CytoAutoCluster

**1. Dataset Overview**

* The dataset includes multiple features like Cell\_length, DNA1, DNA2, CD45RA, CD133, and more.
* The dataset contains 38 columns and numerous rows (265,627 entries).
* Labels are included to classify data, along with indicators for labeled (label) and unlabeled (NaN labels) data.

**2. Data Preprocessing**

* **Column Selection**:
  + Irrelevant columns like Time, file\_number, and event\_number are dropped to focus on essential features.
* **Corrupted Data Generation**:
  + A binary mask is created with a certain probability (keep\_probability).
  + Data is shuffled and masked to generate corrupted datasets used for self-supervised learning.

**3. Feature Splitting**

* The dataset is divided into:
  + **Labeled Features (x\_labeled)**: Data with associated labels.
  + **Unlabeled Features (x\_unlabeled)**: Data without labels.
* Target variables (y\_labeled) are extracted for supervised learning.

**4. Modeling**

* **Train-Test Split**:
  + Labeled data is split into training (70%) and testing (30%) sets.
* **Logistic Regression**:
  + Logistic regression is applied to predict probabilities.
  + Log loss metric is used to evaluate performance.
* **XGBoost**:
  + XGBoost is employed as another classification model with log loss as an evaluation metric.
* **Self-Supervised Learning**:
  + Neural network with two outputs:
    - **Mask Estimation**: Predicts corrupted data points.
    - **Feature Estimation**: Recovers original features.
  + Parameters:
    - Input layer matches the number of features.
    - Hidden layer with ReLU activation.
    - Output layers with Sigmoid activation for mask and feature predictions.
  + Training setup:
    - 50 epochs with batch size 128.
    - Loss function combines binary cross-entropy (mask) and mean squared error (feature) weighted by alpha.

**5. Model Performance Evaluation**

* **Logistic Regression Results**:
  + Provides predicted probabilities and log loss for evaluation.
* **XGBoost Results**:
  + Shows probabilities and achieves a lower log loss compared to logistic regression.
* **Neural Network Training**:
  + Loss values for both mask estimation and feature estimation are tracked across 50 epochs.
  + Training time and loss reduction show model convergence over epochs.

**6. Key Outputs**

* Probabilities from logistic regression and XGBoost.
* Loss values across models (e.g., log loss for both XGBoost and logistic regression).
* Neural network weights and encoder architecture were saved for further use.

**7. Technical Implementations**

* Tools and Libraries:
  + pandas, numpy for data handling.
  + scikit-learn for logistic regression and preprocessing.
  + XGBoost for gradient boosting.
  + tensorflow.keras for building the self-supervised learning model.
* **Data Scaling**:
  + Features were standardized using StandardScaler before feeding into models.