Group 12 Project Documentation
Sales Analytics

Advanced Data Cleaning

Objective
To ensure data quality by identifying and handling outliers, addressing missing values, and correcting data type inconsistencies.

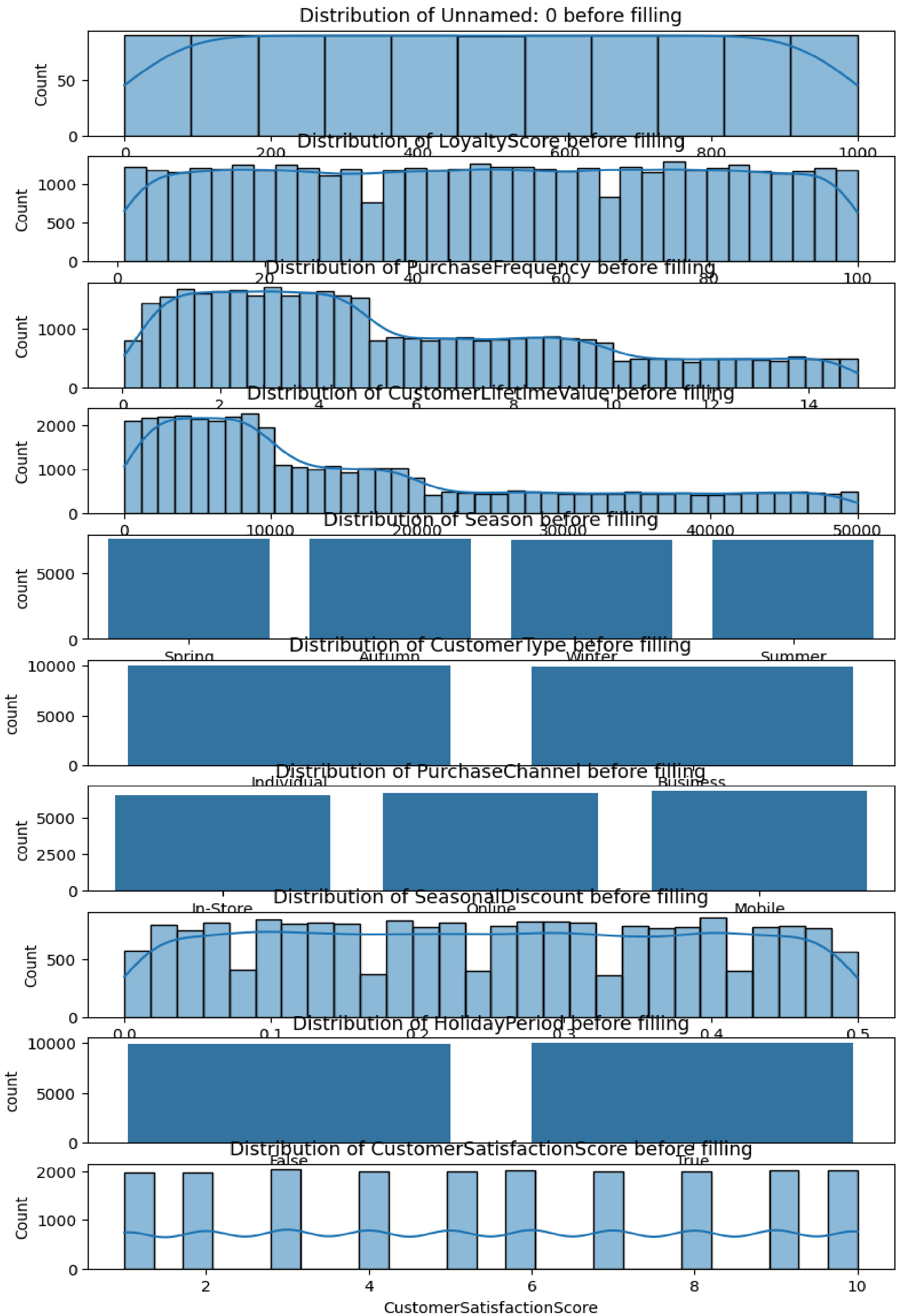
Steps and Justifications:

1. **Loading the Dataset**
   - Loaded the dataset using Pandas to facilitate data manipulation and cleaning.

2. **Identifying Outliers**
   - Used the Interquartile Range (IQR) method to detect outliers. This method is chosen for its robustness in identifying extreme values without being influenced by them.
   - Calculated the first quartile (Q1) and third quartile (Q3) and determined the IQR as Q3 - Q1.
   - Outliers are defined as data points that fall below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR.

Output:

```
Missing values before imputation:
 Unnamed: 0                    40000
CustomerID                        0
Age                               0
Gender                            0
Location                          0
ProductCategory                   0
PurchaseDate                      0
PurchaseAmount                    0
PaymentMethod                     0
Quantity                          0
DiscountPercentage                0
IsReturned                        0
Rating                            0
IsPromotion                       0
CustomerSegment                   0
ShippingDuration                  0
Region                            0
LoyaltyScore                   1000
PurchaseFrequency              1000
CustomerLifetimeValue          1000
Season                        11000
CustomerType                  21000
PurchaseChannel               21000
SeasonalDiscount              21000
HolidayPeriod                 21000
CustomerSatisfactionScore     21000
dtype: int64
```

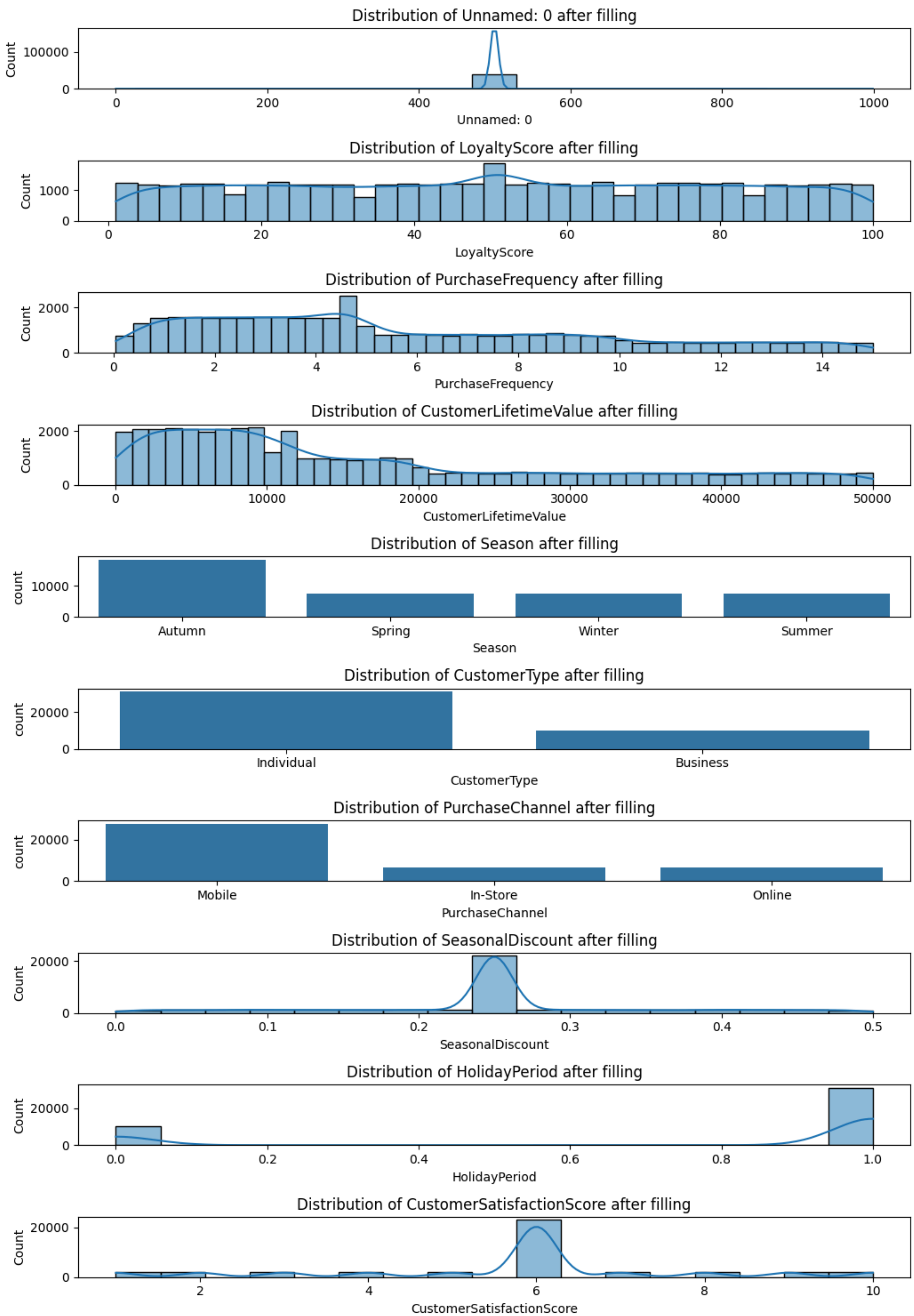

Distribution of Unnamed: 0 before filling
Count
0
50
0
200
400
600
800
1000
Distribution of LoyaltyScore before filling
Count
0
500
1000
0
20
40
60
80
100
Distribution of PurchaseFrequency before filling
Count
0
1000
0
2
4
6
8
10
12
14
Distribution of CustomerLifetimeValue before filling
Count
0
1000
2000
0
10000
20000
30000
40000
50000
Distribution of Season before filling
count
0
5000
Spring
Autumn
Winter
Summer
Distribution of CustomerType before filling
count
0
5000
10000
Individual
Business
Distribution of PurchaseChannel before filling
count
0
2500
5000
In-Store
Online
Mobile
Distribution of SeasonalDiscount before filling
Count
0
500
0.0
0.1
0.2
0.3
0.4
0.5
Distribution of HolidayPeriod before filling
count
0
5000
10000
False
True
Distribution of CustomerSatisfactionScore before filling
Count
0
1000
2000
2
4
6
8
10
CustomerSatisfactionScore

3. **Handling Missing Values**

- Applied different imputation techniques based on the nature of the data:
  - **Mean Imputation**: Used for numerical data where the mean is appropriate.
  - **Median Imputation**: Chosen for skewed numerical data to avoid mean distortion.
  - **Mode Imputation**: Used for categorical data to fill in the most frequent value.

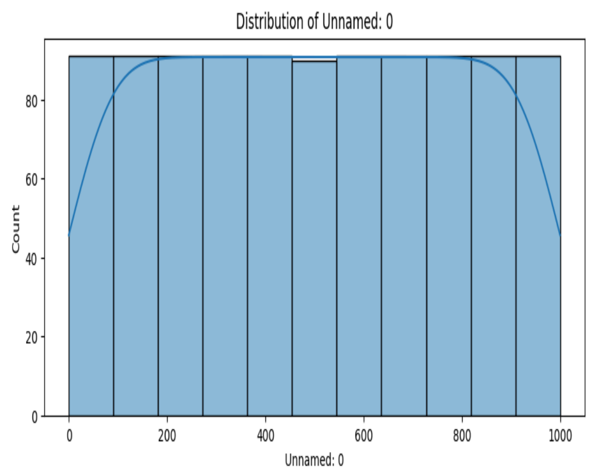
Output:

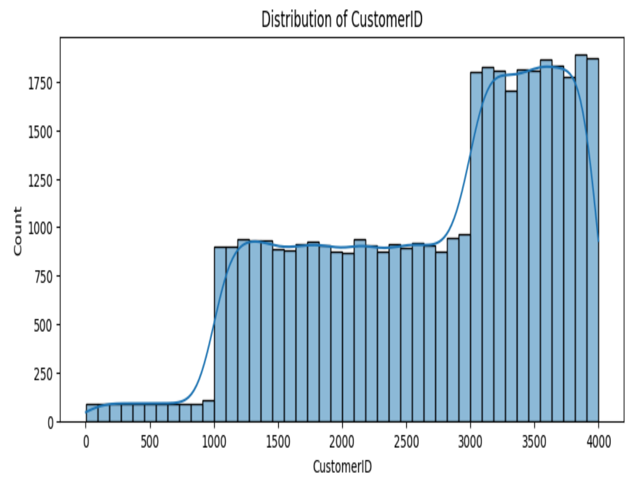
```
Missing values after imputation:
 Unnamed: 0                    0
CustomerID                    0
Age                           0
Gender                        0
Location                      0
ProductCategory               0
PurchaseDate                  0
PurchaseAmount                0
PaymentMethod                 0
Quantity                      0
DiscountPercentage            0
IsReturned                    0
Rating                        0
IsPromotion                   0
CustomerSegment               0
ShippingDuration              0
Region                        0
LoyaltyScore                  0
PurchaseFrequency             0
CustomerLifetimeValue         0
Season                        0
CustomerType                  0
PurchaseChannel               0
SeasonalDiscount              0
HolidayPeriod                 0
CustomerSatisfactionScore     0
dtype: int64
```

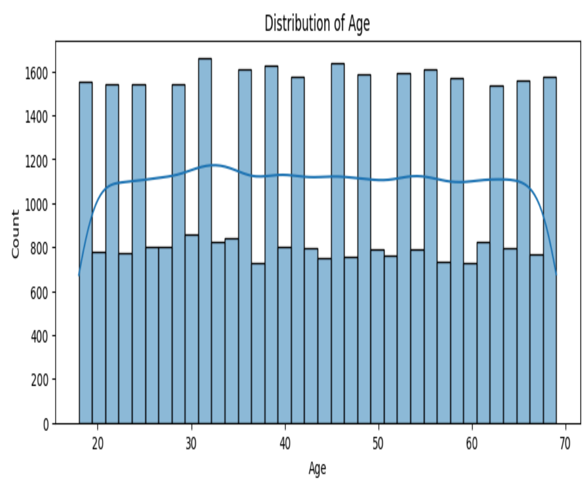


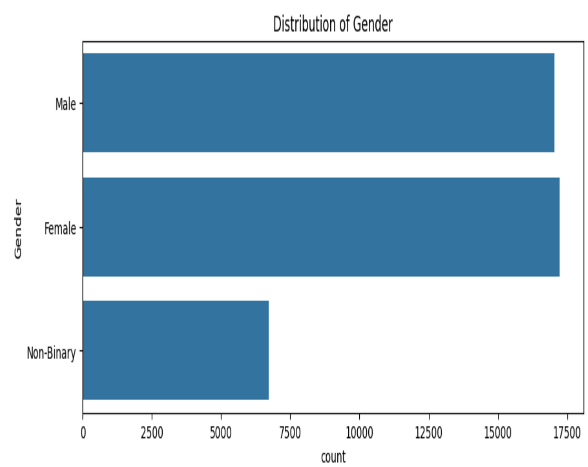
Distribution of Unnamed: 0 after filling
Count
100000
0
0
200
400
600
800
1000
Unnamed: 0
Distribution of LoyaltyScore after filling
Count
1000
0
0
20
40
60
80
100
LoyaltyScore
Distribution of PurchaseFrequency after filling
Count
2000
0
0
2
4
6
8
10
12
14
PurchaseFrequency
Distribution of CustomerLifetimeValue after filling
Count
2000
0
0
10000
20000
30000
40000
50000
CustomerLifetimeValue
Distribution of Season after filling
count
10000
0
Autumn
Spring
Winter
Summer
Season
Distribution of CustomerType after filling
count
20000
0
Individual
Business
CustomerType
Distribution of PurchaseChannel after filling
count
20000
0
Mobile
In-Store
Online
PurchaseChannel
Distribution of SeasonalDiscount after filling
Count
20000
0
0.0
0.1
0.2
0.3
0.4
0.5
SeasonalDiscount
Distribution of HolidayPeriod after filling
Count
20000
0
0.0
0.2
0.4
0.6
0.8
1.0
HolidayPeriod
Distribution of CustomerSatisfactionScore after filling
Count
20000
0
2
4
6
8
10
CustomerSatisfactionScore

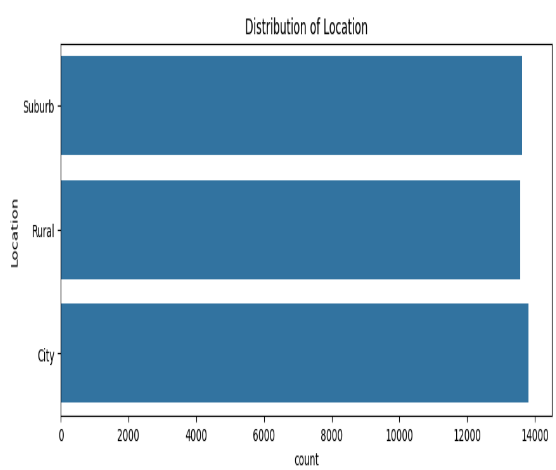
4. **Correcting Data Type Inconsistencies**
 - Ensured that each column had a consistent data type:
   - Converted date columns to datetime objects.
   - Changed numerical columns stored as strings to appropriate numerical types.
 - Verified and corrected any misclassified data types.

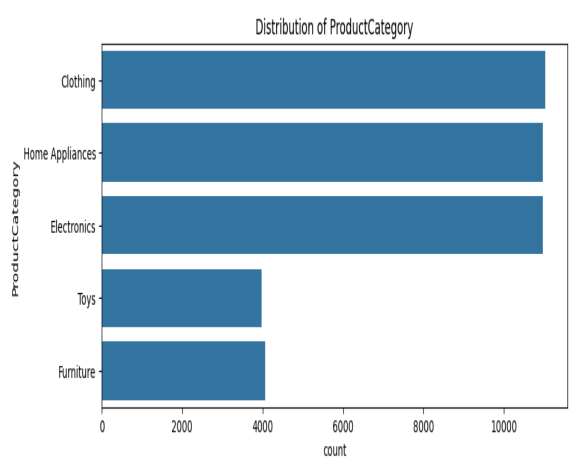


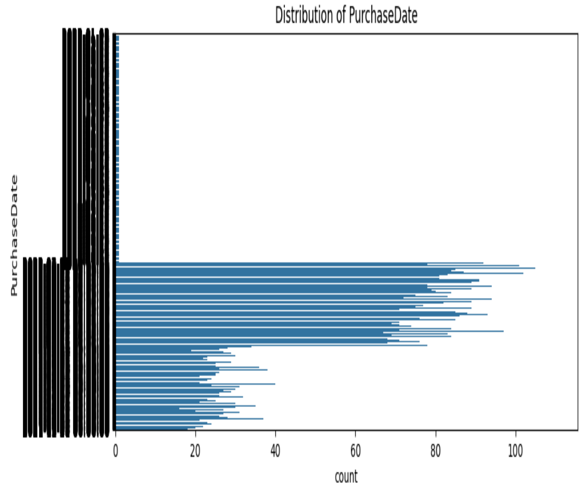


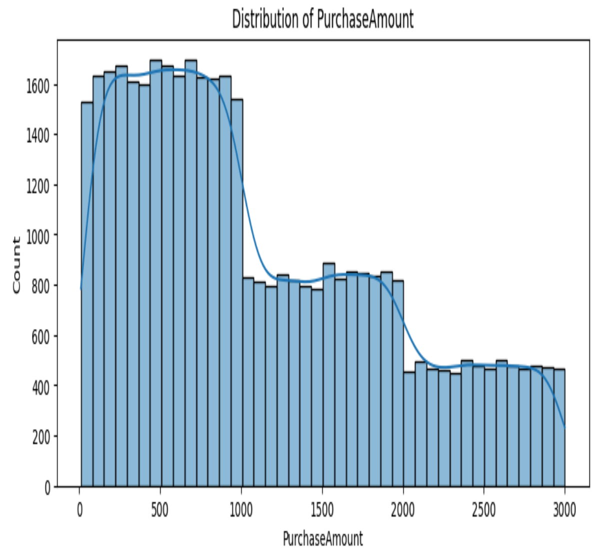


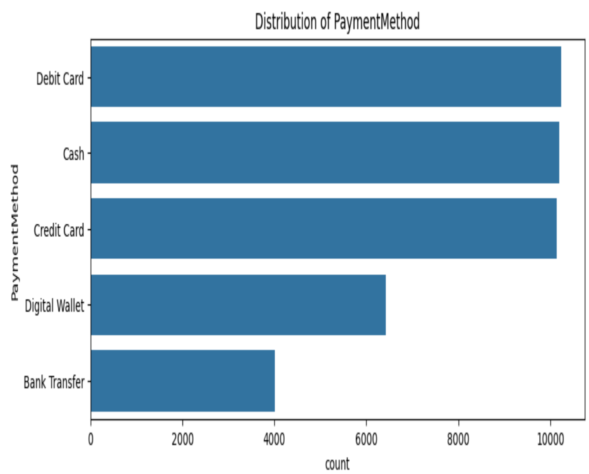


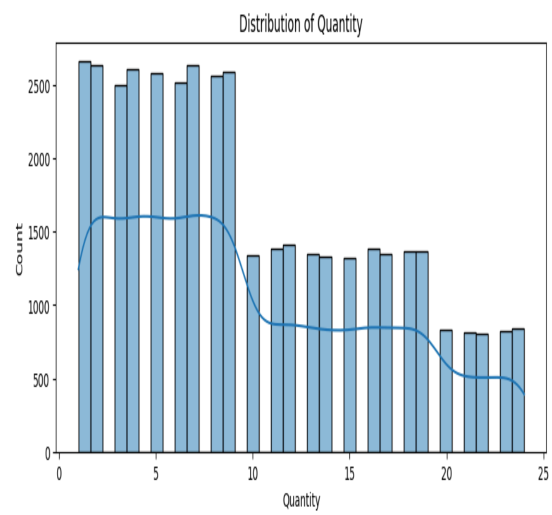


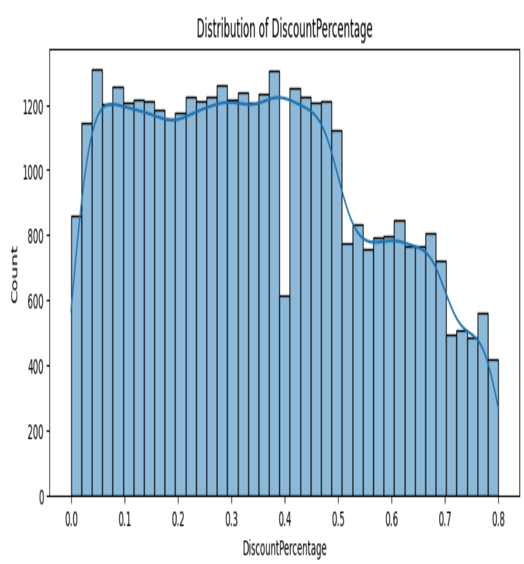


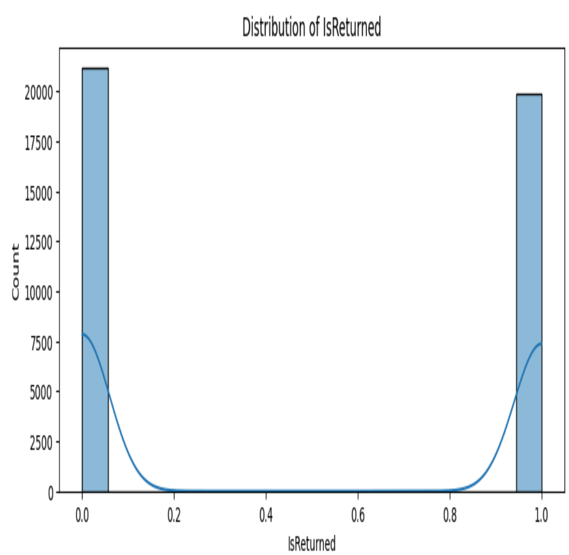

Distribution of PurchaseDate
PurchaseDate
count
0
20
40
60
80
100

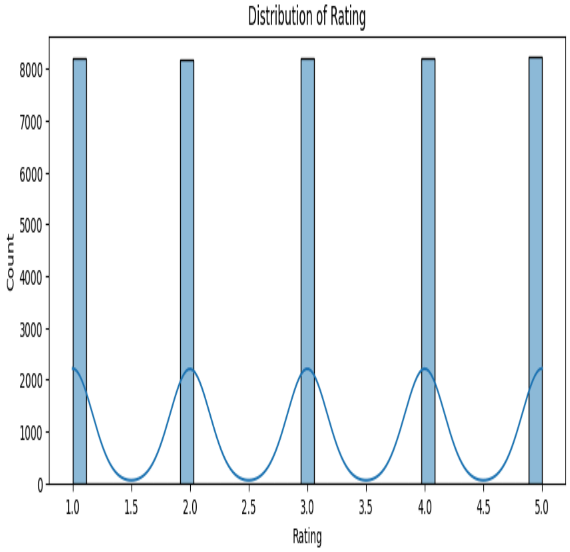

Distribution of PurchaseAmount
Count
PurchaseAmount
0
200
400
600
800
1000
1200
1400
1600
0
500
1000
1500
2000
2500
3000

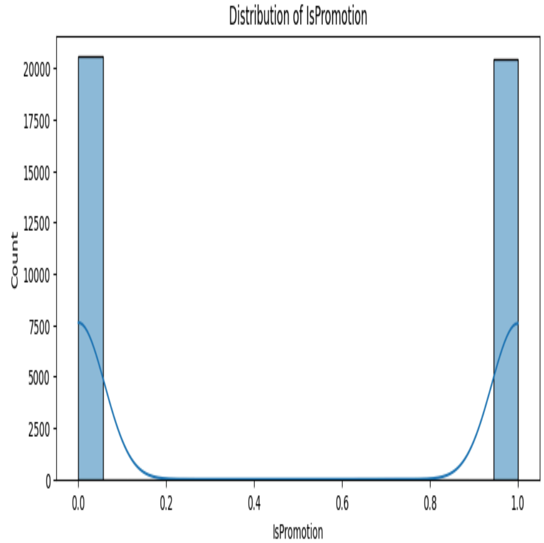

Distribution of PaymentMethod
PaymentMethod
Debit Card
Cash
Credit Card
Digital Wallet
Bank Transfer
count
0
2000
4000
6000
8000
10000

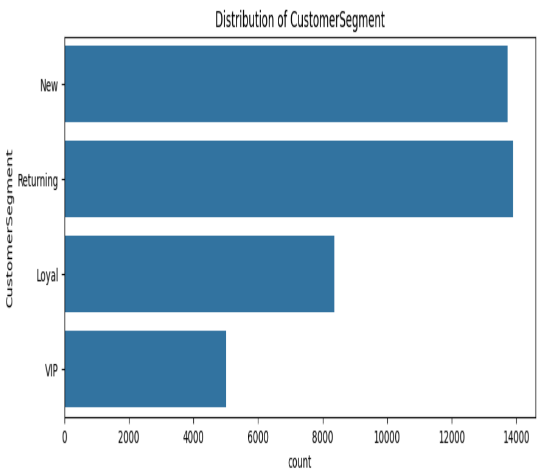

Distribution of Quantity
Count
Quantity
0
500
1000
1500
2000
2500
0
5
10
15
20
25

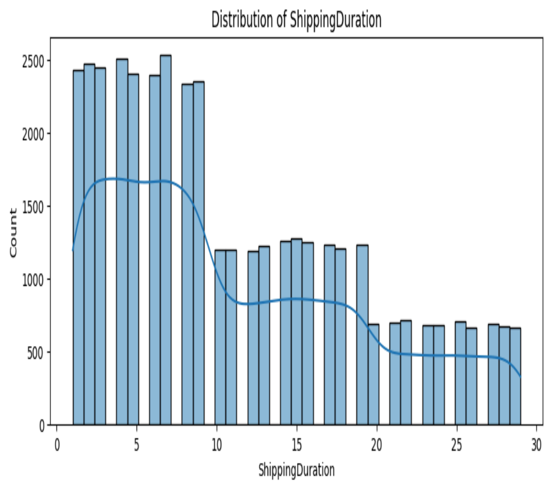

Distribution of DiscountPercentage
Count
DiscountPercentage
0
200
400
600
800
1000
1200
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

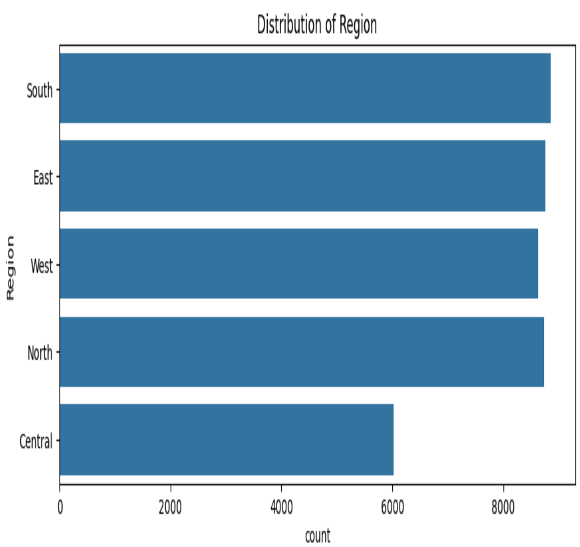

Distribution of IsReturned
Count
IsReturned
0
2500
5000
7500
10000
12500
15000
17500
20000
0.0
0.2
0.4
0.6
0.8
1.0

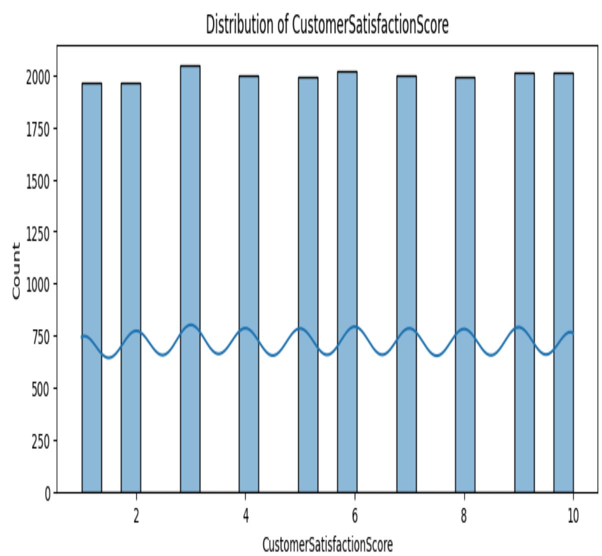

Distribution of Rating
Count
0
1000
2000
3000
4000
5000
6000
7000
8000
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Rating


Distribution of IsPromotion
Count
0
2500
5000
7500
10000
12500
15000
17500
20000
0.0
0.2
0.4
0.6
0.8
1.0
IsPromotion


Distribution of CustomerSegment
CustomerSegment
New
Returning
Loyal
VIP
0
2000
4000
6000
8000
10000
12000
14000
count


Distribution of ShippingDuration
Count
0
500
1000
1500
2000
2500
0
5
10
15
20
25
30
ShippingDuration


Distribution of Region
Region
South
East
West
North
Central
0
2000
4000
6000
8000
count


Distribution of CustomerSatisfactionScore
Count
0
250
500
750
1000
1250
1500
1750
2000
2
4
6
8
10
CustomerSatisfactionScore

Data Augmentation

Objective
To enhance the dataset by generating additional samples while maintaining the statistical properties of the original data.

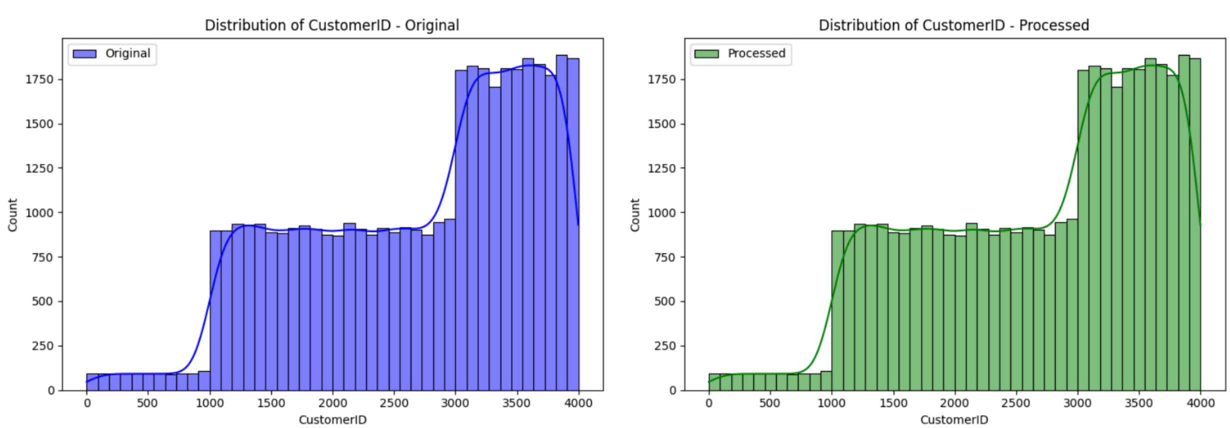
Steps and Justifications

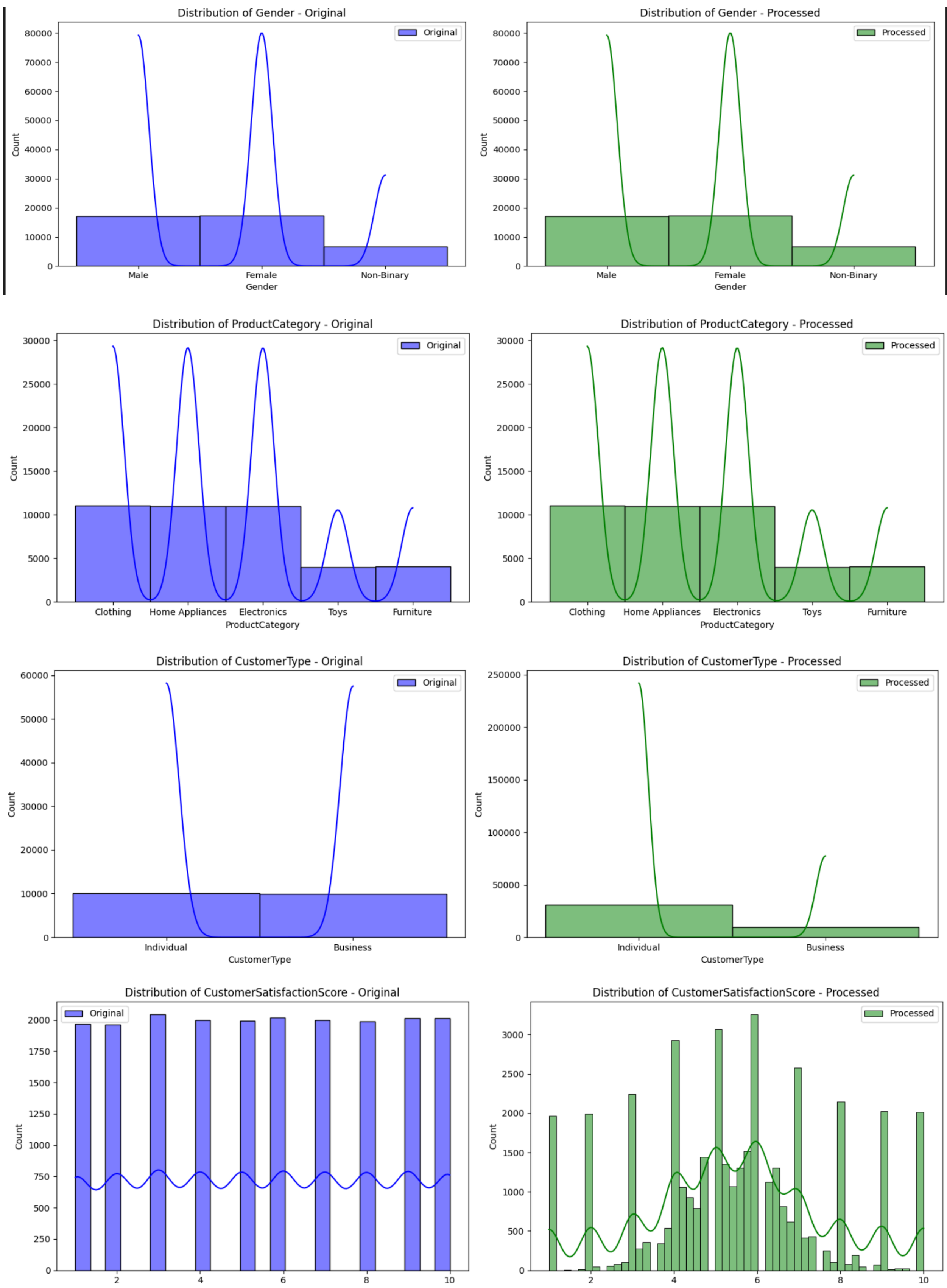
1. **Analyzing Data Distribution**
   - Conducted an in-depth analysis of the existing data's distribution to understand its characteristics.
   - Identified key statistical properties such as mean, variance, skewness, and kurtosis.

2. **Data Augmentation Techniques**
   - Applied bootstrapping to create additional samples. Bootstrapping is a resampling technique that generates new data points by sampling with replacement from the existing data.
   - Ensured that the augmented data followed the original dataset's statistical distribution.

3. **Integrating Augmented Data**
   - Merged the newly generated samples with the original dataset to create an expanded dataset.
   - Maintained the integrity of the original data while ensuring the augmented data enhanced the dataset.

4. **Validation**
   - Performed rigorous validation to ensure the augmented data met quality standards.
   - Compared statistical properties of the augmented dataset with the original to confirm consistency.




Distribution of Gender - Original
Original
Count
Male
Female
Non-Binary
Gender
Distribution of Gender - Processed
Processed
Distribution of ProductCategory - Original
Clothing
Home Appliances
Electronics
Toys
Furniture
ProductCategory
Distribution of ProductCategory - Processed
Distribution of CustomerType - Original
Individual
Business
CustomerType
Distribution of CustomerType - Processed
Distribution of CustomerSatisfactionScore - Original
CustomerSatisfactionScore
Distribution of CustomerSatisfactionScore - Processed

Real-time Data Ingestion

Objective

To set up a real-time data ingestion pipeline using Apache Kafka and ensure optimized data flow into SQL databases.

Steps and Justifications

1. **Setting Up Apache Kafka**
   - Configured an Apache Kafka environment to manage real-time data streams.
   - Established Kafka brokers, topics, and partitions to facilitate efficient data flow.

2. **Creating Kafka Producers**
   - Developed Kafka producers to simulate real-time data streams.
   - Configured producers to send data to the appropriate Kafka topics.

3. **Developing Kafka Consumers**
   - Used Python to create Kafka consumers that ingest data from Kafka topics into SQL databases.
   - Ensured consumers were optimized for high throughput and low latency to handle real-time data efficiently.

4. **Optimizing Data Ingestion**
   - Implemented strategies to minimize latency and maximize throughput.
   - Used batching and compression techniques to enhance performance.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57400 entries, 0 to 57399
Data columns (total 21 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   CustomerID                 57400 non-null  float64
 1   Age                        57400 non-null  float64
 2   Gender                     57400 non-null  object 
 3   Location                   57400 non-null  object 
 4   ProductCategory            57400 non-null  object 
 5   PurchaseDate               57400 non-null  object 
 6   PurchaseAmount             57400 non-null  float64
 7   PaymentMethod              57400 non-null  object 
 8   Quantity                   57400 non-null  float64
 9   DiscountPercentage         57400 non-null  float64
 10  IsReturned                 57400 non-null  bool   
 11  Rating                     57400 non-null  float64
 12  IsPromotion                57400 non-null  bool   
 13  CustomerSegment            57400 non-null  object 
 14  ShippingDuration           57400 non-null  float64
 15  Region                     57400 non-null  object 
 16  LoyaltyScore               57400 non-null  float64
 17  PurchaseFrequency          57400 non-null  float64
 18  CustomerLifetimeValue      57400 non-null  float64
 19  Season                     57400 non-null  object 
 20  CustomerSatisfactionScore  57400 non-null  float64
dtypes: bool(2), float64(11), object(8)
memory usage: 8.4+ MB
```

```
Inserted 1 rows into PostgreSQL
Inserted 2 rows into PostgreSQL
Inserted 3 rows into PostgreSQL
Inserted 4 rows into PostgreSQL
Inserted 5 rows into PostgreSQL
Inserted 6 rows into PostgreSQL
Inserted 7 rows into PostgreSQL
Inserted 8 rows into PostgreSQL
Inserted 9 rows into PostgreSQL
Inserted 10 rows into PostgreSQL
Inserted 11 rows into PostgreSQL
Inserted 12 rows into PostgreSQL
Inserted 13 rows into PostgreSQL
Inserted 14 rows into PostgreSQL
Inserted 15 rows into PostgreSQL
Inserted 16 rows into PostgreSQL
Inserted 17 rows into PostgreSQL
Inserted 18 rows into PostgreSQL
Inserted 19 rows into PostgreSQL
Inserted 20 rows into PostgreSQL
Inserted 21 rows into PostgreSQL
Inserted 22 rows into PostgreSQL
Finished inserting 20 rows into PostgreSQL
```

Storage Optimization

Objective

To evaluate and optimize storage formats for better efficiency and performance.

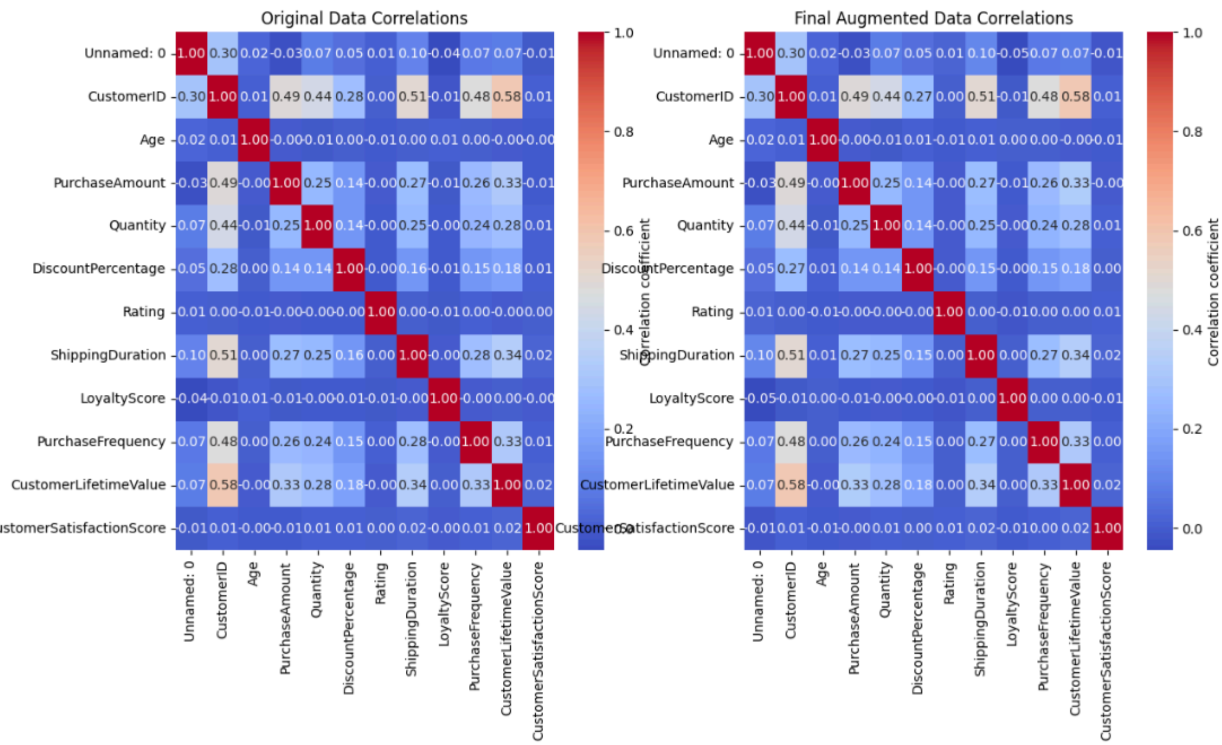
Steps and Justifications

1. **Evaluating Columnar Storage Formats**
   - Assessed columnar storage formats such as Parquet and ORC for their storage efficiency and performance.
   - Compared these formats with traditional row-based storage.

2. **Converting Dataset**
   - Converted the dataset to Parquet and ORC formats.
   - Evaluated the storage space required and the query performance for each format.

3. **Comparison and Analysis**
   - Conducted a detailed comparison of storage efficiency and query performance between columnar and row-based storage.
   - Analyzed metrics such as storage size, read/write speeds, and query response times.

```
(1.0, 56.0, 'Male', 'Suburb', 'Clothing', datetime.datetime(2022, 1, 1, 0, 0, tzinfo=zoneinfo.ZoneInfo(key='UTC')), 948.211
1457, 'Debit Card', 5.0, 0.300396075, False, 2.0, False, 'New', 2.0, 'South', 57.6, 3.07, 5476.866, 'Autumn', 4.2)
(2.0, 46.0, 'Female', 'Rural', 'Home Appliances', datetime.datetime(2022, 2, 1, 0, 0, tzinfo=zoneinfo.ZoneInfo(key='UTC')),
81.59331051, 'Cash', 5.0, 0.350432849, False, 3.0, False, 'Returning', 4.0, 'East', 69.6, 2.5340000000000003, 2392.576, 'Au
tumn', 6.2)
(3.0, 32.0, 'Female', 'Suburb', 'Home Appliances', datetime.datetime(2022, 3, 1, 0, 0, tzinfo=zoneinfo.ZoneInfo(key='UT
C')), 955.5640552, 'Debit Card', 8.0, 0.142602901, False, 5.0, False, 'Returning', 5.0, 'South', 52.2, 3.4120000000000004,
5059.606, 'Autumn', 4.2)
(4.0, 60.0, 'Female', 'Suburb', 'Electronics', datetime.datetime(2022, 4, 1, 0, 0, tzinfo=zoneinfo.ZoneInfo(key='UTC')), 52
7.3508333, 'Cash', 9.0, 0.31538521, False, 4.0, False, 'New', 2.0, 'West', 83.2, 3.732, 6884.0340000000015, 'Autumn', 6.0)
(5.0, 25.0, 'Male', 'Suburb', 'Clothing', datetime.datetime(2022, 5, 1, 0, 0, tzinfo=zoneinfo.ZoneInfo(key='UTC')), 306.570
0212, 'Credit Card', 9.0, 0.227824775, False, 5.0, False, 'Returning', 5.0, 'North', 62.2, 3.284, 4223.393999999999, 'Autum
n', 6.2)
(6.0, 38.0, 'Male', 'Suburb', 'Home Appliances', datetime.datetime(2022, 6, 1, 0, 0, tzinfo=zoneinfo.ZoneInfo(key='UTC')),
86.09345452, 'Credit Card', 6.0, 0.24100867, False, 4.0, False, 'Returning', 1.0, 'East', 69.6, 2.534, 2392.576, 'Autumn',
6.2)
(7.0, 56.0, 'Male', 'City', 'Home Appliances', datetime.datetime(2022, 7, 1, 0, 0, tzinfo=zoneinfo.ZoneInfo(key='UTC')), 50
5.6180254, 'Cash', 6.0, 0.37888828, False, 4.0, False, 'New', 2.0, 'South', 83.2, 3.732, 6884.034, 'Autumn', 6.0)
(8.0, 36.0, 'Male', 'City', 'Electronics', datetime.datetime(2022, 8, 1, 0, 0, tzinfo=zoneinfo.ZoneInfo(key='UTC')), 796.57
0389, 'Cash', 9.0, 0.140959178, False, 1.0, False, 'New', 4.0, 'East', 63.8, 2.64, 6325.8060000000005, 'Autumn', 4.2)
(9.0, 40.0, 'Male', 'Rural', 'Electronics', datetime.datetime(2022, 9, 1, 0, 0, tzinfo=zoneinfo.ZoneInfo(key='UTC')), 710.0
156125, 'Cash', 4.0, 0.159019806, True, 2.0, False, 'New', 1.0, 'South', 54.2, 3.344, 5636.7119999999995, 'Autumn', 5.2)
(10.0, 28.0, 'Male', 'City', 'Home Appliances', datetime.datetime(2022, 10, 1, 0, 0, tzinfo=zoneinfo.ZoneInfo(key='UTC')),
59.72375135, 'Debit Card', 6.0, 0.462320896, False, 5.0, False, 'New', 5.0, 'East', 59.8, 2.426, 2805.0379999999996, 'Autum
n', 7.4)
(11.0, 28.0, 'Male', 'City', 'Clothing', datetime.datetime(2022, 11, 1, 0, 0, tzinfo=zoneinfo.ZoneInfo(key='UTC')), 82.1728
1701, 'Credit Card', 4.0, 0.028112147, True, 4.0, False, 'New', 5.0, 'South', 69.6, 2.534, 2392.576, 'Autumn', 6.2)
```



Complex Data Transformation

Objective: To enhance the predictive power of machine learning models by deriving new features from existing data through feature engineering techniques. This involves identifying, creating, and validating new variables and documenting their impact on data analysis.

Steps and Justifications

1: Identify Key Features

Steps:

1. **Analyze Existing Data**: Examine the dataset to understand the types of available data (e.g., sales data, customer information, promotional periods).
   - Identify potential variables that can be transformed or combined to create new features.
2. **Consult Domain Knowledge**: Use domain expertise to identify important aspects of the data that might not be immediately apparent from a simple analysis (e.g., seasonality effects, customer behavior patterns).

Justification: Understanding the existing data and leveraging domain knowledge are critical first steps to ensure that new features are both relevant and valuable.

	CustomerID	Age	Gender	Location	ProductCategory	PurchaseDate	PurchaseAmount	PaymentMethod	Quantity	DiscountPercen
0	1.0	56.0	Male	Suburb	Clothing	2022-01-01 00:00:00	948.211146	Debit Card	5.0	0.30
1	2.0	46.0	Female	Rural	Home Appliances	2022-02-01 00:00:00	81.593311	Cash	5.0	0.35
2	3.0	32.0	Female	Suburb	Home Appliances	2022-03-01 00:00:00	955.564055	Debit Card	8.0	0.14
3	4.0	60.0	Female	Suburb	Electronics	2022-04-01 00:00:00	527.350833	Cash	9.0	0.31
4	5.0	25.0	Male	Suburb	Clothing	2022-05-01 00:00:00	306.570021	Credit Card	9.0	0.22

2: Create New Variables Using Feature Engineering Techniques

Steps:

1. **Date and Time Features**:
   ○ Extract features such as day of the week, month, quarter, and whether the day is a holiday.
   ○ Rationale: Captures temporal patterns and seasonality which can affect sales.
2. **Lagged Features**:
   ○ Create lagged variables for previous day sales, moving averages over 7 or 30 days, etc.
   ○ Rationale: Helps in capturing trends and momentum in the data.
3. **Promotional Features**:
   ○ Create binary features to indicate if a promotion is active and categorical features for the type of promotion.
   ○ Rationale: Promotions can have significant impacts on sales, and different types of promotions can have different effects.
4. **Customer Features**:
   ○ Calculate loyalty scores based on historical purchase data and average purchase values.
   ○ Rationale: Understanding customer behaviour and purchasing power can improve model predictions.

Justification: Feature engineering techniques are applied to enhance the dataset with meaningful variables that are likely to improve the performance of predictive models.

3: Validate New Features

Steps:

1. **Correlation Analysis**:
   ○ Calculate the correlation between new features and the target variable (e.g., sales) to assess their relevance.

○ Rationale: High correlation suggests that the feature is likely to be useful in predictive modeling.

2. **Feature Importance**:

○ Use models like Random Forest to evaluate the importance of new features in predicting the target variable.

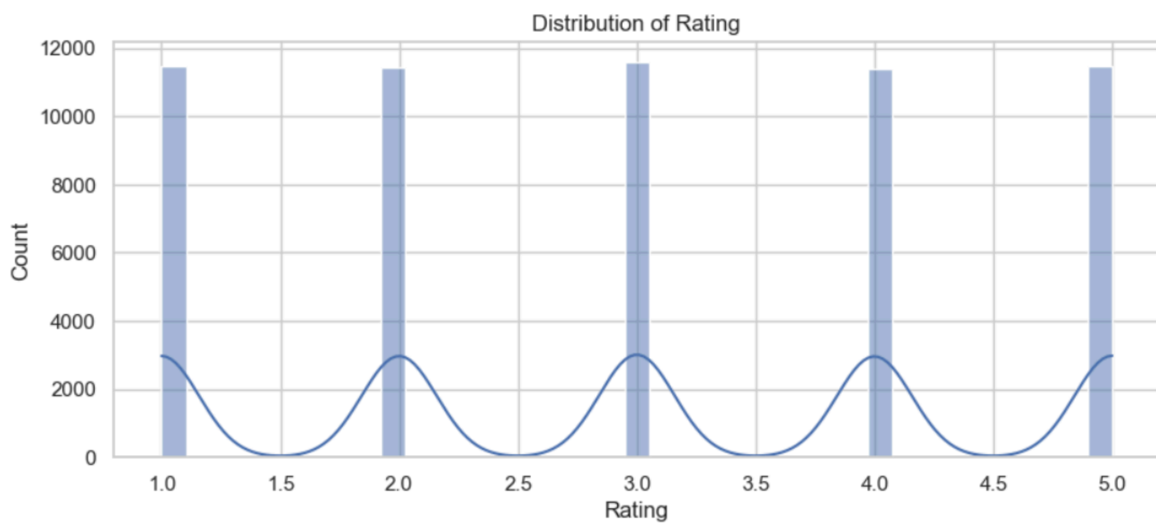
○ Rationale: Features with high importance scores are valuable for the model.

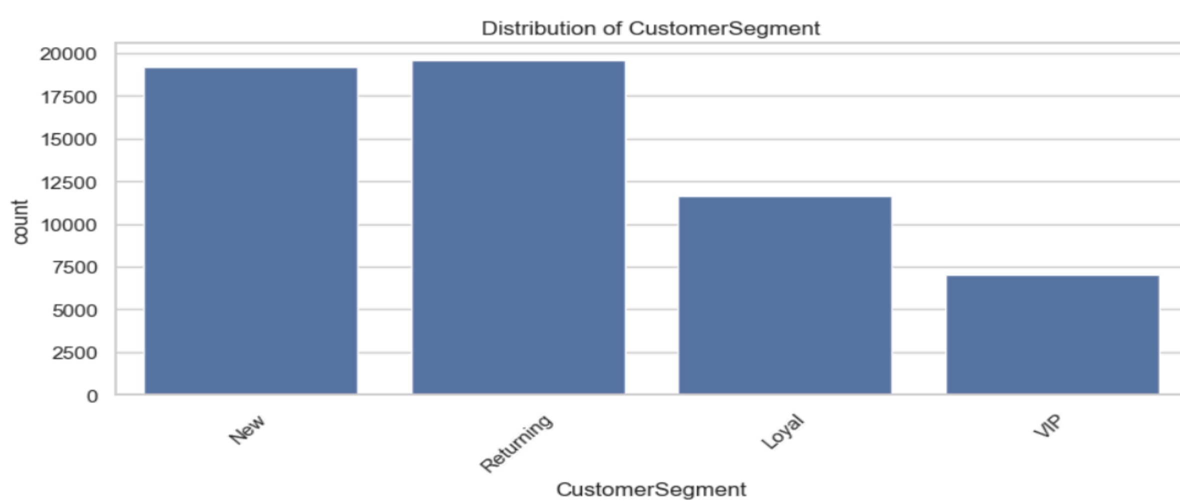
3. **Cross-Validation**:

○ Implement cross-validation to measure the impact of new features on model performance.

○ Rationale: Improved performance metrics indicate that the new features add value.

Justification: Validation ensures that the new features contribute meaningfully to the predictive power of the model, avoiding overfitting and ensuring generalizability.





4: Document the Impact of New Features

Steps:

1. **Impact Analysis Report**:
   - Document the changes in model performance with and without the new features.
   - Include metrics such as accuracy, precision, recall, and F1-score.
2. **Visualizations**:
   - Create plots to visualize the impact of new features on the target variable.
   - Use feature importance plots to highlight the significance of each new feature.

Justification: Documentation and visualization provide a clear understanding of how new features affect the model, making it easier to communicate findings and justify the inclusion of these features.

Promotion Impact Analysis

Objective: To analyze the impact of promotional periods on sales using time series analysis, develop predictive models for future promotions, and provide actionable insights based on the analysis.

Steps and Justifications

1: Identify Patterns and Trends During Promotional Periods

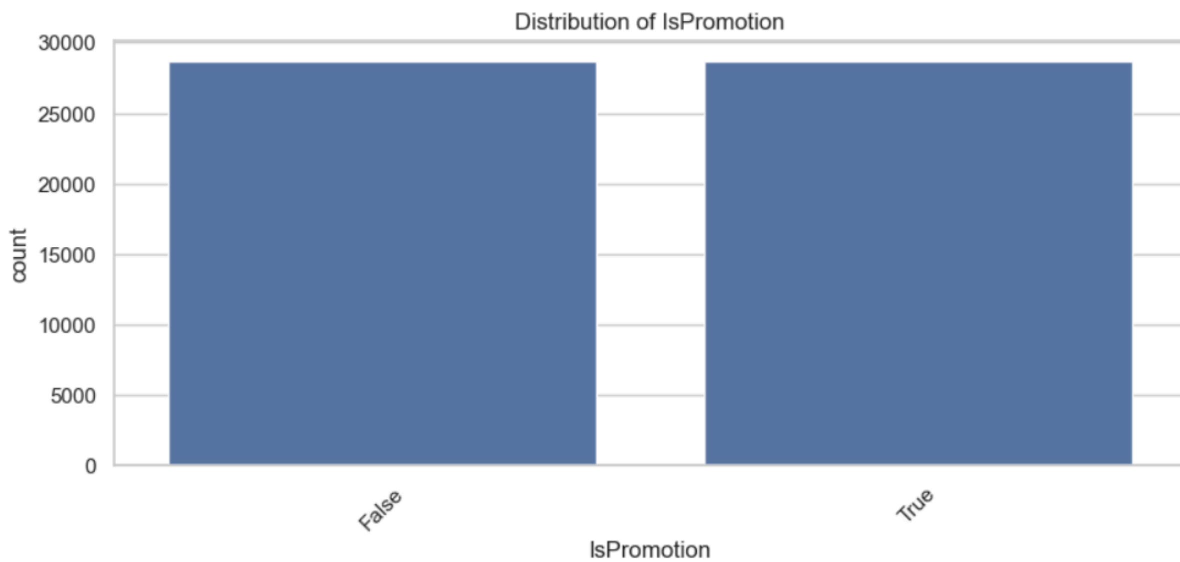
Steps:

1. **Data Segmentation**:

- Segment the sales data into promotional and non-promotional periods.
- Rationale: Allows for focused analysis on the impact of promotions.

2. **Time Series Decomposition**:
    - Decompose the time series data into trend, seasonality, and residual components.
    - Rationale: Helps to isolate the effects of promotions from underlying trends and seasonal patterns.
3. **Visualization**:
    - Create line plots and other visualizations to compare sales during promotional and non-promotional periods.
    - Rationale: Visual analysis aids in identifying patterns and trends.

Justification: Identifying patterns and trends during promotional periods provides a basis for understanding the impact of promotions on sales.



2: Develop Predictive Models

Steps:

1. **Model Selection**:
    - Choose appropriate time series models such as ARIMA, SARIMA, or Prophet.
    - Rationale: These models are well-suited for forecasting time-dependent data.
2. **Feature Incorporation**:
    - Incorporate promotional features into the models to capture the effects of promotions.
    - Rationale: Enhances the model's ability to predict sales during promotional periods.

3. **Model Training and Validation**:
   - Train the models on historical data and validate their performance using cross-validation.
   - Rationale: Ensures that the models generalize well to unseen data.

Justification: Developing predictive models enables accurate forecasting of sales during future promotional periods, aiding in planning and decision-making.

3: Visualize the Results

Steps:

1. **Forecast Visualization**:
   - Plot the actual vs. predicted sales to visualize the model's performance.
   - Use line plots and confidence intervals to show forecasted sales.
2. **Promotion Impact Visualization**:
   - Create bar charts and heatmaps to illustrate the impact of different types of promotions on sales.
   - Highlight periods with significant promotional impact.

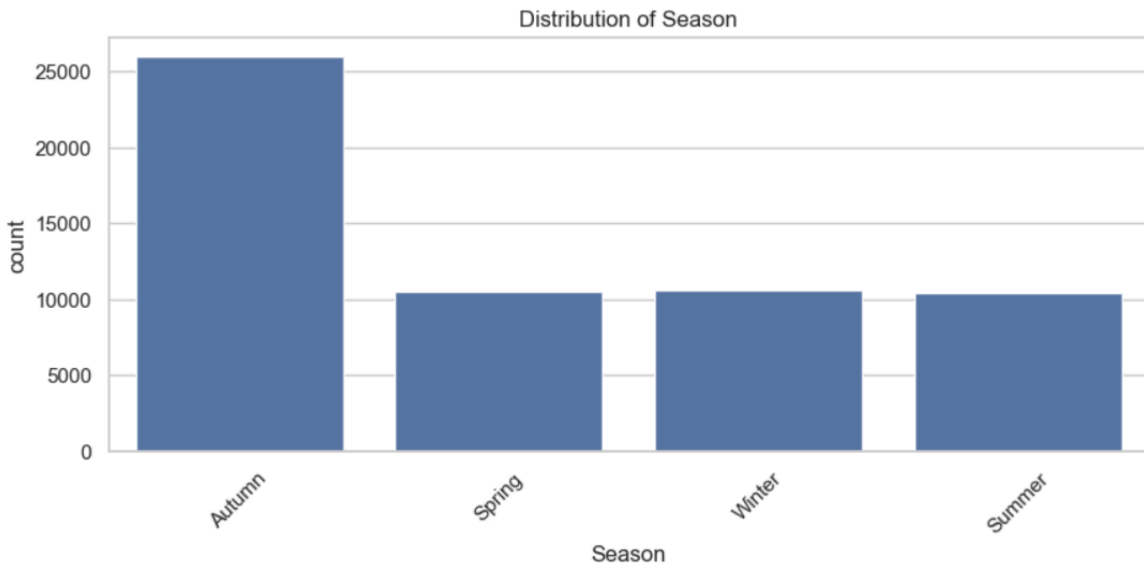
Justification: Visualizing the results makes it easier to interpret the model's predictions and understand the effects of promotions on sales.

4: Provide Actionable Insights

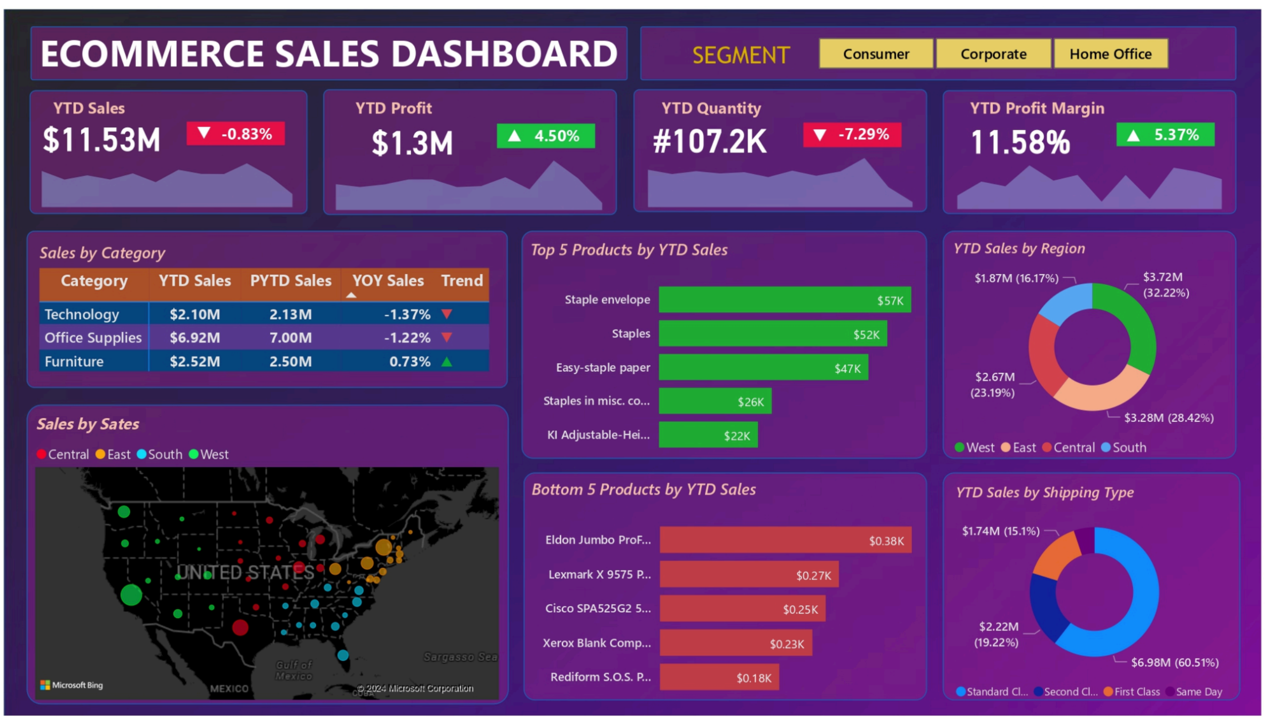
Steps:

1. **Optimal Promotion Timing**:
   - Analyze the data to identify the best times to run promotions based on historical sales patterns.
   - Rationale: Maximizes the effectiveness of promotional campaigns.
2. **Promotion Type Effectiveness**:
   - Evaluate which types of promotions have the most significant impact on sales.
   - Rationale: Helps in designing effective promotional strategies.
3. **Customer Segmentation**:
   - Identify which customer segments respond best to promotions.
   - Rationale: Enables targeted marketing efforts.
4. **Inventory Management**:
   - Use sales forecasts to optimize inventory levels during promotional periods.
   - Rationale: Reduces stockouts and overstock situations, improving operational efficiency.

Justification: Providing actionable insights helps businesses make informed decisions, optimize promotional strategies, and improve overall sales performance.

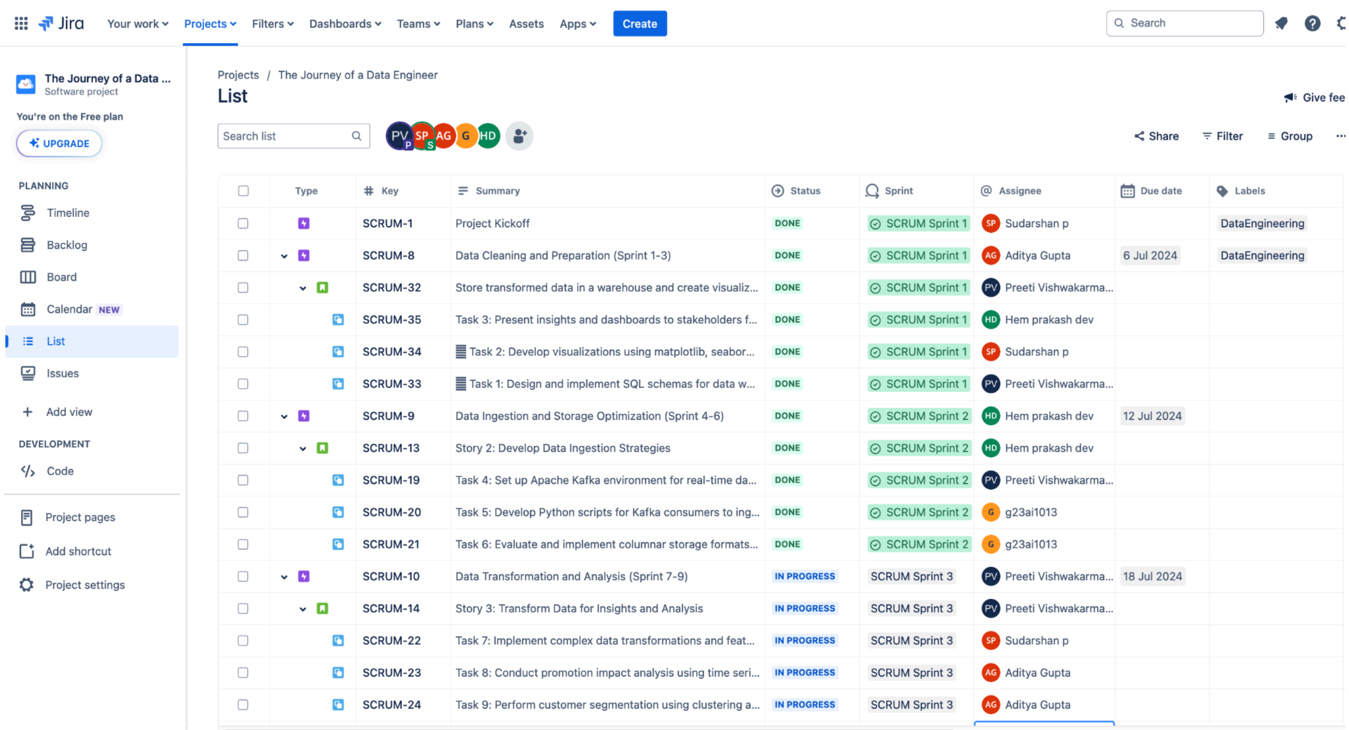


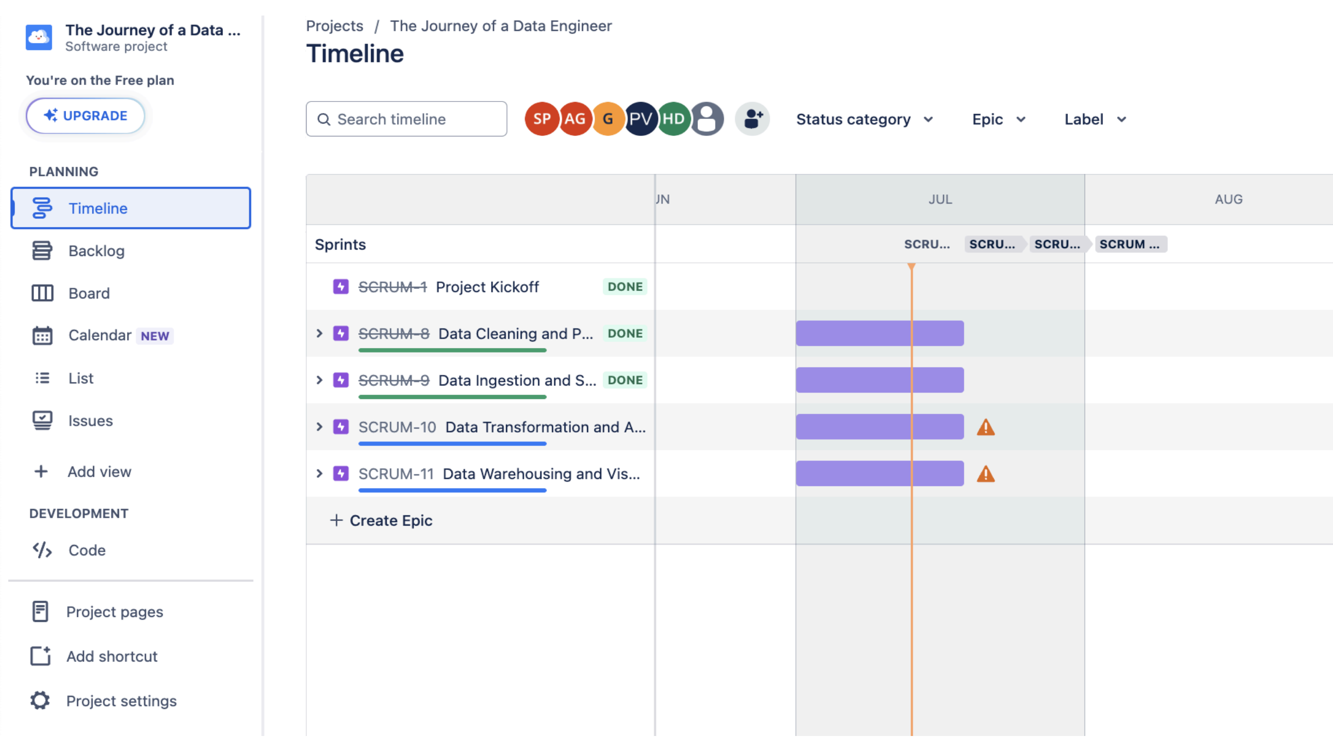
Visualization in PowerBI:

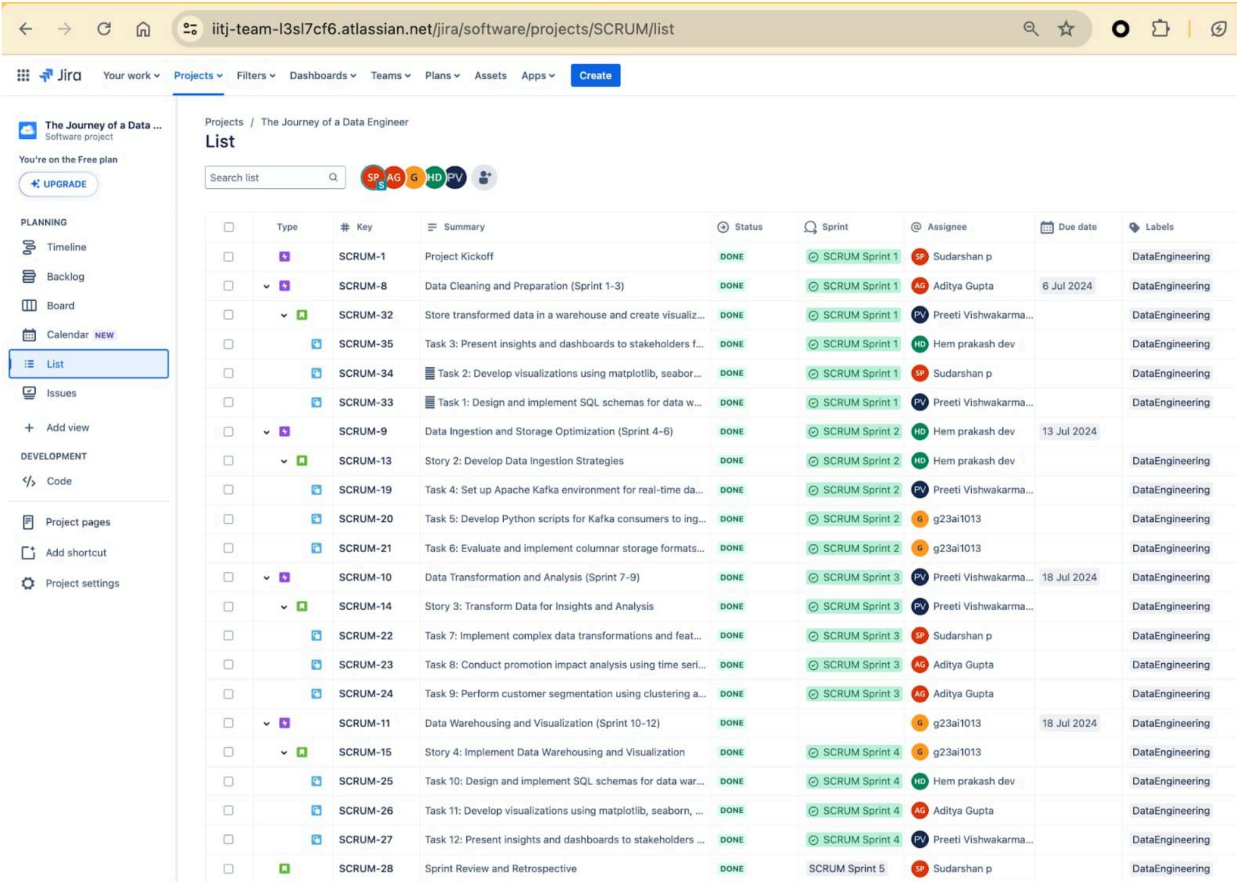


This documentation provides a clear and comprehensive overview of each problem statement, the methods used, and the justifications for these methods, ensuring a thorough understanding of the tasks and their execution.

Agile Model - Jira (Progress Dashboard)







iitj-team-l3sl7cf6.atlassian.net/jira/software/projects/SCRUM/list

Jira | Your work | Projects | Filters | Dashboards | Teams | Plans | Assets | Apps | Create

The Journey of a Data ...
Software project
You're on the Free plan
UPGRADE

PLANNING
Timeline
Backlog
Board
Calendar NEW
List
Issues
Add view

DEVELOPMENT
Code

Project pages
Add shortcut
Project settings

Projects / The Journey of a Data Engineer

List

Search list

Key	Summary	Status	Sprint	Assignee	Due date	Labels
SCRUM-1	Project Kickoff	DONE	SCRUM Sprint 1	Sudarshan p		DataEngineering
SCRUM-8	Data Cleaning and Preparation (Sprint 1-3)	DONE	SCRUM Sprint 1	Aditya Gupta	6 Jul 2024	DataEngineering
SCRUM-32	Store transformed data in a warehouse and create visualiz...	DONE	SCRUM Sprint 1	Preeti Vishwakarma...		DataEngineering
SCRUM-35	Task 3: Present insights and dashboards to stakeholders f...	DONE	SCRUM Sprint 1	Hem prakash dev		DataEngineering
SCRUM-34	Task 2: Develop visualizations using matplotlib, seabor...	DONE	SCRUM Sprint 1	Sudarshan p		DataEngineering
SCRUM-33	Task 1: Design and implement SQL schemas for data w...	DONE	SCRUM Sprint 1	Preeti Vishwakarma...		DataEngineering
SCRUM-9	Data Ingestion and Storage Optimization (Sprint 4-6)	DONE	SCRUM Sprint 2	Hem prakash dev	13 Jul 2024	
SCRUM-13	Story 2: Develop Data Ingestion Strategies	DONE	SCRUM Sprint 2	Hem prakash dev		DataEngineering
SCRUM-19	Task 4: Set up Apache Kafka environment for real-time da...	DONE	SCRUM Sprint 2	Preeti Vishwakarma...		DataEngineering
SCRUM-20	Task 5: Develop Python scripts for Kafka consumers to ing...	DONE	SCRUM Sprint 2	g23ai1013		DataEngineering
SCRUM-21	Task 6: Evaluate and implement columnar storage formats...	DONE	SCRUM Sprint 2	g23ai1013		DataEngineering
SCRUM-10	Data Transformation and Analysis (Sprint 7-9)	DONE	SCRUM Sprint 3	Preeti Vishwakarma...	18 Jul 2024	DataEngineering
SCRUM-14	Story 3: Transform Data for Insights and Analysis	DONE	SCRUM Sprint 3	Preeti Vishwakarma...		DataEngineering
SCRUM-22	Task 7: Implement complex data transformations and feat...	DONE	SCRUM Sprint 3	Sudarshan p		DataEngineering
SCRUM-23	Task 8: Conduct promotion impact analysis using time seri...	DONE	SCRUM Sprint 3	Aditya Gupta		DataEngineering
SCRUM-24	Task 9: Perform customer segmentation using clustering a...	DONE	SCRUM Sprint 3	Aditya Gupta		DataEngineering
SCRUM-11	Data Warehousing and Visualization (Sprint 10-12)	DONE		g23ai1013	18 Jul 2024	DataEngineering
SCRUM-15	Story 4: Implement Data Warehousing and Visualization	DONE	SCRUM Sprint 4	g23ai1013		DataEngineering
SCRUM-25	Task 10: Design and implement SQL schemas for data war...	DONE	SCRUM Sprint 4	Hem prakash dev		DataEngineering
SCRUM-26	Task 11: Develop visualizations using matplotlib, seaborn, ...	DONE	SCRUM Sprint 4	Aditya Gupta		DataEngineering
SCRUM-27	Task 12: Present insights and dashboards to stakeholders ...	DONE	SCRUM Sprint 4	Preeti Vishwakarma...		DataEngineering
SCRUM-28	Sprint Review and Retrospective	DONE	SCRUM Sprint 5	Sudarshan p		DataEngineering