



Navigating the Digital Marketplace: A Data-Driven Analysis of E-Commerce Performance and Consumer Trends

EN-NADIF Moad & ANOUAR Naoufal

Module : Statistiques Descriptives
Supervised By : Dr ELKHECHAFI Mariam
January 1, 2025

E-Commerce Sales Data Analysis

Contents

I) Strategic Analytics for E-commerce Growth: Insights and Implications	3
1 Introduction	3
2 Data Visualization	3
3 Customer Segment Performance	3
3.1 Occasional Customers	3
3.2 Premium Customers	3
3.3 Regular Customers	3
4 Product Category Profitability	4
4.1 Sports	4
4.2 Toys	4
4.3 Home Decor & Electronics	4
5 Revenue vs. Discounts	4
6 Marketing Spend Efficiency	4
7 Key Metrics and Performance Indicators	4
7.1 Overall Financial Metrics	4
7.2 Category-Specific Metrics	4
7.3 Marketing Metrics	5
7.4 Sales Metrics	5
8 Meaningful Interpretations and Recommendations	5
 II) Data Analysis with Python	 7
9 Data collection	7
9.1 Importing Necessary Python Libraries	7
9.2 Importing Data	7
10 Data Cleaning	8
10.1 Check Missing Values	8
10.2 Duplicated Data	8
10.3 Data Types	8
11 Data Analysis	9
11.1 Sales and Revenue Analysis	9
11.1.1 Units Sold	9
11.1.2 Revenue	9
11.1.3 Discounted Revenue	10
11.1.4 KPIs :	12

11.2 Marketing Performance Analysis	12
11.2.1 Marketing Total Marketing Spend	12
11.2.2 Sales per Marketing Dirham	13
11.2.3 Avg Marketing Spend Per Unit Per Product	13
11.3 Profitability Analysis	13
11.3.1 Gross Profit	13
11.3.2 Gross Margin	13
11.3.3 Profit per Customer Segment	13
11.3.4 Profit per Product Category per Customer Segment	14
11.3.5 Profit per Customer Segment per Product Category	14
11.4 Data Correlation	15
12 Revenue and marketing spend Forecast	16

I) Strategic Analytics for E-commerce Growth: Insights and Implications

1 Introduction

In the ever-evolving world of e-commerce, making informed decisions is critical to maintaining a competitive edge and achieving sustainable growth. This report aims to provide a comprehensive analysis of an e-commerce business by leveraging statistical tools and techniques discussed during the course. The study encompasses an evaluation of central tendency measures, position, and dispersion to gain insights into customer behavior, sales performance, and profitability. Furthermore, bivariate analysis is employed to identify relationships between key variables, and predictive models are developed to forecast future trends and optimize decision-making. By integrating these analytical elements, this report seeks to highlight actionable insights, identify opportunities for growth, and recommend strategies for enhancing operational efficiency and profitability.

2 Data Visualization

[Click here to view the Ecommerce Sales Dashboard.](#)

3 Customer Segment Performance

3.1 Occasional Customers

- Profit Contribution: \$2,259,686.30 (Highest among all segments).
- Behavior: Infrequent purchasers, but generate substantial revenue per transaction.
- Category Preferences: Spread across multiple categories.

3.2 Premium Customers

- Profit Contribution: \$1,980,552.20.
- Behavior: Consistent high spenders, with a focus on Sports and Toys.
- Profitability: Slightly lower total profit than Occasional customers but strong performance in specific categories.

3.3 Regular Customers

- Profit Contribution: \$2,085,222.90.
- Behavior: Steady and moderate spenders across various categories.
- Potential: May respond well to strategies aimed at increasing frequency or basket size.

4 Product Category Profitability

4.1 Sports

- Profit Contribution: \$510,359.70 (Highest among all categories).
- Customer Segments: Strong performance, especially with Premium customers.

4.2 Toys

- Discounted Revenue: \$2,403,555 (Highest among all categories).
- Profitability: Significant revenue driven by deep discounts, though margins are reduced.

4.3 Home Decor & Electronics

Performed well across multiple segments, with steady contributions to total profit.

5 Revenue vs. Discounts

- High Discount Categories: Sports and Toys.
- Profit Impact: Lower profit per unit after discounts; close monitoring is necessary to avoid diminishing margins.

6 Marketing Spend Efficiency

- Highest Spend Categories: Sports and Toys received the most marketing focus.
- Alignment with Revenue: Marketing spend is justified by revenue generation in these categories.
- Efficiency Metrics: Customer Acquisition Cost (CAC) and Customer Lifetime Value (CLV) analysis needed to refine spend efficiency.

7 Key Metrics and Performance Indicators

7.1 Overall Financial Metrics

- Total Revenue: \$2,259,686.30
- Total Profit Contribution by Segment:
 - Occasional: \$2,259,686.30
 - Premium: \$1,980,552.20
 - Regular: \$2,085,222.90

7.2 Category-Specific Metrics

- Average Profit Margin by Category:
 - Electronics: 96.3%

- Toys: 97.5%
- Sports: 76.8%
- Home Decor: 83.9%
- Fashion: 93.4%

7.3 Marketing Metrics

- Total Marketing Spend by Segment:
 - Occasional: \$17,524.51
 - Premium: \$29,792.71
 - Regular: \$6,148.94
- Average Customer Acquisition Cost (CAC): \$221.40
- Return on Marketing Investment (ROMI): 2.99

7.4 Sales Metrics

- Average Order Value (AOV): \$592.12
- Units Sold by Category:
 - Electronics: 40
 - Toys: 62
 - Sports: 58
 - Home Decor: 66
 - Fashion: 34

8 Meaningful Interpretations and Recommendations

- Seasonal Pricing: Increase prices during peak seasons (e.g., holidays, back-to-school) for high-demand categories like Toys and Electronics. Offer competitive pricing during off-peak seasons to maintain customer engagement.
- Segmented Pricing: Differentiate pricing strategies for customer segments:
 - Premium Customers: Offer exclusive, higher-priced products or early access.
 - Price-Sensitive Customers: Focus on value-based discounts to attract occasional buyers.
- Expand Marketing Efforts:
 - Social Media Campaigns: Leverage platforms like Instagram and TikTok to showcase trending products, especially in categories like Fashion and Toys.
 - Influencer Partnerships: Collaborate with influencers to drive sales for high-margin and seasonal products.

- Email Marketing: Focus on targeted email campaigns for promotions and new product launches.

II) Data Analysis with Python

9 Data collection

9.1 Importing Necessary Python Libraries

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from prophet import Prophet
```

9.2 Importing Data

```
[2]: Data = pd.read_excel('Ecommerce_Sales_Prediction_Dataset.xlsx')
Data
```

```
[2]:
```

	Date	Product_Category	Price	Discount	Customer_Segment	\
0	2023-01-01 00:00:00	Sports	932.80	35.82	Occasional	
1	2023-02-01 00:00:00	Toys	569.48	3.60	Premium	
2	2023-03-01 00:00:00	Home Decor	699.68	3.56	Premium	
3	2023-04-01 00:00:00	Toys	923.27	0.61	Premium	
4	2023-05-01 00:00:00	Toys	710.17	47.83	Premium	
..	
995	22-09-2025	Fashion	740.06	49.06	Regular	
996	23-09-2025	Electronics	517.10	28.48	Premium	
997	24-09-2025	Electronics	683.43	12.98	Occasional	
998	25-09-2025	Sports	51.26	21.85	Occasional	
999	26-09-2025	Home Decor	93.94	29.68	Regular	

	Marketing_Spend	Units_Sold
0	6780.38	32
1	6807.56	16
2	3793.91	27
3	9422.75	29
4	1756.83	17
..
995	1890.83	34
996	8011.66	21
997	3408.51	19
998	7335.62	26
999	4258.11	39

[1000 rows x 7 columns]

10 Data Cleaning

10.1 Check Missing Values

```
[3]: missing_values = Data.isnull().sum()  
missing_values
```

```
[3]: Date                0  
Product_Category       0  
Price                  0  
Discount               0  
Customer_Segment       0  
Marketing_Spend         0  
Units_Sold              0  
dtype: int64
```

==> There is no missing value

10.2 Duplicated Data

```
[4]: Dup_Data= Data.duplicated().sum()  
Dup_Data
```

```
[4]: 0
```

==> There is no Duplicated Data

10.3 Data Types

- Data types Detection

```
[5]: Types_Data = Data.dtypes  
Types_Data
```

```
[5]: Date                object  
Product_Category       object  
Price                  float64  
Discount               float64  
Customer_Segment       object  
Marketing_Spend         float64  
Units_Sold              int64  
dtype: object
```

==> The 'date' variable should be of the date data type

- Data types Modification

```
[6]: Data['Date'] = pd.to_datetime(Data['Date'], errors='coerce')  
Data.dtypes
```

```
[6]: Date                datetime64[ns]
     Product_Category    object
     Price               float64
     Discount            float64
     Customer_Segment    object
     Marketing_Spend     float64
     Units_Sold          int64
     dtype: object
```

11 Data Analysis

11.1 Sales and Revenue Analysis

11.1.1 Units Sold

```
[7]: Units_Sold_per_Customer_Segment = Data.groupby('Customer_Segment')['Units_Sold'].
     ↪sum().reset_index()
     Units_Sold_per_Customer_Segment.sort_values(by=['Units_Sold'], ascending=False)
```

```
[7]: Customer_Segment  Units_Sold
     2      Regular      10222
     0    Occasional      9842
     1      Premium      9567
```

```
[8]: Units_Sold_per_Product_Category = Data.groupby('Product_Category')['Units_Sold'].
     ↪sum().reset_index()
     Units_Sold_per_Product_Category.sort_values(by=['Units_Sold'], ascending=False)
```

```
[8]: Product_Category  Units_Sold
     0    Electronics      6210
     3      Sports      6125
     4      Toys      6003
     1      Fashion      5650
     2    Home Decor      5643
```

Total Units Sold is :

```
[9]: Total_Units_Sold= Data['Units_Sold'].sum()
     Total_Units_Sold
```

```
[9]: 29631
```

11.1.2 Revenue

```
[10]: Data['Revenue']= Data.Price * Data.Units_Sold
     Data
```

```
[10]:      Date Product_Category  Price  Discount Customer_Segment \
     0  2023-01-01      Sports  932.80    35.82    Occasional
```

1	2023-02-01	Toys	569.48	3.60	Premium
2	2023-03-01	Home Decor	699.68	3.56	Premium
3	2023-04-01	Toys	923.27	0.61	Premium
4	2023-05-01	Toys	710.17	47.83	Premium
..
995	2025-09-22	Fashion	740.06	49.06	Regular
996	2025-09-23	Electronics	517.10	28.48	Premium
997	2025-09-24	Electronics	683.43	12.98	Occasional
998	2025-09-25	Sports	51.26	21.85	Occasional
999	2025-09-26	Home Decor	93.94	29.68	Regular

	Marketing_Spend	Units_Sold	Revenue
0	6780.38	32	29849.60
1	6807.56	16	9111.68
2	3793.91	27	18891.36
3	9422.75	29	26774.83
4	1756.83	17	12072.89
..
995	1890.83	34	25162.04
996	8011.66	21	10859.10
997	3408.51	19	12985.17
998	7335.62	26	1332.76
999	4258.11	39	3663.66

[1000 rows x 8 columns]

Total Revenue is :

```
[11]: Revenue = Data['Revenue'].sum()
Revenue
```

```
[11]: 15002484.56
```

11.1.3 Discounted Revenue

```
[12]: Data['Discount'] = Data['Discount'] / 100
Data['Discounted_Revenue']=( Data.Revenue * (1-Data.Discount)).round(1)
Data['Profit'] =( Data['Discounted_Revenue'] - Data['Marketing_Spend']).round(1)

Data
```

```
[12]:      Date Product_Category  Price  Discount Customer_Segment \
0   2023-01-01      Sports  932.80    0.3582      Occasional
1   2023-02-01      Toys  569.48    0.0360      Premium
2   2023-03-01  Home Decor  699.68    0.0356      Premium
3   2023-04-01      Toys  923.27    0.0061      Premium
4   2023-05-01      Toys  710.17    0.4783      Premium
..      ...      ...      ...      ...      ...
```

995	2025-09-22	Fashion	740.06	0.4906	Regular
996	2025-09-23	Electronics	517.10	0.2848	Premium
997	2025-09-24	Electronics	683.43	0.1298	Occasional
998	2025-09-25	Sports	51.26	0.2185	Occasional
999	2025-09-26	Home Decor	93.94	0.2968	Regular

	Marketing_Spend	Units_Sold	Revenue	Discounted_Revenue	Profit
0	6780.38	32	29849.60	19157.5	12377.1
1	6807.56	16	9111.68	8783.7	1976.1
2	3793.91	27	18891.36	18218.8	14424.9
3	9422.75	29	26774.83	26611.5	17188.8
4	1756.83	17	12072.89	6298.4	4541.6
..
995	1890.83	34	25162.04	12817.5	10926.7
996	8011.66	21	10859.10	7766.4	-245.3
997	3408.51	19	12985.17	11299.7	7891.2
998	7335.62	26	1332.76	1041.6	-6294.0
999	4258.11	39	3663.66	2576.3	-1681.8

[1000 rows x 10 columns]

3-1) Total Discounted Revenue:

```
[13]: Discounted_Revenue = Data['Discounted_Revenue'].sum().round(1)
Discounted_Revenue
```

```
[13]: 11238292.6
```

3-2 Discounted Revenue per Customer Segment

```
[14]: Dis_Rev_per_Customer_Segment = Data.
      ↳groupby('Customer_Segment')['Discounted_Revenue'].sum().reset_index()
Dis_Rev_per_Customer_Segment.sort_values(by=['Discounted_Revenue'],
      ↳ascending=False)
```

```
[14]: Customer_Segment  Discounted_Revenue
0      Occasional      3914791.2
2      Regular        3805949.4
1      Premium        3517552.0
```

3-3) Discounted Revenue per Product Category

```
[15]: Dis_Rev_per_Product_Category = Data.
      ↳groupby('Product_Category')['Discounted_Revenue'].sum().reset_index()
Dis_Rev_per_Product_Category.sort_values(by=['Discounted_Revenue'],
      ↳ascending=False)
```

```
[15]: Product_Category  Discounted_Revenue
3      Sports        2403555.0
```

0	Electronics	2265784.7
2	Home Decor	2192501.2
1	Fashion	2188679.1
4	Toys	2187772.6

3-4) Discounted Revenue per Product Category per Customer Segment

```
[16]: Dis_Rev_per_Product_Category_per_Customer_Segment = Data.
      ↳groupby(['Product_Category', 'Customer_Segment'])['Discounted_Revenue'].sum().
      ↳reset_index()
      Dis_Rev_per_Product_Category_per_Customer_Segment.
      ↳sort_values(by=['Product_Category', 'Discounted_Revenue'], ascending=False)
```

```
[16]: Product_Category Customer_Segment Discounted_Revenue
12      Toys Occasional 775960.0
13      Toys Premium 732562.6
14      Toys Regular 679250.0
11      Sports Regular 841826.9
10      Sports Premium 829645.4
9       Sports Occasional 732082.7
6       Home Decor Occasional 867778.6
8       Home Decor Regular 716574.5
7       Home Decor Premium 608148.1
3       Fashion Occasional 847477.4
4       Fashion Premium 672897.0
5       Fashion Regular 668304.7
2       Electronics Regular 899993.3
0       Electronics Occasional 691492.5
1       Electronics Premium 674298.9
```

11.1.4 KPIs :

Average Order Value (AOV)

```
[17]: AOV=(Discounted_Revenue/Data['Units_Sold'].sum()).round(1)
      AOV
```

```
[17]: 379.3
```

11.2 Marketing Performance Analysis

11.2.1 Marketing Total Marketing Spend

```
[18]: Total_Marketing_Spend = Data['Marketing_Spend'].sum().round(1)
      Total_Marketing_Spend
```

```
[18]: 4912830.2
```

11.2.2 Sales per Marketing Dirham

```
[19]: SMD= ( Discounted_Revenue/Data.Marketing_Spend.sum()).round(2)
SMD
```

```
[19]: 2.29
```

11.2.3 Avg Marketing Spend Per Unit Per Product

```
[20]: Data['Marketing_Spend_per_Unit'] = Data['Marketing_Spend'] / Data['Units_Sold']
avg_marketing_spend_per_unit = Data.
    ↳groupby('Product_Category')['Marketing_Spend_per_Unit'].mean().reset_index()
avg_marketing_spend_per_unit
```

```
[20]:  Product_Category  Marketing_Spend_per_Unit
0      Electronics          181.036316
1         Fashion          176.067336
2      Home Decor          179.104795
3         Sports          168.942017
4          Toys          183.202207
```

11.3 Profitability Analysis

11.3.1 Gross Profit

```
[21]: Gross_Profit= Data['Profit'].sum().round(1)
Gross_Profit
```

```
[21]: 6325461.4
```

11.3.2 Gross Margin

```
[22]: Gross_margin=(Gross_Profit/Discounted_Revenue).round(2)
Gross_margin
```

```
[22]: 0.56
```

11.3.3 Profit per Customer Segment

```
[23]: Profit_per_Customer_Segment = Data.groupby('Customer_Segment')['Profit'].sum().
    ↳reset_index()
Profit_per_Customer_Segment.sort_values(by=['Profit'], ascending=False)
```

```
[23]:  Customer_Segment  Profit
0      Occasional  2259686.3
2         Regular  2085222.9
1         Premium  1980552.2
```

11.3.4 Profit per Product Category per Customer Segment

```
[24]: Profit_per_Product_Category_per_Customer_Segment = Data.  
      ↳groupby(['Product_Category', 'Customer_Segment'])['Profit'].sum().reset_index()  
Profit_per_Product_Category_per_Customer_Segment.  
      ↳sort_values(by=['Product_Category', 'Profit'], ascending=False)
```

```
[24]:   Product_Category Customer_Segment   Profit  
13         Toys      Premium  405274.9  
14         Toys      Regular  389421.0  
12         Toys  Occasional  382528.5  
10        Sports      Premium  510359.7  
9         Sports  Occasional  466621.5  
11        Sports      Regular  448325.3  
6        Home Decor  Occasional  520599.7  
8        Home Decor      Regular  410555.1  
7        Home Decor      Premium  322620.3  
3         Fashion  Occasional  533461.7  
4         Fashion      Premium  380711.4  
5         Fashion      Regular  353551.1  
2        Electronics      Regular  483370.4  
1        Electronics      Premium  361585.9  
0        Electronics  Occasional  356474.9
```

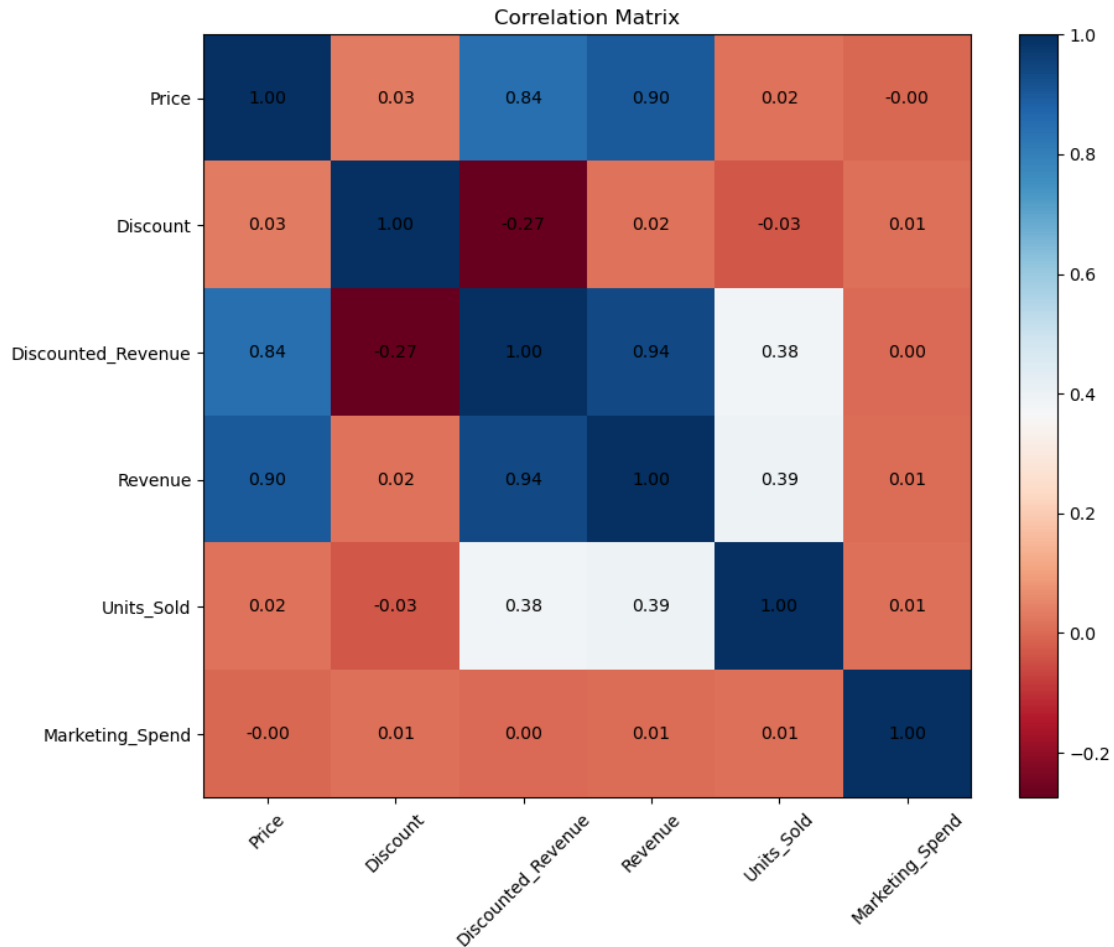
11.3.5 Profit per Customer Segment per Product Category

```
[25]: Profit_per_Customer_Segment_Product_Category = Data.  
      ↳groupby(['Customer_Segment', 'Product_Category'])['Profit'].sum().reset_index()  
Profit_per_Customer_Segment_Product_Category.sort_values(by=['Customer_Segment',  
      ↳'Profit'], ascending=False)
```

```
[25]:   Customer_Segment Product_Category   Profit  
10         Regular      Electronics  483370.4  
13         Regular           Sports  448325.3  
12         Regular      Home Decor  410555.1  
14         Regular           Toys  389421.0  
11         Regular      Fashion  353551.1  
8          Premium           Sports  510359.7  
9          Premium           Toys  405274.9  
6          Premium      Fashion  380711.4  
5          Premium      Electronics  361585.9  
7          Premium      Home Decor  322620.3  
1    Occasional      Fashion  533461.7  
2    Occasional      Home Decor  520599.7  
3    Occasional           Sports  466621.5  
4    Occasional           Toys  382528.5  
0    Occasional      Electronics  356474.9
```

11.4 Data Correlation

```
[26]: correlation_data = Data[['Price', 'Discount', 'Discounted_Revenue', 'Revenue',  
    ↪ 'Units_Sold', 'Marketing_Spend']]  
correlation_matrix = correlation_data.corr()  
  
plt.figure(figsize=(10, 8), edgecolor='Black')  
plt.imshow(correlation_matrix, cmap='RdBu', interpolation='none',  
    ↪ aspect='auto', alpha=1)  
plt.colorbar()  
plt.xticks(range(len(correlation_matrix.columns)), correlation_matrix.columns,  
    ↪ rotation=45)  
plt.yticks(range(len(correlation_matrix.columns)), correlation_matrix.columns)  
plt.title('Correlation Matrix')  
  
for i in range(len(correlation_matrix.columns)):  
    for j in range(len(correlation_matrix.columns)):  
        plt.text(j, i, f'{correlation_matrix.iloc[i, j]:.2f}', ha='center',  
    ↪ va='center', color='black')  
  
plt.show()
```

12 Revenue and marketing spend Forecast

```
[27]: def create_prophet_model(data, seasonality='monthly'):
    model = Prophet()
    if seasonality == 'monthly':
        model.add_seasonality(name='monthly', period=30.5, fourier_order=5)
    model.fit(data)
    return model

future_dates = pd.date_range(start='2026-12-01', periods=12, freq='MS').
    ↳to_frame(index=False, name='ds')

# Forecast total revenue
revenue_forecasts = {}
for category in Data['Product_Category'].unique():
    category_data = Data[Data['Product_Category'] == category][['Date',
    ↳'Discounted_Revenue']].rename(columns={'Date': 'ds', 'Discounted_Revenue':
    ↳'y'})
    model = create_prophet_model(category_data)
```

```

forecast = model.predict(future_dates)
revenue_forecasts[category] = forecast

# Forecast marketing spend
marketing_forecasts = {}
for category in Data['Product_Category'].unique():
    category_data = Data[Data['Product_Category'] == category][['Date',
↳ 'Marketing_Spend']].rename(columns={'Date': 'ds', 'Marketing_Spend': 'y'})
    model = create_prophet_model(category_data)
    forecast = model.predict(future_dates)
    marketing_forecasts[category] = forecast

# Plotting the forecasts
plt.figure(figsize=(14, 7))
for category, forecast in revenue_forecasts.items():
    plt.plot(forecast['ds'], forecast['yhat'], label=f'{category} Revenue')
plt.title('Total Revenue Forecast for Next 12 Months')
plt.xlabel('Date')
plt.ylabel('Revenue')
plt.legend()
plt.grid(True)
plt.show()

plt.figure(figsize=(14, 7))
for category, forecast in marketing_forecasts.items():
    plt.plot(forecast['ds'], forecast['yhat'], label=f'{category} Marketing_
↳ Spend')
plt.title('Marketing Spend Forecast for Next 12 Months')
plt.xlabel('Date')
plt.ylabel('Marketing Spend')
plt.legend()
plt.grid(True)
plt.show()

```

