

A Machine Learning Framework for Performing Binary Classification on Tabular Biomedical Data

Running title: **An ML Framework for Binary Classification**

Ádám Szijártó, M.Sc.¹

Alexandra Fábián, M.D.¹

Bálint Károly Lakatos, M.D. Ph.D.¹

Máté Tolvaj¹

Béla Merkely, M.D., Ph.D., D.Sc.¹

Attila Kovács, M.D., Ph.D.¹

Márton Tokodi, M.D. Ph.D.¹

¹Heart and Vascular Center, Semmelweis University, Budapest, Hungary

CORRESPONDENCE

Márton Tokodi, M.D., Ph.D.

Heart and Vascular Center, Semmelweis University

Address: 68 Városmajor Street, Budapest 1122, Hungary

Phone: +3630-465-23-56

Fax: +361-458-68-42

E-mail: tokmarton@gmail.com

WORD COUNT: 1,569 words

ABSTRACT

Background and aim: Over the past decades, we have witnessed an immense expansion in the arsenal and performance of machine learning (ML) algorithms. One of the most important fields that could benefit from these advancements is biomedical science. To streamline the training and evaluation of binary classifiers, we constructed a universal ML framework that uses tabular biomedical data as input.

Methods and results: Our framework requires the input data to be provided as a comma-separated values file, in which rows correspond to subjects and columns represent different features. After reading the content of this file into a data frame, the framework enables the users to perform outlier detection, handle missing values, rescale features, and tackle class imbalance. Then, hyperparameter tuning, feature selection, and internal validation are performed using nested cross-validation. If an additional dataset is available, the framework also provides the option for external validation. Users may also compute SHapley Additive exPlanations values to interpret the individual predictions of the model and identify the most important features. Our ML framework was implemented in Python (version 3.9), and its source code is freely available via GitHub. In the second part of this paper, we also demonstrate the usage of the framework through an example use case from the field of cardiovascular imaging.

Conclusions: The proposed ML framework enables the efficient training and evaluation of binary classifiers on tabular biomedical data. We hope our framework will serve as a useful resource for both learning and research purposes and will promote further innovation.

Keywords: machine learning, supervised learning, binary classification, tabular data

INTRODUCTION

With data-heavy technologies entering the clinical arena, a staggering volume of data is generated along the diagnostic and therapeutic pathways that patients follow. Taming this data tsunami poses a great challenge to physicians, and we may declare that the appropriate interpretation of the steadily growing amount of medical data has surpassed the capacity of the human mind [1].

Modern data analysis techniques – such as machine learning (ML) – can aid physicians in overcoming this challenge as they are aptly suited for digesting the tremendous amount of information encompassed within structured and unstructured biomedical datasets [2]. Seeing all the innovations sparked by ML in other areas of our life, there is little doubt that ML will also revolutionize healthcare. By automating several time-consuming subtasks and augmenting diagnostics, therapeutics, and prognostication, ML-based tools will also improve the allocation of healthcare resources and allow multitasking physicians to remain focused on the patient and delivering high-quality care, even in scenarios where specialists or high-end diagnostic equipment are out of reach [3, 4]. Despite their capabilities and usefulness, the most widely accepted view is that these ML-based tools will not replace physicians but will be used extensively as decision support systems [5]. Therefore, it will be inevitable for physicians to become acquainted with ML, most preferably through hands-on learning.

To provide a well-documented resource for both learning and research purposes, we implemented a framework in Python for training and evaluating ML models that perform binary classification on tabular biomedical data. In addition, we also demonstrated the usage of the proposed framework through an example from cardiovascular imaging.

THE MACHINE LEARNING FRAMEWORK

Purpose of the framework

Our main motivation for creating this framework was to provide a well-documented and extensible codebase for the training and evaluation of binary classifiers that use tabular biomedical data as input. Our framework was designed and implemented in accordance with the Proposed Requirements for Cardiovascular Imaging-Related MachineLearning Evaluation (PRIME) checklist (Supplementary Table 1) [6]. We implemented the framework in Python (version 3.9) and used the scikit-learn [7] (version 1.0.1) and XGBoost [8] (version 1.6.2) libraries for ML. The source code of the framework is available via GitHub (<https://github.com/szadam96/framework-for-binary-classification>). Although the framework is extensively documented in this repository, we provide a concise description and highlight its major components in the following subsections.

Input data

The framework requires the input data to be provided as a comma-separated values (CSV) file, in which rows correspond to subjects (*i.e.*, patients) and columns represent different features.

Outlier detection

Outliers are data points significantly different from the rest of the dataset. To ensure that the trained ML model generalizes well to the valid range of test data, it's pivotal to detect and handle outliers. In our framework, outlier detection is performed in an unsupervised fashion based on the local outlier factor values [9]. However, given the complexity of outlier handling,

it is the user's responsibility to review each detected outlier manually and confirm whether it is a true outlier and how it should be dealt with (*e.g.*, correct or remove it manually).

Handling missing data

In biomedical research, we often encounter missing values in our datasets, which must be handled appropriately as many ML algorithms require complete datasets. In our framework, we implemented a two-step approach to tackle missing values. First, features and patients with a higher percentage of missing values than a user-defined threshold are discarded. Then, the user may choose from different imputation methods (continuous features: Multiple Imputation by Chained Equations [MICE] [10], mean imputation, arbitrary value imputation; categorical features: mode imputation or arbitrary value imputation) to replace missing values.

Feature scaling

In biomedical datasets, continuous variables of different domains (*e.g.*, anthropometric measurements and imaging-derived parameters) may lie in vastly different ranges. To eliminate the possibility of model bias caused by these differences, scaling transformations should be applied. In our framework, the user may choose from the following scaling methods: (1) Z-score transformation (*i.e.*, standardization), (2) L1 or L2 normalization, (3) min-max scaling, and (4) robust scaling.

Handling class imbalance

A significant imbalance in data classes is commonly observed in biomedical datasets. If left unaddressed, the performance of ML models trained on such datasets may be skewed, as models predominantly learn the characteristics of the majority class. In our framework, the user may choose to tackle class imbalance using the Synthetic Minority Oversampling Technique

(SMOTE) [11] or by assigning different weights to the majority and minority classes during learning.

Feature selection

With feature selection, we aim to reduce the input space by identifying the optimal subset of features that could potentially maximize our model’s performance while reducing the computational cost. For this purpose, we are using the minimum Redundancy Maximum Relevance (mRMR) algorithm [12], which selects a subset of features having the most correlation with a class (relevance) and the least correlation between themselves (redundancy). To identify the optimal subset of features, models are trained recursively with a decreasing number of features.

Model selection and hyperparameter tuning

Several experiments with different ML algorithms and myriads of hyperparameter settings should be performed to identify the model that yields the best performance in the given classification task. The current version of the framework supports seven commonly used classification algorithms (i.e., logistic regression, k -nearest neighbors, support vector machines, random forest, gradient boosting, extreme gradient boosting, and multilayer perceptron), but other algorithms may be added to the codebase by the users if desired. For each classifier, probability calibration is performed during training. Regardless of the choice of the algorithm, it is imperative to perform hyperparameter tuning. In our framework, grid search is used to search exhaustively through a user-defined hyperparameter space.

Evaluation of model performance

Our framework allows the user to evaluate the models' performance internally with cross-validation (CV) and externally using additional datasets from other centers. For internal validation, we implemented a nested CV procedure in which the k -fold CV procedure for feature selection and hyperparameter tuning is nested inside the k -fold CV procedure for model selection and evaluation. In our framework, numerous metrics can be calculated to evaluate the predictive performance of the trained models, such as the area under the receiver operating characteristic curve (ROC-AUC), average precision (AP), accuracy, balanced accuracy, Brier score, logistic loss, and F1 score.

Interpretability

Our framework also provides the option for the users to compute SHapley Additive exPlanations (SHAP) values [13], which quantify the magnitude and direction (positive or negative) of a feature's effect on a given prediction. By averaging across samples, this method helps estimate the contribution of each feature to overall model predictions for the entire dataset. The mean absolute SHAP value also expresses the importance of a given feature.

AN EXAMPLE USE CASE

Prediction task

In this analysis, our objective was to predict 2-year all-cause mortality from tabular echocardiographic data (*i.e.*, 2D and 3D echocardiography-derived parameters). Of note, the sole purpose of this analysis was to demonstrate the usage of the proposed framework rather than providing an ML model with high clinical relevance. The analysis conforms with the principles outlined in the Declaration of Helsinki, and its protocol was approved by the Semmelweis University Regional and Institutional Committee of Science and Research Ethics.

Description of the dataset

We used an echocardiographic dataset presented in one of our previously published papers [14]. This dataset included 174 patients (62 ± 13 years, 72% males) with different left-heart diseases (*i.e.*, heart failure with reduced ejection fraction patients, heart transplant recipients, and patients with severe mitral valve regurgitation). For further information about the dataset and a detailed description of the echocardiographic protocol, please refer to our previously published paper [14]. To keep it simple, we only used echocardiographic features ($n=54$) from this dataset in the current analysis.

Machine learning analysis

Outliers were detected using the above-described method and were corrected or removed manually. Features with more than 30% missing values ($n=18$) were discarded, and we also excluded patients with more than 30% missing data ($n=31$). Missing values of the remaining continuous features were imputed using the MICE algorithm, whereas missing values of categorical features were replaced with -1. Z-score transformation was performed

after imputation. To address the class imbalance in our dataset, SMOTE was applied. We trained and evaluated the above-listed seven ML algorithms. Feature selection, hyperparameter tuning, and internal validation were performed using nested CV (with 5 folds in both the inner and outer loop), as described in the previous section. ROC-AUC was used as the scoring parameter. We reported our methods and results in compliance with the PRIME checklist (Supplementary Table 1) [6].

Results

The final dataset comprised 143 patients from whom 29 (20.2%) died in the first two years following the echocardiographic examination. The results of the ML analysis are presented in Table 1. Among the evaluated classifiers, random forest (with 18 input features) exhibited the best performance with a ROC-AUC of 0.75 (95% CI: 0.65-0.85) (Figure 1A) and an AP of 0.50 (95% CI: 0.40-0.59) (Figure 1B).

Model interpretability

The beeswarm plot in Figure 1C depicts how the final 18 input features impact the model's output (i.e., the predicted probability of 2-year mortality). Feature values referring to systolic or diastolic dysfunction, adverse remodeling, and more severe valvular regurgitations are associated with an increased risk of death.

CONCLUSIONS

In this paper, we presented an ML framework that we designed to train and evaluate binary classifiers on tabular biomedical data. We hope our framework will serve as a useful resource for both learning and research purposes and will promote further innovation.

AUTHORS' CONTRIBUTION

AS, AK, BM, and MTok designed the framework; AS, AK, and MTok drafted the manuscript; AF, BKL, MTol, and BM reviewed and approved the manuscript; BM and AK acquired the funding; AK, BKL, AF, and MTol performed the echocardiographic measurements and built the dataset used in the example; AS wrote the source code, MTok reviewed and approved the source code. All authors reviewed the final version of the manuscript and agreed to submit it to IMAGING for publication.

FUNDING SOURCES

Project no. RRF-2.3.1-21-2022-00004 (MILAB) has been implemented with the support provided by the European Union. This project was also supported by a grant from the National Research, Development, and Innovation Office (NKFIH) of Hungary (FK 142573 to AK).

CONFLICT OF INTERESTS

AS, AF, BKL, and AK report personal fees from Argus Cognitive, Inc., outside the submitted work. BM reports grants from Boston Scientific and Medtronic and personal fees from Biotronic, Abbott, Astra Zeneca, Novartis, and Boehringer-Ingelheim, outside the submitted work. MTok was a former employee of Argus Cognitive, Inc. MTol has no conflict of interest to disclose.

REFERENCES

1. Obermeyer, Z. and T.H. Lee, *Lost in Thought — The Limits of the Human Mind and the Future of Medicine*. New England Journal of Medicine, 2017. **377**(13): p. 1209-1211.
2. Rajpurkar, P., et al., *AI in health and medicine*. Nature Medicine, 2022. **28**(1): p. 31-38.
3. Rajkomar, A., J. Dean, and I. Kohane, *Machine Learning in Medicine*. New England Journal of Medicine, 2019. **380**(14): p. 1347-1358.
4. Kagiya, N., M. Tokodi, and P.P. Sengupta, *Machine Learning in Cardiovascular Imaging*. Heart Fail Clin, 2022. **18**(2): p. 245-258.
5. Topol, E.J., *High-performance medicine: the convergence of human and artificial intelligence*. Nature Medicine, 2019. **25**(1): p. 44-56.
6. Sengupta, P.P., et al., *Proposed Requirements for Cardiovascular Imaging-Related Machine Learning Evaluation (PRIME): A Checklist*. JACC: Cardiovascular Imaging, 2020. **13**(9): p. 2017-2035.
7. Pedregosa, F., et al., *Scikit-learn: Machine Learning in Python*. J. Mach. Learn. Res., 2011. **12**: p. 2825-2830.
8. Chen, T. and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, Association for Computing Machinery: San Francisco, California, USA. p. 785–794.
9. Breunig, M.M., et al., *LOF: identifying density-based local outliers*. SIGMOD Rec., 2000. **29**(2): p. 93–104.
10. van Buuren, S. and K. Groothuis-Oudshoorn, *mice: Multivariate Imputation by Chained Equations in R*. Journal of Statistical Software, 2011. **45**(3): p. 1 - 67.

11. Chawla, N.V., et al., *SMOTE: synthetic minority over-sampling technique*. The Journal of Artificial Intelligence Research, 2002. **16**: p. 321-357.
12. Ding, C. and H. Peng, *Minimum redundancy feature selection from microarray gene expression data*. J Bioinform Comput Biol, 2005. **3**(2): p. 185-205.
13. Lundberg, S.M. and S.-I. Lee, *A unified approach to interpreting model predictions*, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017: Long Beach, California, USA. p. 4768–4777.
14. Tolvaj, M., et al., *Added predictive value of right ventricular ejection fraction compared with conventional echocardiographic measurements in patients who underwent diverse cardiovascular procedures*. Imaging, 2021. **13**(2): p. 130-137.

TABLES

Table 1. Results of the machine learning analysis in the example use case

Model	ROC-AUC↑	Average precision↑	Accuracy↑	Balanced accuracy↑	Brier score↓	Logistic loss↓	F1 score↑
Logistic regression	0.74 [0.62-0.87]	0.47 [0.25-0.69]	0.78 [0.65-0.92]	0.77 [0.66-0.88]	0.23 [0.18-0.28]	1.10 [-0.12-2.32]	0.60 [0.42-0.78]
<i>k</i>-nearest neighbor classifier	0.68 [0.59-0.78]	0.40 [0.24-0.56]	0.69 [0.54-0.84]	0.72 [0.67-0.78]	0.22 [0.17-0.26]	0.85 [0.24-1.45]	0.52 [0.44-0.60]
Support vector machine	0.74 [0.63-0.85]	0.43 [0.25-0.61]	0.76 [0.60-0.92]	<u>0.78</u> <u>[0.70-0.87]</u>	0.23 [0.18-0.28]	0.66 [0.53-0.79]	<u>0.61</u> <u>[0.44-0.77]</u>
Random forest	<u>0.75</u> <u>[0.65-0.85]</u>	<u>0.50</u> <u>[0.40-0.59]</u>	0.74 [0.53-0.94]	0.75 [0.67-0.84]	<u>0.19</u> <u>[0.15-0.24]</u>	<u>0.57</u> <u>[0.48-0.67]</u>	0.57 [0.44-0.70]
Gradient boosting	0.73 [0.62-0.84]	0.45 [0.35-0.56]	<u>0.80</u> <u>[0.73-0.88]</u>	0.75 [0.64-0.85]	0.21 [0.17-0.25]	1.26 [-0.61-3.13]	0.57 [0.44-0.70]
Extreme gradient boosting	0.72 [0.61-0.83]	0.48 [0.24-0.71]	0.72 [0.50-0.93]	0.74 [0.65-0.83]	0.20 [0.16-0.23]	0.59 [0.50-0.67]	0.57 [0.40-0.74]
Multilayer perceptron	0.62 [0.46-0.78]	0.36 [0.17-0.55]	0.69 [0.46-0.92]	0.66 [0.51-0.81]	0.19 [0.12-0.27]	2.79 [0.29-5.28]	0.41 [0.09-0.72]

The performance of the models was assessed using 5-fold cross-validation (i.e., the outer loop of the nested cross-validation procedure). Each cell contains the average value of a given performance metric with 95% confidence intervals.

ROC-AUC – area under the receiver operating characteristic curve

FIGURE LEGENDS

Figure 1. Results of the machine learning analysis in the example use case

(A) Receiver operating characteristic (ROC) curves of the best-performing random forest model. A mean ROC curve with a 95% confidence band and the individual curves of the folds are also plotted. (B) Precision-recall (PR) curves of the best-performing random forest model. A mean PR curve with a 95% confidence band and the individual curves of the folds are also plotted. (C) Beeswarm summary plot of the SHapley Additive exPlanations (SHAP) values. From top to bottom, features are presented in descending order of importance. Each dot denotes a SHAP value for a given feature value of a single patient. The position of a dot on the x -axis shows the impact of a feature value on the predicted probability of death of a given patient. The color represents the high (red) to low (blue) values of the feature.

A – late mitral inflow velocity, AI – aortic insufficiency, AP – average precision, AUC – area under the receiver operating characteristic curve, E – early mitral inflow velocity, e' – early diastolic mitral annular velocity, IVS – thickness of the interventricular septum, LVEDD – left ventricular end-diastolic diameter, LVEF – left ventricular ejection fraction, LVESV – left ventricular end-systolic volume, LVGCS – left ventricular global circumferential strain, LVGLS – left ventricular global longitudinal strain, LVPW – thickness of the left ventricular posterior wall, RV – right ventricular, RVEDV – right ventricular end-diastolic volume, RVESV – right ventricular end-systolic volume, RVFWLS – right ventricular free wall longitudinal strain, SHAP – SHapley Additive exPlanations, s' – systolic mitral annular velocity, TAPSE – tricuspid annular plane systolic excursion, TI – tricuspid insufficiency