Predicting Heart Attacks

Mehdi Bouchoucha, Mohamed Hedi Hidri, Ali Ridha Mrad
School of Computer and Communication Sciences, EPFL, Switzerland

***Abstract*—In this project, we make use of machine learning techniques to predict cardiovascular disease (CVD) risks by analyzing a dataset of health and lifestyle factors. With an aging global population and the associated increase in cardiovascular conditions, proactive risk prediction is essential to mitigate the impact of these diseases. Using the Behavioral Risk Factor Surveillance System (BRFSS) dataset, our models aim to estimate an individual's likelihood of developing myocardial infarction and coronary heart disease (MICHD).**

I. Introduction

Cardiovascular diseases (CVDs), such as myocardial infarction and coronary heart disease (MICHD), are among the leading causes of mortality globally. Early prediction and prevention are crucial in reducing their impact on healthcare systems and improving patient outcomes. Exploiting data from the Behavioral Risk Factor Surveillance System (BRFSS), this project applies machine learning techniques to estimate the risk of developing MICHD based on personal health and lifestyle factors. This report details our approach to data preprocessing, feature engineering, and model evaluation for effective prediction of CVD risk.

II. Data Exploration

A rigorous exploration of the BRFSS dataset was essential to guide our modeling choices. The dataset includes a training set of 328,135 data points and a testing set of 109,379 data points, each containing 321 features that include a mix of categorical and numerical variables. The features are consist of health indicators and personal information. Our initial analysis focused on handling missing values, constant features, and identifying correlations with CVD outcomes.These steps provided key insights that informed both feature selection and model development.

III. Data Preprocessing

Data preprocessing is crucial for preparing the dataset for modeling. The steps taken are outlined below.

A. Removing Constant Features

Features with constant values provide no information for prediction. We identified and removed six such features from both the training and testing datasets, reducing the feature count to 316.

B. Managing Missing Values

Columns with more than 60% missing values were removed to improve data quality. This step further reduced the feature count to 241.

C. Processing Specific Features

For features like `PHYSHLTH`, `MENTHLTH`, and `POORHLTH`, specific values (e.g., 88 indicating 'None') were replaced with 0, and missing indicators (77 and 99) were replaced with NaN.

D. Encoding Categorical Features

The `_STATE` feature, representing the state of residence, is a categorical variable. We converted it into dummy variables using one-hot encoding, resulting in additional features corresponding to each unique state. We identified categorical features as those with integer values between 1 and 9 and with up to 8 unique values. Values of 7 and 9, representing 'Don't know' or 'Refused', were treated as missing and not encoded. We applied one-hot encoding to these features, expanding the feature set to better represent categorical data. This step increases the feature count to 402.

E. Imputing Missing Values

Missing values were imputed using the median of each feature, a robust measure less sensitive to outliers. This ensured that all NaN values were replaced, facilitating the application of algorithms that require complete data.

F. Feature Selection Based on Correlation

We computed the Pearson correlation coefficient between each feature and the target variable. Features with an absolute correlation less than 0.01 were considered weak predictors and were removed. This step reduced the feature count to 301, retaining those most relevant for prediction and removing noise.

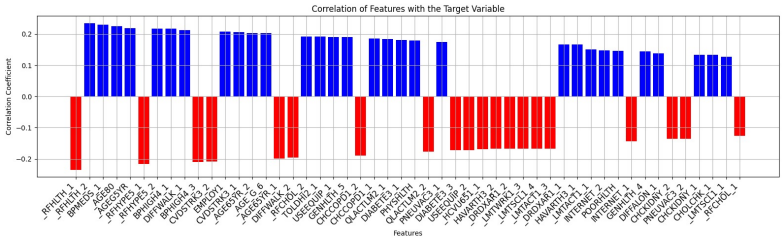

Fig. 1: Correlation of top 50 features with the target variable

G. Feature Scaling

To standardize the feature scales, we applied min-max normalization to the features using the training data statistics. This step ensures that all features contribute equally to the model training.

IV. Models

A. All Implementations

Some implementation details were particularly notable. For instance, the Stochastic Gradient Descent approach turned out to be more computationally heavy than the Gradient Descent method. For the Ridge Regression, the L2 regularisation handles the model complexity by focusing more on the important features which contribute more to the overall error than the less important features. Regularized Logistic Regression added a penalty term to logistic loss, balancing prediction confidence with generalization. These nuances guided our model selection by balancing computational efficiency and generalization capability with accuracy and F1 score performance.

TABLE I: Comparison of Accuracy and F1 Score across different methods (all used 1000 iterations)

Method	γ	λ	**Accuracy**	**F1 Score**
Gradient Descent	0.01	-	0.869	0.412
Stochastic Gradient Descent	0.0015	-	0.856	0.375
Least Squares	-	-	0.865	0.413
Ridge Regression	-	0.015	0.865	0.411
Logistic Regression	0.9	-	0.859	0.422
Reg. Logistic Regression	0.9	0.0001	0.861	**0.434**

B. Finding the best parameters for our models

1) Hyperparameter Tuning: We conducted hyperparameter tuning by exploring various combinations of the learning rate (γ) and the regularization strength (λ). We explored a range of values for both γ and λ . For each combination we trained the model on the training set and evaluated it using the F1 score on the test set. This metric was specifically chosen due to its balance between precision and recall, critical in applications where both false positives and false negatives carry significant weight. By fine-tuning these hyperparameters, we aimed to identify a model that preserves accuracy while ensuring strong generalizability. Our grid search led us to an optimal γ, λ pair, producing the highest F1 score.

2) Threshold Selection: To optimize our binary classification model's performance, we implemented a threshold selection method focused on maximizing the F1 score. After obtaining predicted scores from the model, we evaluated a range of thresholds derived from these scores. For each threshold, we generated binary predictions and calculated the corresponding F1 score and accuracy. We identified the threshold that achieved the highest F1 score, ensuring our model balances precision and recall effectively.

V. Results

For our binary classification task, we opted for *regularized logistic regression* as our primary model, given its effectiveness in managing binary outcomes with an optimal balance of interpretability and computational efficiency. This approach aligns with our objectives, and ensure robust performance. However, we integrated a regularization term controlled by λ, in order to have a better modle stability by handling multicollinearity. Regularization penalizes excessively large coefficients, nudging the model towards simpler and more robust patterns that generalize beyond our training data. This model outperformed the others in terms of F1 score.

TABLE II: Regularized Logistic Regression results with hyperparameter tuning

γ	λ	**F1 Score**	**Accuracy**
0.1	1e-05	0.413	0.857
0.5	1e-05	0.421	0.864
0.7	1e-05	0.419	0.869
0.9	1e-05	0.418	0.858
0.1	0.0001	0.413	0.832
0.5	0.0001	0.420	0.832
0.7	0.0001	0.419	0.832
0.9	0.0001	**0.434**	**0.861**
0.1	0.001	0.412	0.832
0.5	0.001	0.417	0.832
0.7	0.001	0.417	0.832
0.9	0.001	0.418	0.832

To further validate our model's performance, we included the ROC curve, highlighting the trade-off between true positive and false positive rates. With an AUC of 0.8586, the curve confirms the model's strong ability to distinguish between classes, reinforcing the robustness of our model choice.

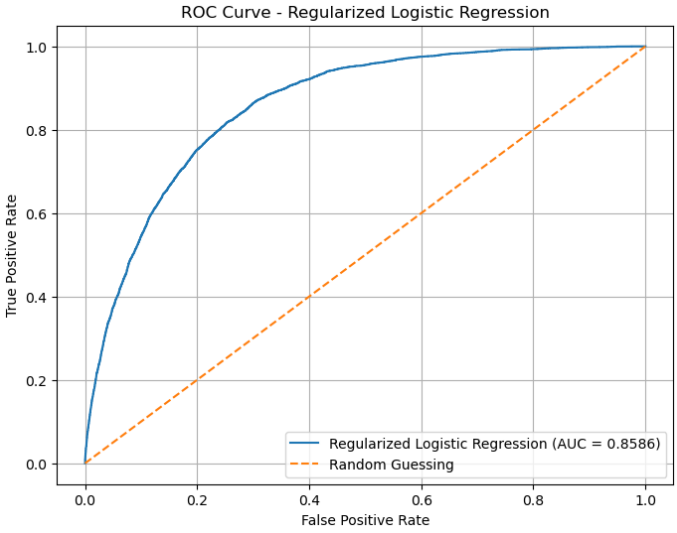

Fig. 2: Regularized Logistic Regression ROC Curve

VI. Conclusion & Future Work

Our regularized logistic regression model achieved solid results in predicting cardiovascular disease risk, with an F1 score of 0.434, an accuracy of 0.861 and an AUC of 0.8586, as confirmed by the ROC curve. Future work could explore advanced models like XGBoost to capture complex patterns potentially missed by logistic regression. Additionally, enriching the dataset with more diverse health metrics or tracking patients over time could enhance model robustness and accuracy, ultimately supporting more effective preventive healthcare applications.