Project 1

Dupont-Roc Maud, Grosjean Barbara, Ingster Abigaïl,
CS-433 Machine learning, EPFL, Switzerland

***Abstract*—The project involves using machine learning techniques and theory on a real-world dataset. The dataset uses lumped profiles of US residents classified as suffering from coronary heart disease (MICHD). The aim of the project is to explore the dataset, process and engineer its features, and finally train a machine learning model capable of predicting, given a certain clinical situation and lifestyle, whether a person is likely to develop MICHD.**

I. Introduction

The project is based on the data set from the Behavioral Risk Factor Surveillance System. Those data lump 321 different clinical and lifestyle features among 437514 different people, and for 328135 of them whether they have or not MICHD. Medical progress in hard-to-cure pathologies generally relies on prevention and/or early detection. This is where machine learning tools become interesting in the clinical field. Machine learning models are built in such a way as to extract the rules underlying an observable mechanism. After training, these models can be used for prediction purposes, i.e. predict an output event for a set of input conditions. The task to be implemented here is a two-labeled classification task, i.e. MICHD or non-MICHD.

II. Models and Methods

A. Data Preprocessing

Three relevant properties of the data set should be considered; The data set is unbalanced, i.e. there are more "non-MICHD" than "MICHD" labels (-1 and 1 resp.); Some features are homogeneous across all the samples (data points), hence, they do not carry any supplementary information; Finally, the data set is incomplete, i.e. some data are missing and implemented as NaN.

Our strategy for feature engineering first consisted of removing features with more than 50% missing data as well as homogeneous features, and replacing the missing dimensions of each data point with the median of the corresponding variable. In order to mitigate the imbalance of the training set, we decided to under-sample the majority class (non-MICHD), yielding a $2:1$ ratio between non-MICHD and MICHD labels (-1 and 1 resp.), privileging the removal of data points with the most missing values. Thereafter, we standardized the data along features in order to avoid exploding gradient, which would eventually lead to feature-dependent convergence rates, given a fixed learning rate γ.

The computational complexity of our optimization algorithm, gradient descent, depends on the dimension D of the data set. Indeed, the complexity per iteration is $\mathcal{O}(N \cdot D)$, where N is the number of samples. This is the reason why we decided to linearly reduce the dimension of our data by applying a Principal Component Analysis (PCA). The cumulative explained variance exceeded 90% with 83 principal components (*cf.* 1). At this step of preprocessing, the data consisted of 86925 samples of 83 dimensions.

We decided to include an offset term w_0 in the target weights of our model (*cf.* II-B). This required adding an artificial dimension to the data, i.e. a column vector of ones.

Finally, to enable more convenient computations, we converted {-1,1} labels to {0,1} labels.

Our preprocessing pipeline yielded a remodeled data set of 86925 samples with 84 dimensions, further denoted as $\tilde{\mathbf{X}}$.

B. The Model

The logistic regression model we used behaves as follows; For a given data point $\tilde{\mathbf{x}}$, the logistic regression outputs $p(1|\tilde{\mathbf{x}})$, the probability that sample $\tilde{\mathbf{x}}$ has label 1, i.e. :

$$\mathbb{P}(Y = 1|X = \mathbf{x}) = \sigma(\tilde{\mathbf{x}}^T\mathbf{w}), \tag{1}$$

where $\mathbf{w} = (w_0, w_1, ..., w_D)$. The bias term w_0 shifts the sigmoid curve of probabilities along $\vec{e_x}$, thereby artificially changing the probability threshold for belonging to one class or another. Thereafter, $\tilde{\mathbf{x}}$ is classified according to :

$$y_{pred} = \begin{cases} 1 & \text{if} \quad p(1|\tilde{\mathbf{x}}) \geq 1/2 \\ 0 & \text{if} \quad p(1|\tilde{\mathbf{x}}) < 1/2 \end{cases}. \tag{2}$$

In other words, this model defines a hyperplane $\mathbf{w}^\perp$ (*cf.* 3), normal to a sigmoid function's inflection point, and tends to separate the data into two classes upon optimization :

$$\mathbf{w}^\perp = \{\mathbf{x}|\tilde{\mathbf{x}}^T\mathbf{w} = 0\}, \tag{3}$$

where $\mathbf{w} \in \mathbb{R}^D$ are the weights to be optimized. The loss of our model is defined as follows :

$$L(\mathbf{w}) := \frac{1}{N}\sum_{n=1}^{N} b_{y_n}(-y_n\tilde{\mathbf{x}}_n^T\mathbf{w} + \log(1 + e^{\tilde{\mathbf{x}}_n^T\mathbf{w}})) + \lambda||\mathbf{w}||_2^2, \tag{4}$$

where y_n is the label of the data point $\tilde{\mathbf{x}}_n$, $\mathbf{w}$ the current weights, b_{y_n} the class weight of y_n, i.e. the inverse of y_n's class frequency, λ the regularization parameter aiming at penalizing large $||\mathbf{w}||_2^2$ (L_2-norm of $\mathbf{w}$), and thereby mitigate overfitting. Such class-dependant scaling of the loss (weighting with b_n) is called cost-sensitive learning, i.e. it penalizes more mis-classifications occurring in the minority class (here 1). The optimization problem $\mathbf{w}^* = \min_{\mathbf{w}} L(w)$ was solved by gradient descent as follows :

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \gamma \cdot \nabla L(\mathbf{w}_t) \tag{5}$$

$$\nabla L(\mathbf{w}_t) = \frac{1}{N}\tilde{\mathbf{X}}^T(\sigma(\tilde{\mathbf{X}}\mathbf{w}_t) - \mathbf{y}) \odot \mathbf{b}, \quad (6)$$

where $\mathbf{b}$ is the vector of class weights and γ the learning rate, set to 0.5 empirically. The convergence threshold was set to 1×10^{-12}, $\mathbf{w}_0$ to $\mathbf{1}$ and the maximum number of steps allowed to 1000. Gradient descent's computational complexity per step is N times higher than that of stochastic gradient descent. However, its convergence is smoother, which is preferred given the relatively small maximum number of steps allowed in order to converge to $\mathbf{w}^*$.

C. Model validation

The hyperparameter λ is chosen according to K-fold stratified cross-validation. This algorithm divides the training set into K subsets which individually conserves the same distribution as the original data set. K-1 subsets will be used for training and one subset for testing. This algorithm is therefore suitable for imbalanced training sets. λ was selected so to prevent overfitting of the training set. Additionally, such stratified version of the cross-validation algorithm aims at reducing the bias in the estimate of the generalization error. Indeed, the test set may contain a different negative to positive cases ratio from the training set. The metric used for the selection is the F1-score :

$$\text{F1} = 2 \cdot \frac{TP}{2TP + FP + FN}, \quad (7)$$

where TP is the number of true positives, FP the number of false positives, and FN the number of false negatives. The data set is unbalanced, thus the model can tend to a systematic classification in the majority class, in order to minimize the losses. The F1 score especially allows control over the false negative instances.

15 different regularization parameters λ, linearly spaced between 0 (no reg.) and 1.5 were tested. λ_{opt} was selected as follows :

$$\lambda_{opt} = \max_{\lambda} \frac{1}{K_{fold}} \sum_{k=1}^{K_{fold}} F1(\mathbf{w}_\lambda, \tilde{\mathbf{X}}_{test,k}, \mathbf{y}_{test,k}), \quad (8)$$

where $\mathbf{w}_\lambda$ and $\tilde{\mathbf{X}}_{test,k}$ were used to compute the prediction $\hat{\mathbf{y}}_{test,k}$ according to 2. Thereafter, the logistic regression was trained on the whole preprocessed training set with λ_{opt}, yielding $\mathbf{w}^*$.

III. Results

	without class weight	with class weight
λ (*cf.* 2)	0	0.32143
F1 score (train)	0.799	0.775
F1 score (test)	0.209	0.245
Prediction loss	0.67405	0.80253

IV. Discussion

Even though the F1 score for the training set are better for the model developed without class weighting and with λ=0, the F1 score for the testing set increases if we implement class weighting, which, by cross-validation model selection also induces L_2-penalization. Yet, it is not clear whether this score improved because of overfitting the testing set or thanks to a more generalizable model. However, as the F1 scores differ much between train and test set, we can clearly state that our model overfits the training set. A solution to that could be to retain less information from the data, e.g. using less principal components.

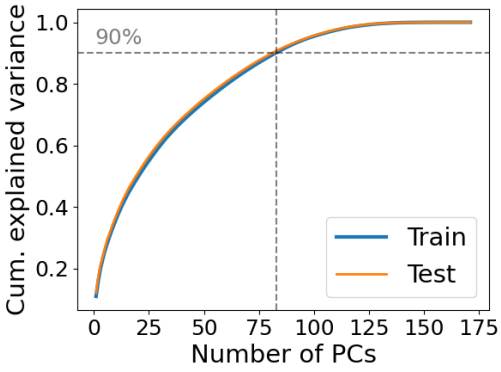

Fig. 1. Cumulative explained variance vs. number of Principal Components (PC) used to describe the data.

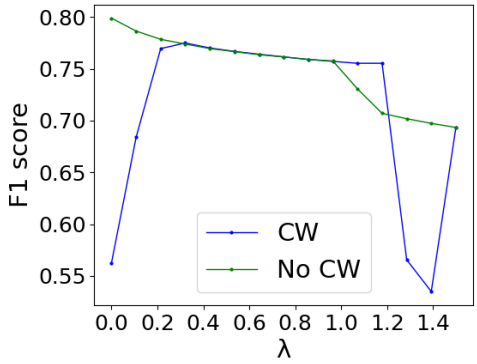

Fig. 2. F1 score of the classification found by logistic regression for different regularization parameters λ, with and without class weighting (CW and No CW resp.).

V. Summary

The task consisted in predicting, on the basis of clinical situations and lifestyles, whether a person was likely to develop MICHD. Our logistic regression model was trained by gradient descent, including L_2 regularization and increasingly penalizing false negatives thanks to class weighting. Finally, despite tuning the model's hyperparameter λ by stratified cross-validation, the selected model still exhibited overfitting of the training set. A further step would be to train a non-linear classifier, as the data may not be linearly separable. k-Nearest Neighbors would however be unsuitable to such a large data set.