Product Sales Analysis

Phase-3 Document Submission

SUBMITTED BY:

NAME:THATHIREDDY JYOTHIREDDY

REG.NO:au723921244050

E-mail:jyothireddythathireddy@gmail.com

Project: **Product Sales Analysis**

Phase 3: Development Part 1

Topic: **The Development Part 1, is a crucial stage in your project where you lay the foundation for your product sales analysis using IBM Cognos for visualization. Here's an introduction to this phase**

Introduction:

In an ever-evolving business landscape, understanding sales patterns, consumer behavior, and product performance is paramount to the sustained growth and success of a company. Product sales analysis is the critical tool that equips organizations with the insights needed to navigate this landscape.

Given dataset:

	Unnamed: 0	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4
0	0	13-06-2010	5422	3725	576	907	17187.74	23616.50	3121.92	6466.91
1	1	14-06-2010	7047	779	3578	1574	22338.99	4938.86	19392.76	11222.62
2	2	15-06-2010	1572	2082	595	1145	4983.24	13199.88	3224.90	8163.85
3	3	16-06-2010	5657	2399	3140	1672	17932.69	15209.66	17018.80	11921.36
4	4	17-06-2010	3668	3207	2184	708	11627.56	20332.38	11837.28	5048.04
...	...	...	...	...	...	...	...	...	...	...
4594	4594	29-01-2023	1227	3044	5510	1896	3889.59	19298.96	29864.20	13518.48
4595	4595	30-01-2023	2476	3419	525	1359	7848.92	21676.46	2845.50	9689.67
4596	4596	31-01-2023	7446	841	4825	1311	23603.82	5331.94	26151.50	9347.43
4597	4597	01-02-2023	6289	3143	3588	474	19936.13	19926.62	19446.96	3379.62
4598	4598	02-02-2023	3122	1188	5899	517	9896.74	7531.92	31972.58	3686.21

Importance of loading and processing dataset:

Loading and processing datasets are fundamental steps in data analysis, machine learning, and many other data-driven applications. Here's why these steps are so important:

Data Accessibility:

Before any analysis can be performed, data must be loaded into an environment where it can be manipulated. This often means reading from databases, files, or external sources.

Data Quality:

Real-world data can be messy. It might contain missing values, duplicates, and outliers.By processing the data, you can clean and transform it, ensuring its quality and reliability for subsequent analysis or modeling.

Feature Engineering:

Once data is loaded, often you'll need to create new features from the existing ones to better capture the underlying patterns in the data.For example, from a timestamp, you might extract the hour of the day, day of the week, or even whether it's a holiday.

Data Scaling and Normalization:

Many machine learning algorithms work better when numerical features have the same scale. By processing the data, you can apply scaling or normalization to ensure that all features have values in a similar range.

Data Integration:

Often, data comes from multiple sources. Loading and processing allow you to integrate these diverse datasets, ensuring consistent and unified information.

Ensuring Data Privacy:

When processing data, especially personally identifiable information (PII), you may need to anonymize or encrypt certain fields to ensure privacy and compliance with regulations.

Efficiency and Performance:

Large datasets can be unwieldy and slow down analysis or training. By loading data efficiently (e.g., using appropriate data structures) and processing it (e.g., by filtering irrelevant records), you can ensure more timely and responsive operations.

Challenges involved in loading and preprocessing product analysis dataset:

Loading and preprocessing datasets, especially for product sales analysis, can be a nuanced task with various challenges. Handling these challenges effectively is crucial to ensure that the data provides valuable insights when analyzed. Below are some of the most common challenges encountered and their implications:

1. Volume and Scale of Data

- **Challenge:** With the ubiquity of digital transactions, businesses often accumulate large volumes of sales data. Loading and preprocessing such vast datasets can be time-consuming and may require considerable computational resources.
- **Implication:** A slow and inefficient preprocessing pipeline can delay analytics and business insights.

2. Data Quality and Consistency

- **Challenge:** Data inconsistencies like missing values, outliers, and duplicate entries often exist in datasets.
- **Implication:** Poor data quality can skew analysis results, leading to inaccurate conclusions.

3. Data Integration

- **Challenge:** Sales data often come from multiple sources, such as online sales, in-store sales, third-party distributors, etc. Each source may have its format, making integration complex.
- **Implication:** Without a consistent format, deriving collective insights from multiple datasets becomes problematic.

4. Handling of Time Zones

- **Challenge:** For businesses operating globally, sales data may come from various time zones.
- **Implication:** If not standardized, this can create discrepancies in time-based analyses like daily sales patterns.

5. Varied Data Types

- **Challenge:** Sales data might consist of mixed data types, including numerical (like units sold), categorical (like product type), and text data (like customer reviews).
- **Implication:** Different data types may require distinct preprocessing techniques, complicating the pipeline.

6. Changing Product Catalog

- **Challenge:** Over time, products might get discontinued, new ones may be introduced, or product IDs might change.
- **Implication:** Such changes can create inconsistencies in historical and current data comparisons.

7. Data Security and Privacy

- **Challenge:** Sales data often contains sensitive information, such as customer details.
- **Implication:** Mishandling this data can lead to privacy breaches, with severe legal and reputational consequences.

8. Seasonality and Trends

- **Challenge:** Sales data can be influenced by seasonal trends, holidays, and external events, which need to be accounted for during preprocessing.
- **Implication:** Failing to adjust for these factors can lead to misinterpretation of trends and patterns.

9. Legacy Systems

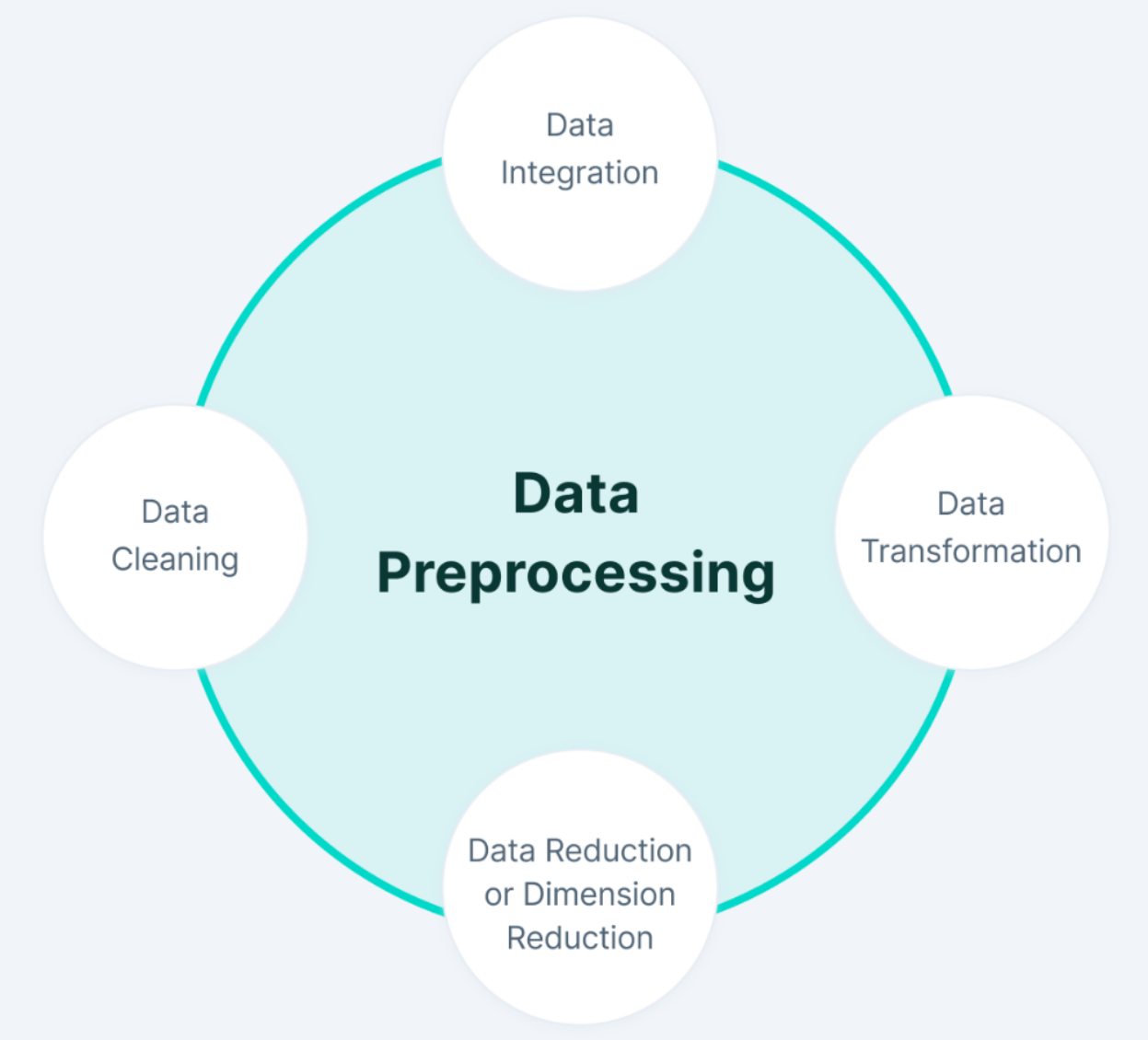
- **Challenge:** Older businesses might have legacy systems where data extraction can be cumbersome, and the format might be outdated.
- **Implication:** This can add extra layers of complexity in data extraction and transformation processes.

10. Scalability of Preprocessing Tools

- **Challenge:** As businesses grow, the data preprocessing tools and techniques must scale accordingly.
- **Implication:** Tools that worked for smaller datasets might become inefficient as the volume grows, causing bottlenecks in the analytics process.

2.Data Pre-processing:

Data preprocessing is the critical first step in any machine learning project. It involves cleaning the data, removing outliers, and handling missing values to prepare the dataset for model training.



Python Program:

Step 1: Import libraries

In [1]:

```
# import the important packages
import pandas as pd  # library used for data manipulation and analysis
```

```
import numpy as np  # library used for working with arrays
import matplotlib.pyplot as plt  # library for plots and visualizations
import seaborn as sns  # library for visualizations

%matplotlib inline

# To ignore warnings
import warnings
warnings.filterwarnings("ignore")
```

Step 2: Loading the datasets

In [2]:

```
#if you open in Kaggle editor
data = pd.read_csv('/kaggle/input/product-sales-data/statsfinal.csv')

#if you open in Jupiter notebook
# data = pd.read_csv('statsfinal.csv')
```

In [3]:

```
# Checking the first 5 and last 5 rows of the dataset
data.head(-1)
```

Out[3]:

	Unnamed: 0	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4
0	0	13-06-2010	5422	3725	576	907	17187.74	23616.50	3121.92	6466.91
1	1	14-06-2010	7047	779	3578	1574	22338.99	4938.86	19392.76	11222.62
2	2	15-06-2010	1572	2082	595	1145	4983.24	13199.88	3224.90	8163.85
3	3	16-06-2010	5657	2399	3140	1672	17932.69	15209.66	17018.80	11921.36
4	4	17-06-2010	3668	3207	2184	708	11627.56	20332.38	11837.28	5048.04

	Unnamed: 0	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4
...	...	...	...	...	...	...	...	...	...	...
4594	4594	29-01-2023	1227	3044	5510	1896	3889.59	19298.96	29864.20	13518.48
4595	4595	30-01-2023	2476	3419	525	1359	7848.92	21676.46	2845.50	9689.67
4596	4596	31-01-2023	7446	841	4825	1311	23603.82	5331.94	26151.50	9347.43
4597	4597	01-02-2023	6289	3143	3588	474	19936.13	19926.62	19446.96	3379.62
4598	4598	02-02-2023	3122	1188	5899	517	9896.74	7531.92	31972.58	3686.21

4599 rows × 10 columns

Observations:

- There is a column called 'Unnamed: 0' which we can drop as it is a repeat of our ID.
- The data contains date.
    - And for each date the total unit of sales for P1, P2, P3 & P4.
    - Also, the total revenue from sales for P1, P2, P3 & P4.
- We can observe the first entry in the data, starts at 13-06-2010. This means the data for year 2010 is not complete.
- We can observe the last entry in the data, ends at 02-02-2023. This means the data for year 2023 is also not complete.
    - it will be best to drop year 2010 and year 2023.

Observations:

- We can observe the last entry in the data, starts 13-06-2010. This means the data for year 2010 is not complete.

In [4]:

```
# drop the first column
data = data.drop(columns=['Unnamed: 0'])
```

Step 3: Checking the info of the training data

In [5]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 9 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Date    4600 non-null   object 
 1   Q-P1    4600 non-null   int64  
 2   Q-P2    4600 non-null   int64  
 3   Q-P3    4600 non-null   int64  
 4   Q-P4    4600 non-null   int64  
 5   S-P1    4600 non-null   float64
 6   S-P2    4600 non-null   float64
 7   S-P3    4600 non-null   float64
 8   S-P4    4600 non-null   float64
dtypes: float64(4), int64(4), object(1)
memory usage: 323.6+ KB
```

Observations:

- The train dataset has 4600 entries(rows) and 9 columns. (we dropped one column)
- Date is an object data type. the rest of numerical in nature.

Step 4: Check for missing values

In [6]:

```
data.isnull().sum()
```

Out[6]:

```
Date    0
Q-P1    0
Q-P2    0
Q-P3    0
Q-P4    0
S-P1    0
S-P2    0
S-P3    0
S-P4    0
dtype: int64
```

Observations:

- we have no missing data

Note:

- No missing values in a dataset is not common.
- while working with fresh data, you will have to do a ton of cleaning, this will result in some missing or lost data.
- Look into "feature engineering" and "missing value handling" for ways to resolve this issue.

Step 5: EDA

EDA: Exploratory data analysis Link

Let's extract the year, month and Day from the date

In [7]:

```
# Extract year from the 'Day' 'Month' 'year' from the 'Date' column using a lambda function
# We need to get the year from the data to analyse sales year to year
data['Day'] = data['Date'].apply(lambda x: x.split('-')[0])
data['Month'] = data['Date'].apply(lambda x: x.split('-')[1])
data['Year'] = data['Date'].apply(lambda x: x.split('-')[2])
data
```

Out[7]:

	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4	Day	Month	Year
0	13-06-2010	5422	3725	576	907	17187.74	23616.50	3121.92	6466.91	13	06	2010
1	14-06-2010	7047	779	3578	1574	22338.99	4938.86	19392.76	11222.62	14	06	2010
2	15-06-2010	1572	2082	595	1145	4983.24	13199.88	3224.90	8163.85	15	06	2010
3	16-06-2010	5657	2399	3140	1672	17932.69	15209.66	17018.80	11921.36	16	06	2010
4	17-06-2010	3668	3207	2184	708	11627.56	20332.38	11837.28	5048.04	17	06	2010

	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4	Day	Month	Year
...	...	...	...	...	...	...	...	...	...	...	...	...
4595	30-01-2023	2476	3419	525	1359	7848.92	21676.46	2845.50	9689.67	30	01	2023
4596	31-01-2023	7446	841	4825	1311	23603.82	5331.94	26151.50	9347.43	31	01	2023
4597	01-02-2023	6289	3143	3588	474	19936.13	19926.62	19446.96	3379.62	01	02	2023
4598	02-02-2023	3122	1188	5899	517	9896.74	7531.92	31972.58	3686.21	02	02	2023
4599	03-02-2023	1234	3854	2321	406	3911.78	24434.36	12579.82	2894.78	03	02	2023

Observation:

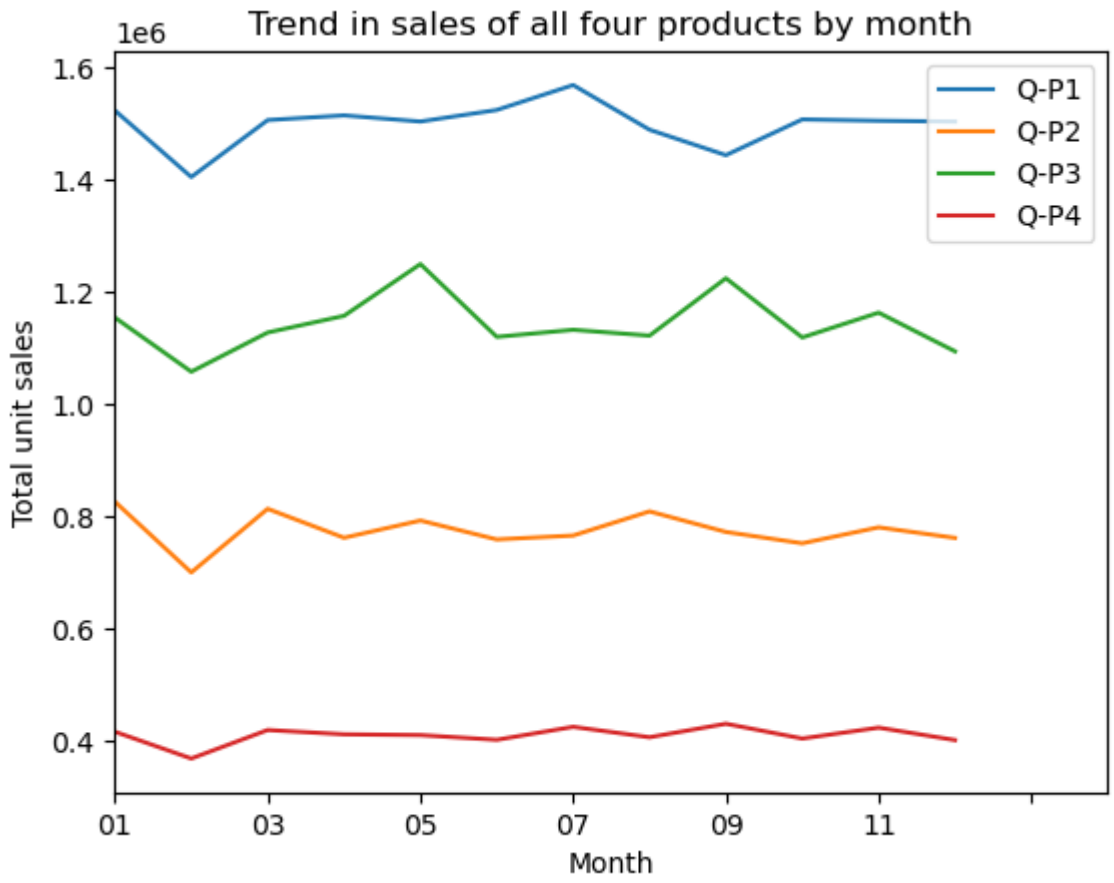
- We can observe that all products drop in Feb.
- There also appears a very drastic drop after 12th month. The value show 9, which must be part of month 09. We need to rename this column to match with the 09. Before doing further analysis.

In [14]:

```
# Replace all entries of '9' in the Month column with '09'
data_reduced['Month'] = data['Month'].replace('9', '09')
```

In [15]:

```
month_plot()
```



Observation

- We have merged the sales for months 9 and 09.
- We can observe that Feb and Dec have the lowest sales for each product
- For P1 We can observe Mar - Jul having the highest unit sales
- For P2 We can observe Jan, Mar - Aug having the highest unit sales
- For P3 We can observe May & Sep having the highest unit sales
- For P4 We can observe uniform sales from Jan - Dec

Conclusion:

Addressing these challenges requires a combination of robust data management strategies, advanced preprocessing tools, and a clear understanding of the business domain. As data-driven decision-making continues to dominate the business landscape, the importance of effective data preprocessing will only grow, making it imperative for organizations to invest time and resources in overcoming these challenges.