**1. Data Overview and Initial Inspection:**

* **Action**: Load the dataset and perform an initial overview to identify missing values, data types, and distribution of values.
* **Justification**: Understanding the structure of the dataset helps inform all subsequent cleaning decisions.
* **Result:**

1. **Shape**: 100,000 rows and 16 columns.
2. **Columns**: 'gender', 'age', 'hypertension', 'diabetes\_pedigree\_function', 'diet\_type', 'star\_sign', 'BMI', 'weight', 'family\_diabetes\_history', 'social\_media\_usage', 'physical\_activity\_level', 'sleep\_duration', 'stress\_level', 'pregnancies', 'alcohol\_consumption', 'diabetes'.
3. **Missing Values**: There are a total of 319,686 missing values across the dataset.
4. **Data Types**: The dataset contains a mix of categorical (object) and numerical (float64) data types.
5. **First 5 Rows**: A snapshot of the first 5 rows reveals some missing values and mixed data for different features.

**2. Handling Missing Values:**

* **Identify Missing Data**: Check the percentage of missing data in each column.
* **Action**:
  + **Remove Columns/Rows**: Drop columns or rows with a high percentage (e.g., >50%) of missing values if they don’t contribute significantly to the analysis.
  + **Impute Missing Values**: Use mean/median for numerical columns and mode for categorical columns, or employ more sophisticated imputation (e.g., k-nearest neighbors).
* **Justification**: This step ensures data integrity and reduces bias while preserving as much useful information as possible.

 **Handled Missing Values**:

* Numerical columns were imputed with the mean.
* Categorical columns were filled with the mode or "Unknown."

 **Outliers**:

* Capped extreme values in sleep\_duration and BMI based on typical ranges.

 **Encoded Categorical Variables**:

* One-hot encoding applied to categorical variables like diet\_type and social\_media\_usage.
* Ordinal encoding used for stress\_level.

 **Feature Engineering**:

* Created a BMI category feature.
* Combined hypertension and family\_diabetes\_history into a single risk score.

 **Normalization**:

* Standardized all numerical columns to ensure consistency.

 **Removed Irrelevant Columns**:

* Dropped the star\_sign column, which was deemed irrelevant.

**3. Outlier Detection and Treatment:**

* **Action**: Identify outliers using box plots or statistical methods (e.g., Z-score, IQR).
* **Justification**: Outliers can distort the analysis and predictions. Deciding whether to remove or transform outliers depends on the context (e.g., rare but legitimate medical cases).

**4. Data Type Conversion:**

* **Action**: Convert columns to appropriate data types (e.g., age as integer, BMI as float).
* **Justification**: Ensuring correct data types helps with proper data handling and computational efficiency.

**5. Encoding Categorical Variables:**

* **Action**: Apply label encoding or one-hot encoding to categorical variables like gender, diet\_type, and social\_media\_usage.
* **Justification**: Machine learning models often require numerical input, so categorical variables must be encoded for compatibility.

**6. Normalization/Standardization:**

* **Action**: Normalize numerical columns like BMI, weight, and age if the data will be used in distance-based algorithms (e.g., KNN) or scale them for models sensitive to variable scales.
* **Justification**: This ensures that no feature dominates due to its scale, which can skew results in machine learning models.

**7. Handling Class Imbalance:**

* **Action**: Check the distribution of the diabetes target variable. Apply techniques like oversampling (e.g., SMOTE) or undersampling if needed.
* **Justification**: Balanced class distributions improve the predictive power and fairness of classification models.

**8. Feature Engineering:**

* **Action**: Create new features that could add value (e.g., BMI category based on thresholds, age groups).
* **Justification**: Well-thought-out features can improve model performance and interpretability.

**9. Data Splitting:**

* **Action**: Split the dataset into training and testing sets.
* **Justification**: Splitting ensures models are evaluated on unseen data to prevent overfitting and measure real-world performance.

**10. Documentation of Each Step:**

* **Action**: For each step taken, document:
  + The specific operation performed.
  + The number or proportion of data points affected.
  + Any assumptions made (e.g., imputing age with the mean).
* **Justification**: A clear record provides transparency and traceability, crucial for collaborative projects and reproducibility.

**Detailed Report Template:**

Create a structured document or Jupyter Notebook with the following sections:

* **Introduction**: Purpose and overview of the dataset.
* **Initial Data Inspection**: Summary of initial findings.
* **Data Cleaning Steps**: Detailed steps and justifications.
* **Preprocessing Techniques**: How and why features were prepared.
* **Visualizations**: Use plots to show distributions, outliers, and missing data patterns.
* **Conclusions**: Summary of the cleaned and prepared dataset, highlighting the data ready for modeling or further analysis.