# CS-433 Machine Learning - Project 1 Report

Lorenzo Drudi, Mikolaj Boronski, Olena Zavertiaieva
École Polytechnique Fédérale de Lausanne, 2023

***Abstract*—We propose a recipe to clean and prepare the data, which results in significant improvement in the training outcomes. We show how both logistic and linear regressions benefit from increasing the number of iterations and lowering the learning rate. We propose a logistic regression model achieving $0.418$ F1 score, $0.855$ accuracy on test (aicrowd) and $0.416$ F1 score, $0.854$ accuracy on training data.**

## I. Introduction

We implement and show the usage of basic machine learning methods – linear and logistic regressions – on data from the Behavioral Risk Factor Surveillance System (BRFSS). The dataset contains various information about the patient's health and associates with it a label whether the patient has heart disease (1) or not (-1). Our goal is to prepare the data, train the implemented models, and evaluate which model is the best at classifying between having a heart disease and not based on the health information. For evaluation, we use three basic metrics: the model's loss, F1 score, and accuracy. We present different approaches to data cleaning and feature engineering and show their impact (or the lack of it) on the learning process. Finally, we take a closer look at the tuning of hyperparameters of a chosen model and how it changes the evaluation metrics. To ensure the reproducibility we set a random seed in the whole environment to 42 and provide source code used to prepare this report, as well as the requirements file containing the versions of used libraries.

## II. Exploratory Analysis And Data Preparation

### A. Exploratory Analysis

The Behavioral Risk Factor Surveillance System (BRFSS) training dataset contains 328135 rows and 322 columns. Columns have information about the place, date, and health of the surveyed person. Authors of the dataset added "calculated" columns which could be either: variables used to stratify and weight the data, intermediate variables used to calculate other variables, or variables used to categorize or classify respondents. The existence of such columns instantly raises the question about the level of correlation between different dataset features, we address that below. Initial analysis shows a high percentage of missing data – 44% of all values. To understand how to clean the data we calculate the complementary cumulative distribution functions (CCDF) of max absolute correlation between columns per column and the fraction of missing data per column. In Figure 1 the high levels of both: correlation between columns and missing values are clearly visible. Around 40% of columns are missing at least 75% of values and have a maximum absolute correlation of at least 75% with one of the other columns (we exclude autocorrelations).

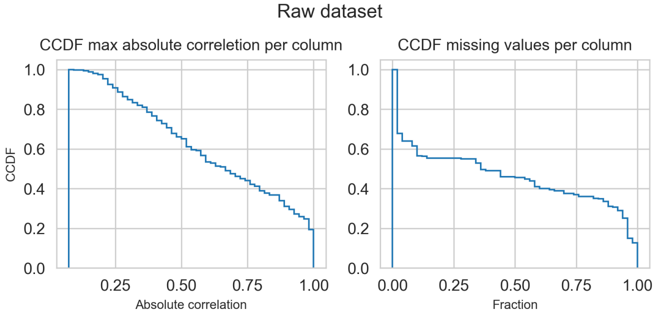

Fig. 1: CCDFs of max absolute correlation between columns per column and missing values per column on raw training dataset

Another problem frequently appearing in the classification tasks is the imbalance of the classes. This over-representation problem is also present in the BRFSS dataset: the rows associated with "-1" label make up 91.16% of the training data.

### B. Data Preparation

To address issues shown in Subsection II-A and enrich the dataset, we propose to sequentially apply the following routine:

1) Drop calculated columns
2) Drop columns containing more NaN values than a set threshold
3) Drop rows containing more NaN values than a set threshold
4) Replace remaining NaN values with either the mean of the column, the most frequent value of the column, random number sampled from a uniform distribution with values range same as the column, or replace with 0
5) Balance the dataset so that we match the number of dataset rows for each label, down-sampling the over-represented class by using a random choice, with a possibility to scale by a balance scale coefficient to keep some Bayesian "natural" disproportion
6) Build a polynomial out of features up to some $N-th$ degree
7) Standardize the data using z-score
8) Drop outliers based on their distance from the mean in the units of standard deviation

In Figure 2 we present an example of how the data changes after applying steps 1 to 3 with both thresholds set to 0.9. Running all these steps for NaN threshold – 0.9, balancing

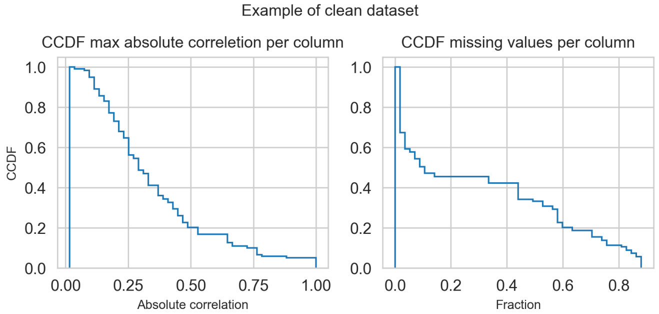

Fig. 2: CCDFs of max absolute correlation between columns per column and missing values per column on raw training dataset

## III. Models and Methods

To get a sense of which direction (data cleaning parameters, specific model, model hyperparameters etc.) is worth pursuing, we first run a grid search on all parameters for Mean Squared Gradient Descent Linear Regression, Logistic Regression, and Regularized Logistic Regression, and create boxplots of their f1 scores – we want to see whether there is a certain combination that is falling further away from the mean of all the runs per model. The results of the grid search are shown in Figure 3. Linear regression clearly does worse, but the shift in its distribution compared to logistic regressions only appears because of the drastic decrease in F1 score when the grid search parameters were set so that the training dataset was not balanced. Interestingly, this parameter did not matter in logistic regression. The specific parameters of the grid search are saved in the source code attached to this report.

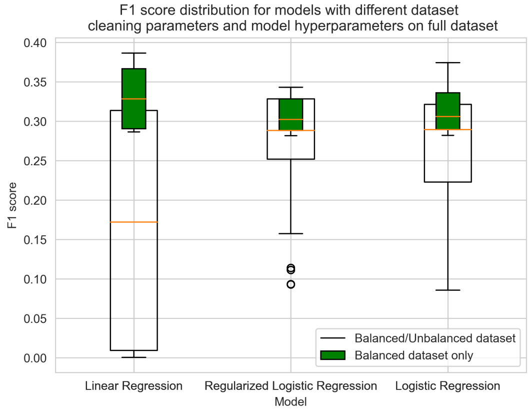

Fig. 3: Distributions of F1 scores per model for different dataset cleaning and model hyperparameters

## IV. Results and Discussion

Based on the results from Figure 3 we decide to apply data balancing, and cleaning, and move further into analyzing the impact of model hyperparameters on the F1 score. We choose the best parameters (drop calculated columns, 0.9 NaN threshold for rows and columns, replace NaN with 0, balanced dataset, 2nd degree polynomial expansion) out of the grid search and create heatmaps containing the F1 scores for the learning rate $\gamma$ and number of iterations. The results are shown in Figure 4.

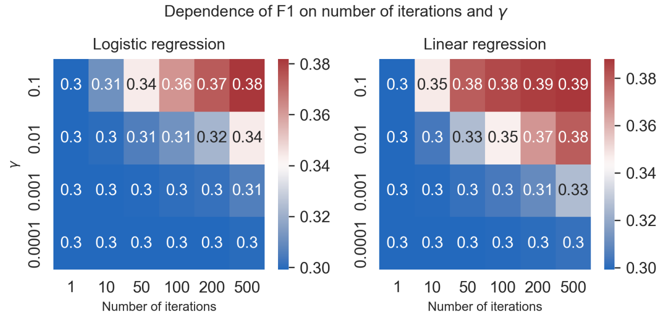

Fig. 4: Dependence of F1 scores on the number of iterations and learning rate $\gamma$ for linear and logistic regressions

Figure 4 clearly shows that increasing the number of iterations and decreasing the learning rate $\gamma$ results in a better F1 score. We implement another grid search just on the number of iterations and learning rate. The results are shown in Figure 5.

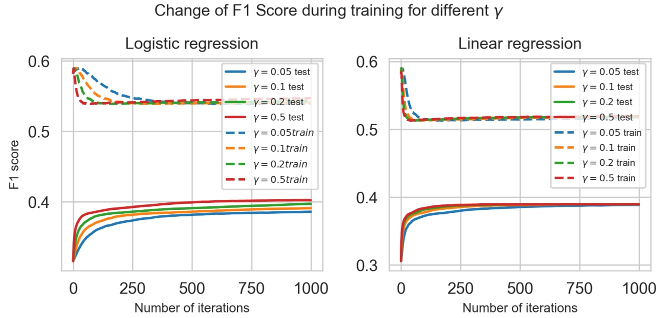

Fig. 5: The change in F1 score in training (balanced and cleaned) dataset and a full one for different learning rates

Based on these results we train our final model to be a logistic regression one.

## V. Summary

Using the methods proposed in the report we clean and balance the dataset. This lets us lower the correlation between columns and bring up the means of F1 scores of our 3 models: linear, logistic, and regularized logistic regressions. We then show that there is a dependence between raising the number of iterations in both linear and logistic models. By additional parameter tuning, we present a logistic regression model achieving 0.418 F1 score, 0.855 accuracy on test (aicrowd) and 0.416 F1 score, 0.854 accuracy on training data. Its specific parameters are inside the attached `run.py` file.