MICHD Detection using Machine Learning

Kamel Charaf — Ivan Pavlov — Michele Smaldone
Department of Computer Science, EPFL Lausanne, Switzerland

I. Introduction

This report presents an end-to-end pipeline focused on effectively predicting the likelihood that MICHD (Myocardial Infarction or Coronary Heart Disease) occurs using a dataset from the Behavioral Risk Factor Surveillance System (BRFSS). The primary goal of the project is to find and evaluate different machine learning algorithms to get meaningful insights about the risks and factors having MICHD.

II. The Dataset

The dataset contains administrative and health-related records about individuals who participated in the surveys. Approximately 44.79% of entries contain missing values (NaN-s) with an average of 143 missing values per sample. Additionally, the dataset shows a huge imbalance towards the number of people not having MICHD compared to the number of people having MICHD with a ratio of $92:8$ (referred to as *class-imbalance*).

A. Cleaning the data

To prepare the dataset for modelling and to enhance the quality and reliability of the data, a comprehensive set of data-cleaning strategies were applied in the following order:

- **Filtering Rows**: Rows with excessive missing data (higher than 0.95 quantiles) are removed. Rows with lower missing data than the threshold are retained.
- **Column Dropping**: In order to reduce the noise in the dataset, columns with more than 0.85% of missing values were removed. The threshold was determined through a linear search.
- **Type Separation**: The dataset includes *categorical* (e.g. marital status), *boolean* (e.g., smoking status, exercise engagement), and *numeric* data (e.g., age, weight). Since *numeric* and *non-numeric* data were treated differently, a separation was applied with the aim of determining which feature belongs to which group.
  - Features that contain maximum of two distinct values were labelled as *boolean*.
  - Features with more than two, but less than then distinct values were labelled as *categorical*.
  - Features with more than 10 features were labelled as *numeric*.

  The NaN values were replaced by the mean for *numeric* data, and by the mode for *non-numeric* data (referred to as *default NaN filling strategy*).
- **Filling the NaN-s**: If a feature contains more than 80% of non-NaN data the default NaN filling strategy is applied.
- **Feature Selection**: Out of several types of variables non-informative (administrative) features were removed, while preserving calculated features for enhanced model performance.
- **New class method**: For *categorical* and *boolean* data, a new category for the NaN values was introduced. The new class is the largest value of the category contained in the feature plus one.

Finally, any remaining NaN values were treated using the default filling strategy, to ensure minimal data loss while maximizing feature completeness.

III. Models and Methods

As for the first part, three simple models were evaluated for predicting MICHD:

- **Ridge Regression** ($\lambda = 0.0001$)
- **Logistic Regression** ($SGD,\ \gamma = 0.1, iteration = 5000, batch\_size = 10000$)
- **Random Classifier**

Model	Accuracy	F1 Score	Recall
Random Classifier	0.688 ± 0.0007	0.133 ± 0.0011	0.268 ± 0.0023
Logistic Regression	0.872 ± 0.0009	**0.426 ± 0.0018**	**0.529 ± 0.0045**
Ridge Regression	**0.878 ± 0.0005**	0.424 ± 0.0018	0.498 ± 0.0029

Table I
Performance comparison between the models

A Random Classifier functions as a baseline by predicting ones with a probability equal to the frequency of one-labels in the training set. As a result of class imbalance, it achieves a high accuracy rate but fails to provide meaningful insight to make reliable predictions. Thus, the **F1-score** was the primary metric to compare models to each other.

As Table I indicates, the Logistic Regression achieved the highest F1-score among the other simple methods. We report mean performance metrics along with 95% confidence intervals, computed using the z-score. These results are based on 5 runs of 5-fold cross-validation (referred to as 5-5-Cross-validation).

A. Downsampling for Class Imbalance

Given the strong class imbalance, we implemented a downsampling strategy [1].

The downsampling proportion constant p controls the ratio of majority to minority class samples during training (i.e. when $p = 2$, the majority class has $2:1$ rate to minority class and so on).

To evaluate the impact of downsampling, a series of proportions were tested using cross-validation and the F1, Recall, and Precision scores were recorded. Figure 1 shows the observed trends, with the F1-score approximating a concave curve (similar to a quadratic function) and reaching its peak at a moderate downsampling rate of approximately $p = 2.7$, after which it declines.

The optimal tuning, as indicated by the peak F1-score, achieves a balanced trade-off between Recall and Precision.

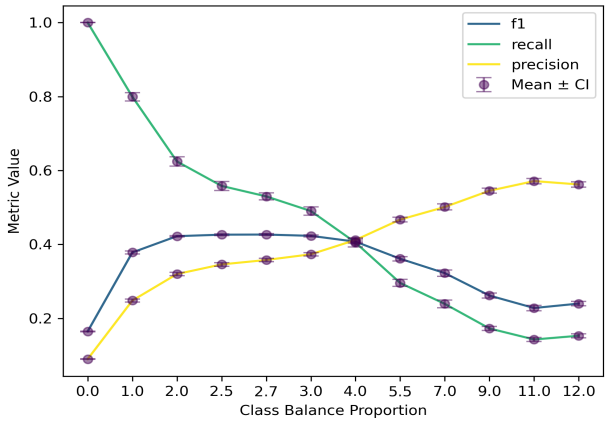

Figure 1. Tuning Curve for balance proportion p. 95% z-score confidence intervals reported (5-5-Cross-validation performed)

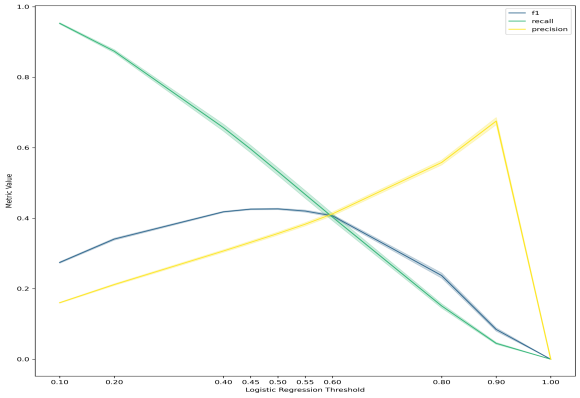

Figure 2. Tuning Curve for Ridge threshold. 95% t-confidence intervals are provided (5-5-Cross-validation performed)

B. Threshold Tuning

In addition to downsampling, threshold tuning is employed to optimize the decision boundary for Logistic Regression. By adjusting the threshold, the model's confidence for predicting positive (minority) labels can be controlled, which directly affects Recall, Precision and F1-score.

In order to evaluate the effect of various thresholds, a range of threshold values were defined: $[0.1, 0.2, 0.4, 0.45, 0.5, 0.55, 0.6, 0.8, 0.9, 1.0]$. For each threshold 5 iterations of 5-fold cross-validation with downsampling applied at a fixed proportion rate $p = 2.7$.

Figure 2 shows the variation in F1-score, Recall, and Precision rates as a function of the threshold. The Recall rate shows a clear downward trend. The F1-score resembles a concave function with a peak around 0.5. However, Precision grows until reaching a critical point near 0.9, after which the number of true positive predictions likely decline sharply.

Given the importance of Recall in the context of medical problems (as false negatives are significantly more dangerous than false positives), adjusting the model's threshold to around 0.4 is recommended for production use. This adjustment would increase Recall while maintaining a similar F1-score.

IV. Results

The results of our analysis demonstrates that the **Random Classifier** achieved an accuracy of approximately `0.688%` with a very low F1-score of `0.133%`, indicating its limitations as a baseline. The application of **Ridge Regression** produced significantly higher accuracy and F1-score compared to the baseline reaching an accuracy of `0.878%` and F1-score of `0.424%`. The **Logistic Regression** model consistently outperformed Ridge Regression across important metrics, achieving an accuracy of `0.872%` and an F1-score of `0.426%`.

Tuning the **downsampling ratio** showed the importance of downsampling across our analysis. Following the original ratio between the classes (which is around $p = 11$), the F1-score was only `0.227%`, while choosing the best $p = 2.7$ the F1-score increased to `0.426%`.

After tuning the **decision boundary threshold** for Logistic Regression we can conclude, that a default value of 0.5 seems to achieve the highest scores.

We achieved **0.437** F1-score with our submission (ID: 275438) on AICrowd.

V. Discussion

There are multiple ways how our models, methods and evaluation can be improved:

The primary goal of the analysis was to reduce the amount of false negatives, since in medical applications predicting negative while the truth is positive can be crucial. This means that a higher Recall rate is prioritized at the cost of the Precision. The F1-score was used since our method was tested on AICrowd based on F1-score but using F_β measurement with $\beta = 2$ could represent our way much better.

The preprocessing part was conducted half-manually with the administrative features selected manually but the labelling of features based on their type was done automatically (based on a threshold). Further understanding of the dataset or talking to a domain expert could improve our feature selection [2].

There are features that contain mixed measurements (like the *WEIGHT2 HEIGHT3*) that could be resolved for more consistency in the dataset.

More powerful models could be also used [3] that are more robust for class imbalance, like Random Forest or Neural Networks. Since we had to implement everything using Python Standard Library (PSL) and NumPy libraries we decided to optimize the simple machine learning algorithms rather than implementing more complex ones.

VI. Summary

In the end, to address these challenges, we implemented key preprocessing steps, including the removal of non-predictive administrative data and the handling of missing values through various imputation strategies. Class imbalance was addressed through the downsampling approach, testing a range of ratios to optimize the balance between Recall and F1-scores.

Our results highlighted that **Logistic Regression**, combined with downsampling and threshold tuning outperformed both the Ridge Regression and the baseline model, achieving a higher F1-score.

REFERENCES

[1] K.-V. Tompra, G. Papageorgiou, and C. Tjortjis, "Strategic machine learning optimization for cardiovascular disease prediction and high-risk patient identification," *Algorithms*, vol. 17, no. 5, 2024. [Online]. Available: https://www.mdpi.com/1999-4893/17/5/178

[2] M. A. Naser, A. A. Majeed, M. Alsabah, T. R. Al-Shaikhli, and K. M. Kaky, "A review of machine learning's role in cardiovascular disease prediction: Recent advances and future challenges," *Algorithms*, vol. 17, no. 2, 2024. [Online]. Available: https://www.mdpi.com/1999-4893/17/2/78

[3] S. B. Akter, R. Hasan, S. Akter, M. M. Hasan, and T. Sarkar, "Improving heart disease probability prediction sensitivity with a grow network model," *medRxiv*, 2024. [Online]. Available: https://www.medrxiv.org/content/early/2024/03/05/2024.02.28.24303495