The MICHD Classification Project

Belvisi Andrea, D'Andrea Michelangelo, Ferrazzi Matteo
Machine Learning (CS-433) Project - Fall 2023
EPFL, Switzerland

***Abstract*— In this project, mandatory part of the Machine Learning course at EPFL, we applied machine learning techniques to real life medical data gathered by BRFFS with the aim of predicting which individuals have coronary heart disease (MICHD). In the report, we present the final results of the analysis, as well as the methods we used to perform it and the way we managed the large original dataset**

I. Introduction

The World Health Organization reports that Cardiovascular Diseases (CVD), including conditions like strokes and heart attacks, have emerged as one of the primary global causes of death. As the average age of the population rise, heart and circulatory conditions have become increasingly common. Machine learning methods can help in the early identification and prevention of CVD development. And that is why the goal of the project is to assess an individual's risk of developing CVD by analyzing their personal lifestyle factors. After understanding and preprocessing the dataset, we employ five different ML algorithms to carry out the predictions. Then, we show the results of all the models in terms of accuracy and F1 score, and we highlight the best model and the relative hyperparameters. In conclusion, we present our final considerations.

II. DATA ANALYSIS AND PREPROCESSING

A. Exploratory data analysis

The dataset employed in this research project comprises a total of *n* observations, with 328,135 allocated for the training phase and 109,379 designated for the testing phase. Each observation within the dataset is characterized by a set of 321 distinct features.

B. Features removal

Upon initial examination, it was noticed that the training dataset contained a substantial proportion of missing values, accounting for 44.75% of the total entries. Consequently, we made the decision to remove any feature that exhibited a missing value rate exceeding 25%. Additionally, in the interest of refining our analysis, we identified certain features deemed uninformative (e.g., telephone number or identification code of the respondent), and others that displayed redundancy. These redundant and non-informative features were excluded to mitigate the risk of overfitting.

C. Adjusting Values

For the subset of features that successfully cleared our initial filtration criteria, we identified instances of non-sensical values, i.e. the presence of '5' indicating a dataset compilation error in a categorical variable assuming binary values (0 or 1). In light of these identified inconsistencies, we opted to treat these non-sensical values as missing data points.

D. Filling missing values

Even after pruning features primarily afflicted by missing values, our dataset continued to exhibit a substantial 7.8% proportion of null values. In response to this challenge, we embarked on a two-tiered strategy by categorizing the features into numerical and categorical subsets, each demanding distinct treatment:

- For numerical features, we resolved the issue of missing values by imputing them with the corresponding feature's median. This approach was chosen to enhance the dataset's resilience against potential outliers.
- In the case of categorical features, we opted to address missing values by imputing them with the mode, as both the mean and median are inadequately informative within the categorical context.

E. Management of categorical features

All variables categorized as categorical were systematically subjected to a process of decomposition, wherein each was transformed into 'k-1' dummy variables, where 'k' represents the total number of distinct categories within the respective variable. The omission of one category, the base category, was chosen to reduce concerns about collinearity with the constant feature.

F. Management of numerical features

We implemented a series of transformations for the numerical features, comprising the following steps:

- Winsorization at the 95% level was applied to these features to mitigate the influence of outliers. It's important to note that the outliers were not automatically dropped, preserving the entirety of the corresponding data rows.
- Following the completion of the initial preprocessing phase, the numerical data underwent a standardization process, a deliberate choice aimed at enhancing the overall efficiency of our algorithms.
- To enhance prediction accuracy, we expanded the feature space by employing a polynomial expansion function denoted as $\psi : [X] \rightarrow [X, X^2, ..., X^d]$ for each numerical

column. The determination of the polynomial degree, denoted as 'd,' was considered as one of the model's hyperparameters.
- Our initial plan was to incorporate interaction terms between features. However, upon carefully examining the dataset description, we observed multiple variables already encompassing interactions between different features. These interactions had already been introduced by those who constructed the dataset. To prevent overfitting and excessive computational cost, we resolved against introducing additional interaction terms.

III. METHODS

After preprocessing the dataset, we decided to build the prediction model by implementing five different methods:

- **Gradient Descent**
- **Stochastic Gradient Descent**
- **Ridge Regression**, using: $min_w \frac{1}{2N}||y - x^T w||_2^2 + \lambda||w||_2^2$
- **Logistic Regression** using gradient descent to maximize the *Log Likelihood* ($y \in 0, 1$)
- **Regularized Logistic Regression** using gradient descent to maximize the *Log Likelihood*, the regularization term is $\lambda||w||_2^2$ ($y \in 0, 1$)

It is worth noticing that we also implemented **Least Squares**, but we decided not to include it in the final analysis. In fact, due to the large dimension of the dataset and its sparsity, the matrix $X^T X$ arise to be incredibly ill conditioned, i.e. conditioning number of the order of 10^{30}. Hence, when the method tries to invert it, the resulting loss diverges, i.e. the method does not converge.

We tested the methods by tuning them a grid of four hyperparameters (note that not all methods include all four hyperparameters), that are:

- λ: the regularization parameter. Used to reduce overfitting
- γ: method's learning rate
- d: the highest degree of the polynomials included as regressors
- the *threshold*

We decided to tune the *threshold*, i.e. the value $\bar{t}$ such that if $y_{predicted} > \bar{t}$ then $y_{predicted}$ is set at 1, because the original dataset is very unbalanced (way more zeros than ones) and we thought that setting it to the traditional 0.5 would have produces biased results.

IV. RESULTS

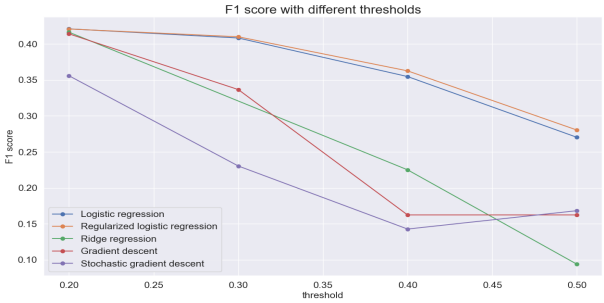
Table 1 shows the results we got for each method both in the train set (using a k-fold cross-validation to optimize the hyperparameters) and in the test set.

Table 1: Accuracy and F1

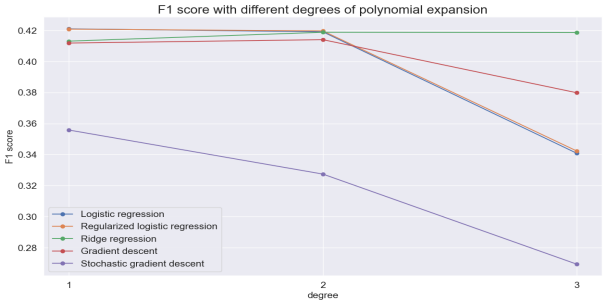
Method	accuracy	accuracy*	F1	F1*
Gradient Descent	0.853	0.855	0.414	0.420
Stochastic Gradient Descent	0.826	0.830	0.356	0.344
Ridge Regression	0.865	0.867	0.419	0.425
Logistic Regression	0.870	0.872	0.421	0.427
Regularized Logistic Regression	0.870	0.872	0.421	0.427

*those values are calculated on the test set

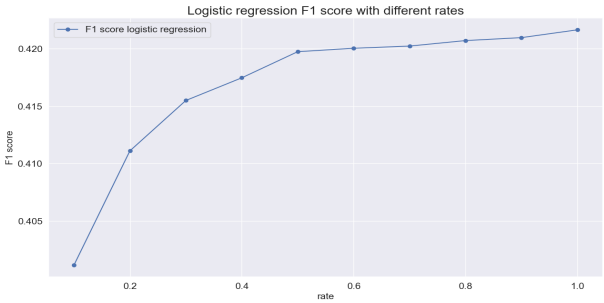
We show the best F1 scores of the different methods as a function of the threshold. As we can see, the score decreases as the threshold increases. It is an expected result since the dataset is heavily unbalanced toward zero. In addition, we tried thresholds values lower than 0.2, but values of both F1 score and accuracy were not satisfactory.



We also want to show how the F1 score relates with the d hyperparameter. We see that all models (excluding *Ridge Regression*) behave better with $d = 1$ or $d = 2$, rather than $d = 3$. This could mean that the real decision boundary is not highly nonlinear.



At the end, the best perfoming model is Logistic Regression. In the graph below, we highlight how the F1 score increases with the learning rate. After a steep increase, the model reaches a *plateaux* around $\gamma = 0.5$



V. CONCLUSIONS

We attained our highest F1 score, 0.421 on the k-fold cross-validated train set and of 0.427 on the AIcrowd test set, by using the **Logistic Regression** with hyperparameters $d = 1$, $\gamma = 0.9$ and $\bar{t} = 0.2$. We want to emphasize that data cleaning and feature engineering were crucial in achieving a significant enhancement in the model's F1 score.