Ishtiak Ahmed Moyen

2131580642

Question 1: Which of the following statements best describes a dataset?

C) A structured collection of data points representing some aspect of the real world.

Question 2: Why is data preprocessing an important step in data analysis?

C) It reduces noise and inconsistencies in the data, improving the quality of analysis.

Question 3: Which of the following is considered categorical data?

C) Colors of flowers (e.g., red, blue, yellow).

Question 4: What is one common method for handling missing data in a dataset?

B) Removing the entire row or column containing missing values.

Question 5: What does feature engineering involve in data analysis?

C) It involves creating or transforming new features to improve the model's performance.

Question 6: Why is splitting a dataset into training and testing sets important?

C) To ensure that the model's performance is evaluated on unseen data.

Question 7: What is a common technique to handle categorical data before feeding it into a machine learning model?

C) One-Hot Encoding, where each category becomes a binary column.

Question 8: Why might scaling numerical features in a dataset be necessary?

C) To ensure that all numerical features have the same unit of measurement.

Question 9: What is an outlier in the context of data analysis?

C) Unusual or extreme data points that significantly differ from the rest.

Question 10: What does data imputation involve?

D) Filling in missing values with estimated or calculated values.

Question 11: What is a consideration when dealing with time-series data in data analysis?

C) The order and timing of data points matter.

Question 12: What is the primary goal of dimensionality reduction techniques in data analysis?

D) To reduce the number of features while preserving relevant information.

Question 13: Why is addressing imbalanced classes important when building models?

C) Imbalanced classes can bias the model towards the majority class.

Question 14: Which preprocessing step is commonly used for text data before analysis?

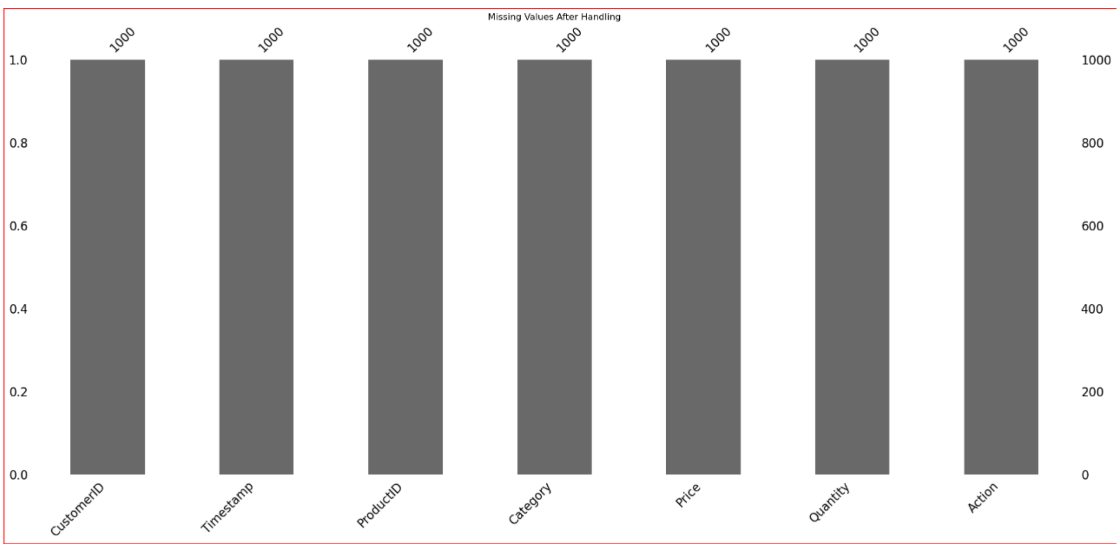
A) Converting text data to numerical values using encoding technique

Loading Data

We began by importing the raw `ecommerce_data.csv` into a pandas DataFrame. A quick `.info()` and `.head()` revealed 1,000 rows across seven columns—CustomerID, Timestamp, ProductID, Category, Price, Quantity, and Action—and helped us confirm basic types and spot formats needing conversion or cleaning.

Handling Missing Values

An initial null-value check showed 100 gaps each in the Category and Price fields. We visualized these with a missing-value bar chart to understand their scope. For Category, we imputed the most frequent label (mode). For Price, we first filled missing entries using the median price **within** each Category; any remaining blanks (where Category itself had been missing) were given the global median price. A follow-up check and bar chart confirmed all nulls were resolved.

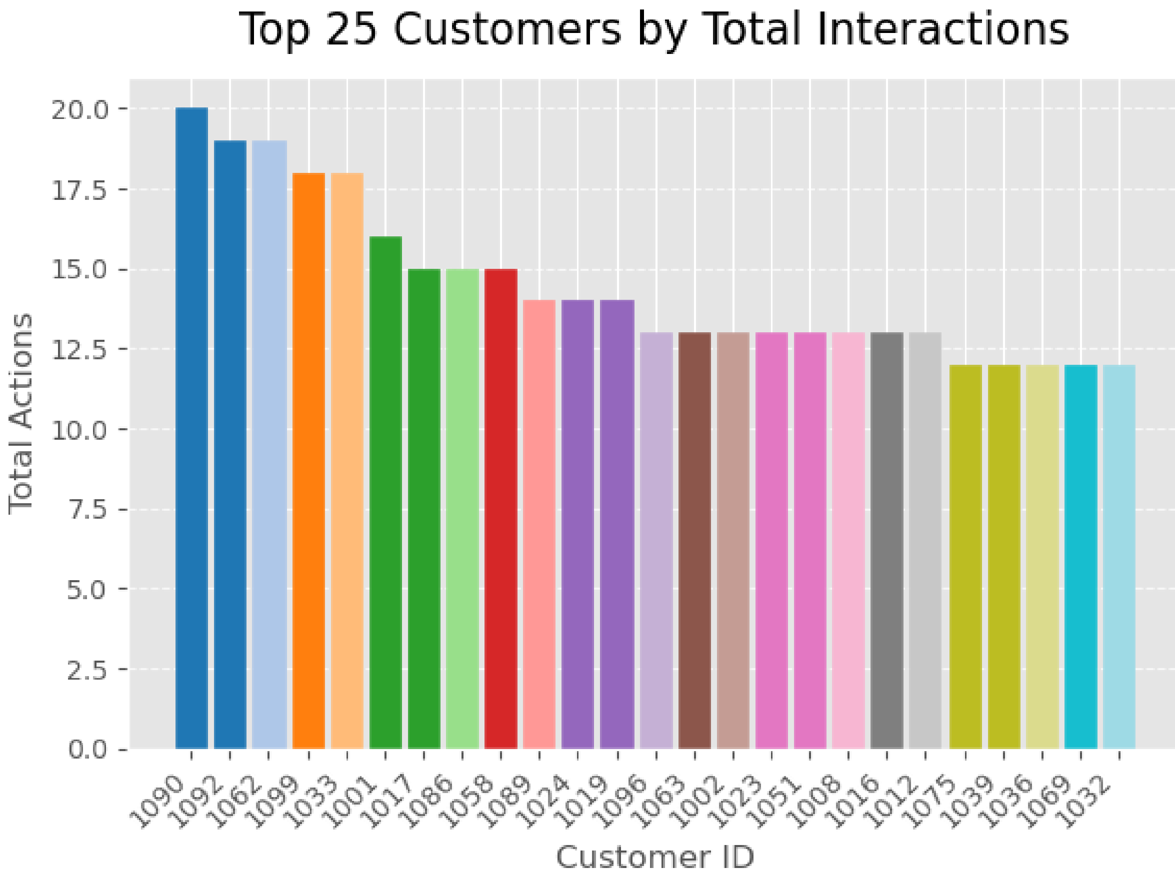


Pic-1:Missing value

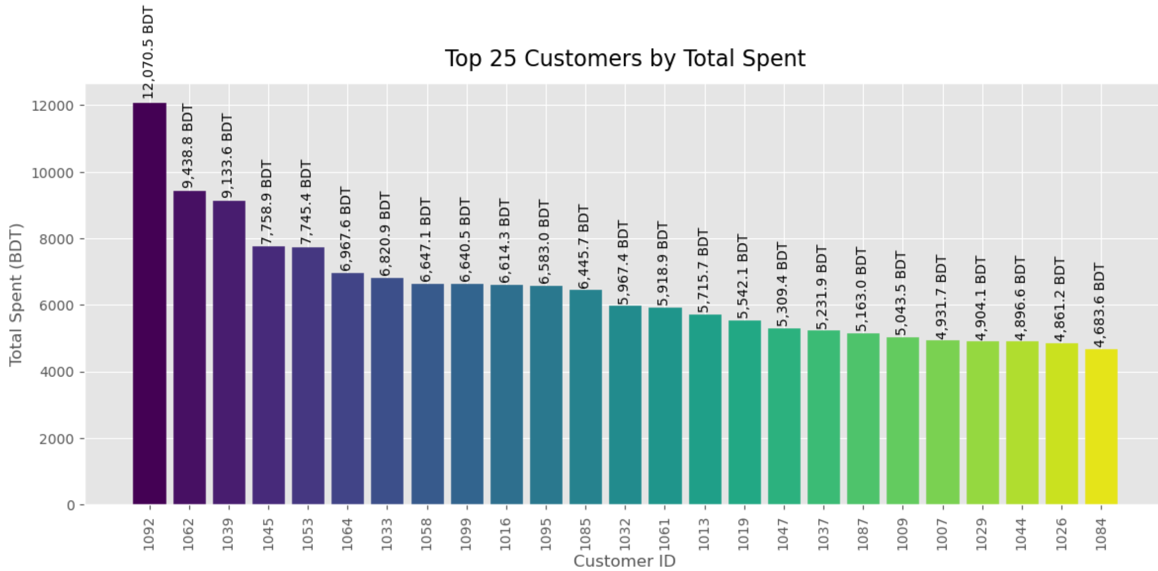
Feature Engineering

To capture customer behavior, we created two key summaries:

- **TotalActions**: the count of every interaction (view, add-to-cart, purchase) grouped by CustomerID.

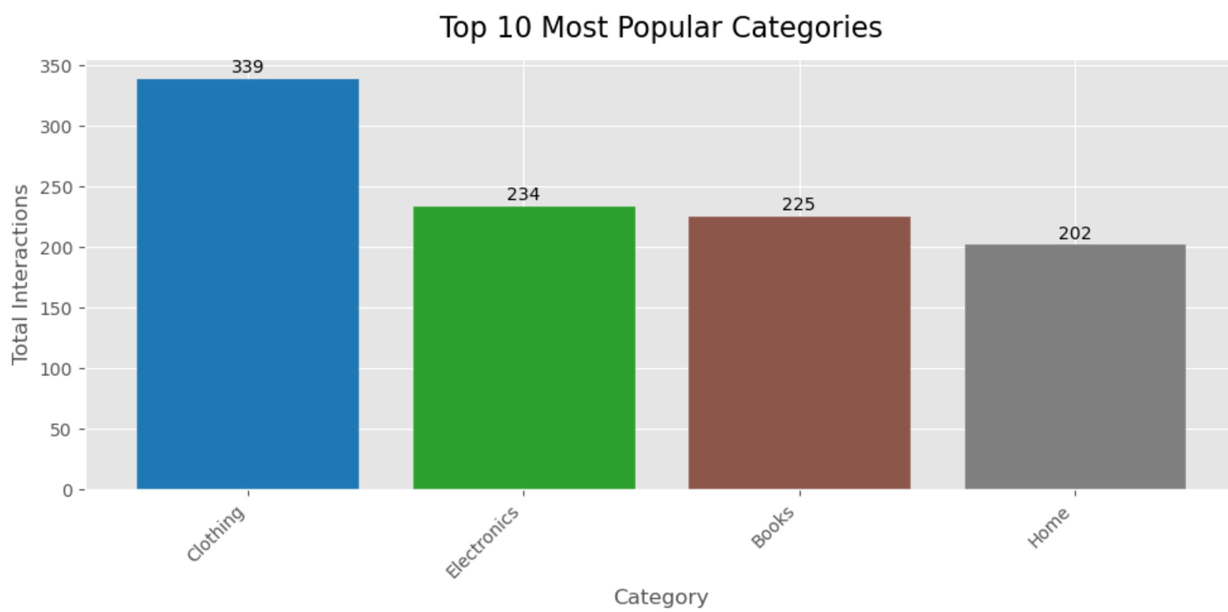


- **TotalSpent**: filtered for “Purchase” actions, computed per-row spending as Price × Quantity, then summed by CustomerID.
  We plotted bar charts of the top customers in each dimension—activity and spend—using colormaps and on-bar annotations for clarity.



Category Analysis & Price Distribution

We grouped interactions by Category to identify the most popular product types, visualizing the top ten in a colored bar chart with interaction counts. We also computed the average Price per Category, and produced a box plot of the overall Price distribution to highlight medians, quartiles, and outliers.



Encoding & Scaling

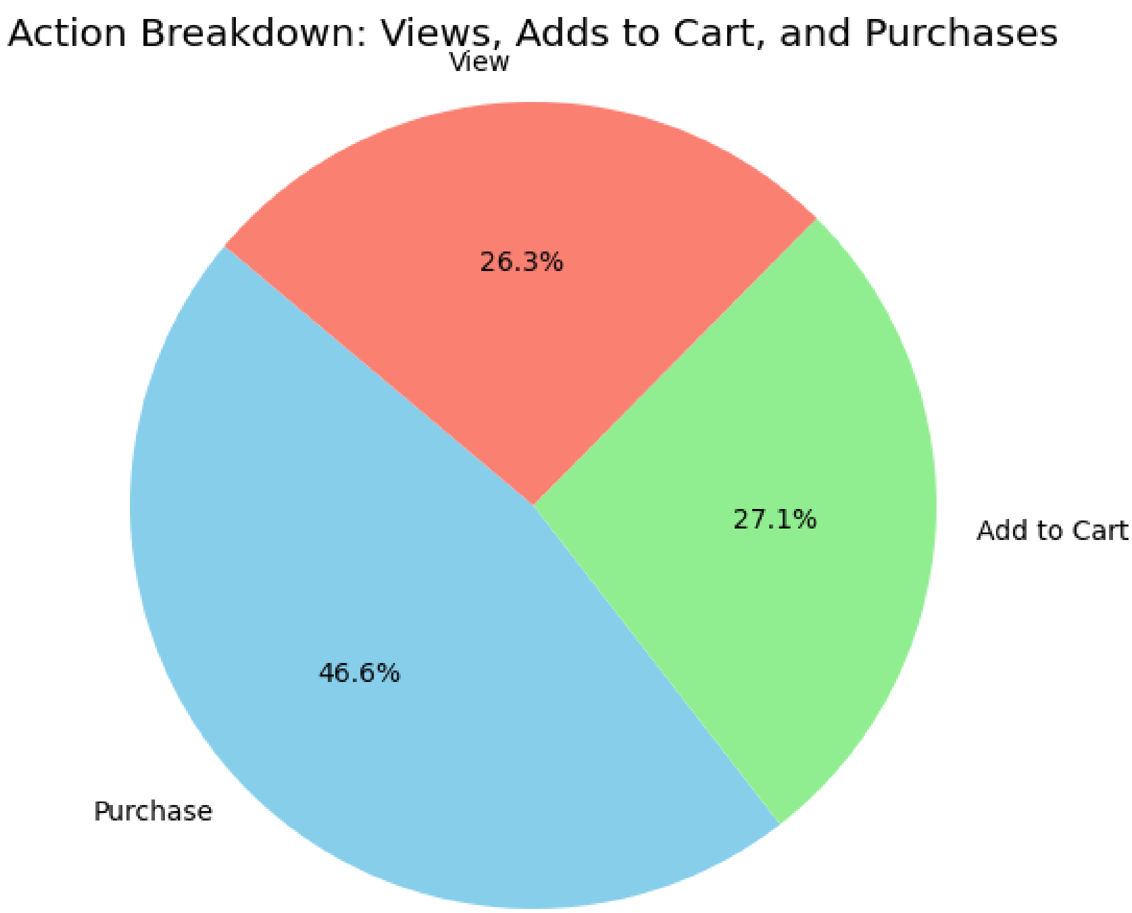
To prepare for modeling, we converted categorical fields (Category, Action) into one-hot (0/1) numeric columns using `pd.get_dummies(dtype=int)`. Then we merged the TotalSpent values back into the main DataFrame and applied Z-score standardization (mean 0, std 1) to Price, Quantity, and TotalSpent via scikit-learn's `StandardScaler`, creating new `_z` columns.

Train/Test Split

Finally, we dropped identifier and raw-feature columns, chose our target variable (`TotalSpent_z`), and split the dataset into 80% training and 20% testing sets with a fixed random seed. This completes the preprocessing pipeline, resulting in clean, numeric, and standardized data ready for model training and evaluation.

<u>Highlight any trends or patterns you observed in the data.</u>

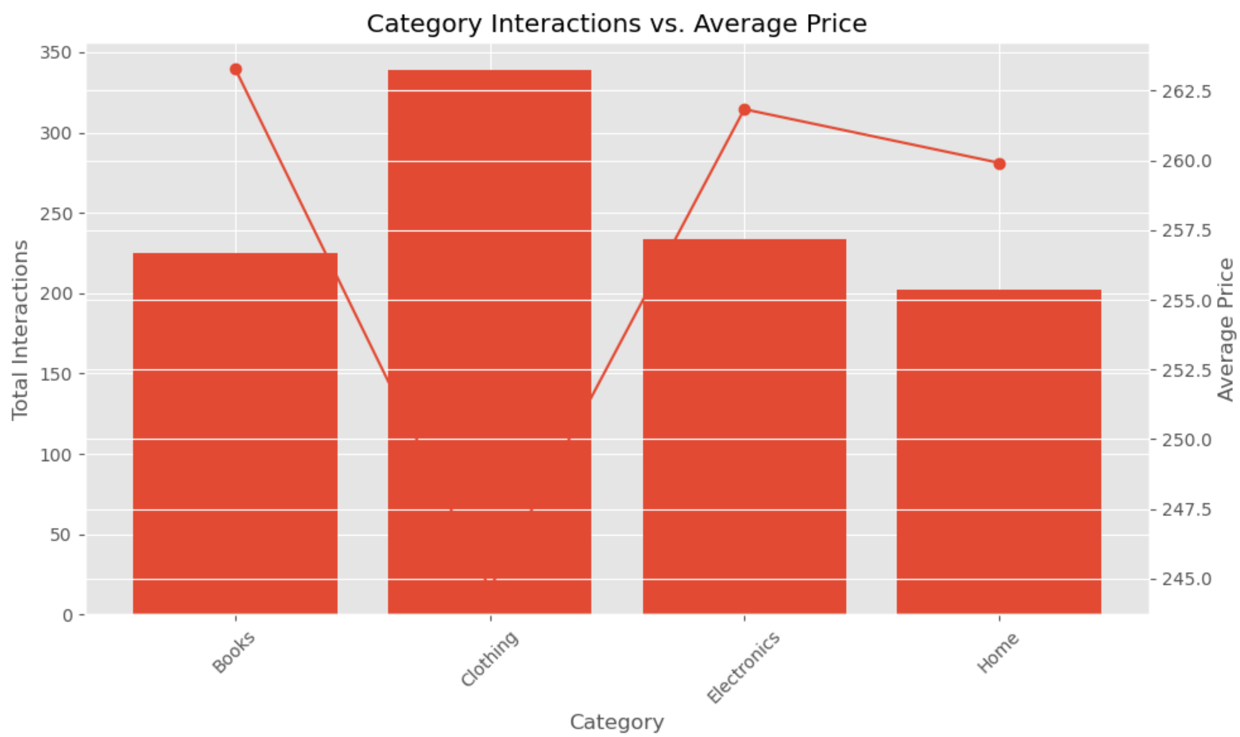
1. Action Breakdown:



Pic-4

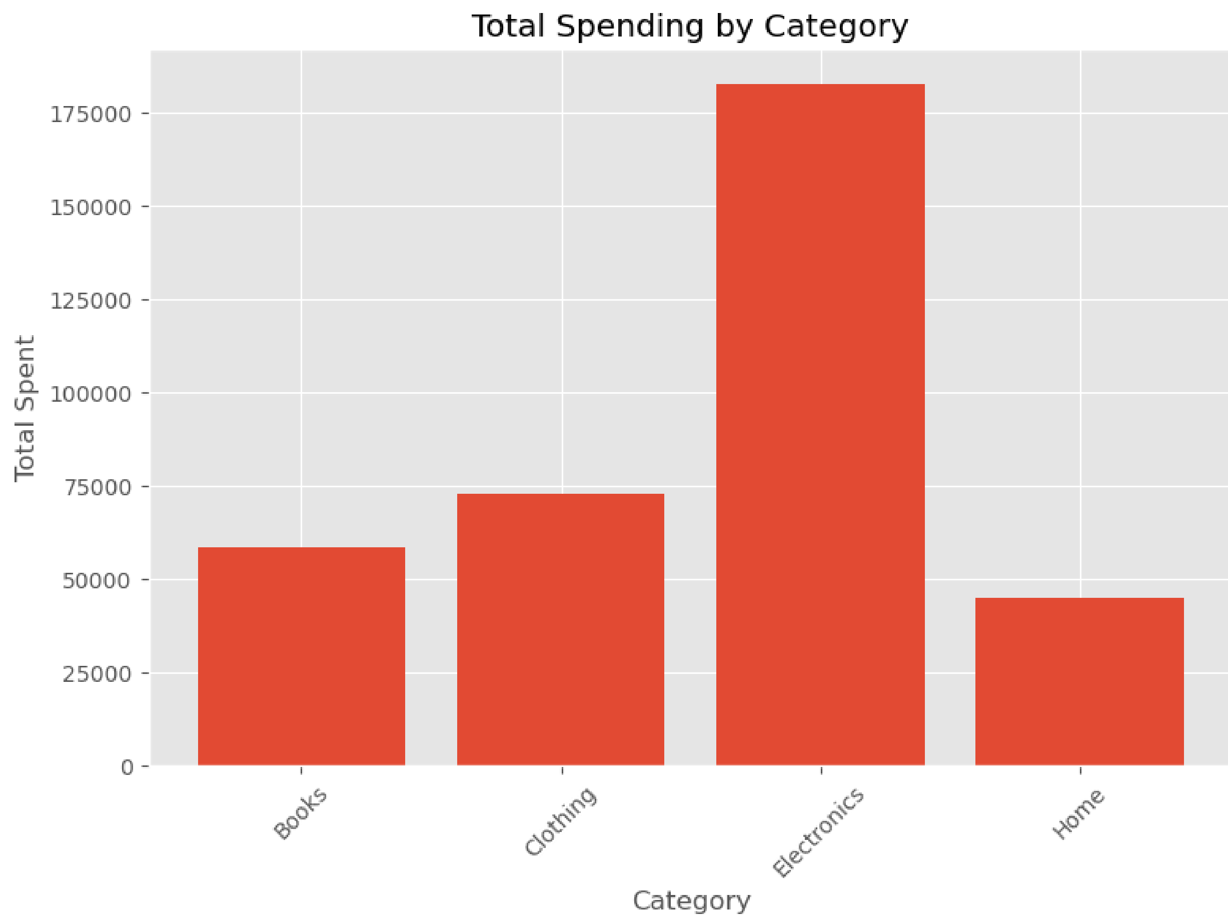
The pie chart illustrates that nearly half of all user actions culminate in a purchase (46.6%), reflecting a highly effective conversion funnel. Viewing and adding items to the cart account for 26.3% and 27.1% of actions, respectively, showing that users not only explore products but also move decisively toward buying.

2. Category Interactions vs. Average Price



A dual-axis chart reveals a nuanced relationship between engagement and pricing across categories. Clothing leads in total interactions, suggesting strong user interest, yet it holds the lowest average price ($252). Conversely, Electronics commands a premium price ($263) while maintaining solid interaction levels, establishing it as a high-value segment that maximizes revenue per engagement.

3. Total Spending by Category



When considering actual revenue (price × quantity for purchases), Electronics emerges as the dominant category, generating approximately \$190 K—more than three times the spending seen in any other category. Books and Clothing follow as mid-range contributors at around \$60 K each, while Home products lag at about \$48 K, signaling an area for targeted growth initiatives.

Feature Engineering and Preprocessing:

we applied strategic feature engineering and preprocessing to prepare the e-commerce dataset for meaningful analysis and potential machine learning applications. We handled **missing values** using a context-aware approach: categorical features like *Category* were filled using the most frequent category (mode), while numerical fields like *Price* were imputed using median values grouped by category, preserving contextual pricing behavior. A new feature, **TotalSpent**, was engineered by multiplying *Price* with *Quantity*, enabling direct analysis of customer value and purchase volume. To capture relationships between categorical variables and the target features, we applied **one-hot encoding** to variables like *Category* and *Action*, allowing models to interpret them numerically without assuming order. For numerical stability, features like *Price*, *Quantity*, and *TotalSpent* were **standardized using Z-score normalization** to center them around a mean of 0 and standard deviation of 1—essential for many machine learning algorithms. Finally, the dataset was **split into training (80%) and testing (20%) subsets** to support model development and evaluation while preventing overfitting. These choices were made to ensure data quality, preserve semantic meaning, and prepare the data in a scalable and model-ready format.