## Project Documentation: Data Preprocessing and Exploratory Data Analysis (EDA)

### Step 1: Importing Necessary Libraries

To begin, all essential libraries were imported to facilitate data manipulation, transformation, and exploratory analysis. \*\*Pandas\*\* was chosen for its versatility in data handling and manipulation, while \*\*NumPy\*\* aided in performing mathematical operations efficiently. \*\*Scikit-Learn\*\* provided robust tools for data preprocessing, including imputation, encoding, and scaling, crucial for setting up a streamlined and consistent workflow. Additionally, \*\*Sweetviz\*\* was used for generating automated EDA reports.

### Step 2: Loading the Dataset and Initial Data Inspection

The dataset was loaded from a local directory, and the first few rows were printed to gain a preliminary understanding of the structure, columns, and content of the data. This initial inspection was essential for identifying potential inconsistencies and provided a glimpse into the variables we would analyze further.

### Step 3: Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is fundamental in understanding the dataset’s characteristics, detecting patterns, and identifying potential issues. The EDA process was executed as follows:

1. \*\*Data Overview\*\*: A summary of the dataset was generated, showing data types, column names, and the number of non-null entries per column. Descriptive statistics were also applied to examine central tendencies, spread, and the range of values, helping in preliminary observations of the data’s structure.

2. \*\*Feature Engineering\*\*: This stage involved analyzing existing variables for potential transformations or combinations that might enhance model performance. Feature engineering could include deriving new features from existing ones or applying transformations that help reveal insights in the data.

3. \*\*Handling Missing Values\*\*: Missing data patterns were evaluated, revealing which columns had null entries and the extent of the missingness. This analysis guided the imputation strategy for replacing missing values appropriately.

4. \*\*Duplicate Entries\*\*: The dataset was checked for duplicate records, as duplicates could distort model training and lead to incorrect inferences. Identifying and addressing duplicates ensured the integrity and reliability of the dataset.

5. \*\*Partitioning Columns by Type\*\*: The columns were divided into numeric and categorical categories. This partitioning was necessary as numeric and categorical data require distinct preprocessing techniques, such as scaling for numeric data and encoding for categorical data.

### Step 4: Data Preprocessing Pipeline

A data preprocessing pipeline was developed to apply multiple transformations consistently. Each transformation was methodically chosen to address specific data characteristics:

1. \*\*Outlier Removal\*\*: The Winsorization technique was applied to numeric features to limit the influence of extreme values. Winsorization capped values at specific percentiles, effectively reducing outliers while retaining the structure of the data.

2. \*\*Mean Imputation\*\*: For handling missing values in numeric columns, mean imputation was employed. This technique replaced missing entries with the mean of the respective column, ensuring a balanced approach that preserved the overall distribution of values.

3. \*\*One-Hot Encoding\*\*: Categorical variables were encoded using One-Hot Encoding, which converts categories into binary columns. This method avoided introducing any ordinal bias in non-numeric features, allowing categorical data to be used effectively in the model.

4. \*\*Scaling\*\*: Min-Max Scaling was used to normalize the range of numeric features to between 0 and 1. Scaling is crucial for algorithms sensitive to the magnitude of values and helped align the dataset’s numeric columns to a common scale, improving model stability and performance.

The entire data preprocessing process was managed within a \*\*Pipeline\*\* and \*\*ColumnTransformer\*\*, which allowed each transformation to be efficiently applied to relevant columns. This structure also enabled easy adjustments and reproducibility in future data processing workflows.

### Step 5: Converting and Concatenating Processed Data

After preprocessing, all transformed data were converted back into a DataFrame format. Numeric features and One-Hot Encoded categorical features were then combined into a single, cohesive DataFrame. This unified structure was essential for maintaining compatibility with subsequent analytical and modeling steps.

### Step 6: Post-Processing Analysis

Following data preprocessing, further analysis was conducted to validate the transformations and explore relationships within the data:

1. \*\*Skewness and Kurtosis\*\*: The skewness and kurtosis of the data were examined to understand the symmetry and peakedness of distributions. Assessing these properties helped identify any remaining imbalances and confirmed the effectiveness of outlier handling and scaling.

2. \*\*Correlation Analysis\*\*: Correlations between features were computed to investigate potential relationships and multicollinearity issues. This step is vital for understanding feature dependencies and guiding further feature selection or engineering processes, ensuring that the model does not become biased by highly correlated inputs.

3. \*\*Automated EDA Report with Sweetviz\*\*: An automated EDA report was generated using Sweetviz, providing comprehensive, visual insights into the dataset. The report included analyses of distributions, relationships, and summary statistics, streamlining the documentation process and offering a structured overview of the dataset.