NOVARTIS

NEST

Nurturing Excellence,
Strengthening Talent.

PS(04)-Utilizing data to predict recruitment rate (RR) in clinical trial for benchmarking

README FILE

File Preparation for Modelling

02

Note: ***We used a single google colab notebook for the entire workflow instead of separate ones for different tasks. This made the code cleaner, more consistent, and easy to follow. It also ensured that anyone could run all the cells at once without errors.***

Analyzing Hackathon Metadata

The hackathon metadata included detailed definitions of each feature along with their biomedical relevance. This analysis was critical to understanding the domain and ensuring the inclusion of meaningful features.

Relevant features for the model were chosen based on their direct influence on recruitment rates, as defined by prior research and established criteria. The rationale for selecting these features was documented, ensuring both transparency and reproducibility.

Switching to CPU for Model Training

Once embeddings and preprocessing were complete, the workflow was switched to CPU usage for model training due to the limited GPU availability on Google Colab. This ensured efficient resource management while continuing the training process.

Insights were derived during training by evaluating the model's performance using key metrics (e.g., RMSE, R-squared) to ensure robustness.

Test Data Analysis

The test dataset was analyzed for completeness, revealing no missing values.

This ensured that no additional imputation or cleaning was required during testing.

The cleaned and preprocessed test data was ready for model inference.

Prediction on Test Data

The **gbm_model_log.pkl** file, which contained the best-trained model, was loaded for predictions on the test dataset.

The predictions were generated using the trained model, leveraging all relevant features and embeddings to ensure high accuracy.

Embeddings Generation and Preprocessing

The processed dataset was uploaded to Google Colab, leveraging its T4 GPU for computational tasks like embeddings generation and data preprocessing.

Note: The limited free time available on Colab was utilized efficiently by generating embeddings for textual features and performing other preprocessing steps.

After embeddings were generated, all intermediate results, including **.npy** files, were stored both on the local runtime and on Colab. These files were also downloaded locally from the runtime for future use and backup.

Date Transformation

Since 46 rows in the **Primary Completion Duration of Trial** column contained non-standard date formats that **NumPy's datetime** functions could not recognise properly. Specifically, the formula ***=X2 - V2*** was applied to calculate the duration between two relevant date columns. This formula was then dragged down the entire dataset, ensuring consistent calculations for all rows. Excel's intuitive interface made it easier to validate and correct any irregularities, minimizing errors and ensuring uniform date representations for both the training and testing datasets.

Note : *We tried to keep the report as clear and to the point as possible, avoiding unnecessary details at the same time ensuring that each and every point holds up a significant reason.*

File Conventions

Below is an overview of the naming conventions and purposes for each file and folder:

- **Hackathon Metadata (Excel file)**: Provides metadata and notes relevant to the hackathon project.
- **regressor_training_PREifinal (Jupyter Notebook)**: The primary notebook for model training, including data preprocessing, model configuration, and all related code.
- **DETAILED REPORT** (PDF): Summarizes the project's methodology, results, visualizations, and conclusions.
- **bayes_opt_fine_tuning (Jupyter Notebook)**: Performs Bayesian optimization and hyperparameter tuning for the model using a log-transformed target variable.
- **X_train_combined.npy (NumPy file)**: Consolidated training dataset (features) containing both numerical data and BioBERT embeddings.
- **X_test_combined.npy (NumPy file)**: Consolidated test dataset (features) containing both numerical data and BioBERT embeddings.
- **X_train_embeddings.npy / X_test_embeddings.npy (NumPy files)**: Precomputed BioBERT embeddings for the training and test sets, respectively.
- **gbm_model_log.pkl (Pickle file)**: Log or checkpoint file with the best parameters for the Gradient Boosting Model (GBM) in the log-transformed target variable scenario.
- **gbm_model.pkl (Pickle file)**: Log file with best model weights for GBM for the case of target StandardScaled Target variable scenerio.
- **PLOTS (Directory)**: Contains charts, graphs, and other visual outputs utilized during analysis.
- **testing (Directory)**: Includes test scripts, test data, and model predictions ,embeddings utilized while performing prediction on test data.
  - **data_preprocessed.xlsx :** The data_preprocessed.xlsx file is the final preprocessed dataset that was subsequently used for model training.
  - **test_prediction_final (Jupyter Notebook)**:Contains the final prediction pipeline or analysis steps specifically for test data.
  - **final_test_data_predictions (Excel file)**:Holds the final prediction outputs generated by the model on the test set.
  - **test_data_preprocessed (Excel file)**: The preprocessed test dataset, prepared (cleaned and transformed) for the model.
  - **X_test_data_combined.npy (NumPy file)**: Consolidated test feature set combined (numerical and embeddings) in .npy format.
  - **X_test_data_embeddings.npy (NumPy file)**: Embeddings extracted (e.g., using BioBERT) for the test data, stored for efficient loading.
  - **Use_Case_4_Test_dataset_To_be_shared (Excel file)**: Another excel file which had the test datasets on which predictions are to be done using the trained model.
- **usecase_4_ (Excel file)**: Contains training data with date columns transformed via Excel features, while other columns remain unchanged.