Model Documentation Overview

This documentation provides an overview of the modules and files used in our model development. The following Python libraries and their respective versions were utilized to ensure compatibility and proper functioning of the code.

Module Versions

- **Python version**: 3.11.9
- **Pandas version**: 2.2.2
- **NumPy version**: 1.26.4
- **Seaborn version**: 0.13.2
- **Matplotlib version**: 3.9.2
- **Scikit-learn version**: 1.5.1
- **XGBoost version**: 2.1.1
- **LightGBM version**: 4.5.0
- **CatBoost version**: 1.2.5
- **Dill version**: 0.3.8

These libraries are essential for data manipulation, model building, evaluation, and saving/loading the model pipelines.

Code and Notebooks

1. **Model Development**
   To understand our detailed approach to model development, please refer to the final_kernel.ipynb notebook. This file contains all the code, methodologies, and steps we followed for data preprocessing, feature engineering, model training, evaluation, and final model selection.
2. **Training and Testing**
   To train or test the model on new datasets, you can use the inference_final.ipynb notebook. This file is designed for applying the trained model to new data and generating predictions.

Experiment Tracking

All experiments conducted during the model development phase are stored in the **experiment** folder. This includes hyperparameter tuning, model comparisons, and performance metrics across different models.

Reports and Presentations

For a summary of the project, along with visual reports and slides, refer to the **Reports_and_Presentation** folder. It contains presentations and written reports outlining the project's methodology, findings, and conclusions.

Results

Screenshots of the model's results, including key metrics and outputs, are saved in the **Our_Results** folder for quick reference. This structure ensures easy access to key materials and a clear understanding of the model's workflow from development to deployment.

Steps to Train and Test the Model on Your Data

To train and test the model on your own dataset, follow these steps. This section outlines the process for replacing our sample data with your data and generating predictions and evaluation metrics using our pre-built model.

Steps to Train and Test the Model

1. **Prepare Your Data**
   Replace the following files with your own data files (ensure that the format and column structure remain consistent):
   - X_Train_Data_Input.csv: Replace this with your training data features.
   - Y_Train_Data_Target.csv: Replace this with your training true output labels (target values).
   - X_Test_Data_Input.csv: Replace this with your testing data features.
   - Y_Test_Data_Target.csv: Replace this with your testing true output labels (target values).

These files should have the exact same names as listed above when placed in the folder where the code will be executed. Ensure that your data is properly formatted as per our original dataset (features and labels in separate CSV files).

2. **Run the inference_final.ipynb Notebook**
   Once you've replaced the data files, open and run the inference_final.ipynb notebook. This notebook will:
   - Load the pre-trained model.
   - Process the new training and testing data.
   - Train the model on your new training data.
   - Make predictions on your testing data.
3. **Generate Results and Evaluation Metrics**
   After running the notebook, it will output the results, including predictions and key evaluation metrics such as Accuracy, Precision, Recall, F1 Score, and ROC AUC for your testing data. These metrics will help you assess the performance of the model on your dataset. The notebook is designed to handle the entire workflow from loading your data to evaluating the model, so you can focus on interpreting the results.

By following these steps, you can effectively train and test our pre-built model on your own dataset while obtaining performance metrics tailored to your specific data.

This structure provides a clear, organized presentation of your model documentation, making it easy for readers to understand the components and processes involved.