Project 1: Prediction of Myocardial Infarct or Coronary Heart Disease

Sara Zatezalo
Section of
Electrical Engineering

Marija Zelic
Section of
Life Sciences Engineering

Elena Mrdja
Section of
Life Sciences Engineering

***Abstract*—The goal of this project was to implement different Machine Learning models such as Linear regression, Logistic regression, and Ridge regression, to predict whether a person is at a risk of developing heart disease or getting a heart attack. The predictions are made based on the labeled BRFSS dataset, containing the health information of over 300,000 U.S. residents. Since this dataset contains many missing values as well as features with non-informative entries, the success of the Machine Learning models heavily relied on data preprocessing. The final result of our project was obtained using Regularized Logistic Regression on the preprocessed data, which resulted in a validation accuracy of 0.847 and an F1 score of 0.413.**

I. Introduction

Coronary heart disease (CHD) is caused by a blocked blood supply in the heart and is a major cause of death every year in the US and worldwide. Early detection of a patient's predisposition to heart disease is key in order to prevent Myocardial Infract (MI) and CHD. Even though we know of several main risk factors that are correlated with a high predisposition to MICHD, such as high cholesterol levels and high blood pressure, a more comprehensive and complete method to predict the MICHD risk in patients is needed for successful prevention. This task is known to pose a great challenge in the healthcare sector, mostly due to a lack of comprehensive and useful data. The motivation of our project is to help create such a method, by analyzing and preprocessing one of the largest labeled datasets containing a broad range of health-related factors. We aimed to implement well-known Machine learning techniques to establish their potential when dealing with a complex classification task such as MICHD prediction.

II. Methods and Models

We have approached this problem by first looking into the given dataset and preprocessing its samples and features and then building, training, and testing Machine learning algorithms, in order to yield the best metrics possible. As we will see, data preprocessing plays a crucial role in achieving this goal.

A. Data preprocessing

1) Extracting meaningful features: Our training dataset contains more than 300 thousand samples with over 300 features, with a large amount of them being irrelevant to the specific task. Consequently, the first approach we took was to extract meaningful features. According to the paper [1] and BRFSS Codebook [2], we harvested 21 features with some of them being records of previous heart diseases, diabetes, smoking habits, mental and general health state, etc. It is important to note that we also modified entries of these features as per Codebook, by replacing missing or not responded answers. On the remaining features, we performed polynomial augmentation by expanding a feature up to a certain degree and by multiplying randomly selected columns. However, this did not yield a pleasing classification result.

2) Removing features with missing values: Our next approach was to work with the whole set of features. Following the Codebook led us to the conclusion that there are many missing values or the values assigned with labels 7, 9, 77, 99, etc. which correspond to the subjects who failed to give any meaningful information. We replaced those values with Nan. Furthermore, we dropped every feature that contains more than 25% of Nan entries.

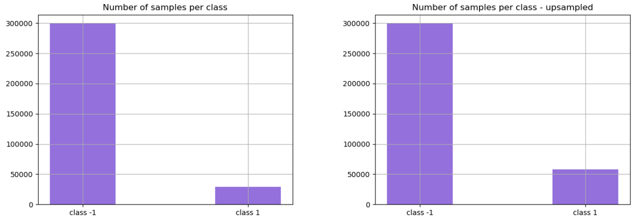
3) Median imputation: We approached we remaining missing values by replacing them with the median value of a corresponding feature, since according to the Codebook, most of our features are categorical, i.e. the range of possible values they take is discrete and rather small.

4) Adapting outliers: Considering that outliers can lead to poor model performance and overfitting, we ought to clip the values of our features to a certain range. The strategy that yielded the best performance was the clipping range of $[mean(x) \pm std(x)]$, where x represents the values of one feature.

5) Data standardization: We performed standardization of training and test sets by subtracting the mean values and dividing them by the standard deviation of the train set. Data standardization is important for the model performance as it requires data that is consistent in format and scale. This ensures faster convergence of gradient descent-based algorithms.

6) Dropping correlated columns: Given that we started off with a large set of unexplored features, we sought to extract the most information from our data with the least computational complexity. As highly correlated features do not bring any additional information to our learning process, we dropped columns with a correlation coefficient higher than 0.85.

7) Data balancing: Since this dataset consists of heavily imbalanced classes, we attempted to resolve this issue by oversampling the minority class, by repeating it once again in the dataset, making it 16% of it, as presented in Figure 1.



(a) Before oversampling. (b) After oversampling.

Fig. 1: Number of samples per class.

8) Adding bias column: In order to increase the range of possibilities for the regression function, we added a bias column of all ones.

B. Model selection and evaluation

We built and tested 2 main models: Ridge Regression and Reg Logistic Regression.
The only hyperparameter in Ridge Regression is λ (regularization factor), for which we performed tuning using grid search. As for Reg Logistic Regression, we observed the model performance over 4 different hyperparameters: λ (regularization factor), γ (learning rate) and w_0 and w_1, which are class weights. The concept of class weights is very common in the problems of unbalanced data and it works in such a manner that it penalizes the misclassification of the minority class by assigning its classification error a larger weight in the loss function. The loss function for Reg Logistic Regression with class weights incorporated is the following:

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N}(w_0 y_n \log\sigma(x_n^T w) + w_1(1-y_n)\log(1-\sigma(x_n^T w))).$$

To improve model performance and maximize the F1 score, hyperparameter tuning was implemented, using 5-fold cross-validation.

III. Experiments and discussion

In Table I, we report the best hyperparameters obtained by hyperparameter tuning and corresponding accuracy and F1 score. The λ parameter was searched over a range $[10^{-4}, 0.1]$ and γ parameter over $[10^{-5}, 0.1]$, both spaced evenly on a log scale. In the first setup, we observed 4 different scenarios:

- Ridge regression and Reg Logistic Regression with preprocessed data attained by the procedures explained in the sections II-A1, II-A3, II-A5, II-A6, II-A7 and II-A8. The results achieved by this approach are labeled with 1 in the table.
- Ridge regression and Reg Logistic Regression with preprocessed data attained by the procedures explained in the sections II-A2 to II-A8. The results achieved by this approach are labeled with 2 in the table.

Method	λ	$degree$	γ	Accuracy	F1 score
Ridge regression[1]	10^{-4}	3	-	0.91	0.1
Ridge regression[2]	1	-	-	0.89	0.37
Reg Logistic regression[1]	0.1	-	0.1	0.87	0.33
Reg Logistic regression[2]	0.1	-	0.1	**0.84**	**0.4**

TABLE I: Performance of the models

As we can clearly infer, the Reg Logistic Regression in the preprocessing scenario 2 performed the best. Therefore, we further explored its behavior, when approaching the imbalanced data in different manners.
As demonstrated in the section II-B, we attempted to incorporate class weights in order to tackle the problem of imbalanced data. The general suggestion for calculating the class weight parameters is:

$$w_i = \frac{N}{2N_i},$$

where w_i represents weight for the i-th class, N total number of samples and N_i number of samples of i-th class. Following the formula, class weights should have values of $w_{-1} = 0.54$ and $w_1 = 5.66$. However, these values did not produce a good classifier, so we turned to tuning these parameters by grid search. Regarding the parameters λ and γ, we stuck to the optimal values obtained in the Table I. For the class weights, we used several different combinations, which are depicted in Table II with the corresponding F1 score and accuracy (upper value and lower value, respectively).

w_1 w_{-1}	1	1.25	1.5
0.75	0.4 0.86	0.38 0.89	0.34 0.9
0.9	0.4 0.85	0.4 0.88	0.37 0.9
1	**0.41** **0.84**	0.4 0.87	0.38 0.89

TABLE II: Comparison of metrics for different class weights

As we can see from Table II, the best results are obtained with the ”basic” Reg Logistic Regression, meaning that both of the class weights are equal to 1. This might be due to the fact that the training dataset was previously upsampled by repeating the minority class one more time. With this being said, we report the best overall accuracy to be 0.847 and the F1 score of 0.413 (results of submission in AIcrowd).

IV. Conclusion

The results of our work demonstrate the importance of understanding of the dataset and thorough data preprocessing before applying any Machine learning algorithm. After data analysis, handling missing values, and dealing with outliers and a heavily imbalanced dataset we achieved a decent F1 score of 0.413 with Reg Logistic Regression.

References

[1] B. Akkaya, E. Sener, and C. Gursu, “A comparative study of heart disease prediction using machine learning techniques,” in *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2022, pp. 1–8.

[2] C. for Disease Control and Prevention. (2015) Behavioral risk factor surveillance system. [Online]. Available: https://www.cdc.gov/brfss/annual_data/annual_2015.html