**Dataset Overview:**

The dataset consists of **299 records** and **13 columns** related to clinical records of heart failure patients. Here's a brief description of the columns:

1. **age**: Patient's age (numeric, float).
2. **anaemia**: Whether the patient has anemia (binary, 0 = no, 1 = yes).
3. **creatinine\_phosphokinase**: Level of CPK enzyme in the blood (numeric, int).
4. **diabetes**: Whether the patient has diabetes (binary, 0 = no, 1 = yes).
5. **ejection\_fraction**: Percentage of blood leaving the heart after each contraction (numeric, int).
6. **high\_blood\_pressure**: Whether the patient has high blood pressure (binary, 0 = no, 1 = yes).
7. **platelets**: Platelets count in the blood (numeric, float).
8. **serum\_creatinine**: Level of creatinine in the blood (numeric, float).
9. **serum\_sodium**: Level of sodium in the blood (numeric, int).
10. **sex**: Gender of the patient (binary, 0 = female, 1 = male).
11. **smoking**: Whether the patient is a smoker (binary, 0 = no, 1 = yes).
12. **time**: Follow-up period in days (numeric, int).
13. **DEATH\_EVENT**: Whether the patient died during the follow-up period (binary, 0 = no, 1 = yes).

**Exploratory Data Analysis (EDA):**

**Steps for EDA:**

1. **Summary Statistics**:
   * Examine distributions of numerical variables (e.g., age, creatinine levels).
   * Use .describe() to understand central tendencies, spread, and outliers.
2. **Null Values and Data Types**:
   * Check for missing or null values.
   * Verify the data types of each column.
3. **Class Distribution**:
   * Analyze the target variable (DEATH\_EVENT) to understand class balance.
4. **Correlation Analysis**:
   * Compute correlations between numeric variables.
   * Use a heatmap to visualize relationships.
5. **Outlier Detection**:
   * Identify outliers using boxplots for numerical features.

**Visualization:**

1. **Using Matplotlib**:
   * Histogram of age, serum\_creatinine, and platelets.
   * Boxplot to detect outliers in ejection\_fraction or serum\_sodium.
   * Bar plots for binary variables like anaemia and smoking.
2. **Using Plotly**:
   * Interactive scatter plots (e.g., age vs. creatinine\_phosphokinase).
   * Pie chart for the DEATH\_EVENT distribution.
   * Bubble plots to analyze multiple variables simultaneously.
3. **Correlation Heatmap**:
   * Use Seaborn or Plotly for a correlation heatmap of numeric variables.

**Applying a Machine Learning Algorithm:**

**Model: Linear Regression or Logistic Regression**

1. **Linear Regression**:
   * **Objective**: Predict continuous outcomes like time (follow-up period).
   * **Features**: Use relevant predictors like age, serum\_creatinine, platelets, etc.
   * **Outcome**: A regression line/model that predicts follow-up time.
2. **Logistic Regression**:
   * **Objective**: Predict the binary outcome DEATH\_EVENT (0 or 1).
   * **Features**: Use predictors such as age, anaemia, diabetes, ejection\_fraction, etc.
   * **Outcome**: Classification model with probabilities for survival or death.

**Steps:**

1. Preprocess the dataset (scale numerical variables, handle class imbalance if needed).
2. Split the dataset into training and testing sets.
3. Train a simple model (e.g., Logistic Regression).
4. Evaluate using metrics such as accuracy, precision, recall, and AUC-ROC for classification.