Cam_pred's Project 1 Report

Cyrine Akrout, Ahmed Zguir, Mohamed Mokhtar Sellami
EPFL, Switzerland

Abstract—This project applies machine learning to predict cardiovascular diseases (CVDs) using data from the Behavioral Risk Factor Surveillance System (BRFSS).

1 Introduction

We explore the BRFSS survey to predict coronary heart disease. The dataset contains hundreds of features describing respondents' health, habits, and demographics. Before modeling, we examine balance, missingness, feature granularity, and univariate analysis.

2 Data Exploration

The training set has 328,135 samples and 321 features; at this scale we rely on both systematic and precise exploration.

2.1 Class imbalance

As shown in Figure 1, the dataset exhibits a strong class imbalance, with a dominance of negative cases ($y = -1$) over positive cases ($y = +1$). This imbalance indicates that a naive classifier could achieve high accuracy by mostly predicting the majority class, while failing to detect true risk cases effectively. Therefore, we should use a model that is not highly sensitive to class imbalance.

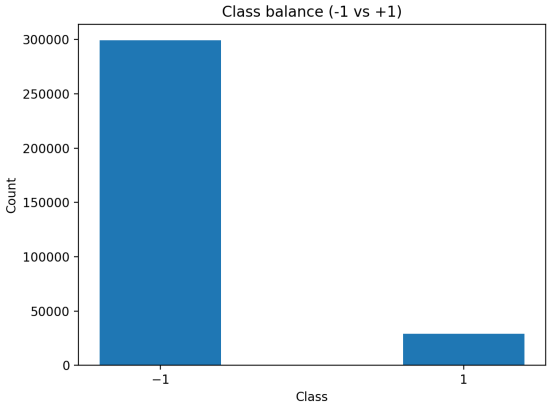

Figure 1: Class counts for $y = -1$ and $y = +1$.

2.2 Missingness structure

Figure 2 presents the distribution of features according to their fraction of missing values. As shown in the plot, a large portion of the features contain more than 75% missing values. Many of these columns do not provide meaningful insights and may even introduce noise into the model. Therefore, such features were removed to reduce dimensionality and improve prediction quality. However, certain variables with a high proportion of missing values, such as CVDASPRN and STREHAB1, carry important medical information. To avoid losing this signal, we handled these features separately through targeted imputation strategies.

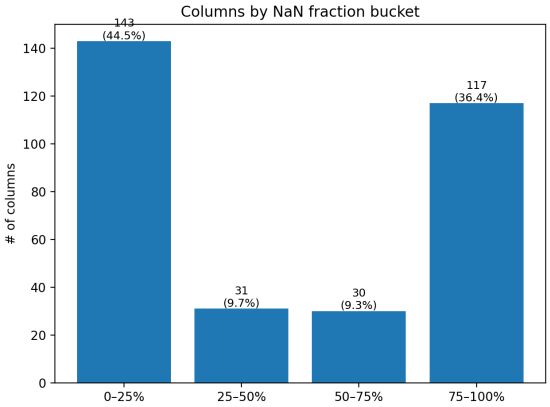

Figure 2: Number of features per missingness bucket.

2.3 Feature granularity

Another interesting aspect to examine is whether the features behave as categorical or continuous variables. This distinction is important, as certain regression techniques (e.g. logistic regression) are better suited for discrete inputs, while others perform better on continuous features. Figure 3 shows that most features have a relatively small number of unique values, indicating that they likely represent discrete categories or ordinal levels rather than continuous measures. This insight guides our preprocessing decisions, such as applying one-hot encoding for categorical features.

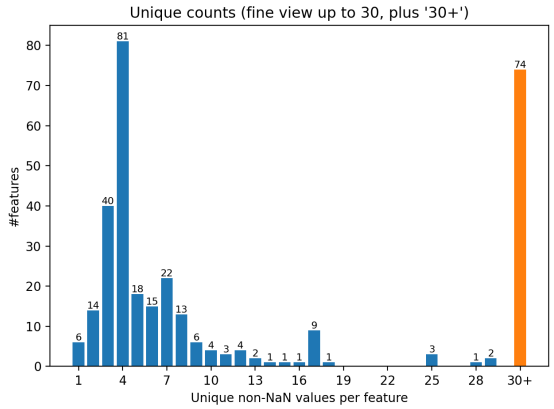

Figure 3: Unique-value counts (fine view up to 30 with "30+").

2.4 Correlation analysis

Figure 4 shows how strongly each continuous feature is related to the target variable. We can see that most features have only a very weak relationship with the outcome. This means that no single feature on its own can explain much of the result — the useful information is spread out across many different variables. Because of this, we use techniques like Principal Component Analysis (PCA), which combine several features together to capture patterns that only appear when considering them jointly, rather than one by one.

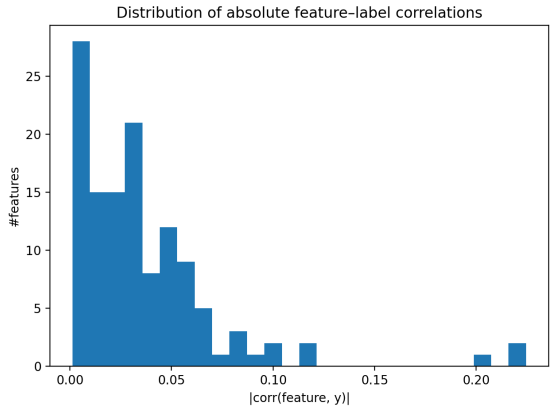

Figure 4: Distribution of $|\,\mathrm{corr}(\mathrm{feature}, y)\,|$.

3 Data Preprocessing

Based on the insights gained from the data exploration, we now prepare the dataset for modeling. This stage involves cleaning the data, handling missing values, encoding categorical variables, normalizing features, and applying dimensionality reduction where appropriate. All preprocessing steps are done similarly in both the training and test sets.

3.1 Targeted Filling of Medically Meaningful Variables

Before any pruning, we applied targeted imputations to a few clinically important variables (e.g., diabetes and cardiac-rehabilitation indicators) to preserve useful signal for later analysis.

Missing values in continuous features were imputed with their median values (e.g., DIABAGE2), while categorical features were filled with zero when the missing value indicated that the question or test was not applicable. For instance, in PREDIAB1, empty entries corresponded to respondents already diagnosed with diabetes, so these were set to zero.

3.2 Manual Column Filtering

We removed obviously irrelevant identifiers and administrative fields (e.g. FMONTH, SEQNO) which showed very low correlation with the target variable.

3.3 Special code Handling

We removed special placeholder codes in the dataset (e.g., 7, 9, 77, 99, 888, 999), which represent responses such as "Don't know," "Refused," or "Not applicable." These codes do not correspond to real measurements and can distort statistical computations.

For each feature, placeholder codes were identified by comparing values against the realistic range observed in the training data. Any out-of-range code was replaced with `NaN`, ensuring that subsequent steps such as median imputation or scaling were not affected by invalid values.

3.4 Dropping High-Missingness Columns

We removed columns with a missingness fraction above 60% to reduce dimensionality and noise, resulting in the removal of 128 columns.

3.5 Dropping Zero-Variance Columns

We removed columns with (near) zero variance ignoring `NaN` values, resulting in the removal of 4 columns.

3.6 Categorical vs. Continuous Distinction

Since the dataset is entirely numeric, we distinguished between categorical and continuous variables based on the number of unique non-missing values. Empirically, a threshold of seven unique values provided the most performant separation between the two types. As a result, features with ≤ 7 treated as categorical and the rest as continuous

3.7 One-Hot Encoding of Categorical Features

We one-hot encoded categorical features to prevent the misinterpretation of numeric labels as ordinal values within a Euclidean feature space (e.g., implying that class 2 represents a higher level than class 1). After this step, the dataset contained a total of 502 columns, combining continuous and one-hot-encoded categorical features.

3.8 Dropping Linearly Dependent Categorical Columns

Following one-hot encoding, several categorical features introduced perfect linear dependencies among dummy variables. To address this, we applied a QR decomposition on the categorical block of the training set to detect and remove linearly dependent columns, eliminating 172 columns.

3.9 Median Imputation (Continuous Only)

Missing values in the remaining continuous features were imputed with the training-set median for each column. The learned medians were then applied to the test set to ensure consistency.

3.10 Standardization

We observed significant variation in the scales of different features, which could cause certain variables to dominate the learning process. To mitigate this, all features were standardized using training-set statistics (for both sets), subtracting the mean and dividing by the standard deviation for each column.

3.11 PCA on Continuous Features

We applied Principal Component Analysis (PCA) to the standardized *continuous* block of the training set and selected the smallest number of components that achieved at least 95% cumulative explained variance, resulting in the removal of 24 columns.

3.12 Final Design Matrix and Bias Term

The final input matrix concatenates the one-hot categorical block and PCA-reduced continuous block. We append an explicit bias term (a column of ones) as the last step. The final shape is (328135, 304)

4 Model Results

4.1 Comparative Evaluation of Models

Since our target variable y takes binary values, the prediction task is a binary classification problem. We therefore focused on models capable of estimating class probabilities rather than continuous values. When training the logistic and regularized logistic regression models, the preprocessing pipeline included a step that transformed the target labels from $\{-1, +1\}$ to $\{0, 1\}$. For the other models, however, the original labels were kept unchanged, since their loss functions operate directly on $\{-1, +1\}$ and do not require conversion to $\{0, 1\}$ as in logistic regression. We compared multiple regression and classification algorithms using both **accuracy** and **F1-score** as evaluation metrics. We chose the F1 score to be the main metric, as it provides the most reliable insight given the significant class imbalance present in the dataset.

Table 1 summarizes the performance of all tested models. Logistic Regression models achieved excellent performance, outperforming baseline linear models.

Model	Accuracy	F1 Score
Mean Squared Error GD ($\gamma = 10^{-4}$)	0.818	0.395
Mean Squared Error SGD ($\gamma = 10^{-4}$)	0.787	0.365
Least Squares	0.916	0.131
Ridge Regression ($\lambda = 0.1$)	0.917	0.167
Logistic Regression	0.884	0.448
Regularized logistic Regression	**0.879**	**0.449**

Table 1: Model performance on the AICROWD test data, based on accuracy and F1-score.

4.2 Justification for Choosing Regularized Logistic Regression

To ensure compatibility with the loss function and sigmoid activation, the target labels were transformed from $y \in \{-1, +1\}$ to $y \in \{0, 1\}$, enabling proper interpretation of output probabilities.

Standard Logistic Regression was used as a baseline, while an L2-regularized variant was also tested to constrain model weights and enhance generalization. Both achieved comparable accuracy and F1-scores, likely because the dataset's large sample size relative to its feature count inherently reduced overfitting.

Given its slightly improved F1 score, the **regularized logistic regression model** was selected as our final approach.

4.3 Model Fine-Tuning and Cross-Validation

We fine-tuned both the **logistic** and **regularized logistic regression** models using 5-fold cross-validation to maximize the validation F1-score.

For each model, we jointly optimized the learning rate (γ) and the classification threshold, and for the regularized version, we also tuned the regularization strength (λ). Threshold values between 0.02 and 0.98 were tested to handle data imbalance.

The optimal parameters obtained were $\gamma = 0.1$, threshold $= 0.212$ for logistic regression, and $\lambda = 10^{-3}$, $\gamma = 0.1$, threshold $= 0.208$ for the regularized model.

4.4 Impact of preprocessing on the model

To evaluate the contribution of each preprocessing step, we trained a regularized logistic regression model on the validation set obtained from an 80/20 split of the training data. Starting from the baseline configuration, which included imputing missing values with zero, standardizing features, and adding a bias term, each additional step was incrementally added in the same order as implemented in the final preprocessing pipeline. This ensured that the effect of each step was measured in a realistic, cumulative setting rather than in isolation. The F1-score was used as the main evaluation metric, as it effectively captures model performance under the dataset's strong class imbalance.

The preprocessing steps can be grouped into two main objectives: Goal 1 – Improving Model Performance: Steps such as filling important features, treating special codes as missing values, and one-hot encoding categorical variables led to consistent improvements in the F1-score. Goal 2 – Reducing Computational Cost: Operations such as applying PCA, removing administrative or low-information columns, and dropping columns with many missing values were primarily designed to reduce feature redundancy and improve efficiency.

Although dropping high-NaN columns initially reduced performance, this effect was later compensated by filling key features identified as important to retain. As shown in Figure 5, the final configuration, constructed following the same step order as described in section 3, outperformed the baseline, achieving a balanced trade-off between preserving informative but sparse features and reducing noise and dimensionality, resulting in a model that remained both efficient and accurate.

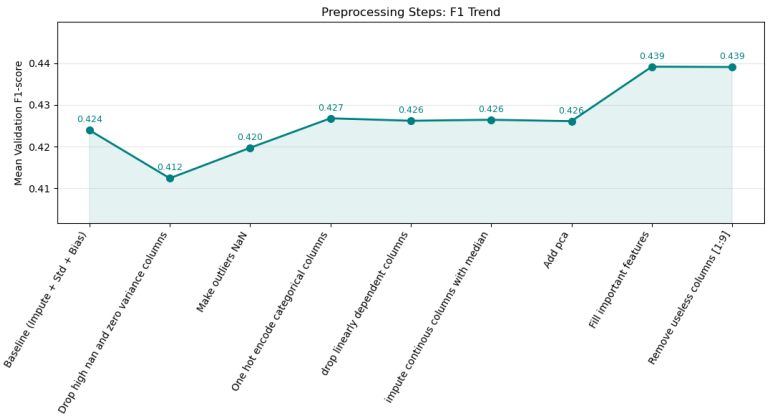

Figure 5: Incremental effect of preprocessing steps on validation F1-score

Conclusion

This report highlights the essential role of Data Analysis, Feature Engineering and fine-tuning in building effective Machine Learning systems. Through systematic data exploration, we addressed key issues such as missing values, class imbalance, and categorical variables. Both Logistic Regression and its regularized variant achieved similar results, reflecting the dataset's strong generalization due to its large sample size. Overall, the study emphasizes that careful preprocessing and model tuning are critical for achieving robust and interpretable predictions.