Machine Learning Automator

Contents

Context

This report explains the steps involved in the machine learning pipeline designed to train and evaluate various classification models. The goal is to provide an understanding of how different parts of the code contribute to building and validating these models.

Requirements

Inputs

The CSV files must not contain missing values.

For the Docker version an Input volume will be a folder with Train.csv and Test.csv files within. CSVs must not contain missing values and they need to have the target as the last column

1. **Train.csv:** A csv file with the features and the target as the last column. Here the K-Fold cross validation will be performed, and the thresholds are going to be tuned.

2. **Test.csv:** A csv file with the features and the target as the last column. Here only the validation of the threshold, the metrics, the roc curves and the shapley analysis are going to be performed

Outputs

Materials Folder:

For the Docker Version Output Volume would be an empty folder on your system that is going to be filled with the following files after the execution

1. **ROC CURVES.PNG file** with the ROC curves on each algorithm on the test set
2. **ShapFeatures folder** where a number of png images with the shap values on each algorithm are shown.
3. **Excel file** with the metrics for the algorithm on the internal k-fold
4. **Excel file** with the metrics for the algorithm on the external set
5. **Models' folder** where the pickle models are shown. The ones that the external data were evaluated. The Pickle files are pipelines and not individual models. Therefore the user can use it as is without performing manual feature selection or preprocessing.

Main Advantages

- Automated feature selection & preprocessing
-
- K-Fold Stratified Cross validation on the **Train.csv**
- Threshold automated calculation based on the validation splits from K-Fold
- Hyperparameter Tuning on the Stratified K-Fold
- Testing on the Test.csv with the best hyperparameters from the internal stratified K-Fold and the Average threshold across Folds
- 5 Metric reported on the **Internal Stratified K-Fold** & **The external Set (test.csv)**
- **ROC** Curves (Diagram provided)
- **Shapley Analysis** on external Set (Diagram Provided)

Overview

The pipeline evaluates six different classification models:

Logistic Regression

Support Vector Machines

Random Forest

AdaBoost

Decision Trees

XGBoost

Each model undergoes:

1. Hyperparameter tuning on a stratified automated calculated k-fold stratified scheme on the train.csv set. It calculates the best parameters from a predefined grid.

2. K-Fold Training on the best parameters across folds

3. Evaluation using a stratified k-fold cross-validation approach.

Key Concepts

Hyperparameters: These are parameters that are set before the training process begins and control the behavior of the training algorithm.

Cross-validation: A technique used to evaluate the performance of a model by splitting the data into multiple folds and training/testing the model on different combinations of these folds.

Pipeline: A sequence of data processing and model training steps that are applied consistently across all models.

Feature Selection

Feature Selection: Identifies and retains important features based on correlation.

The user can select across a variety of feature selection strategies.

"featurewiz" (Default): Automated Feature selection based on correlation matrix and XGBoost selection

Params:

- **corr_limit (defaults to 0.6):** Threshold for the correlation matrix

"rfe" Recursive Feature Elimination using Logistic regression algorithm

Params:

- **n_features_to_select(defaults to 5):** Number of Features to keep

“lasso”: Lasso algorithm

“random_forest” : Automated Feature selection based on correlation matrix and XGBoost selection

“xgboost”: Automated Feature selection based on correlation matrix and XGBoost selection

Preprocessing

Prepares the data for training (e.g., scaling, handling missing values).

Tabular: One Hot encoding

Numeric: Z-Score

Hyperparameter Tuning & Training

Hyperparameter Tuning: Uses exhaustive grid search to find the best hyperparameters.

Training: Trains the model on the training data.

Evaluation

Threshold Optimizer: Finds the optimal threshold on the train set for each fold, based on **AUC** metric as default and stores them.

The user can select across 5 metrics

- AUC
- F-Score
- Accuracy
- Sensitivity
- Specificity

Evaluation: Evaluates the model on the validation data on each fold

Testing on the External Set

Context: The models are retrained overall train.csv file which is the training dataset and the hyperparameters are set based on the K-Fold selection in the previous step. Also, the Threshold for the models are set as the average of the thresholds on the K-Fold on the previous step as well

- Metrics computed on the test.csv file for the optimal threshold identified previously
  The metrics are:
  - **AUC**
  - F-Score

    - Accuracy
    - Sensitivity
    - Specificity
- **Shapley Analysis** is performed for each algorithm on a fraction of the test set (100 or length of test.csv if the number of instances are less than 100)
- **ROC Curves**: Finally the ROC Curves on the testing dataset are reported for each algorithm on a single diagram