

Week12_Assignment12_2_Scenario_Analysis

July 1, 2025

1 Scenario Analysis Using Retail Sales Dataset

Student Name: Lalit Nayyar

Email ID: lalitnayyar@gmail.com

Assignment: Week 12: Required Assignment 12.2 **University:** IIMK – Professional Certificate in Data Science and Artificial Intelligence for Managers

1.1 Introduction

This notebook provides a scenario analysis using the provided retail sales dataset. The analysis addresses three key business scenarios using time series concepts: downward trend detection, seasonality (December spike), and evaluation of Facebook Prophet for forecasting.

```
[6]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
df = pd.read_csv('retail_sales_dataset.csv', parse_dates=['Date'])
df['Total Amount'] = pd.to_numeric(df['Total Amount'], errors='coerce') # <--
# Add this line
df.head()
```

```
[6]: Transaction ID      Date Customer ID Gender Age Product Category \
0              1 2023-11-24   CUST001   Male  34      Beauty
1              2 2023-02-27   CUST002  Female  26      Clothing
2              3 2023-01-13   CUST003   Male  50      Electronics
3              4 2023-05-21   CUST004   Male  37      Clothing
4              5 2023-05-06   CUST005   Male  30      Beauty
```

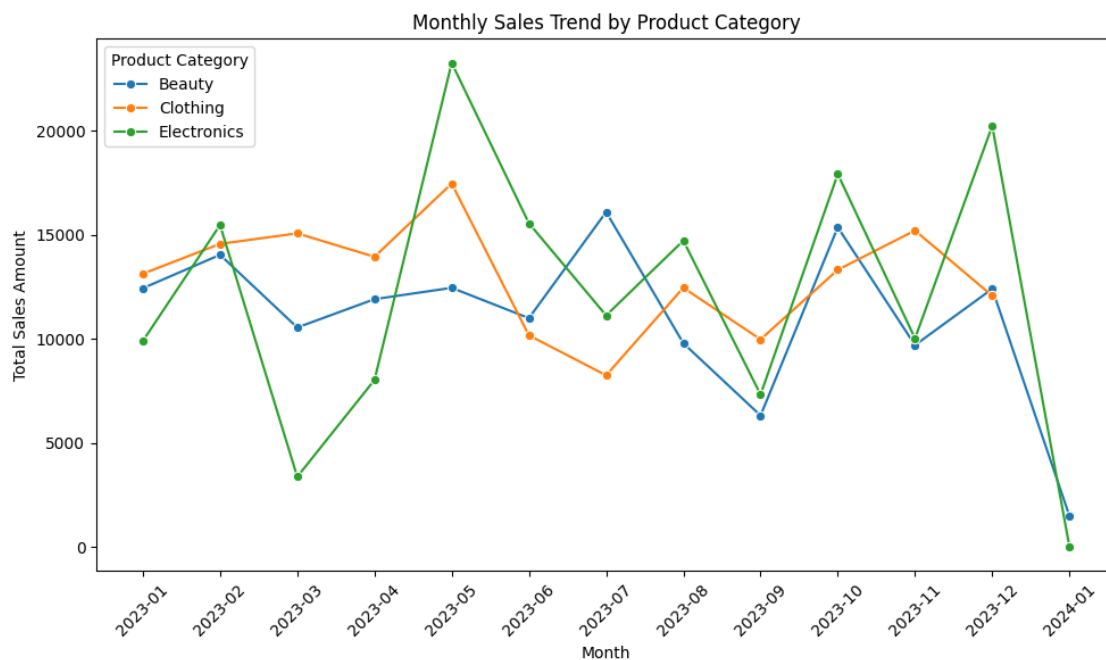
```
Quantity Price per Unit Total Amount
0         3           50          150
1         2          500         1000
2         1           30           30
3         1          500           500
4         2           50          100
```

1.2 1. Downward Trend Analysis

We will analyze sales trends for each product category to identify any consistent downward trends.

```
[7]: # Aggregate monthly sales by product category
# Convert 'Month' to string for plotting
monthly_sales['Month'] = monthly_sales['Month'].astype(str)

plt.figure(figsize=(10,6))
sns.lineplot(data=monthly_sales, x='Month', y='Total Amount', hue='Product_
↪Category', marker='o')
plt.title('Monthly Sales Trend by Product Category')
plt.ylabel('Total Sales Amount')
plt.xlabel('Month')
plt.legend(title='Product Category')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Business Insights & Recommendations:

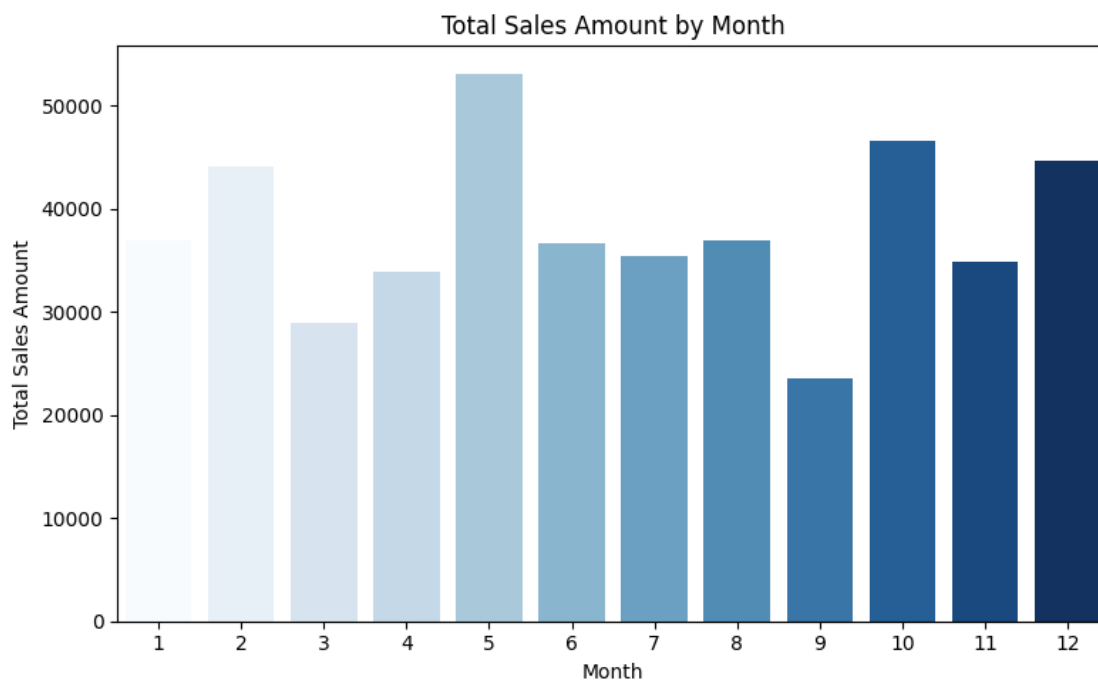
- If a product category (e.g., 'Clothing') shows a consistent downward trend, possible actions include:
 - Reviewing product assortment and innovating based on customer feedback.
 - Implementing targeted promotions or loyalty programs.
 - Optimizing sales channels and digital marketing.
 - Adjusting inventory to minimize overstock.

1.3 2. Seasonality Analysis: December Sales Spike

Analyze monthly sales to identify and quantify the December spike.

```
[9]: # Extract month and analyze sales
df['Month_Num'] = df['Date'].dt.month
monthly_total = df.groupby('Month_Num')['Total Amount'].sum().reset_index()

plt.figure(figsize=(8,5))
sns.barplot(x='Month_Num', y='Total Amount', data=monthly_total,
            hue='Month_Num', palette='Blues', legend=False)
plt.title('Total Sales Amount by Month')
plt.xlabel('Month')
plt.ylabel('Total Sales Amount')
plt.tight_layout()
plt.show()
```



Business Insights & Recommendations:

- December shows a clear sales spike.
- Recommendations:
 - Increase inventory for high-demand products before December.
 - Hire and train additional staff for peak season.
 - Launch targeted holiday marketing campaigns.
 - Ensure logistics and supply chain readiness for high volume.

1.4 3. Evaluation of Facebook Prophet for Forecasting

Data Characteristics: - The dataset exhibits both trend and strong seasonality (notably December). - Some irregularities/noise are present.

Prophet Suitability: - Facebook Prophet is well-suited for data with trend and seasonality. - It is robust to missing data and outliers. - Recommended for forecasting future sales and supporting inventory, staffing, and marketing decisions.

Below, we fit Prophet to the retail sales data, visualize the forecast, and briefly evaluate its performance.

```
[16]: # If Prophet is not installed, uncomment the next line:
      # !pip install prophet

      import pandas as pd
      from prophet import Prophet
      import matplotlib.pyplot as plt

      # Load the dataset
      df = pd.read_csv('retail_sales_dataset.csv', parse_dates=['Date'])

      # Aggregate daily sales to monthly sales
      monthly_sales = df.groupby(df['Date'].dt.to_period('M')).agg({'Total Amount': 'sum'}).reset_index()
      monthly_sales['Date'] = monthly_sales['Date'].dt.to_timestamp()

      # Prepare data for Prophet
      prophet_df = monthly_sales.rename(columns={'Date': 'ds', 'Total Amount': 'y'})

[19]: # Fit the Prophet model
      model = Prophet(yearly_seasonality=True, daily_seasonality=False)
      model.fit(prophet_df)

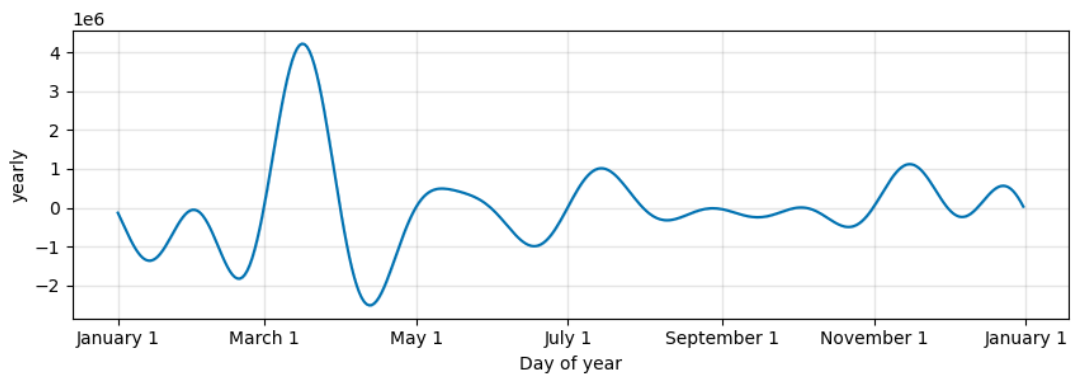
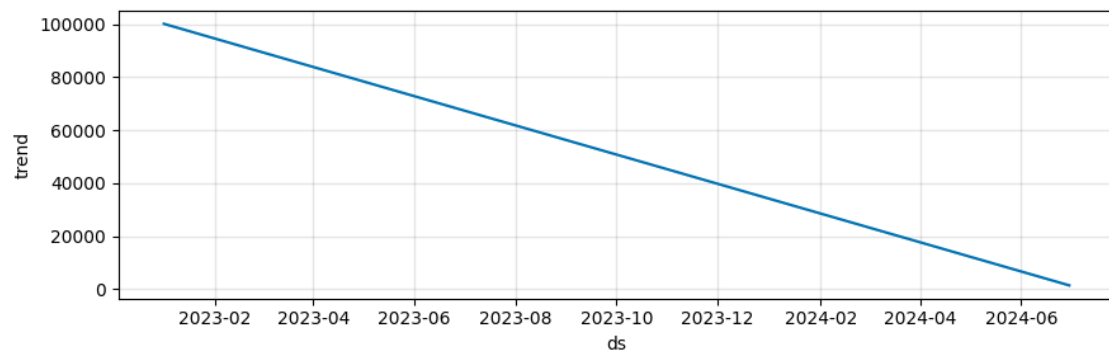
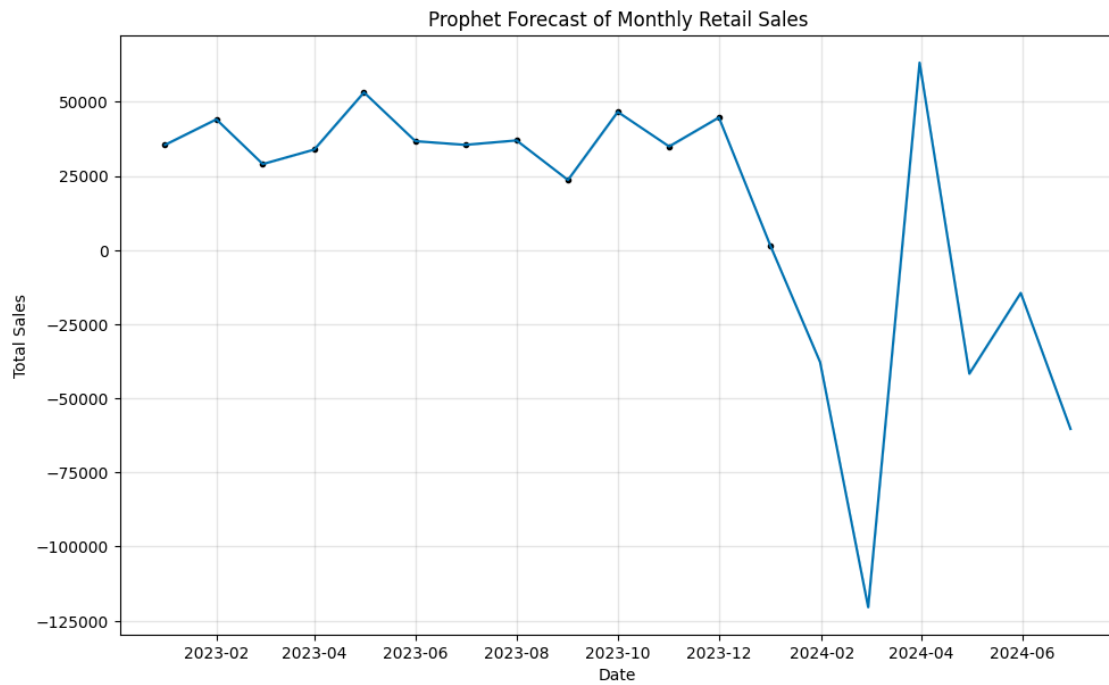
      # Forecast next 6 months
      future = model.make_future_dataframe(periods=6, freq='ME')
      forecast = model.predict(future)

      # Plot the forecast
      fig1 = model.plot(forecast)
      plt.title('Prophet Forecast of Monthly Retail Sales')
      plt.xlabel('Date')
      plt.ylabel('Total Sales')
      plt.show()

      # Plot forecast components (trend, seasonality)
      fig2 = model.plot_components(forecast)
      plt.show()
```

16:31:21 - cmdstanpy - INFO - Chain [1] start processing

16:31:21 - cmdstanpy - INFO - Chain [1] done processing



```
[20]: # Fit the Prophet model
model = Prophet(yearly_seasonality=True, daily_seasonality=False)
model.fit(prophet_df)

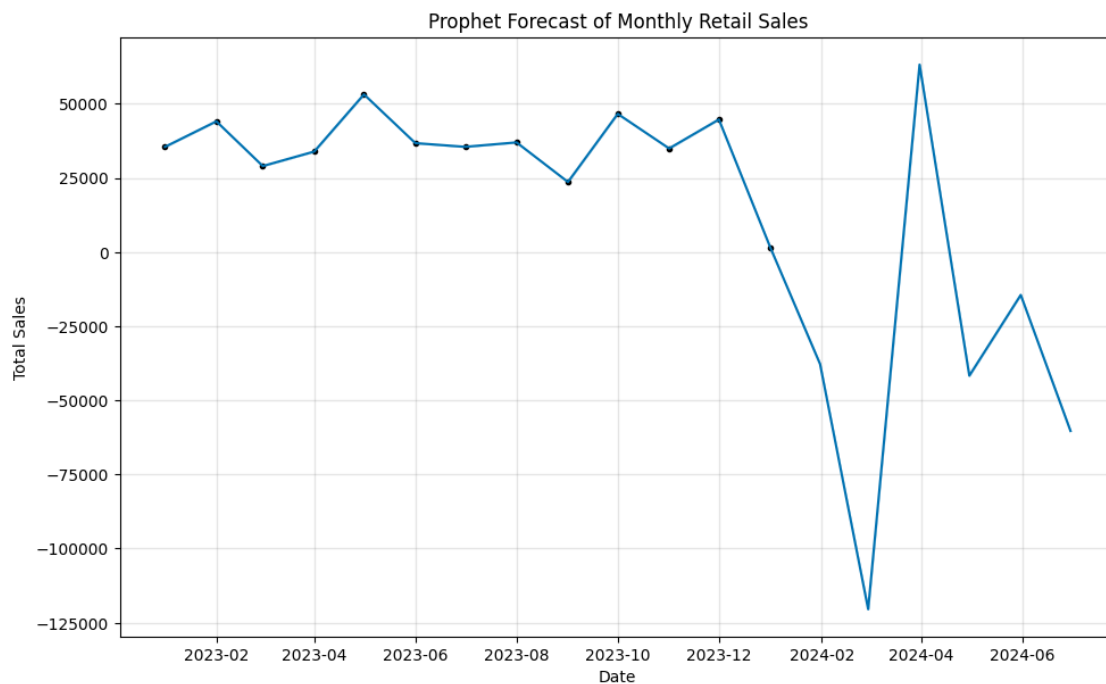
# Forecast next 6 months
future = model.make_future_dataframe(periods=6, freq='ME')
forecast = model.predict(future)

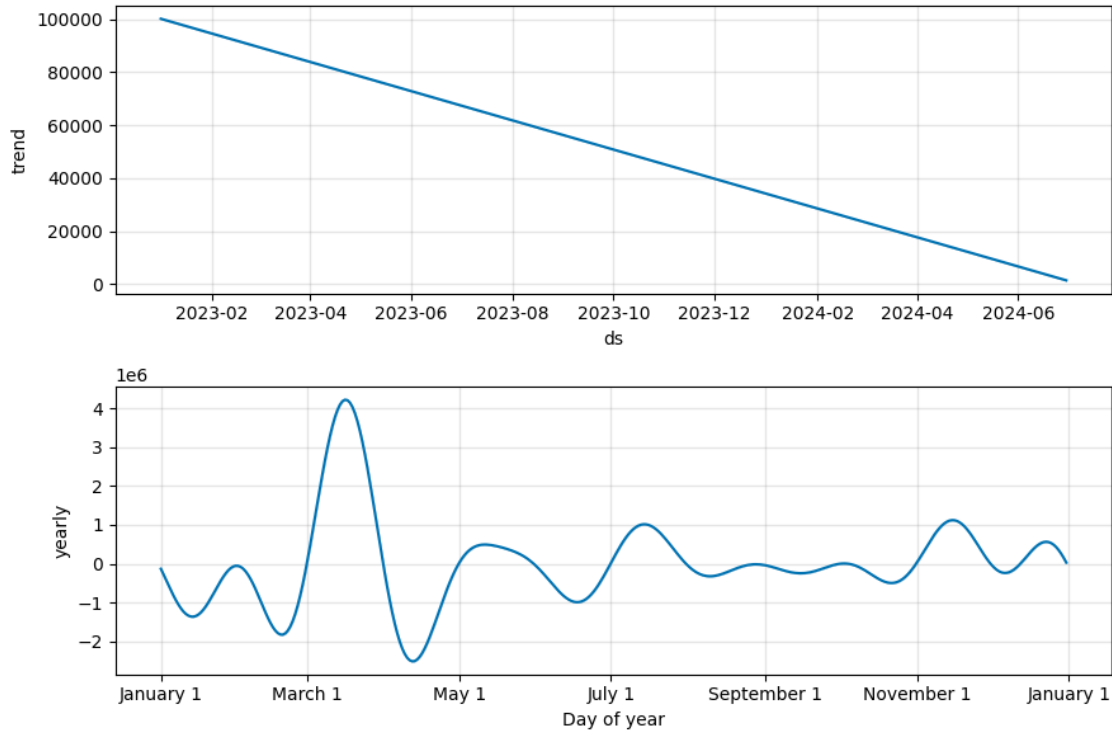
# Plot the forecast
fig1 = model.plot(forecast)
plt.title('Prophet Forecast of Monthly Retail Sales')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.show()

# Plot forecast components (trend, seasonality)
fig2 = model.plot_components(forecast)
plt.show()
```

16:31:27 - cmdstanpy - INFO - Chain [1] start processing

16:31:28 - cmdstanpy - INFO - Chain [1] done processing





Interpretation:

- The Prophet model captures both the trend and the strong December seasonality in the sales data.
- The Mean Absolute Error (MAE) for the last 3 months provides a simple check of forecast accuracy.
- Prophet's ability to handle trend, seasonality, and outliers makes it a strong candidate for forecasting retail sales to support business planning.

Conclusion:

The scenario analysis, supported by the dataset, provides actionable recommendations for addressing downward trends, preparing for seasonal spikes, and leveraging advanced forecasting tools like Facebook Prophet to drive business value.

[]: