Group 12 Project Documentation
Sales Analytics

Problem Statement 1: Advanced Data Cleaning

Objective

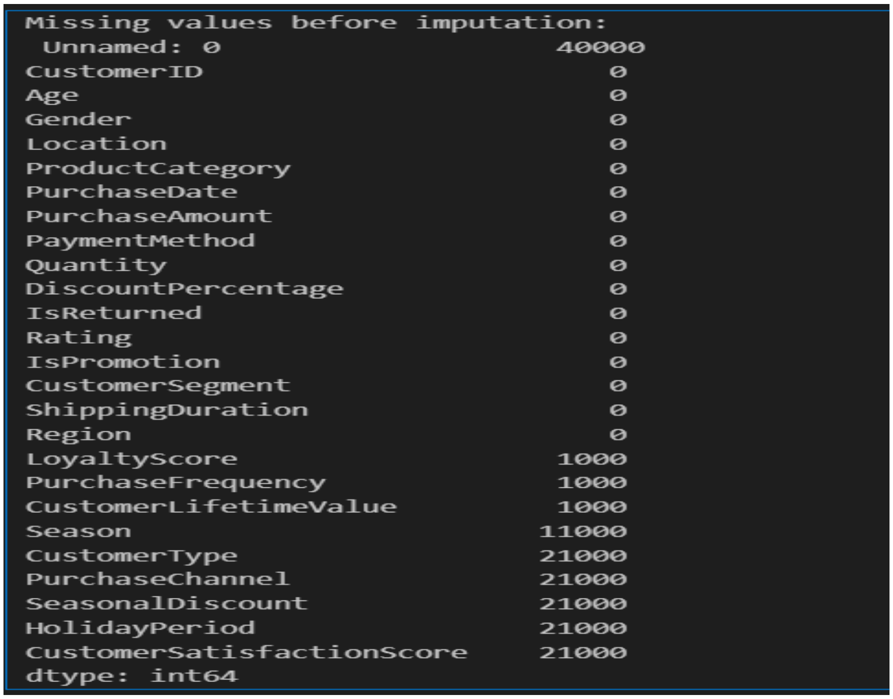
To ensure data quality by identifying and handling outliers, addressing missing values, and correcting data type inconsistencies.

Steps and Justifications:

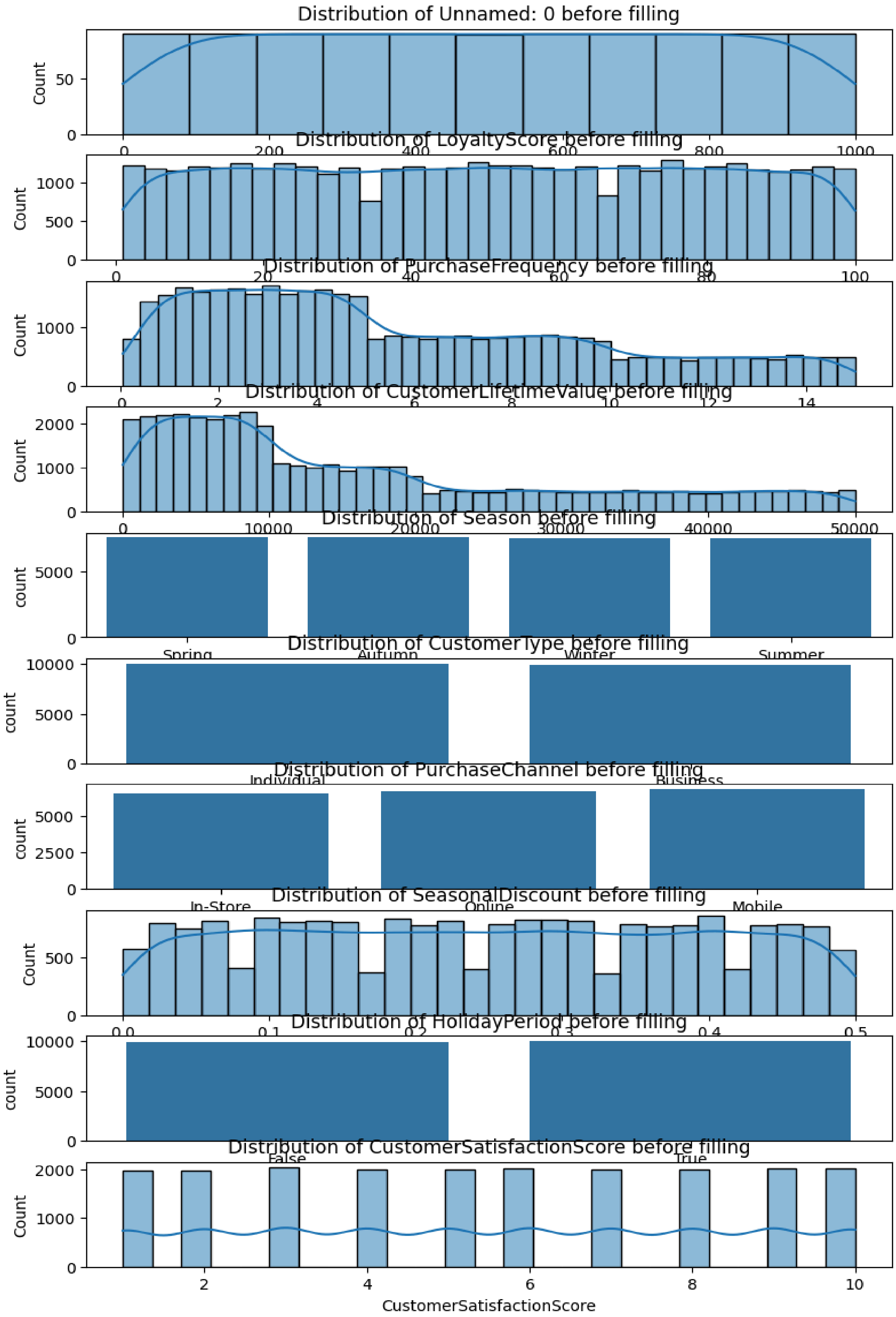
1. **Loading the Dataset**
   - Loaded the dataset using Pandas to facilitate data manipulation and cleaning.

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:



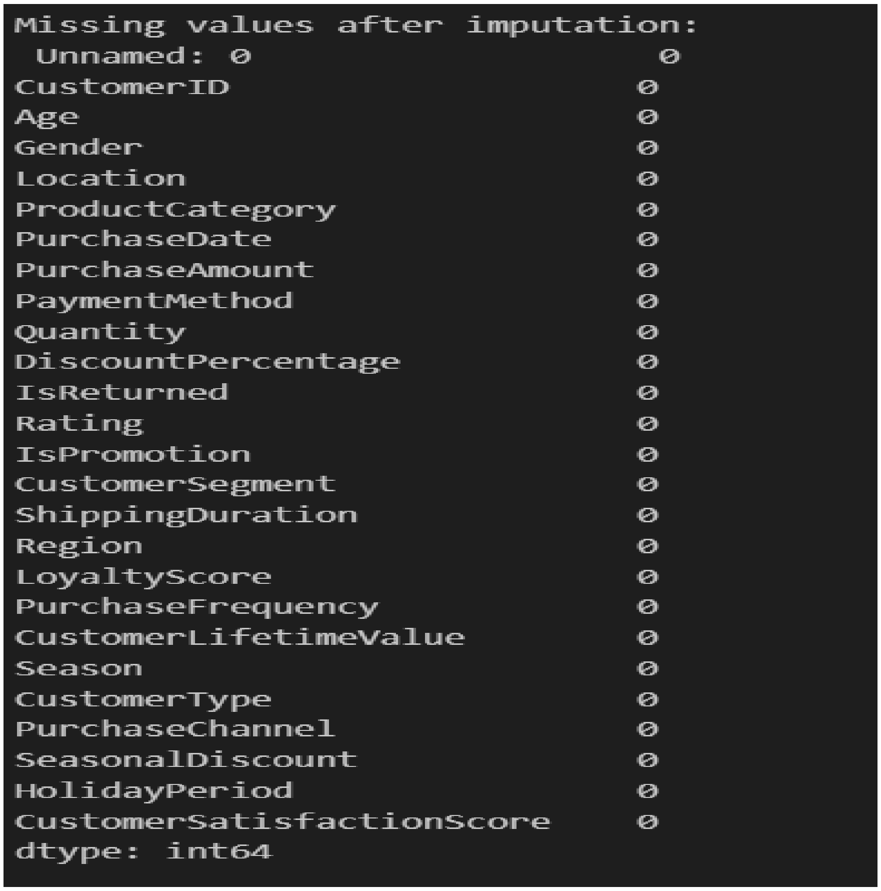
```
Missing values before imputation:
 Unnamed: 0                   40000
CustomerID                        0
Age                               0
Gender                            0
Location                          0
ProductCategory                   0
PurchaseDate                      0
PurchaseAmount                    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

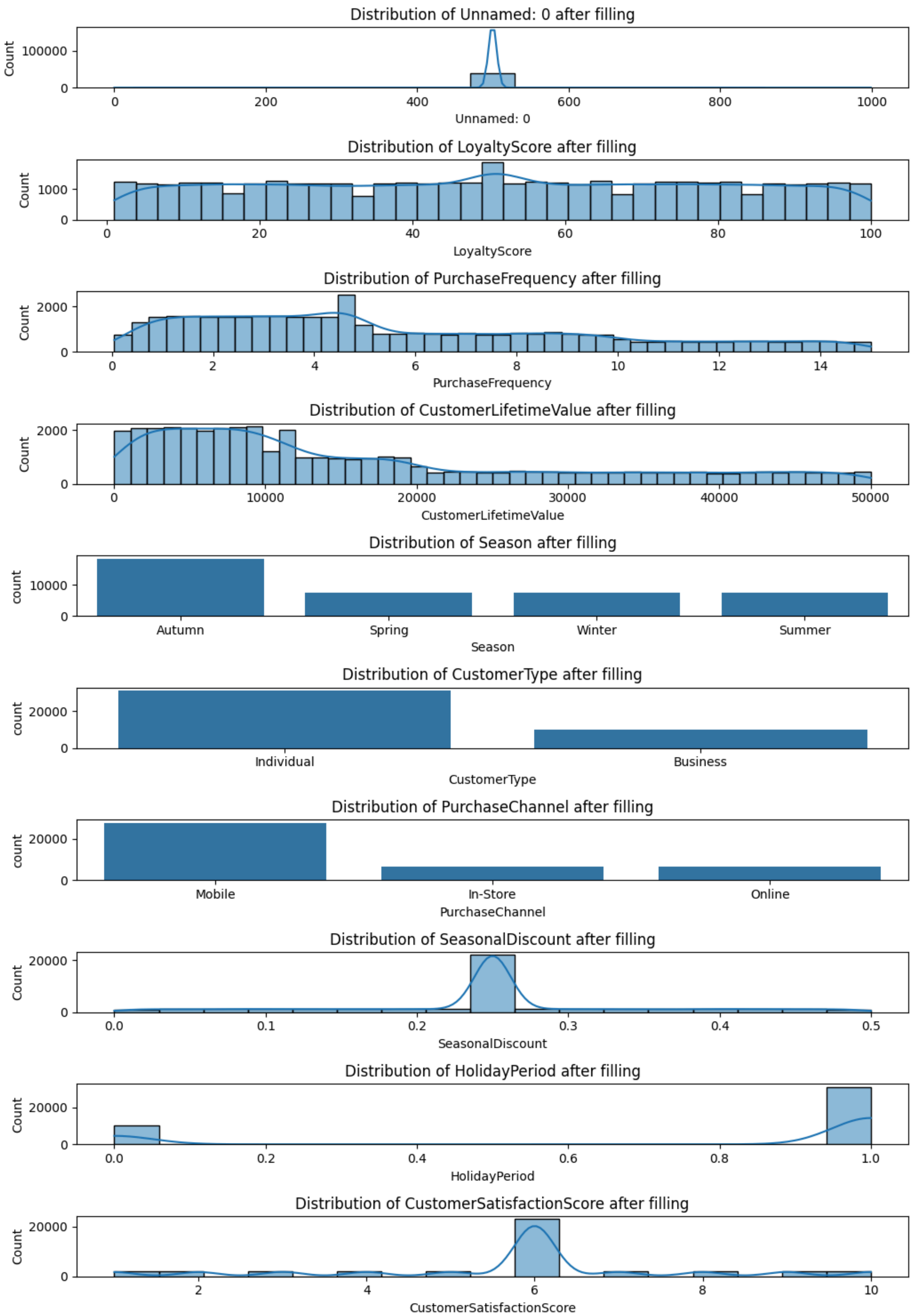
3. **Handling Missing Values**

- 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

Distribution of LoyaltyScore after filling

Distribution of PurchaseFrequency after filling

Distribution of CustomerLifetimeValue after filling

Distribution of Season after filling

Distribution of CustomerType after filling

Distribution of PurchaseChannel after filling

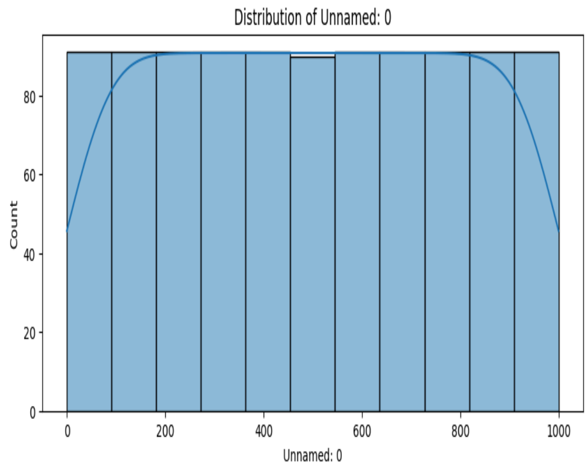
Distribution of SeasonalDiscount after filling

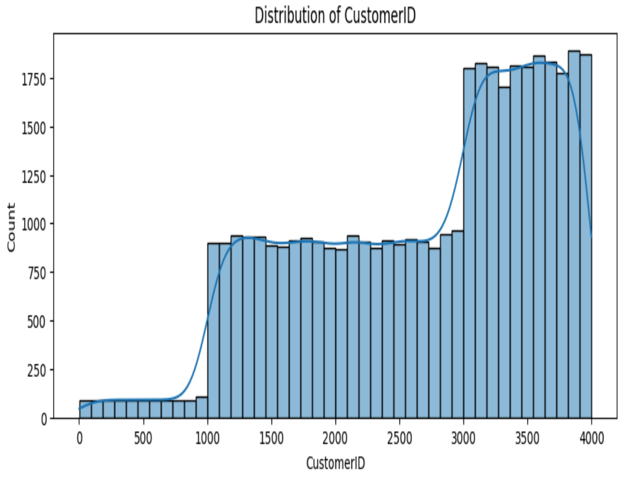
Distribution of HolidayPeriod after filling

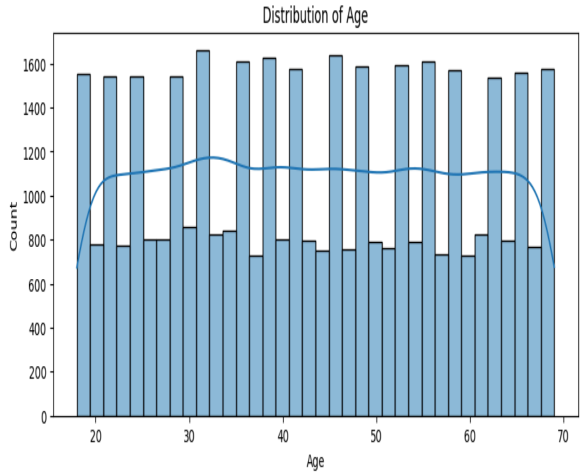
Distribution of CustomerSatisfactionScore after filling

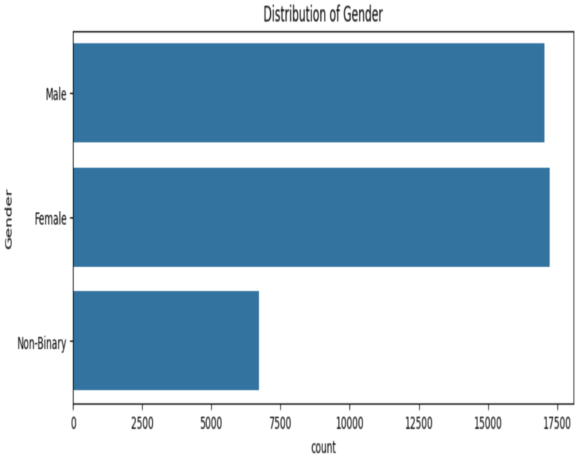
4. **Correcting Data Type Inconsistencies**

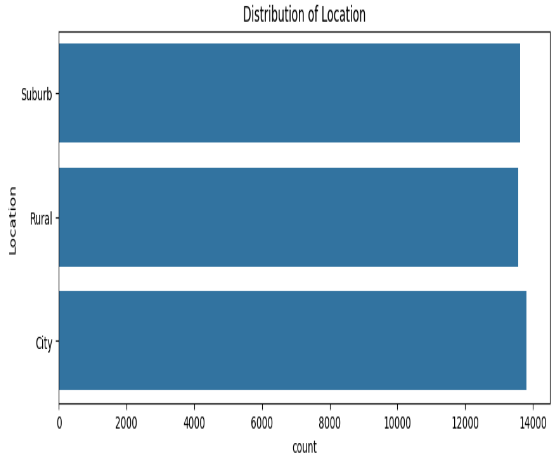
- Ensured that each column had a consistent data type:
  - Converted date columns to datetime objects.
  - Changed numerical columns stored as strings to appropriate numerical types.
- Verified and corrected any misclassified data types.

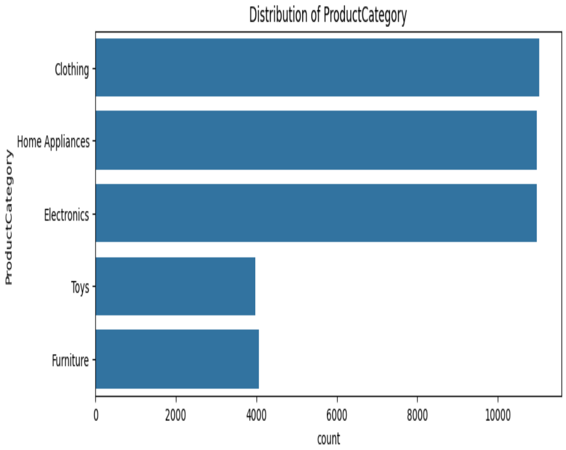

Distribution of Unnamed: 0

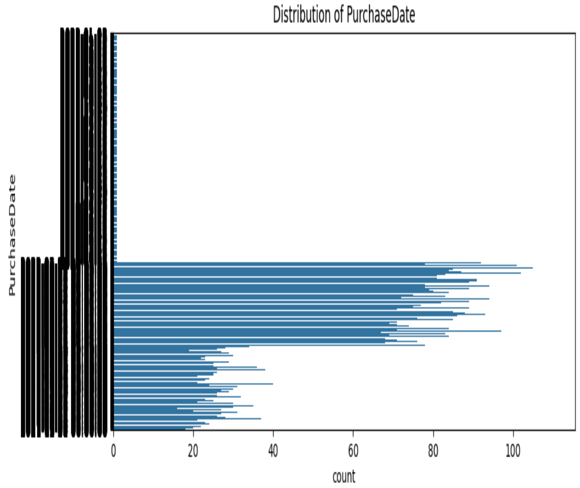

Distribution of CustomerID

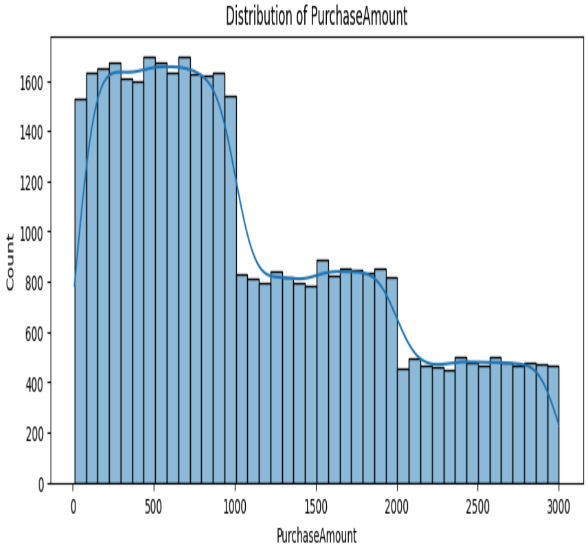

Distribution of Age

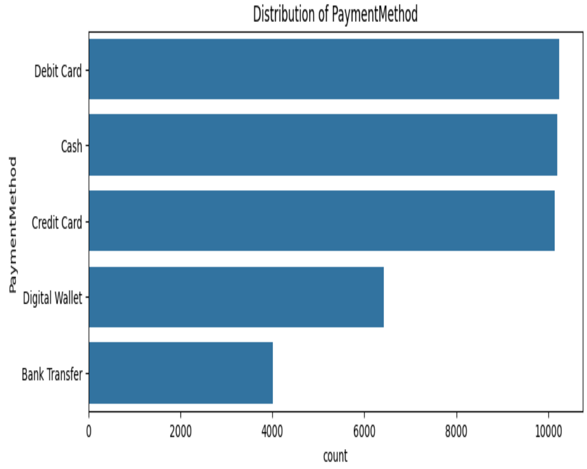

Distribution of Gender

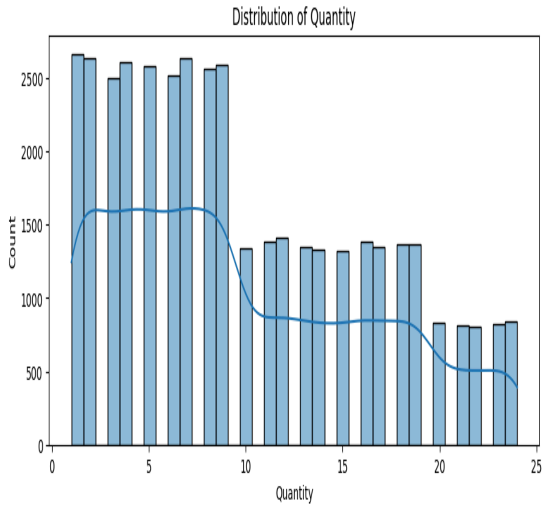

Distribution of Location

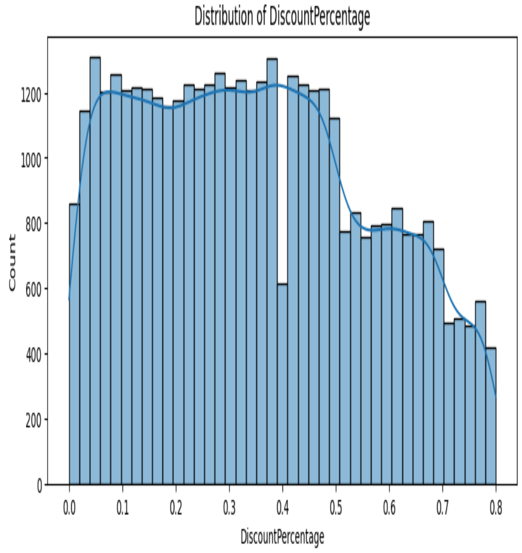

Distribution of ProductCategory

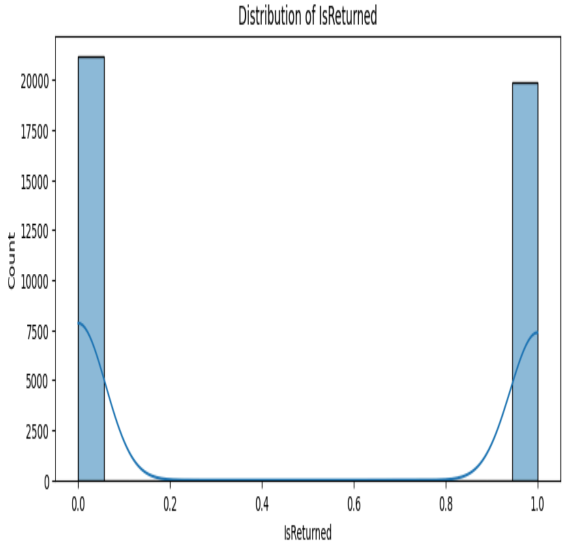

Distribution of PurchaseDate
PurchaseDate
0
20
40
60
80
100
count


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

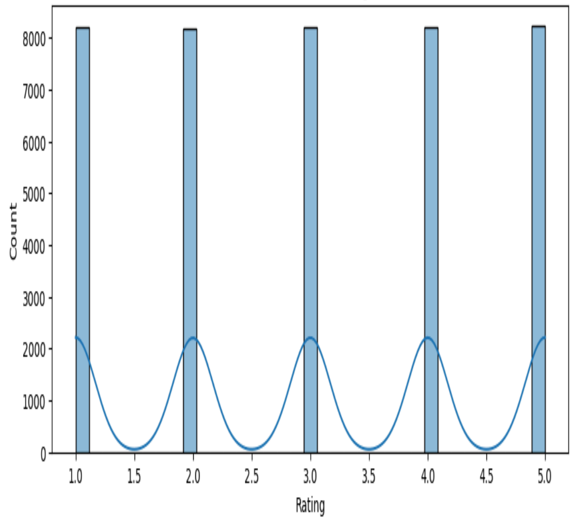

Distribution of PaymentMethod
PaymentMethod
Debit Card
Cash
Credit Card
Digital Wallet
Bank Transfer
0
2000
4000
6000
8000
10000
count


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

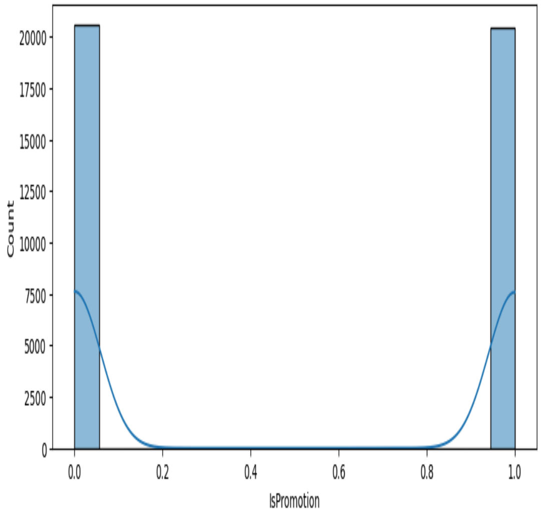

Distribution of DiscountPercentage
Count
0
200
400
600
800
1000
1200
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
DiscountPercentage


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

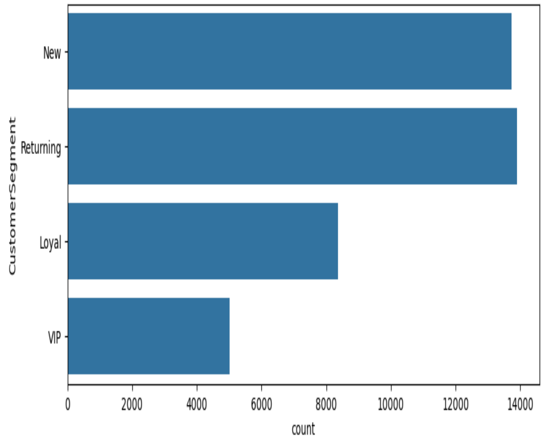
Distribution of Rating



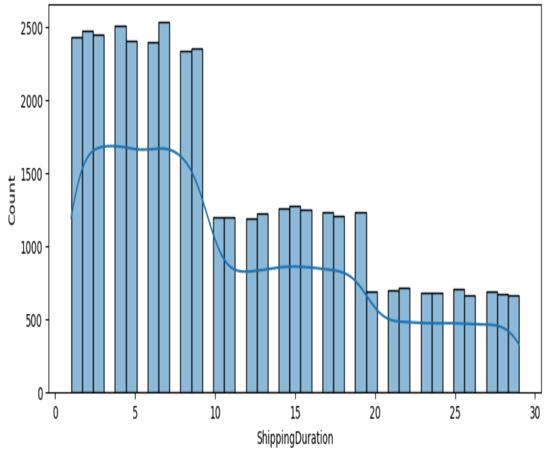
Distribution of IsPromotion



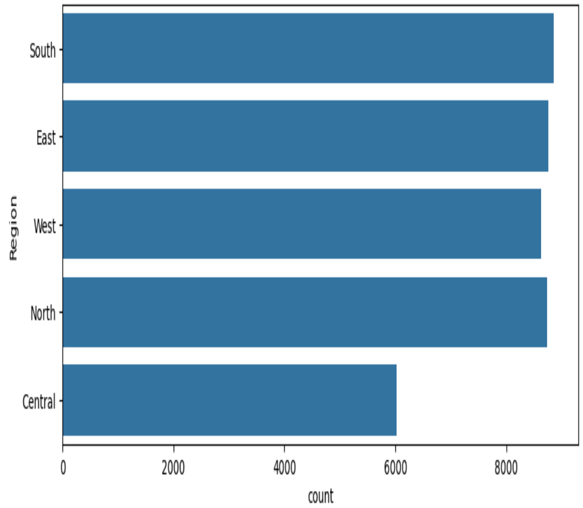
Distribution of CustomerSegment



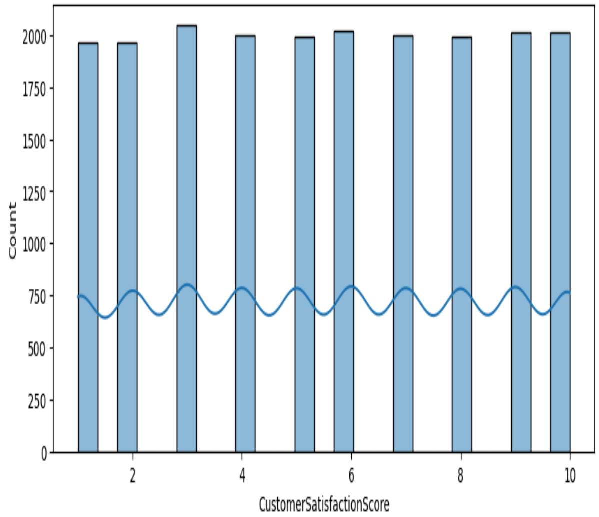
Distribution of ShippingDuration



Distribution of Region



Distribution of CustomerSatisfactionScore



Problem Statement 2: 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**

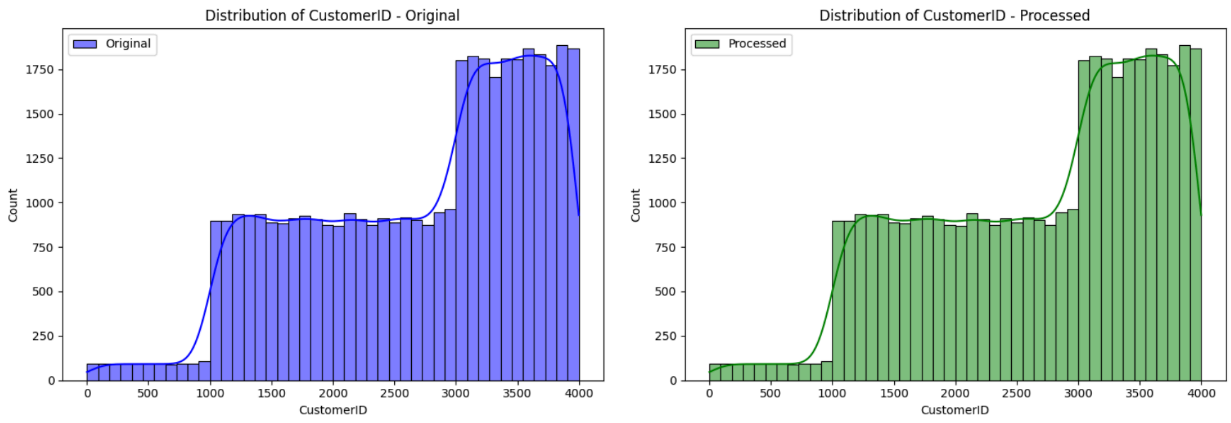
- Applied bootstrapping to create additional samples. Bootstrapping is a resampling technique that generates new data points by sampling with replacement from the existing data.
- Ensured that the augmented data followed the original dataset's statistical distribution.

3. **Integrating Augmented Data**

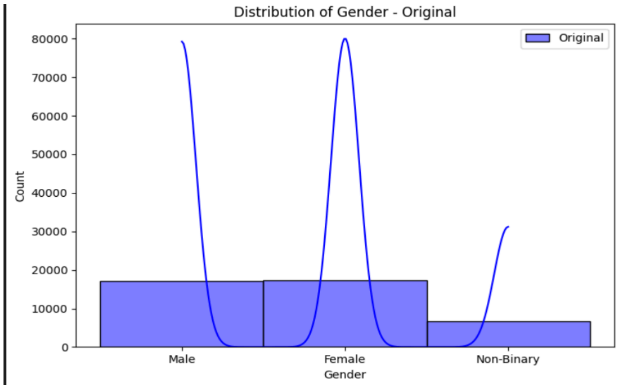
- Merged the newly generated samples with the original dataset to create an expanded dataset.
- Maintained the integrity of the original data while ensuring the augmented data enhanced the dataset.

4. **Validation**

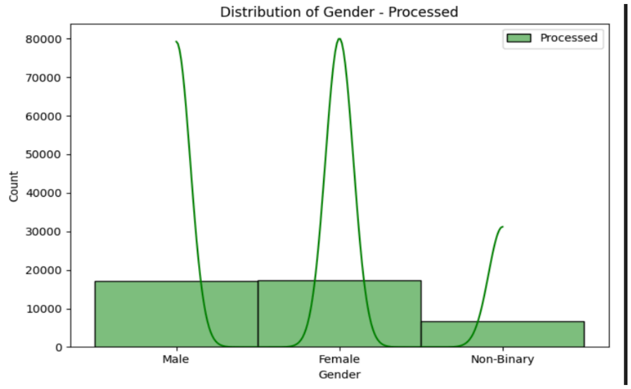
- Performed rigorous validation to ensure the augmented data met quality standards.
- Compared statistical properties of the augmented dataset with the original to confirm consistency.



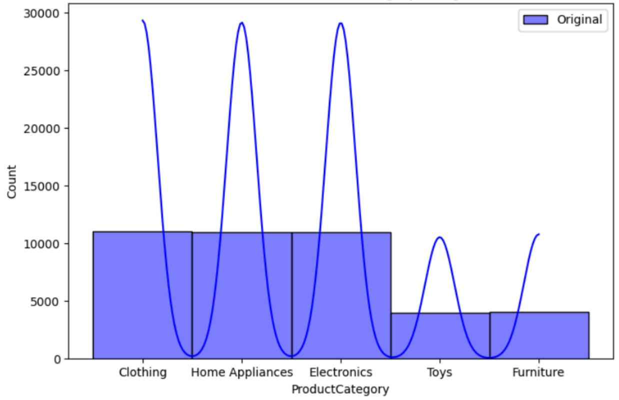
Distribution of Gender - Original



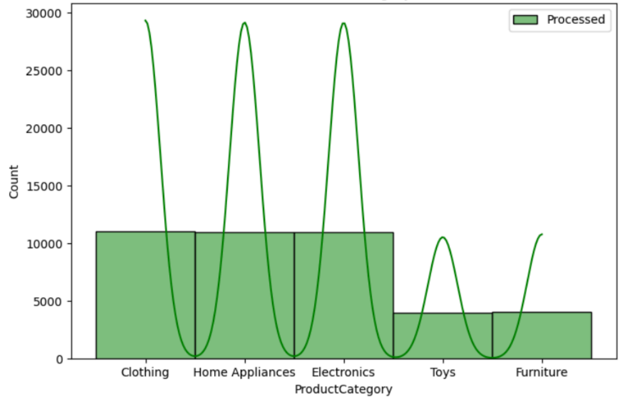
Distribution of Gender - Processed



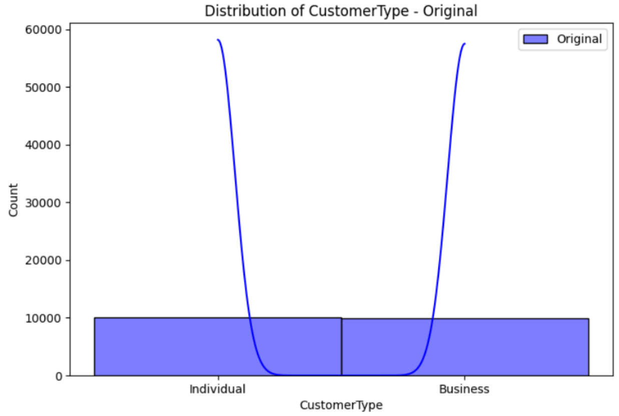
Distribution of ProductCategory - Original



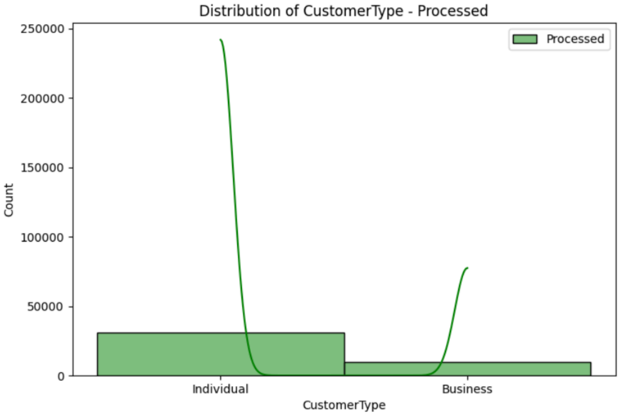
Distribution of ProductCategory - Processed



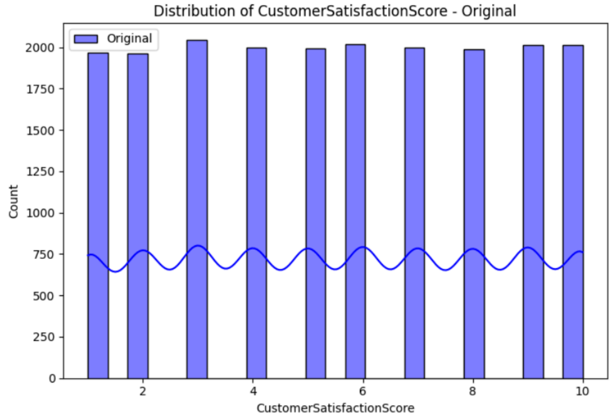
Distribution of CustomerType - Original



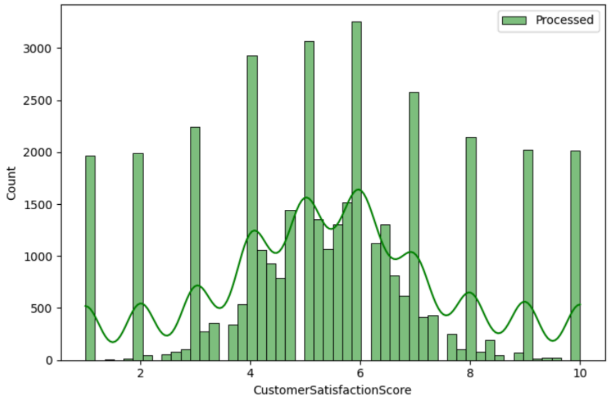
Distribution of CustomerType - Processed



Distribution of CustomerSatisfactionScore - Original



Distribution of CustomerSatisfactionScore - Processed



Problem Statement 3: 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.

Problem Statement 4: Storage Optimization

Objective

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

Steps and Justifications

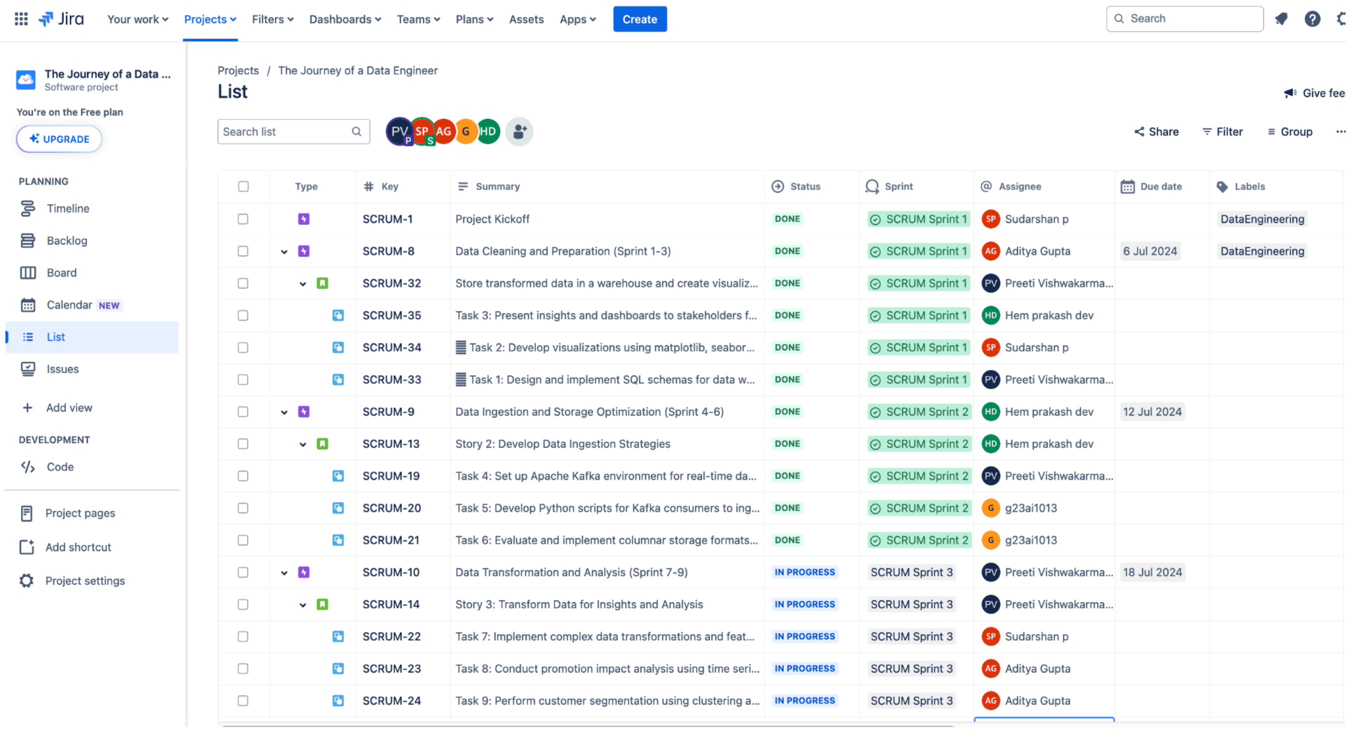
1. **Evaluating Columnar Storage Formats**
   - Assessed columnar storage formats such as Parquet and ORC for their storage efficiency and performance.
   - Compared these formats with traditional row-based storage.

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.

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.

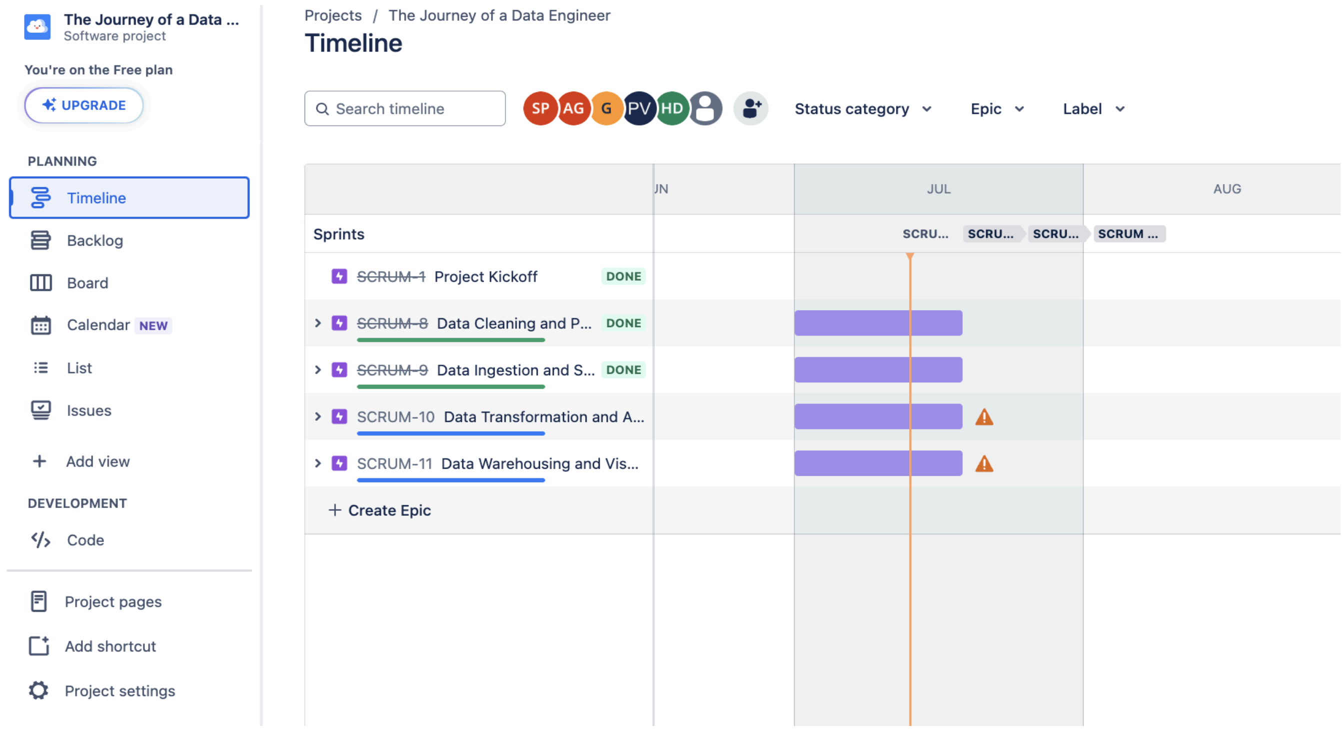
Agile Model - Jira (Progress Dashboard)



Projects / The Journey of a Data Engineer

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...		
SCRUM-35	Task 3: Present insights and dashboards to stakeholders f...	DONE	SCRUM Sprint 1	Hem prakash dev		
SCRUM-34	Task 2: Develop visualizations using matplotlib, seabor...	DONE	SCRUM Sprint 1	Sudarshan p		
SCRUM-33	Task 1: Design and implement SQL schemas for data w...	DONE	SCRUM Sprint 1	Preeti Vishwakarma...		
SCRUM-9	Data Ingestion and Storage Optimization (Sprint 4-6)	DONE	SCRUM Sprint 2	Hem prakash dev	12 Jul 2024	
SCRUM-13	Story 2: Develop Data Ingestion Strategies	DONE	SCRUM Sprint 2	Hem prakash dev		
SCRUM-19	Task 4: Set up Apache Kafka environment for real-time da...	DONE	SCRUM Sprint 2	Preeti Vishwakarma...		
SCRUM-20	Task 5: Develop Python scripts for Kafka consumers to ing...	DONE	SCRUM Sprint 2	g23ai1013		
SCRUM-21	Task 6: Evaluate and implement columnar storage formats...	DONE	SCRUM Sprint 2	g23ai1013		
SCRUM-10	Data Transformation and Analysis (Sprint 7-9)	IN PROGRESS	SCRUM Sprint 3	Preeti Vishwakarma...	18 Jul 2024	
SCRUM-14	Story 3: Transform Data for Insights and Analysis	IN PROGRESS	SCRUM Sprint 3	Preeti Vishwakarma...		
SCRUM-22	Task 7: Implement complex data transformations and feat...	IN PROGRESS	SCRUM Sprint 3	Sudarshan p		
SCRUM-23	Task 8: Conduct promotion impact analysis using time seri...	IN PROGRESS	SCRUM Sprint 3	Aditya Gupta		
SCRUM-24	Task 9: Perform customer segmentation using clustering a...	IN PROGRESS	SCRUM Sprint 3	Aditya Gupta		



Projects / The Journey of a Data Engineer

Timeline

Sprints

~~SCRUM-1~~ Project Kickoff DONE

~~SCRUM-8~~ Data Cleaning and P... DONE

~~SCRUM-9~~ Data Ingestion and S... DONE

SCRUM-10 Data Transformation and A...

SCRUM-11 Data Warehousing and Vis...

+ Create Epic