Myocardial Infarction and Coronary Heart Disease Prediction using Machine Learning Techniques

Elisa BILLARD
Life Sciences Engineering Section

Souleyman BOUDOUH
Computer Science Section

François DUMONCEL
Data-Science Section

***Abstract*—Cardiovascular diseases (CVD) compromise the functionality of the heart and blood vessels, frequently resulting in mortality or physical debilitation. Thus, the prompt and automated identification of CVD can assist medical professionals and enhance the diagnostic quality. In this investigation, we demonstrate the capability of a linear classifier to prognosticate CVD, achieving a F_1-score of 0.436 and an accuracy rate of 87%. Further optimizations are possible through the application of more advanced methods [1].**

I. Introduction

According to World Health Organization, Cardiovascular Diseases (CVD) are becoming one of the leading causes of death globally. In our project we apply machine learning techniques to determine the risk of a person developing CVDs based on features of their personal lifestyle. In an attempt to produce the best possible classifier, we optimised 6 different models on the binary classification task of predicting whether or not the person will develop a CVD.

II. Pre-processing

A. Feature engineering

The original data set contains 321 features, representing all the questions asked by the Behavioral Risk Factor Surveillance System (BRFSS) to the participants. The majority of the questions are not irrelevant to the topic we are investigating, therefore we manually chose 58 of them [2].

Then, to decide which medical features to choose, we looked at known risk factors in coronary heart diseases (CHD) and myocardial infraction (MI). The most important ones are hypertension, high cholesterol levels, diabetes, smoking, obesity, physical inactivity, unhealthy diet and stress. All those factors appear in the questions asked to the participant and we therefore chose the corresponding features. We paid attention to choose features with the least possible missing values and chose meaningful computed variables that summaries several previous features.

B. Data Cleaning

After choosing the most appropriate features, we cleaned the data by changing the values of some questions. For some binary results we needed to change the twos into zeros, for other questions we replaced the sevens and nines that accounted for 'Don't know/Not sure' or 'Refused' with standard NaN values.

C. Handling Missing Values

We adopted distinct approaches for managing missing values in *categorical* and *continuous* variables. In the case of a categorical variable, we decided to substitute the missing values with a value representing the missing category : -1. For continuous variables, we replace the missing values with the mean of the feature.

D. Standardisation

To ensure that the different features equally contribute to the training of our model, we standardize the continuous variables. However, the data-set presents a many categorical features, many of which are nominal and thus can not be directly mapped to a standardized scale. Instead, we one-hot encode every label of each nominal feature, effectively reducing the sets of possible values to $\{0, 1\}$ in order to fit them in the [0, 1] range enforced for all other features.

E. Balancing Data Set

The initial data-set is characterized by an imbalance, with over 90% of the individuals not expected to develop CVDs. This disproportion often leads to adverse effects during the model training phase. Consequently, we have chosen to experiment with data-set balancing through the down-sampling of the majority class, and subsequently training models on these four diversified data-sets:

- Dataset $\mathcal{D}_1$: Original unbalanced dataset (90%-10%)
- Dataset $\mathcal{D}_2$: Perfectly balanced dataset (50%-50%)
- Dataset $\mathcal{D}_3$: $2/3$ semi-balanced dataset (67%-33%)
- Dataset $\mathcal{D}_\star$: Maximizer of F_1-score (63%-37%)

Dataset $\mathcal{D}_\star$ was chosen specifically to maximise the F_1-score. The down-sampling proportion factor leading to this distribution of classes was determined heuristically.

III. Machine Learning Models [3]

The models proposed for our predictions can be split into linear and logistic regressions. They are fitted using stochastic gradient descent algorithms with their associated learning rate γ, and share two extra hyper-parameters: the regularization factor λ and the classification threshold $\mathcal{T}$ to map the continuous predictions to the discrete set of categories through an inequality test.

Table I: RESULTS SUMMARY FOR SIX ML MODELS. In the Model Parameters section, γ is the learning-rate, N the number of iterations, λ the regularizer and $\mathcal{T}$ the classification threshold.

Methods	Model hyper-parameters				Dataset $\mathcal{D}_1$		Dataset $\mathcal{D}_2$		Dataset $\mathcal{D}_3$		Dataset $\mathcal{D}_\star$	
	γ	N	λ	$\mathcal{T}$	F_1	Accuracy	F_1	Accuracy	F_1	Accuracy	F_1	Accuracy
Linear regression GD	0.01	10^3	-	0	0.0322	91.36%	0.3496	72.74%	0.4061	88.84%	0.4003	81.80%
Linear regression SGD	0.001	10^4	-	0	**0.1188**	91.50%	0.3426	71.82%	0.4055	88.09%	0.3438	72.17%
Least squares	-	-	-	0	0.0991	91.47%	0.3582	74.00%	**0.4133**	88.84%	0.4069	82.42%
Ridge regression	-	-	10^{-7}	0.151	0.0403	91.38%	**0.3936**	80.06%	0.3724	90.48%	**0.4224**	86.53%
Logistic regression	0.005	10^4	-	0.5	0.0020	91.27%	0.3721	78.87%	0.2886	90.50%	0.3903	86.29%
ℓ_2 - Logistic regression	0.005	10^4	10^{-4}	0.5	0.0022	91.27%	0.3726	78.95%	0.2848	90.48%	0.3901	86.33%

A. *Linear Regression Models*

The linear regression models have in common the goal of minimizing the mean square error, i.e:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2N}\sum_{n=1}^{N}(y_n - \mathbf{x}_n^\top \mathbf{w})^2$$

When using the Least Squares method, we ran into the issue of the Gram matrix $\mathbf{X}^\top\mathbf{X}$ being ill-conditioned ($\mathbf{X}$ is often rank deficient), leading to numerical issues when solving the linear system. We therefore used the pseudo inverse of Moore-Penrose when necessary.

B. *Logistic Regression Models*

For the logistic regression model we minimize the negative log-likelihood loss function, i.e:

$$\mathcal{L}(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N} y_n\mathbf{x}_n^\top\mathbf{w} + \log(1 + e^{\mathbf{x}_n^\top\mathbf{w}})$$

We decided to map the labels y_n from $\{-1, 1\}$ to $\{0, 1\}$ to simplify the calculations.

C. *Weights initialization optimisation*

When training the models, we can enhance the quality of their fit with a judicious initialization of the weights [4]. We observe improvements in the convergence of all three methods by first sampling the weights from a normal distribution $\mathcal{N}(0, \frac{2}{d})$ with d the feature dimension, prior to updating them.

D. *Hyper-parameters optimisation*

To optimize the hyper-parameters, we cross validated them ($K = 4$) and chose the parameters that were maximizing the F_1-score rather than minimizing the loss.

IV. RESULTS

Table I is a summary of our results for the six ML models we tried. It contains the F_1-score and the accuracy of each model on each dataset $\mathcal{D}_i,\ i \in \{1, 2, 3, \star\}$. As one can see the ridge regression model with $\lambda = 10^{-7}$ and $\mathcal{T} = 0.151$ has a better accuracy up to 86.53% and a better F_1-score up to 0.4224. One can also remark that Linear regression SGD underperforms due to the use of a batch size of 1.

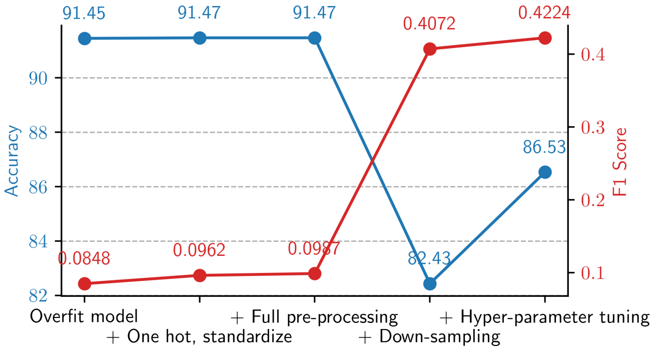

Figure 1: Evaluation of model

V. DISCUSSION

The original BRFSS dataset is significantly unbalanced, making the dataset sub-optimal for machine learning models. Indeed, even a trivial classifier that consistently predicts -1 would achieve an accuracy rate of approximately 92%. To address this issue, we have opted to down-sample the original dataset, aiming to decrease the prevalence of the majority class. Additionally, we have chosen to focus on optimizing the F_1-score, as accuracy proves to be a less relevant metric in this context. As the table I suggests, the model employing ridge regression emerged as our most effective model, surpassing the performance of logistic regression models. Indeed, linear regression with least squares methods, including ridge regression, leverages a closed-form solution to determine the optimal weights, circumventing the need for iterative optimization methods such as Gradient Descent. In Figure 1, one can observe that preprocessing and down-sampling increase F_1-score but good choice of hyper-parameter maintains a high accuracy.

VI. CONCLUSION

With all these techniques, the optimal model we developed, based on Ridge Regression, attained a test F_1-score of 0.436 and demonstrated an accuracy of 87% when evaluated on AiCrowd.

References

[1] G. P. K. D. Madhumita Pal, Smita Parija and R. K. Mohapatra, “Risk prediction of cardiovascular disease using machine learning classifiers,” *Open Medicine*, 2022.

[2] C. for Disease Control and Prevention, “Codebook report landline and cell-phone data,” 2015.

[3] M. J. . N. Flammarion. (2023) CS-433: EPFL Machine Learning Course. [Online]. Available: https://github.com/epfml/ML_course

[4] S. R. J. S. Kaiming He, Xiangyu Zhang, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *ICCV*, 2015.