Model Code and Documentation

Overview

This script demonstrates the process of building, training, and evaluating a classification model using LightGBM with hyperparameter optimization via Hyperopt. The workflow includes data preprocessing, model training, and evaluation of model performance metrics.

Documentation

Libraries Used

- Pandas: For data manipulation and analysis.
- Numpy: For numerical operations and handling arrays.
- Scikit-learn: For machine learning utilities, preprocessing, and evaluation metrics.
- LightGBM: For gradient boosting framework to create the classification model.
- Hyperopt: For hyperparameter optimization.
- Matplotlib & Seaborn: For data visualization.

Workflow

1. Data Loading: Loads training and test datasets and combines features and labels.
2. Data Preprocessing:
    - - Missing values are handled using simple and iterative imputers.
    - - Unnecessary features are dropped to reduce dimensionality.
    - - Outliers are detected and removed from the training set.
    - - Features are standardized using `StandardScaler`.
3. Hyperparameter Optimization:
    - - Defines a search space for hyperparameters.
    - - Uses Hyperopt to find the best parameters by minimizing validation loss.
4. Model Training:
    - Trains a final LightGBM model using the optimized hyperparameters.
5. Model Evaluation:
    - Evaluates the model using various metrics: accuracy, precision, recall, F1 score, ROC AUC, balanced accuracy, and log loss.
    - Visualizes the confusion matrix, ROC curve, and precision-recall curve for model performance insights.
6. Model Accuracy Comparison: Compares train and test accuracy to assess overfitting.

Hyperparameter Search Space

- `n_estimators`: Number of boosting iterations (100 to 1000).
- `learning_rate`: Step size shrinkage (0.01 to 0.3).
- `max_depth`: Maximum tree depth (3 to 10).

- `num_leaves`: Maximum number of leaves per tree (20 to 100).
- `min_child_samples`: Minimum number of samples per leaf (10 to 100).
- `subsample`: Fraction of samples to be used for fitting (0.5 to 1.0).
- `colsample_bytree`: Fraction of features to be used for fitting (0.5 to 1.0).
- `reg_alpha`: L1 regularization (1e-8 to 1.0).
- `reg_lambda`: L2 regularization (1e-8 to 1.0).

Conclusion

The LightGBM model demonstrates strong performance across multiple metrics, indicating it is well-suited for the task at hand. The hyperparameter tuning using Hyperopt effectively optimized the model, contributing to its high accuracy and other performance metrics. Future work may include exploring more advanced modeling techniques, additional feature engineering, and cross-validation strategies to further enhance model robustness.