CS-433 : Machine Learning Project I

Riccardo Carpineto, Elias Nicolas Naha, Kaan Uçar
Department of Computer Science, EPFL, Switzerland

***Abstract*—In this report, we describe our approach to predict Myocardial Infarction and Coronary Heart Disease based on the personal life features. We showcase various machine learning models and compare their accuracy to select the most robust one.**

I. Introduction

Cardiovascular Diseases are a major issue, representing the first noncommunicable cause of death in Europe [1]. With the aging of the population, the number of heart attack and other related diseases is on the rise, making prevention and early detection a primary focus. Extensive data-sets availability has significantly improved Machine Learning models, making them highly efficient and accurate allowing reliable predictions.

In this project, we harness the power of Machine learning to predict likelihood of developing Myocardial Infarction and Coronary Heart Disease (MICHD) in individuals based on features of their personal lifestyles. In order to achieve this, we used data taken from the Behavioral Risk Factor Surveillance System (BRFSS), a major source of health-related data. Our training dataset was composed of 328135 samples and 321 features. Each sample has a binary label corresponding to if the person has had or not a MICHD. To accurately predict the MICHD, first did the key steps of preprocessing on our data from BRFSS such as cleaning, reduction, transformation and partitioning [2]. We developed a classification model, optimized by tuning hyper-parameters and evaluated its performance using diverse metrics such as F1 score, accuracy and recall.

II. Models and Methods

A. Preprocessing and Data cleaning

1) *Cleaning*: The first step of preprocessing is the cleaning of the data. To achieve this we look at the data values and make adjustments to ensure of their meaningfulness. For instance, labels "Don't know" or "Refused" are replaced by NaN values and are further processed as such (see Item 2). Other features need to be converted to provide valuable information such as the feature "WEIGHT2" which contains pounds and kilograms, or "BLDSUGAR" which provides mixed data in days, months, years.
2) *Handling of missing values*: Most of the features contain a significant number of missing value making the option of simple removal impossible due to loss of data. We thus replace them by the median value of each feature. We did try to use one hot encoding for the categorical columns and ended up with more than 1000 columns, and then we used PCA on the new table created. However the results were actually worse than without this technique, so we rejected this idea.
3) *Transformation* : After feature selection we can proceed to standardize our data to get the most balanced weights. We standardize all the features using the Z-score standardization $Z = \frac{X-\mu}{\sigma}$ with μ being the mean and σ the standard deviation. This is also a necessary step for the PCA.
4) *Reduction and Feature selection* : The next step is to reduce the data removing features not containing any information. To select columns we apply a PCA algorithm. Principal Component Analysis works effectively by maximizing data variance, reducing dimensions, and emphasizing orthogonal and uncorrelated features, making it a valuable tool for simplifying data, reducing noise, aiding visualization, and facilitating feature selection. After all these steps we have a dataset composed of only 117 features, a lot less compared to the 321 initial number.
5) *Separation of Dataset* : The dataset was split into training and testing to avoid overfitting and evaluate model performance on unseen data. For this we split the dataset into 80% Training and 20% Testing.

B. Polynomial Expansion

In the optic of improving our performances we implement the polynomial expansion by adding the bias term $[1]$ and expansion $\phi : [X] \rightarrow [X, X^2, \ldots, X^k]$. The latter aims to better fit our data which we believe does not follow a linear relationship. (see Table III-A)

C. Classification Algorithms

In this project we implement the six following models with Gradient Descent (GD) or Normal equations (NE):

1) Linear regression using GD
2) Linear regression stochastic GD
3) Least squares regression using NE
4) Ridge regression using NE
5) Logistic regression using GD
6) Regularized Logistic regression using GD

In our Algorithms we implement two loss functions : Mean Squared Error (MSE) and Logistic Loss. The MSE is minimized by the Linear, Least squares and Ridge regression models whereas the Logistic loss is minimized by Logistic regression models.

D. Hyperparameter tuning

In this project we focus on tuning Logistic regressions and ridge regression more than the others, the binary classification being the underlying reason. We first tried to tune logistic regression with the learning rate γ and the degree of the polynomial using a grid search. Simultaneously we performed the same process on the Ridge regression which led to better results, therefore we went into more depth towards this algorithm. To yield the optimized hyperparameters we performed a 4-fold cross-validation grid-search over the regularisation parameters λ and the polynomial degree. The degrees range from 1 to 4 while the λ from 10^{-4} to 1. The parameters which lead to the best MSE found are :

1) $\lambda = 0.00092 = e^{-3.036212 * log(10)}$ (see Fig. 1)
2) degree = 2.

III. Results

A. Model Comparison

To compare our model's performance we focused on the F1 score and accuracy primarily. F1 score is particularly useful to measure performance of classification tasks where a significant class imbalance is present. The following results are the one yielded on our test set and only the two logistic (normal and regularized) and ridge regressions are optimized.

Table I
F1 & Accuracy score from each model

Classification Model	Accuracy	F1 score
Linear regression using GD	90.91	24.43
Linear regression stochastic GD	90.51	16.17
Least squares regression using NE	91.20	15.03
Ridge regression using NE	85.74	41.90
Logistic regression using GD	84.63	37.53
Regularized Logistic regression using GD	79.98	36.90

From these results we deduct that linearity performs poorly on this classification dataset, thus the use of logistic regression and polynomial feature expansion for better scores. This table also showcases the importance of F1 score, especially when dealing with imbalanced datasets where one class dominates the other. It ensures that both false positives and false negatives are considered, providing a more complete evaluation of the model's performance. Accuracy only takes into account true negatives (TN) and true positive (TP) together which in our case leads to TN overshadowing TP.

B. Ablation study

Use of PCA can also give rise to an increased F1 score. Performing the same run without the use of PCA gives a F1 score of 22%, compared to 41.9% in our best run. Reducing feature complexity frees more computational power for hyperparameter tuning and polynomial expansion.

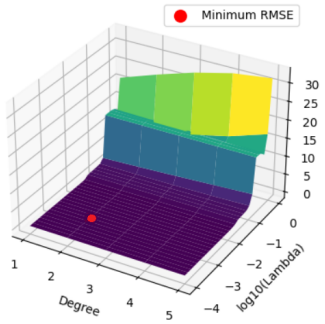

Figure 1. Grid-search over lambdas and degrees

Parameter tuning is also essential to perform well in this classification task. Plotting our grid search showcases the variation of RMSE with respect to lambda and degree combinations.

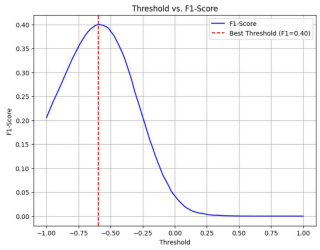

Figure 2. Threshold tuning with F1 score

Tuning the threshold we find that the F1 score significantly changes based on the threshold parameter for the classification. Looking at the figure 2, we find that the best threshold is at -0.576 .

IV. Discussion

Contrary to our anticipation, in our heart attack prediction task, Ridge regression outperformed Logistic regression. Ridge's strength lies in its built in penalty which enables it to handle data complexity and prevent overfitting. Conversely, Logistic regression, with its assumption that the log-odds of the outcome variable linearly relate to the predictors, can be sensitive to certain data patterns. This suggests Ridge regression could better capture the distribution in our dataset, offering a more reliable prediction model.

V. Summary

In this project we exploit the strength of machine learning to forecast the likelihood of developing MICHD based on personal features. Rigorous preprocessing and model selection led to the emergence of Ridge regression as the most performing model to achieve this task, highlighting its adaptability to varied data sets.

REFERENCES

[1] T. M., "Cardiovascular disease: An introduction," *Vasculopathies*, 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7123129/

[2] X. W. Cheng Fan, Meiling Chen, J. Wang, and B. Huang, "A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data," *Frontiers in Energy Research*, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fenrg.2021.652801/full