Importing Libraries

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.impute import SimpleImputer
import datetime as dt
import warnings
warnings.filterwarnings('ignore')
print("All required libraries have been successfully imported!")
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[3], line 5
      3 import matplotlib.pyplot as plt
      4 import seaborn as sns
----> 5 from sklearn.preprocessing import LabelEncoder
      6 from sklearn.impute import SimpleImputer
      7 import datetime as dt

ModuleNotFoundError: No module named 'sklearn'
```

In [9]:
```
!pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\amogh\appdata\local\programs\python\python312\lib\site-packages (1.6.1)
Requirement already satisfied: numpy>=1.19.5 in c:\users\amogh\appdata\local\programs\python\python312\lib\site-packages (from scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.6.0 in c:\users\amogh\appdata\local\programs\python\python312\lib\site-packages (from scikit-learn) (1.15.2)
Requirement already satisfied: joblib>=1.2.0 in c:\users\amogh\appdata\local\programs\python\python312\lib\site-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\amogh\appdata\local\programs\python\python312\lib\site-packages (from scikit-learn) (3.6.0)
[notice] A new release of pip is available: 24.1.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [10]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.impute import SimpleImputer
import datetime as dt
import warnings
warnings.filterwarnings('ignore')
print("All required libraries have been successfully imported!")
```

```
All required libraries have been successfully imported!
```

Creating DataFrame

In [11]:
```
df = pd.read_csv("sales_5000000.csv")
```

In [12]:
```
df
```

Out[12]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost
0	Australia and Oceania	Palau	Office Supplies	Online	H	2020-03-06	517073523	2020-03-26	2401	651.21	524.96	1563555.21	1260428.96
1	Europe	Poland	Beverages	Online	L	2014-04-18	380507028	2014-05-26	9340	47.45	31.79	443183.00	296918.60
2	North America	Canada	Cereal	Online	M	2019-01-08	504055583	2019-01-31	103	205.70	117.11	21187.10	12062.33
3	Europe	Belarus	Snacks	Online	C	2018-01-19	954955518	2018-02-27	1414	152.58	97.44	215748.12	137780.16
4	Middle East and North Africa	Oman	Cereal	Offline	H	2023-04-26	970755660	2023-06-02	7027	205.70	117.11	1445453.90	822931.97
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4999995	Middle East and North Africa	Iraq	Household	Online	L	2018-03-17	940436398	2018-04-23	4884	668.27	502.54	3263830.68	2454405.36
4999996	Europe	Monaco	Meat	Offline	H	2019-11-08	407689177	2019-11-28	3142	421.89	364.69	1325578.38	1145855.98
4999997	Australia and Oceania	Solomon Islands	Clothes	Online	C	2024-06-01	727000367	2024-07-18	4419	109.28	35.84	482908.32	158376.96
4999998	Australia and Oceania	Marshall Islands	Cosmetics	Offline	L	2024-02-12	714043796	2024-03-22	282	437.20	263.33	123290.40	74259.06
4999999	Europe	Greece	Office Supplies	Online	C	2023-05-11	604805791	2023-06-12	4329	651.21	524.96	2819088.09	2272551.84

5000000 rows × 14 columns

Reducing Data Frame

In [13]:
```
df_sampled = df.sample(n=1_000_000, random_state=42)
```

In [14]:
```
df_sampled
```

Out[14]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Co
3577888	Europe	Andorra	Fruits	Online	H	2019-12-21	681978309	2020-01-14	1391	9.33	6.92	12978.03	9625.
4993932	Australia and Oceania	Tonga	Personal Care	Online	L	2020-01-06	542415893	2020-01-24	1771	81.73	56.67	144743.83	100362.
4094900	Australia and Oceania	Tonga	Snacks	Online	C	2018-10-12	492005813	2018-11-18	4308	152.58	97.44	657314.64	419771.
4420497	Europe	Czech Republic	Baby Food	Online	M	2017-11-06	263193643	2017-12-04	3702	255.28	159.42	945046.56	590172.
634465	Asia	Japan	Beverages	Online	L	2015-01-09	632574594	2015-02-01	1734	47.45	31.79	82278.30	55123.
...	...	...	...	...	...	...	...	...	...	...	...	...	
879073	Europe	Luxembourg	Snacks	Online	L	2024-09-03	164502084	2024-09-26	8677	152.58	97.44	1323936.66	845486.
461358	Europe	Greece	Cereal	Offline	C	2016-10-22	709683597	2016-12-09	2	205.70	117.11	411.40	234.
4559407	Europe	Kosovo	Snacks	Online	M	2017-02-28	896485936	2017-04-15	8965	152.58	97.44	1367879.70	873549.
1450969	Middle East and North Africa	Turkey	Office Supplies	Online	H	2018-11-09	392313897	2018-12-22	3255	651.21	524.96	2119688.55	1708744.
4805308	Europe	Slovakia	Baby Food	Online	H	2020-08-30	821599090	2020-10-18	8283	255.28	159.42	2114484.24	1320475.

1000000 rows × 14 columns

In [15]:
```
df_sampled.to_csv("reduced_dataset.csv", index=False)
```

DATA CLEANING

In [16]:
```
null_counts = df_sampled.isnull().sum()
null_percentage = (null_counts / len(df_sampled)) * 100
null_df = pd.DataFrame({'Null Count': null_counts, 'Null Percentage': null_percentage})
null_df[null_df['Null Count'] > 0].sort_values(by='Null Percentage', ascending=False)
```

Out[16]:

	Null Count	Null Percentage

In [18]:
```
threshold = 40
null_percentage = (df_sampled.isnull().sum() / len(df_sampled)) * 100
columns_to_drop = null_percentage[null_percentage > threshold].index.tolist()
print("Columns with high null percentage (> {}%):".format(threshold))
print(columns_to_drop)
```

```
Columns with high null percentage (> 40%):
[]
```

In [19]:
```
df_sampled.drop(columns=columns_to_drop, inplace=True)
print("\nUpdated dataset shape:", df_sampled.shape)
```

```
Updated dataset shape: (1000000, 14)
```

In [20]:
```
row_threshold = 0.3 * df_sampled.shape[1]
rows_to_drop = df_sampled[df_sampled.isnull().sum(axis=1) > row_threshold]

df_sampled = df_sampled.drop(rows_to_drop.index)

print(f"Removed {len(rows_to_drop)} rows with more than 30% missing values.")
print("Updated dataset shape:", df_sampled.shape)
```

```
Removed 0 rows with more than 30% missing values.
Updated dataset shape: (1000000, 14)
```

In [21]:
```
unwanted_columns = ["Column1", "Column2"]
df_sampled.drop(columns=unwanted_columns, inplace=True, errors="ignore")

print("Unwanted columns removed.")
print("Final dataset shape:", df_sampled.shape)
```

```
Unwanted columns removed.
Final dataset shape: (1000000, 14)
```

In [22]:
```
df_sampled['Purchase Date'] = pd.to_datetime(df_sampled['Order Date'], errors='coerce')
df_sampled = df_sampled.sort_values(by='Purchase Date', ascending=True)
print("Dataset sorted by Purchase Date.")
```

```
Dataset sorted by Purchase Date.
```

In [23]:
```
df_sampled
```

Out[23]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Co
4369702	Europe	Bosnia and Herzegovina	Personal Care	Online	C	2014-01-01	514631640	2014-02-01	9794	81.73	56.67	800463.62	555025.
1028195	Sub-Saharan Africa	Guinea	Cereal	Online	L	2014-01-01	402343213	2014-02-14	1153	205.70	117.11	237172.10	135027.
640973	Australia and Oceania	Federated States of Micronesia	Clothes	Online	H	2014-01-01	396085298	2014-02-16	9296	109.28	35.84	1015866.88	333168.
4837943	Europe	Lithuania	Personal Care	Online	C	2014-01-01	699614369	2014-02-14	294	81.73	56.67	24028.62	16660.
2841590	Australia and Oceania	Samoa	Office Supplies	Online	L	2014-01-01	113376605	2014-01-15	9512	651.21	524.96	6194309.52	4993419.
...	...	...	...	...	...	...	...	...	...	...	...	...	
4812427	Middle East and North Africa	Bahrain	Office Supplies	Online	M	2024-09-10	940800321	2024-09-24	2576	651.21	524.96	1677516.96	1352296.
3441561	Asia	Singapore	Cosmetics	Offline	L	2024-09-10	606217706	2024-10-05	2145	437.20	263.33	937794.00	564842.
322106	Asia	India	Office Supplies	Offline	M	2024-09-10	994838893	2024-10-01	3791	651.21	524.96	2468737.11	1990123.
3193251	Asia	Sri Lanka	Fruits	Online	M	2024-09-10	234087598	2024-10-30	577	9.33	6.92	5383.41	3992.
2461025	Australia and Oceania	Tonga	Household	Online	L	2024-09-10	947058236	2024-09-21	4433	668.27	502.54	2962440.91	2227759.

1000000 rows × 15 columns

In [24]:
```
df_sampled.to_csv("reduced_dataset.csv", index=False)
```

In [25]:
```
df_sampled.drop(columns=['Order Date'], inplace=True, errors='ignore')
print("Order Date column removed.")
print("Updated dataset shape:", df_sampled.shape)
```

```
Order Date column removed.
Updated dataset shape: (1000000, 14)
```

In [26]:
```
df_sampled
```

Out[26]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Tot
4369702	Europe	Bosnia and Herzegovina	Personal Care	Online	C	514631640	2014-02-01	9794	81.73	56.67	800463.62	555025.98	24
1028195	Sub-Saharan Africa	Guinea	Cereal	Online	L	402343213	2014-02-14	1153	205.70	117.11	237172.10	135027.83	10
640973	Australia and Oceania	Federated States of Micronesia	Clothes	Online	H	396085298	2014-02-16	9296	109.28	35.84	1015866.88	333168.64	68
4837943	Europe	Lithuania	Personal Care	Online	C	699614369	2014-02-14	294	81.73	56.67	24028.62	16660.98	
2841590	Australia and Oceania	Samoa	Office Supplies	Online	L	113376605	2014-01-15	9512	651.21	524.96	6194309.52	4993419.52	120
...	...	...	...	...	...	...	...	...	...	...	...	...	
4812427	Middle East and North Africa	Bahrain	Office Supplies	Online	M	940800321	2024-09-24	2576	651.21	524.96	1677516.96	1352296.96	32
3441561	Asia	Singapore	Cosmetics	Offline	L	606217706	2024-10-05	2145	437.20	263.33	937794.00	564842.85	37
322106	Asia	India	Office Supplies	Offline	M	994838893	2024-10-01	3791	651.21	524.96	2468737.11	1990123.36	47
3193251	Asia	Sri Lanka	Fruits	Online	M	234087598	2024-10-30	577	9.33	6.92	5383.41	3992.84	
2461025	Australia and Oceania	Tonga	Household	Online	L	947058236	2024-09-21	4433	668.27	502.54	2962440.91	2227759.82	73

1000000 rows × 14 columns

In [27]:
```
df_sampled.drop(columns=['Ship Date'], inplace=True, errors='ignore')
print("Order Date column removed.")
print("Updated dataset shape:", df_sampled.shape)
```

```
Order Date column removed.
Updated dataset shape: (1000000, 13)
```

In [28]:
```
df_sampled
```

Out[28]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order ID	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
4369702	Europe	Bosnia and Herzegovina	Personal Care	Online	C	514631640	9794	81.73	56.67	800463.62	555025.98	245437.64
1028195	Sub-Saharan Africa	Guinea	Cereal	Online	L	402343213	1153	205.70	117.11	237172.10	135027.83	102144.27
640973	Australia and Oceania	Federated States of Micronesia	Clothes	Online	H	396085298	9296	109.28	35.84	1015866.88	333168.64	682698.24
4837943	Europe	Lithuania	Personal Care	Online	C	699614369	294	81.73	56.67	24028.62	16660.98	7367.64
2841590	Australia and Oceania	Samoa	Office Supplies	Online	L	113376605	9512	651.21	524.96	6194309.52	4993419.52	1200890.00
...	...	...	...	...	...	...	...	...	...	...	...	...
4812427	Middle East and North Africa	Bahrain	Office Supplies	Online	M	940800321	2576	651.21	524.96	1677516.96	1352296.96	325220.00
3441561	Asia	Singapore	Cosmetics	Offline	L	606217706	2145	437.20	263.33	937794.00	564842.85	372951.15
322106	Asia	India	Office Supplies	Offline	M	994838893	3791	651.21	524.96	2468737.11	1990123.36	478613.75
3193251	Asia	Sri Lanka	Fruits	Online	M	234087598	577	9.33	6.92	5383.41	3992.84	1390.57
2461025	Australia and Oceania	Tonga	Household	Online	L	947058236	4433	668.27	502.54	2962440.91	2227759.82	734681.09

1000000 rows × 13 columns

In [ ]:
```
df_sampled.to_csv("reduced_dataset.csv", index=False)
```

Most Engaged Products

In [31]:
```
product_engagement = df_sampled['Item Type'].value_counts().reset_index()
product_engagement.columns = ['Item Type', 'Units Sold']

most_engaged_product = product_engagement.iloc[0]
print("Highest Customer Engagement Product:")
print(most_engaged_product)
```

```
Highest Customer Engagement Product:
Item Type     Snacks
Units Sold     83641
Name: 0, dtype: object
```

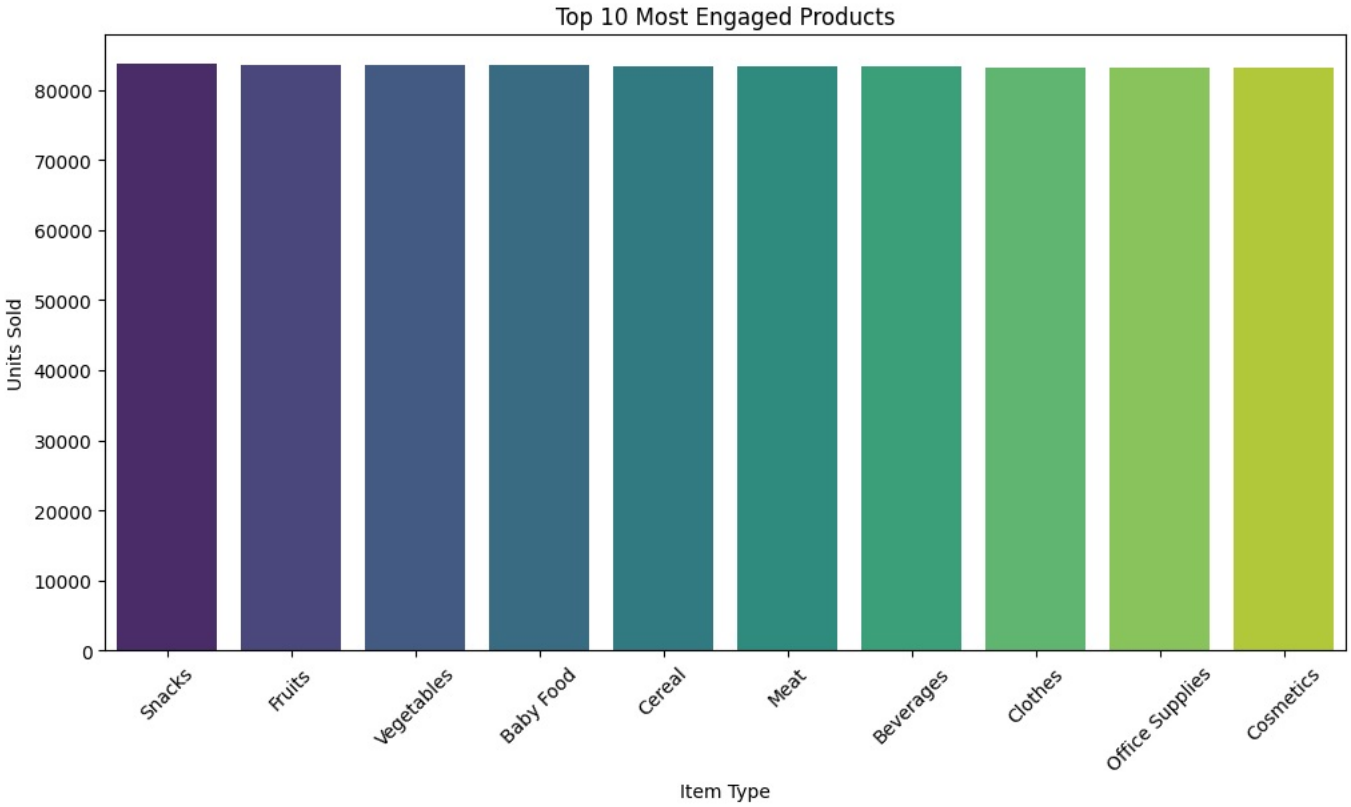
In [32]:
```
product_sales = df_sampled.groupby('Item Type')['Total Revenue'].sum().reset_index()
product_sales = product_sales.sort_values(by='Total Revenue', ascending=False)

top_product = product_sales.iloc[0]
print("Highest Revenue Generating Product:")
print(top_product)
```

```
Highest Revenue Generating Product:
Item Type                  Household
Total Revenue    277496609482.690002
Name: 6, dtype: object
```

In [34]:
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12, 6))
sns.barplot(x=product_engagement['Item Type'][:10], y=product_engagement['Units Sold'][:10], palette="viridis")
plt.xticks(rotation=45)
plt.xlabel("Item Type")
plt.ylabel("Units Sold")
plt.title("Top 10 Most Engaged Products")
plt.show()
```



Most Units Sold and its Profit

In [35]:
```
product_sales_count = df_sampled['Item Type'].value_counts().reset_index()
product_sales_count.columns = ['Item Type', 'Units Sold']
print(product_sales_count.head(10))
```

```
        Item Type  Units Sold
0           Snacks       83641
1           Fruits       83565
2       Vegetables       83532
3        Baby Food       83443
4           Cereal       83420
5             Meat       83410
6        Beverages       83372
7          Clothes       83230
8  Office Supplies       83173
9        Cosmetics       83170
```

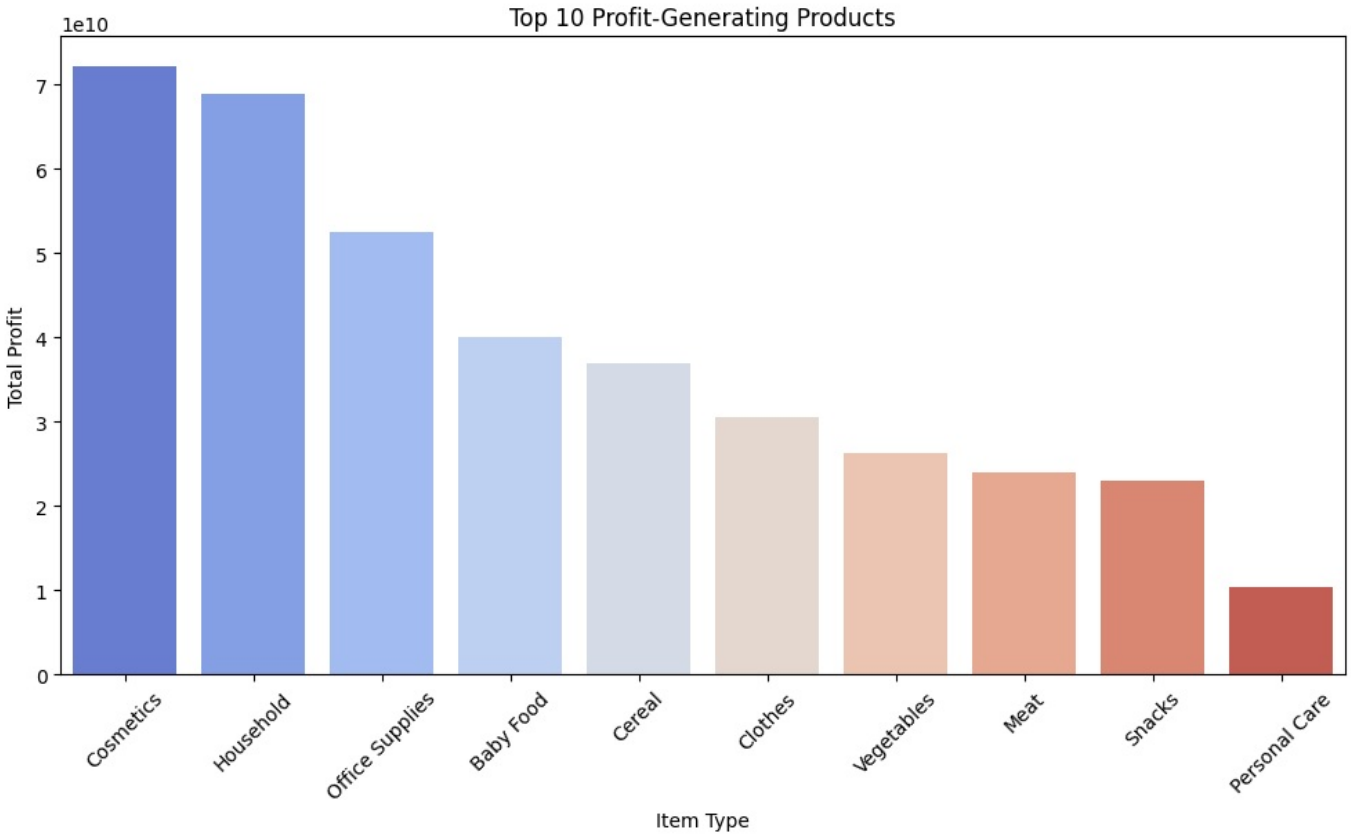
In [36]:
```
product_sales_revenue = df_sampled.groupby('Item Type')['Total Profit'].sum().reset_index()
product_sales_revenue = product_sales_revenue.sort_values(by='Total Profit', ascending=False)

print(product_sales_revenue.head(10))
```

```
         Item Type  Total Profit
4         Cosmetics  7.208481e+10
6         Household  6.881876e+10
8   Office Supplies  5.252351e+10
0         Baby Food  4.005826e+10
2            Cereal  3.686232e+10
3           Clothes  3.051282e+10
11       Vegetables  2.631973e+10
7              Meat  2.390524e+10
10           Snacks  2.304180e+10
9     Personal Care  1.040127e+10
```

In [38]:
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12, 6))
sns.barplot(x=product_sales_revenue['Item Type'][:10], y=product_sales_revenue['Total Profit'][:10], palette="c
plt.xticks(rotation=45)
plt.xlabel("Item Type")
plt.ylabel("Total Profit")
plt.title("Top 10 Profit-Generating Products")
plt.show()
```



Yearly Sales

In [41]:
```
df_sampled['Purchase Date'] = pd.to_datetime(df_sampled['Purchase Date'], errors='coerce')
df_sampled['Year'] = df_sampled['Purchase Date'].dt.year
```

In [42]:
```
df_sampled['Year']
```

Out[42]:
```
4369702    2014
1028195    2014
640973     2014
4837943    2014
2841590    2014
           ...
4812427    2024
3441561    2024
322106     2024
3193251    2024
2461025    2024
Name: Year, Length: 1000000, dtype: int32
```

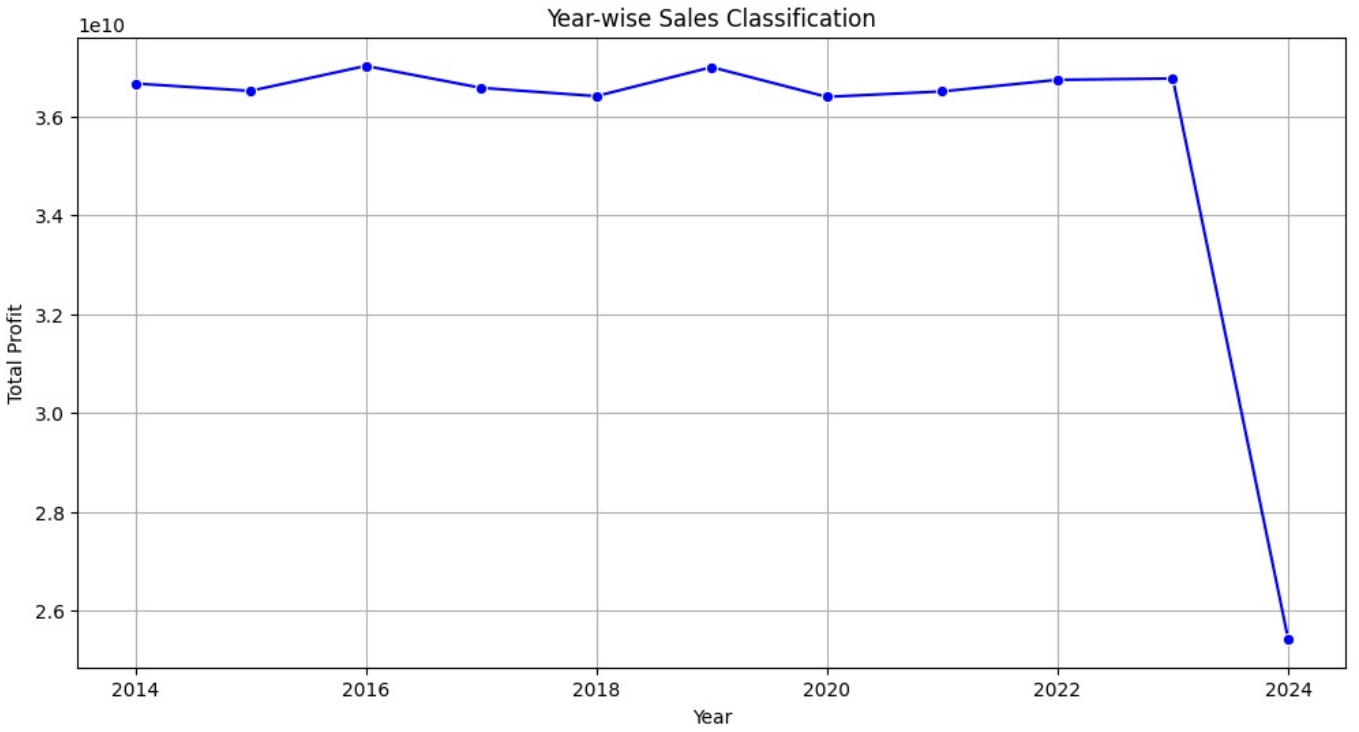
In [44]:
```
yearly_sales = df_sampled.groupby('Year')['Total Profit'].sum().reset_index()
yearly_sales = yearly_sales.sort_values(by='Year', ascending=True)

print("Year-wise Sales Classification:")
print(yearly_sales)
```

```
Year-wise Sales Classification:
    Year  Total Profit
0   2014  3.666861e+10
1   2015  3.651901e+10
2   2016  3.702472e+10
3   2017  3.658070e+10
4   2018  3.641282e+10
5   2019  3.699914e+10
6   2020  3.639818e+10
7   2021  3.650900e+10
8   2022  3.674266e+10
9   2023  3.677060e+10
10  2024  2.543752e+10
```

In [45]:
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12, 6))
sns.lineplot(data=yearly_sales, x='Year', y='Total Profit', marker='o', color='b')
plt.xlabel("Year")
plt.ylabel("Total Profit")
plt.title("Year-wise Sales Classification")
plt.grid(True)
plt.show()
```



Order Priority

In [46]:
```
priority_sales = df_sampled.groupby('Order Priority')['Total Profit'].sum().reset_index()
priority_sales = priority_sales.sort_values(by='Total Profit', ascending=False)

print("Total Sales by Order Priority:")
print(priority_sales)
```

```
Total Sales by Order Priority:
  Order Priority  Total Profit
0              C  9.816042e+10
3              M  9.805954e+10
1              H  9.796677e+10
2              L  9.787622e+10
```

In [47]:
```
priority_orders = df_sampled['Order Priority'].value_counts().reset_index()
priority_orders.columns = ['Order Priority', 'Units Sold']

print("Order Volume by Priority:")
print(priority_orders)
```

```
Order Volume by Priority:
  Order Priority  Units Sold
0              M      250489
1              C      250202
2              L      249732
3              H      249577
```

In [50]:
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 5))
sns.barplot(x=priority_sales['Order Priority'], y=priority_orders['Units Sold'], palette="coolwarm")
plt.xlabel("Order Priority")
plt.ylabel("Units Sold")
plt.title("Sales Distribution by Order Priorit")
plt.show()
```

In [51]:
```
df_sampled['PAT Income'] = df_sampled['Total Revenue'] - df_sampled['Total Cost']
```

In [52]:
```
df_sampled['PAT Income']
```

Out[52]:
```
4369702     245437.64
1028195     102144.27
640973      682698.24
4837943       7367.64
2841590    1200890.00
              ...
4812427     325220.00
3441561     372951.15
322106      478613.75
3193251       1390.57
2461025     734681.09
Name: PAT Income, Length: 1000000, dtype: float64
```

Month Wise Classification

In [53]:
```
df_sampled['Purchase Date'] = pd.to_datetime(df_sampled['Purchase Date'], errors='coerce')
df_sampled['Month'] = df_sampled['Purchase Date'].dt.month_name()
```

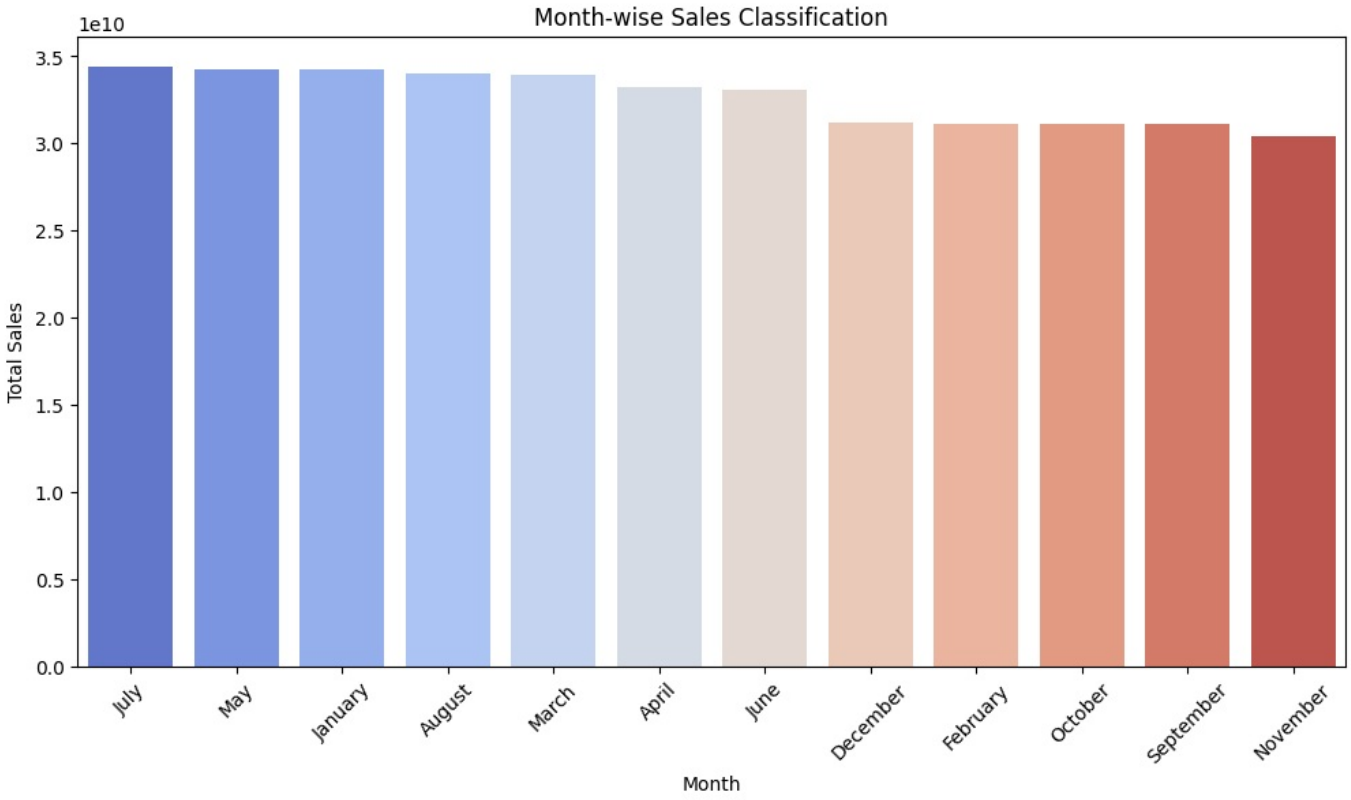
In [54]:
```
monthly_sales = df_sampled.groupby('Month')['Total Profit'].sum().reset_index()
monthly_sales = monthly_sales.sort_values(by='Total Profit', ascending=False)

print("Month-wise Sales Classification:")
print(monthly_sales)
```

```
Month-wise Sales Classification:
        Month  Total Profit
5        July  3.443416e+10
8         May  3.427095e+10
4     January  3.423547e+10
1      August  3.402837e+10
7       March  3.395267e+10
0       April  3.320233e+10
6        June  3.305524e+10
2    December  3.117489e+10
3    February  3.111800e+10
10    October  3.109965e+10
11  September  3.108691e+10
9    November  3.040432e+10
```

In [55]:
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12, 6))
sns.barplot(x=monthly_sales['Month'], y=monthly_sales['Total Profit'], palette="coolwarm")
plt.xticks(rotation=45)
plt.xlabel("Month")
plt.ylabel("Total Sales")
plt.title("Month-wise Sales Classification")
plt.show()
```



In [ ]:
```
df_sampled.to_csv("reduced_dataset.csv", index=False)
```

Thank You

In [ ]: