

Advanced Machine Learning - Project 1

Michał Gromadzki

March 2024

1 Methodology

This section describes methodology used in this project.

1.1 Datasets

Nine datasets have been selected from [UCI](#) and [OpenML](#). Three of those datasets have at most 10 variables and six have more than 10 variables. Figure 1 depicts distribution of datasets' dimensions.

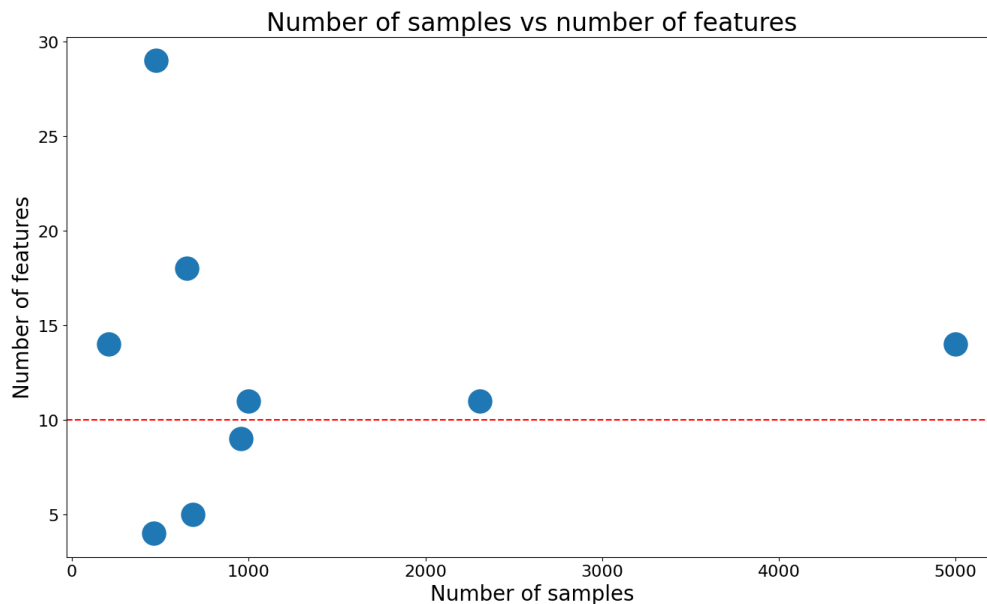


Figure 1: Number of samples and features

Datasets have at most 0.23% missing data.

In the next step, collinear variables have been removed from each dataset. Algorithm for removing collinear variables is described below.

Algorithm for removing collinear variables:

- Algorithm is iteration-based, in each iteration 1 variable can be removed
- In each iteration:
 1. Calculate variance inflation factor (VIF) for all variables
 2. Select all variables with VIF higher than $TH = 5$
 3. Delete variable with the highest VIF
- Algorithm stops when there is no variables with VIF over TH

1.2 Implementation

All optimization algorithms for parameter estimation in logistic regression have been implemented as classes in Python. All optimizers have the following attributes and methods. Attributes:

- `max_iter` (int) - Maximum number of iterations of the optimizer

- *tol* (float) - Minimum value of Frobenius norm of difference in parameters between iterations, used to determine convergence
- *coef_* (array) - Array containing parameters
- *intersections* (bool) - Whether to also include intersections of provided variables

Methods:

- *_add_intersections()* - Adds intersections to the provided variables
- *_sigmoid()* - Returns value of a sigmoid function for the provided input
- *fit()* - Trains the model, return history of training and number of iterations
- *predict_proba()* - Predicts probabilities based on the provided data
- *predict()* - Rounds the probabilities to class labels

Note that some implementations have individual attributes and methods necessary for their implementation.

1.3 Stopping rule

All algorithms have the same stopping rule to ensure fair comparison. At the end of each iteration of the algorithms the following value is computed:

$$diff = norm(coef - prev_coef)$$

where:

- *norm* - The Frobenius norm
- *coef* - Parameters in the current iteration
- *prev_coef* - Parameters in the previous iteration

If the *diff* is smaller than the pre-set *tol* value the training stops.

1.4 Training

All models have been trained on 9 different train-test splits. The *max_iter* and *tol* values have been set to 500 and 0.001, respectively. For SGD and Adam optimizer the *learning_rate* has been set to 0.01. Note that all variables have been scaled using standard scaler. To measure the performance of the models BalancedAccuracy was used on the test set.

2 Convergence analysis

To check for convergence, log-loss (negative log-likelihood) was computed on train set after each iteration. Figure 2 presents how the log-loss changes during training.

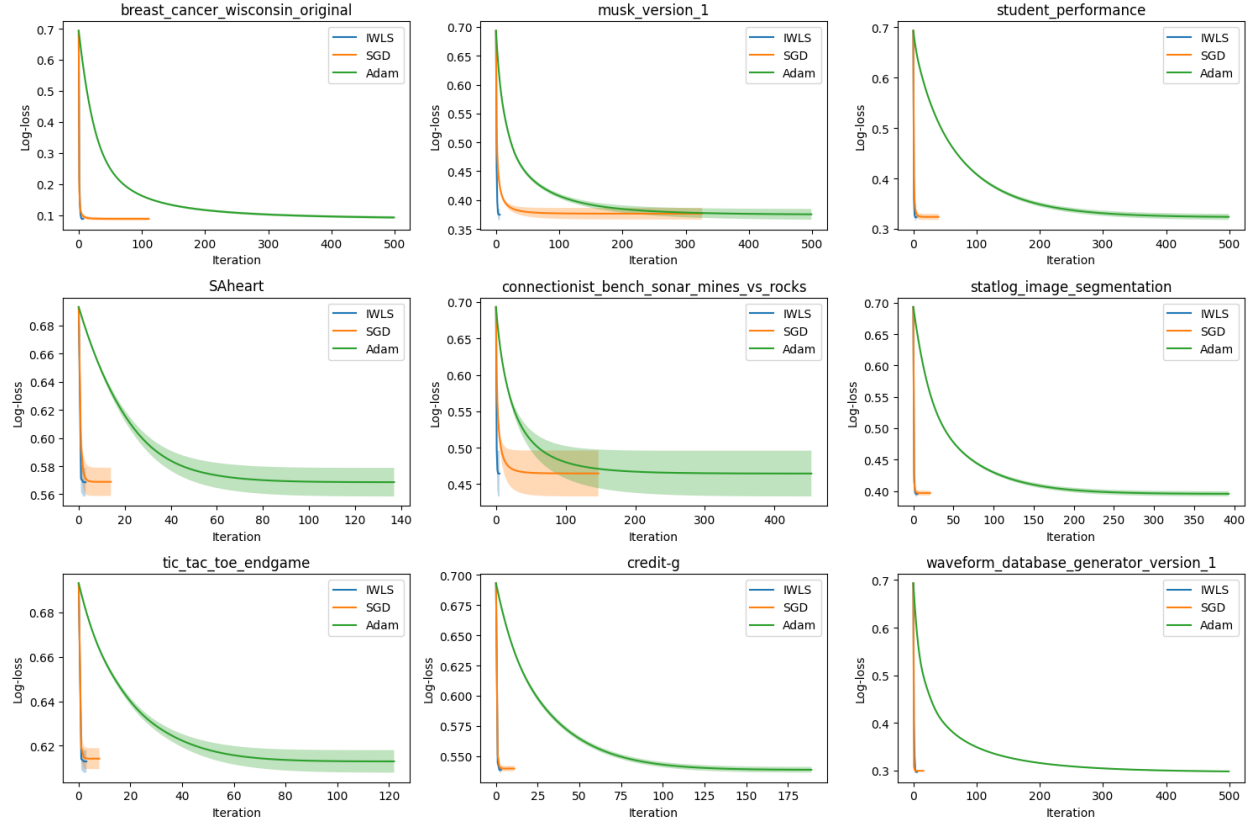


Figure 2: Log-loss value during training

Values at each iteration were computed as a mean of values from 9 different train-test splits. In all datasets final values of log-loss are similar for all methods. However, IWLS on average takes 5.7 iterations to finish training. On the other hand, SGD and Adam optimizers take much longer, 55 and 348 iterations, respectively.

3 Comparison of classification performance

Performance of the optimization algorithms has been tested on 9 datasets, along the implemented algorithms, LDA, QDA, DecisionTree and RandomForest were also used. Figure 3 depicts mean accuracy on each of the datasets.

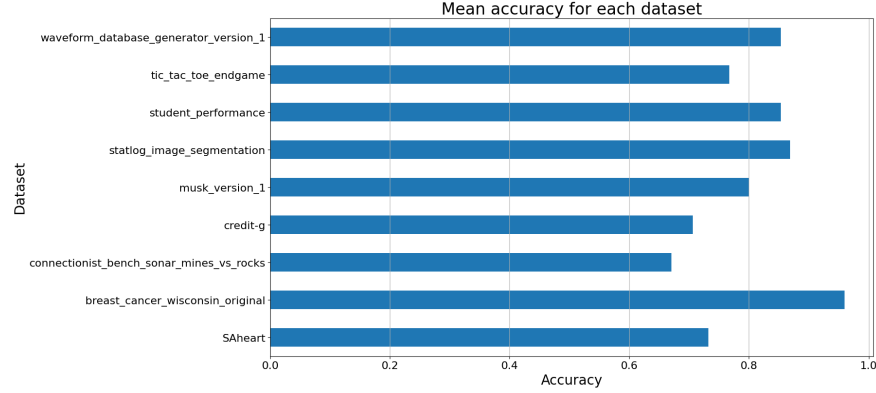


Figure 3: Mean accuracy on each of the datasets

Based on the Figure 3 we can tell which datasets are the easiest and which are the hardest. On average the hardest dataset is connectionist_bench_sonar_mines_vs_rocks and the easiest is breast_cancer_wisconsin_original. Figure 4 presents performance of all methods on all of the datasets.

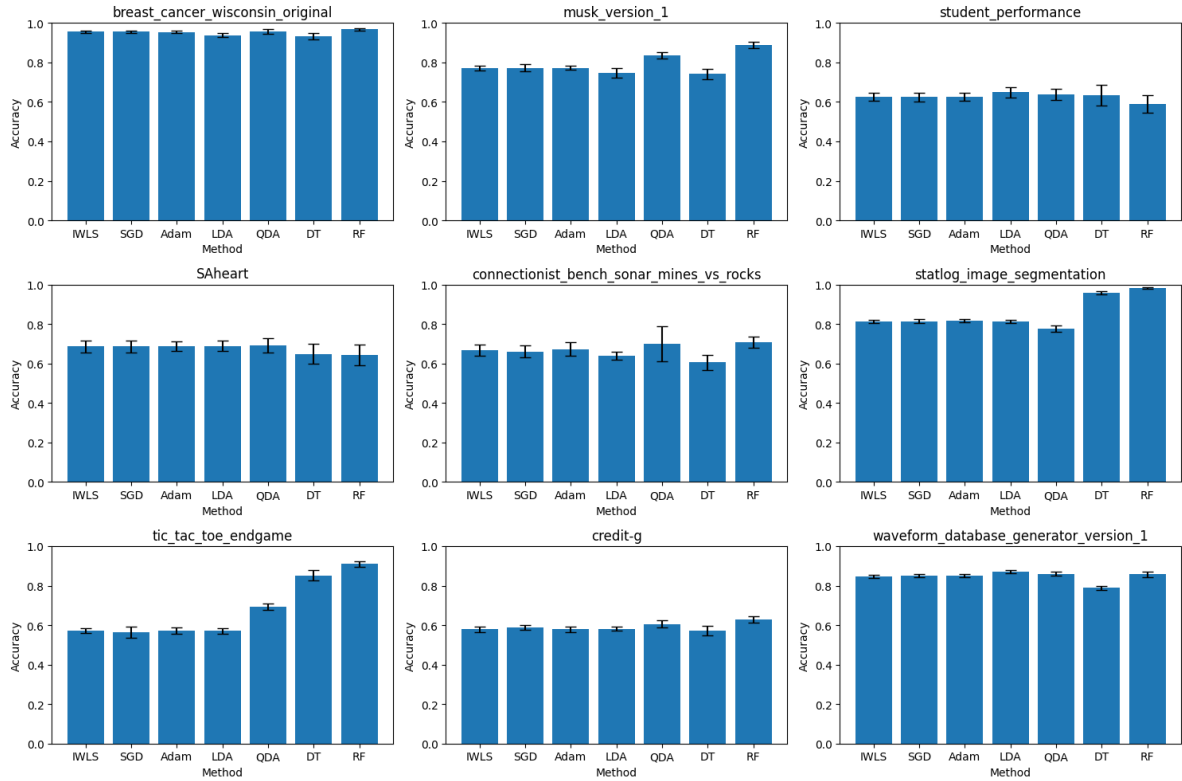


Figure 4: Accuracy of all methods on all datasets

RandomForest was the best performing algorithm in 8 out of 9 datasets. However if we only consider implemented algorithms (IWLS, SGD, Adam), each one was the best performing one 3 times. In all cases the differences were small, the biggest one occurred in the `connectionist_bench_sonar_mines_vs_rocks` dataset and was equal to 1.2%. Moreover, one of the implemented algorithms was the worst only in one of the nine datasets.

4 Comparison of classification performance of models with and without intersections

For small datasets, which have less than 10 variables, two variations of logistic regression were tested. One with and one without intersections of the variables. Figure 5 presents accuracy of models with and without intersections.

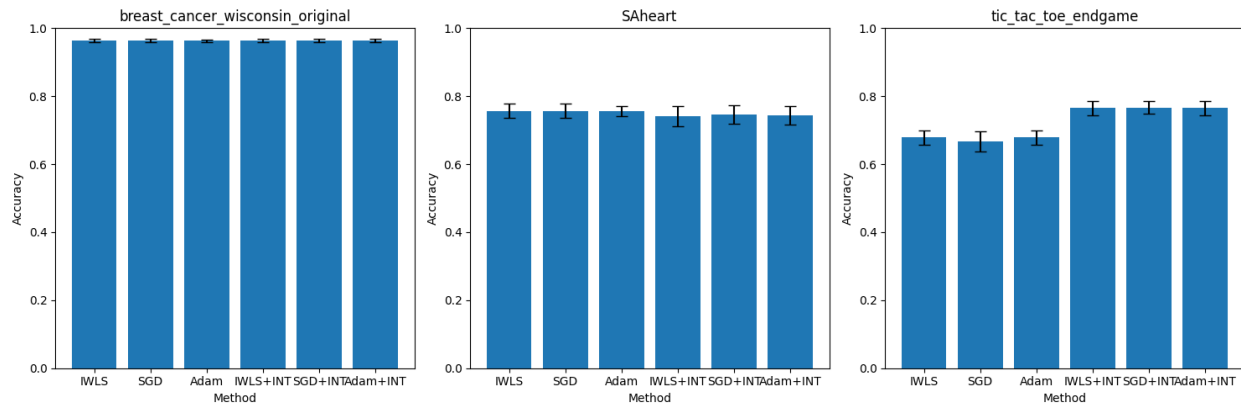


Figure 5: Accuracy of models with and without intersections

In two out of three datasets models with intersections outperformed those without. The biggest difference occurred in the `tic_tac_toe_endgame` dataset, where all models with intersections outperformed their counterparts without by about 10%.