000000	1000.000000	1000.000000	999.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000

Now we identify the top selling products and categories across the year .

```
Ecom["year"]=Ecom.iloc[:,6:].sum(axis=1)
Ecom.head()
```

	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	sales_month_7	sales_month_8
0	1	Product_1	Clothing	190.40	1.7	220	479	449	92	784	604	904	446	603
1	2	Product_2	Home & Kitchen	475.60	3.2	903	21	989	861	863	524	128	610	436
2	3	Product_3	Toys	367.34	4.5	163	348	558	567	143	771	409	290	828
3	4	Product_4	Toys	301.34	3.9	951	725	678	59	15	937	421	670	933
4	5	Product_5	Books	82.23	4.2	220	682	451	649	301	620	293	411	258

Next steps: Generate code with Ecom | View recommended plots | New interactive sheet

```
maximum=Ecom["year"].max()
```

```
Ecom[Ecom["year"]==maximum]["product_name"]#finding maximum selling products name in a year
```

	product_name
223	Product_224

dtype: object

```
#finding maximum selling categories name in a year
total_sale=Ecom.groupby("category")["year"].sum().reset_index()
```

```
total_sale
```

	category	year
0	Books	938229
1	Clothing	826536
2	Electronics	845120
3	Health	834414
4	Home & Kitchen	742141
5	Sports	916371
6	Toys	917101

Next steps: Generate code with total_sale | View recommended plots | New interactive sheet

```
maximum=total_sale["year"].max()
```

```
total_sale[total_sale["year"]==maximum]["category"]
```

	category
0	Books

dtype: object

❖ Data preprocessing and cleaning :

As we seen before in info method that there are no null value in any columns and all the column's has correct data type. But to confirm null value we will call isnull method.

```
Ecom.isnull().sum()
```

	0
product_id	0
product_name	0
category	0
price	0
review_score	0
review_count	0
sales_month_1	0
sales_month_2	0
sales_month_3	0
sales_month_4	0
sales_month_5	0
sales_month_6	0
sales_month_7	0
sales_month_8	0
sales_month_9	0
sales_month_10	0
sales_month_11	0
sales_month_12	0
year	0

dtype: int64

we have confirm that there are no null value in our dataset. So no need to us to clean dataset and call the columns data type has in its on data type, so, we don't need preprocess data . Now we can go to analysis.

❖ Analysis and Visualization:

Now we will try to better data analysis using visualization through graph plotting.

Month-wise sales trends and detect seasonal pattern

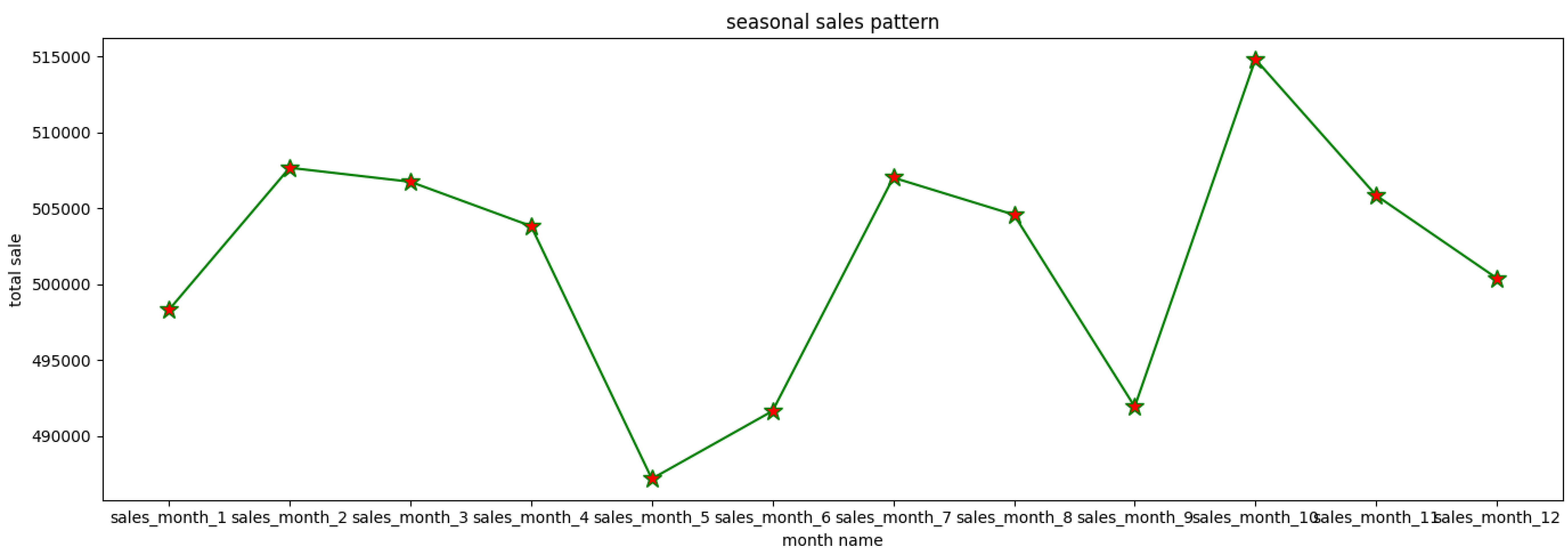
```
mttls=list(Ecom.iloc[:,6:18].sum())
```

```
mname=list(Ecom.columns)[6:18]
mname
```

```
['sales_month_1',
 'sales_month_2',
 'sales_month_3',
 'sales_month_4',
 'sales_month_5',
 'sales_month_6',
 'sales_month_7',
 'sales_month_8',
 'sales_month_9',
 'sales_month_10',
 'sales_month_11',
 'sales_month_12']
```

```
%matplotlib inline
```

```
plt.figure(figsize=(14,5))
plt.plot(mname,mttls,marker="*",markersize=12,markerfacecolor="red",color="green")
plt.title("seasonal sales pattern")
plt.xlabel("month name")
plt.ylabel("total sale")
plt.tight_layout()
```



Now we will see top five categories products using pie chart

```
t5cat=total_sale.head()
t5cat
```

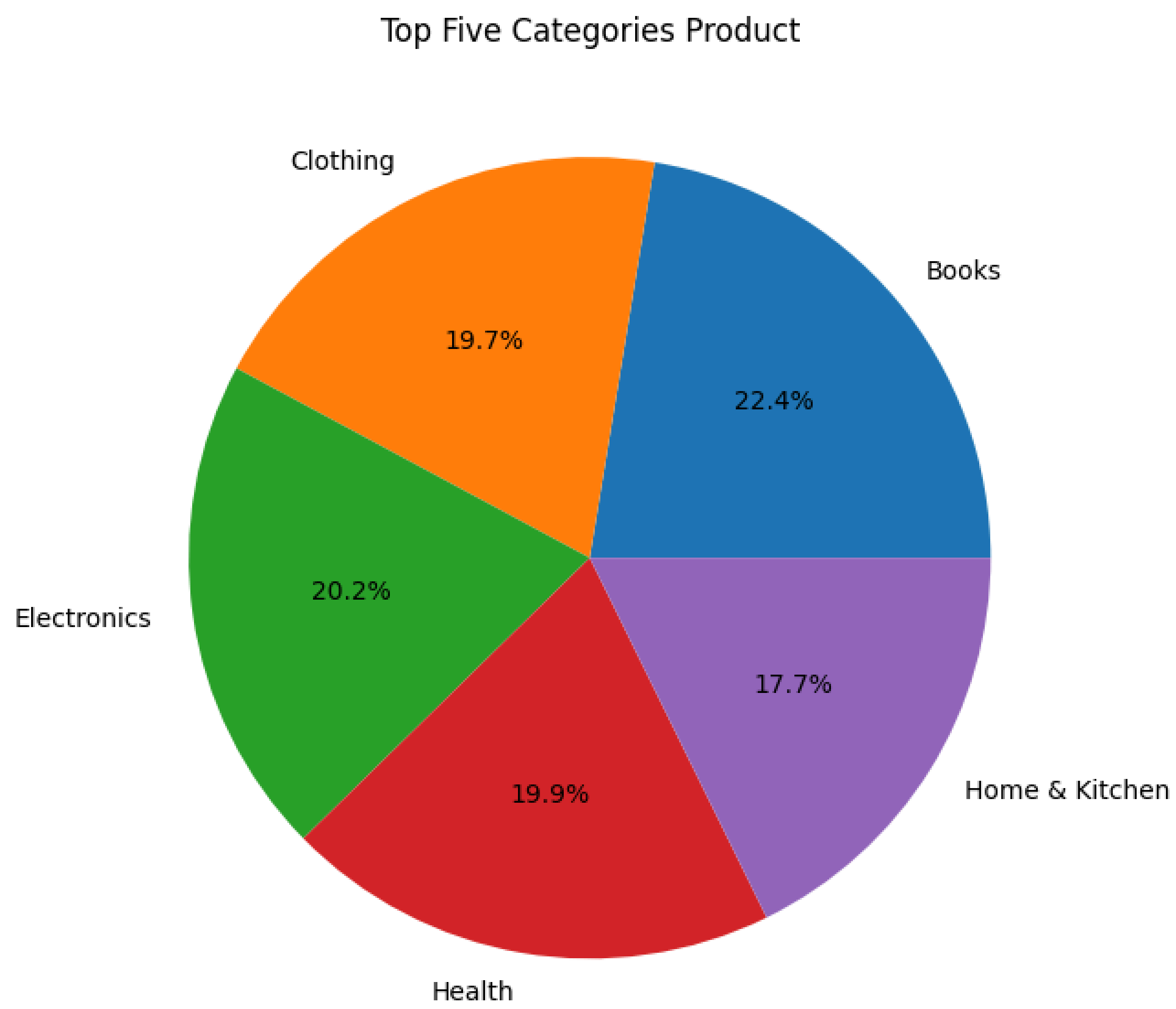
	category	year
0	Books	938229
1	Clothing	826536
2	Electronics	845120
3	Health	834414
4	Home & Kitchen	742141

Next steps: Generate code with t5cat View recommended plots New interactive sheet

```
t5cat.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   category  5 non-null      object
 1   year      5 non-null      int64
dtypes: int64(1), object(1)
memory usage: 212.0+ bytes
```

```
plt.figure(figsize=(16,6))
plt.pie(t5cat["year"],labels=t5cat["category"],autopct="%1.1f%%")
plt.title("Top Five Categories Product")
plt.tight_layout()
plt.show()
```

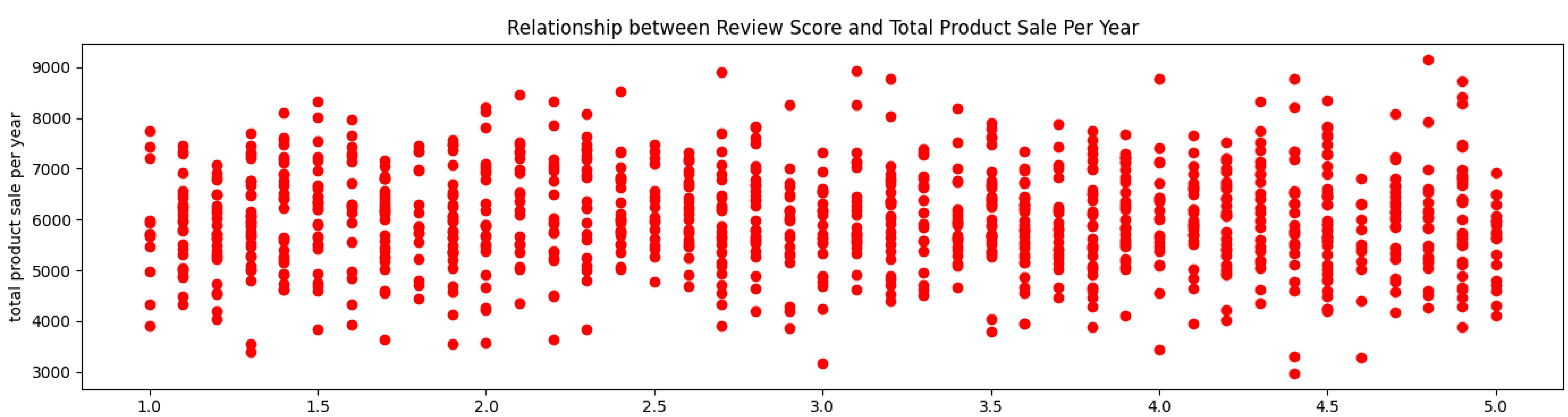


Now we will see relationship between customer review score and total product sale per year.

```
Ecom.columns
```

```
Index(['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', 'sales_month_7',
       'sales_month_8', 'sales_month_9', 'sales_month_10', 'sales_month_11',
       'sales_month_12', 'year'],
      dtype='object')
```

```
plt.figure(figsize=(14,4))
plt.scatter(x=Ecom["review_score"],y=Ecom["year"],color="red")
plt.title("Relationship between Review Score and Total Product Sale Per Year")
plt.xlabel("review score")
plt.ylabel("total product sale per year")
plt.tight_layout()
plt.show()
```



❖ Analysis of Results :

The E-commerce Sales Analysis project yielded several key insights into product performance, customer preferences, and sales trends over a 12-month period. Below is a summary of the findings:

1. *Top-Selling Products and Categories*

   - The analysis identified Product_224 as the highest-selling product for the year, with the maximum total sales.

   - Among product categories, Books emerged as the top-performing category, generating the highest total sales (938,229 units), followed by Toys (917,101 units) and Sports (916,371 units).

   - The pie chart visualization revealed the distribution of sales among the top five categories:

   - Books: 22.4%

   - Clothing: 19.9%

   - Electronics: 17.7%

   - Health: 15.7%

   - Home & Kitchen: 14.3%

2. Month-Wise Sales Trends and Seasonal Patterns

- The line graph depicting monthly sales showed fluctuations across the year, indicating potential seasonal trends.

- Peaks and troughs in sales were observed in specific months, suggesting periods of high demand or slowdowns. For instance, sales in Month 1 and Month 6 were notably higher, possibly due to holiday seasons or promotional events.

- Understanding these patterns can help the e-commerce platform optimize inventory and marketing strategies for peak seasons.

3. Relationship Between Review Scores and Sales

- The scatter plot analysis revealed a positive correlation between customer review scores and total product sales.

- Products with higher review scores (closer to 5.0) generally exhibited higher sales, emphasizing the importance of customer satisfaction and product quality in driving revenue.

- However, some outliers were observed, indicating that other factors (e.g., price, marketing efforts) may also influence sales performance.

4. Data Quality and Preprocessing

- The dataset was found to be clean, with no missing values or incorrect data types, ensuring the reliability of the analysis.

- No preprocessing was required, allowing the team to focus directly on deriving insights.

5. Business Implications

- Inventory Management: Prioritize stocking high-performing categories like Books and Toys, especially during peak sales months.

- Marketing Strategies: Leverage positive customer reviews to promote top-rated products and improve visibility.

- Pricing and Discounts: Consider competitive pricing or discounts for lower-performing categories to boost sales.

Conclusion

- The project successfully identified critical trends and actionable insights for the e-commerce platform.
- By focusing on top-selling categories, optimizing seasonal strategies, and enhancing customer satisfaction, the platform can drive growth and improve overall performance.
- Future work could include deeper segmentation (e.g., by demographics) and predictive modeling to forecast sales trends.