CS-433 Machine Learning: Project 1

Valerio Ardizio, Stefano Viel, Malena Mendilaharzu
Department of Computer Science, EPFL Lausanne, Switzerland

Abstract—**The Behavioral Risk Factor Surveillance System (BRFSS) is the central platform for conducting health-related telephone surveys within the United States. The purpose of this paper is to investigate how machine learning models can be used to analyze the data gathered by the BRFSS, to evaluate an individual's susceptibility to developing MICHD, a type of cardiovascular disease. We trained and tested our model and achieved an accuracy of 0.886 and an F1-score of 0.422.**

I. Introduction

Our study's objective is to build a predictive model that can evaluate an individual's vulnerability to cardiovascular diseases by analyzing their clinical and lifestyle attributes. This report offers a thorough overview of the methodology employed to create such a classifier. Section II explains the selection of our model, while section III outlines the dataset and the exploration of preprocessing techniques. In section IV we delve into the training and fine-tuning of our model's hyperparameters. Subsequently, section V presents the performance and outcomes, and VI offers a concise summary of the work conducted. Ultimately, this model can be used to estimate the likelihood of new patients to develop MICHD.

II. Study of different models

As required by the project instructions, we deployed a range of models including MSE full-batch and stochastic gradient descent, least squares, ridge regression and logistic regression with and without regularization. We began our research by deciding which one of these methods would be the most suitable for the analysis of the data from BRFSS. The selection of our model was primarily driven by the nature of the task at hand, which involves binary classification.

Except for logistic regression, the remaining methods do not inherently scale their output within the 0 to 1 range, making them less suitable for classification tasks due to their reduced interpretability as probabilities and their loss function which isn't suited for classification. For this reason, and supported by further experimental findings, based mainly on accuracy and F1-scores, we employed logistic regression. Specifically, we implemented logistic regression with L2 regularization, and we incorporated $\lambda = 0$ when tuning our hyperparameters to add a representation of the simple logistic regression (refer to section IV).

III. Data Analysis and Pre-Processing

Then, we proceeded to gather additional information about the dataset in order to identify the necessary preprocessing techniques.

The BRFSS training dataset is composed of 328,135 samples, each encompassing 321 distinct features. Upon inspecting the dataset, two key challenges emerge:

1) Firstly, a substantial proportion of the dataset exhibits missing values, accounting for roughly 44% of the total entries.
2) Secondly, there is a noticeable class imbalance with 91.2% of the samples falling into class 0, while only 8.8% falling into class 1.

All experiments in this section use the logistic regression model with no regularization. We compare preprocessing methods by training on various preprocessed data and evaluating performance on a validation set. Limited by computational resources, we use the first 50,000 samples of the dataset, splitting it into 80% training and 20% validation.

A. Dealing with missing values

In addressing the first concern (1), we explored the possibility of substituting these missing values with either zeros or statistical measures of their corresponding feature (such as the mean and the median). As seen in Table 1, despite achieving a higher F1-score when opting for median replacement, the significant reduction in accuracy dissuaded us from adopting this approach. Our decision is guided by the realization that the primary concern lies in not overlooking potential heart disease cases, where the consequences of failing to identify true cases significantly outweigh the introduction of some false positives. Thus, we adopted the zero replacement strategy.

Replacement method	F1-Score	Accuracy
Zeros	**0.2407**	**0.9176**
Median	0.2987	0.6397
Mean	0.2123	0.9155

TABLE I
Results for missing values replacement

B. Dealing with class imbalance

In response to (2), we delved into a variety of methods. These included undersampling, oversampling and introducing artificial data through the injection of noise. The first method resulted in the loss of valuable data, while the latter two considerably extended the training phase. Consequently, we opted to treat the threshold of our prediction function as a hyperparameter to be able to fine-tune our model's leniency to output underrepresented labels, while keeping a low overall complexity (details of this hyperparameter are further explained in section IV).

C. *Removing unnecessary data*

Further, upon a thorough examination of the nature of the 321 features provided by the BFRSS dataset, it became apparent that a significant number of them exhibited correlations, and many features were mainly composed of missing values. Consequently, with the hopes of simplifying the model, we experimented with various approaches to eliminate redundant and missing data. Our attempts included the removal of correlated (defined by Pearson product-moment correlation coefficients greater than 0.95) columns as well as those that were notably sparse and seemed to offer limited information (more than 90% of missing values). The outcomes, as displayed in Table 2, indicate that the most favorable results are attained when eliminating correlated columns. Hence, this was the approach we chose to adopt.

Method	F1-Score	Accuracy
Without sparse columns	0.1976	0.9152
Without correlated columns	**0.2417**	**0.9176**
Raw	0.2393	0.9175

TABLE II
RESULTS FOR UNNECESSARY DATA REMOVAL

D. *Feature scaling*

Given the diverse range of values across the dataset's features, there's a potential risk of the model giving undue importance to features with larger values. To mitigate this, we employed Z-score normalization, represented as:

$$X_{\text{normalized}} = \frac{X - \text{mean}(X)}{\text{std}(X)} \tag{1}$$

This normalization is particularly beneficial when using L2 regularization and gradient-descent optimization, as it ensures equal treatment of all features and facilitates faster convergence.

IV. Model Selection and Hyperparameter Tuning

To choose the optimal hyperparameter, we employed cross-validation with k = 7, as suggested in by Nti et al. [1]. Through an exhaustive grid search, we explored various values for our three key hyperparameters: the regularization coefficient (λ), the learning rate (γ), and the prediction threshold for the construction of decision boundaries.

Fine-tuning λ allowed us to find the right balance between overfitting and underfitting while adjusting γ controlled the convergence speed to prevent issues such as getting stuck in local minima.

We performed grid-search among the following possible set of values:

$$\lambda \in \{0, 10^{-4}, 10^{-5}\} \quad \text{and} \quad \gamma \in \{0.15, 0.2, 0.25, 0.3, 0.35\}$$

The optimal ones were $\lambda = 0$, $\gamma = 0.25$ which is a bit surprising as the model performs better with no regularization.

Additionally, modifying the prediction threshold was crucial for addressing the database imbalance as it allowed us to customize the model's precision-recall trade-off. When using the default one (0.5), we noticed a very high accuracy on the majority class but a very poor performance on the minority one. By tuning this parameter, we were able to make our model more sensitive to the minority class, yielding a significant improvement in our F1-score.

The following figure demonstrates how the optimal prediction threshold was chosen, taking into consideration that maximizing F1 was our first priority.

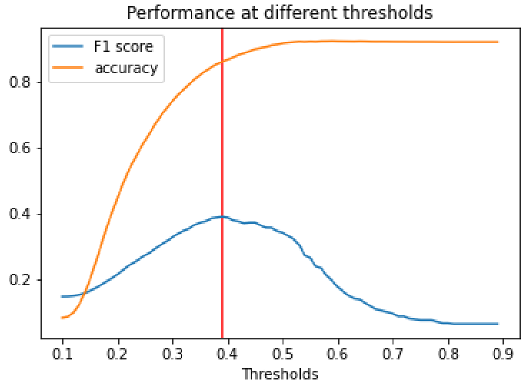

Fig. 1. Performance at different thresholds, red line represents maximum F1 score

Further, the following confusion matrices show the improvement provided by tuning the prediction threshold. As seen below, the different thresholds yields one order of magnitude increase in the numbers of true negative.

Threshold = 0.5

$$\begin{bmatrix} \text{9e+03 (TP)} & \text{7.9e+02 (FN)} \\ \text{15 (FP)} & \text{2.1e+02 (TN)} \end{bmatrix}$$

Threshold = 0.39

$$\begin{bmatrix} \text{7.9e+03 (TP)} & \text{6.8e+02 (FN)} \\ \text{1.2e+02 (FP)} & \text{1.3e+03 (TN)} \end{bmatrix}$$

V. Results

The following results were obtained after applying the techniques outlined in sections III and IV. Through grid search and cross-validation, we determined the optimal values for our hyperparameters to be: $\lambda = 0$, $\gamma = 0.25$, prediction threshold = 0.39.

Employing these values, our AIcrowd submission yielded an accuracy of 0.886 and an F1-score of 0.422.

VI. Summary

In this project, we explored various machine learning models to predict heart diseases based on an individual's clinical and lifestyle factors. Throughout our research, we observed that the default models exhibited a very poor performance in the binary classification of raw data. This highlights the critical role of preprocessing methods, including data cleaning and scaling, as well as the necessity for the fine-tuning of hyperparameters. To improve the success of our model, it could be a matter of interest to further explore the generation of synthetic data derived from the minority class to have a more balanced dataset.

References

[1] Isaac Nti, Owusu Nyarko-Boateng, and Justice Aning. Performance of machine learning algorithms with different k values in k-fold cross-validation. *International Journal of Information Technology and Computer Science*, 6:61–71, 12 2021.