Machine Learning: Project 1

Jean Barenghi, Gergo Berta, Estelle Baup
EPFL, Switzerland

***Abstract*—This project aims to solve the problem of early detection and prevention of cardiovascular diseases. This will be done by applying machine learning methods to data from the Behavioral Risk Factor Surveillance System (BRFSS). Machine learning methods become particularly handy when working with a lot of data, and hence their use has been drastically increased in a variety of fields. For this particular problem, we will use different classification methods and study their behavior with the provided dataset.**

I Introduction

Coronary heart disease (CHD) is a major health concern that the World Health Organization monitors closely. Our aim in this project is to build a model based on the data from the BRFSS, which is able to estimate confidently the likelihood of developing MICHD (myocardial infarction CHD) given certain health-related features. To that end, we will process the data, apply different methods, and improve our model by tuning the parameters.

II Data Pre-processing and Feature Selection

We conducted a small exploratory data analysis to familiarize ourselves with the dataset we used throughout the project. First, the training dataset *x_train* contains 328,135 number of observations and 321 features. We also noted that the dataset is rather imbalanced, as only 8.83% of the observations lead to a prediction of MICHD (those predictions are in *y_train*).

Our first step in pre-processing involved getting rid of the columns that contain too many missing values. We plotted the number of missing values (NaN) for each of the features, and decided to set a threshold of 80%, dropping all features which had a number of non-missing observations below this threshold. This left us with 144 features.

We wanted to make sure that none of our features are perfectly co-linear and remove the ones which have a high correlation with several of the other features. For this reason, we plotted the heatmap of the features across the dataset (see Figure 1) and created functions that removed perfectly colinear features and 30 features that had the largest number of high correlations with others (where high correlation is defined as $\rho > |0.6|$). We show in Figure 2 the heatmap of the correlations after we dropped the highly correlated columns.

From the description of the features, we also realized that some of them were given in different measurement units (e.g. pounds and kilograms), so we replaced these with variables given in standardized units. Furthermore, we dropped features related to the date of the interview as these were irrelevant to our study. In the final dataset, we are left with 96 features. For some categorical features, we also created a logic that

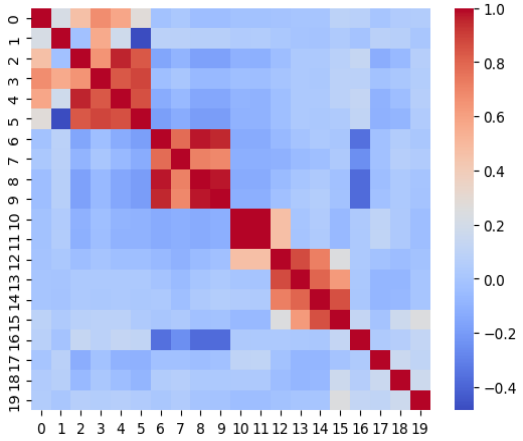

Fig. 1. Heatmap before pre-processing

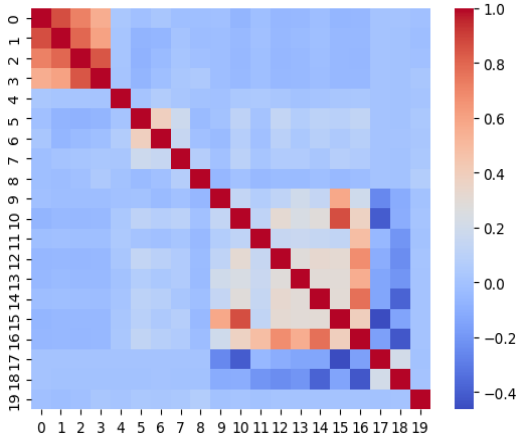

Fig. 2. Heatmap after pre-processing

replaces the 'Did not Respond' and 'Don't know' responses with NaN values to preserve the ordinal structure of the data. Similarly, to prevent extreme values (or simply recording errors) from influencing our estimation we replaced the values of continuous features over 1.5 times their interquartile range with NaNs.

Finally, we replaced all the missing values with their medians across features. We chose the median over other methods (such as the mean) as it gives us results more robust to outliers.

III Models and Methods

We implemented 6 different machine learning algorithms that we saw in the course: Linear Regression with Gradi-

ent Descent (MSE GD), Linear Regression with Subgradient Descent (MSE SGD), Least Squares Regression, Ridge Regression, Logistic Regression with Gradient Descent and Regularized Logistic Regression with Gradient Descent. To compare those methods, we applied them to our pre-processed dataset and obtained the results in Table III.

Methods	F1-score	Accuracy	λ	γ	Max iter
MSE GD	0.2965	62.49%		0.001	50
MSE SGD	0.2089	54.17%		0.001	50
Least squares	0.3203	66.70%			
Ridge Regression	0.3088	62.24%	0.5		
Logistic Regression	0.2956	62.43%		0.001	50
Regularized Logistic Regression	0.2956	62.43%	0.5	0.001	50

TABLE I
PERFORMANCE OF THE SIX IMPLEMENTED METHODS

We can see that using the given parameters, all methods have very similar F1-scores and accuracies. We can still note however that the Mean-Squared Error using the Stochastic Gradient Descent method is performing slightly below the other methods.

IV Cross Validation

We selected the Regularized Logistic Regression method, as we believed it to be the best method where we could fine tune the hyperparameters.

We thus performed a 4-fold Cross Validation in order to find the best possible value for λ. Note that we used a fixed value of 0.5 for γ. To determine this value, we computed the maximum learning rate at which the gradient descent algorithm started to diverge. We then followed the rule of thumb according to which the optimum learning rate corresponds to half of the maximum learning rate.

We then added a polynomial function to our data and estimated the best degree in addition to the best possible λ.

We didn't try to minimize the root mean-squared error as we believed that it wasn't the best indicator to evaluate our model's performance in the case of a classification problem. We rather tried to maximize the F1-score, which corresponds to the harmonic mean of the precision and the recall.

In Figures 3 and 4, we can see the evolution of the validation set's F1-score as a function of the degree of the polynomial and the parameter λ respectively.

After having performed the Cross Validation, we found the optimal values for the polynomial degree and λ to be respectively equal to 1 and 0.5. Those are the parameters that lead to our best submission on AIcrowd.

V Discussion

With the Regularized Logistic Regression method and our way of selecting the features and the hyper-parameters, we reached an accuracy of 86.4%, for an F1 score of 0.399 (our best submit on AIcrowd). We could also reach a higher accuracy by tuning differently the hyperparameters, but the F1 score decreased.

Since we studied Logistic Regression quite late, we did not have the time to improve and test our model with this method as much as we wanted to. For example, another approach we could have taken is to try a non-polynomial transformation of the data. It is also possible that we could have restricted ourselves to fewer columns to simplify the model, but doing that led to a smaller F1-score so we decided to keep a rather larger number of features. Finally, another idea we had in mind was to subsample the dataset to work with a more balanced training set.

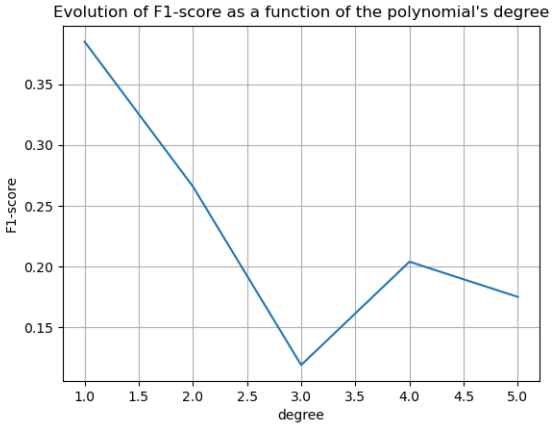

Fig. 3. Cross Validation in order to find the best polynomial degree

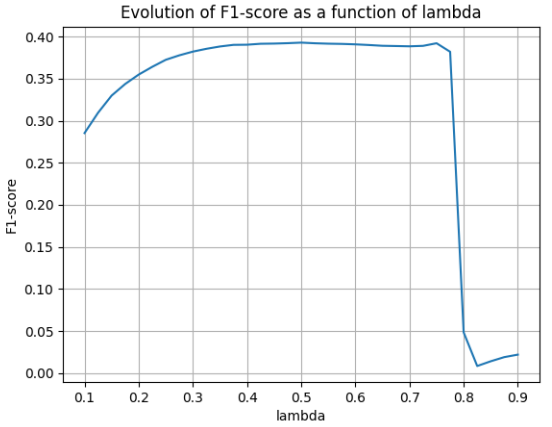

Fig. 4. Cross Validation in order to find the best penalization term λ

VI Summary

During this project, we saw the differences in performance and limitations of the six classical machine learning algorithms we implemented. We learnt how to explore the data, and take pre-processing steps to be able to handle a big dataset and clean it. We are aware that our model is not perfect, but we think that we took logical and necessary steps to develop and improve it.