AML Project 1 report

Adam Czerwoński, Adam Narożniak, Jędrzej Ruciński

Methodology

This project aims to analyze three different parameter optimization methods used for training Logistic Regression models. The optimization algorithms tested in this work are:

- IRLS (Iterative Reweighted Least Squares)
- SGD (Stochastic Gradient Descent)
- ADAM (Adaptive Moment estimation)

To test optimization algorithm performance, we use a selection of nine data sets for the binary classification task. Datasets are either loaded from the downloaded files or loaded directly using the HuggingFace datasets library. See Table 1 for the dataset details.

Every one of our optimization algorithms is created by a Python class designed to perform that specific type of optimization. These classes consist of an update method that iteratively updates weights and biases. For the experiments, an object of an optimizer is created and gets passed into our custom Logistic Regression class which processes data in epochs. This additionally promotes good coding practices by using dependency injection.

The Logistic Regression class, regardless of which optimization algorithm is used, calculates validation loss on each epoch.

Batch Size

Within a single epoch (or pass through the entire data set), we process the training data in batches in the case of SGD (otherwise, it would be GD), Adam, and for IRLS, we use the whole dataset, which is the who the method is primarily used. Otherwise, we would be prone to the multitude of problems related to matrix inversion that the method internally uses.

Stopping Rule

Our stopping rule is constructed in a way that allows the user to pass a patience parameter to the optimizer. If our best validation loss does not improve after a number of epochs equal to the patience value, the optimization stops.

Dataset Division

For our experiments we use a train-test split of 4:1, also cutting out 20% of the training data to use as validation data during parameter optimization. We use 5 different seeds during these splits, to then average out the obtained balanced accuracy, for more stable results.

Data set name	size type	dimensions	description
Body signal of smoking	big	(55692, 25)	This dataset is a collection of basic health biological signal data. The goal is to determine the presence or absence of smoking through bio-signals.
Basketball Players' Career Duration	big	(1340, 19)	The data consists of performance statistics from each player's rookie year. The target variable is a Boolean value that indicates whether a given player will last in the league for five years.
Banana Quality	small	(8000, 7)	Can you identify good bananas by their numerical characteristics?
Blood Transfusion	small	(748, 4)	The center passes their blood transfusion service bus to one university in Hsin-Chu City to gather blood donated about every three months
Online Shoppers	big	(12330, 18)	Online Shoppers Purchasing Intention Dataset contains sessions of different users over 1 year period.
Heart Failure Dataset	big	(299, 13)	Predict patient death from earth failure given some personal medical data
QSAR Biodegradation	big	(1055,42)	Data set containing molecular descriptors used to classify chemicals into 2 classes (ready and not ready biodegradable).
Parkinson’s Disease	big	(195, 22)	Classifying whether someone has parkinsons using his voice recording
Diabetes	small	(2000,8)	Classifying whether someone has diabetes or not using some basic health metrics.

Table 1: Comparison of the dataset used for the experiments.

Convergence Analysis

In this section we will investigate how the loss function (negative log-likelihood) of each optimizer changes over successive epochs and how fast it converges meaning how fast it satisfies the stopping rule (with patience equal to 5).

From the figure below we can draw the following conclusions:

- All optimizers achieve similar values of loss function.
- IRLS converges after very few epochs - in all cases besides the Banana dataset it stopped after the 5th epoch which was the value of patience.
- ADAM is the slowest optimizer - on Blood Transfusion and Heart Failure datasets it didn't converge in 500 epochs.
- On the Smoking dataset ADAM was outperformed by SGD and IRLS, however setting patience to a larger value could maybe improve its result.
- In most cases SGD has a much steeper learning curve than ADAM.

All these observations could lead us to think that IRLS is the best optimizer, however, this isn't necessarily the case. Although it converged in the lowest number of epochs it actually took the most time to train logistic regression with it. We should also remember that ADAM's results might be very dependent on our choice of hyperparameters. Taking all that into account it's hard to definitively choose the best optimizer.

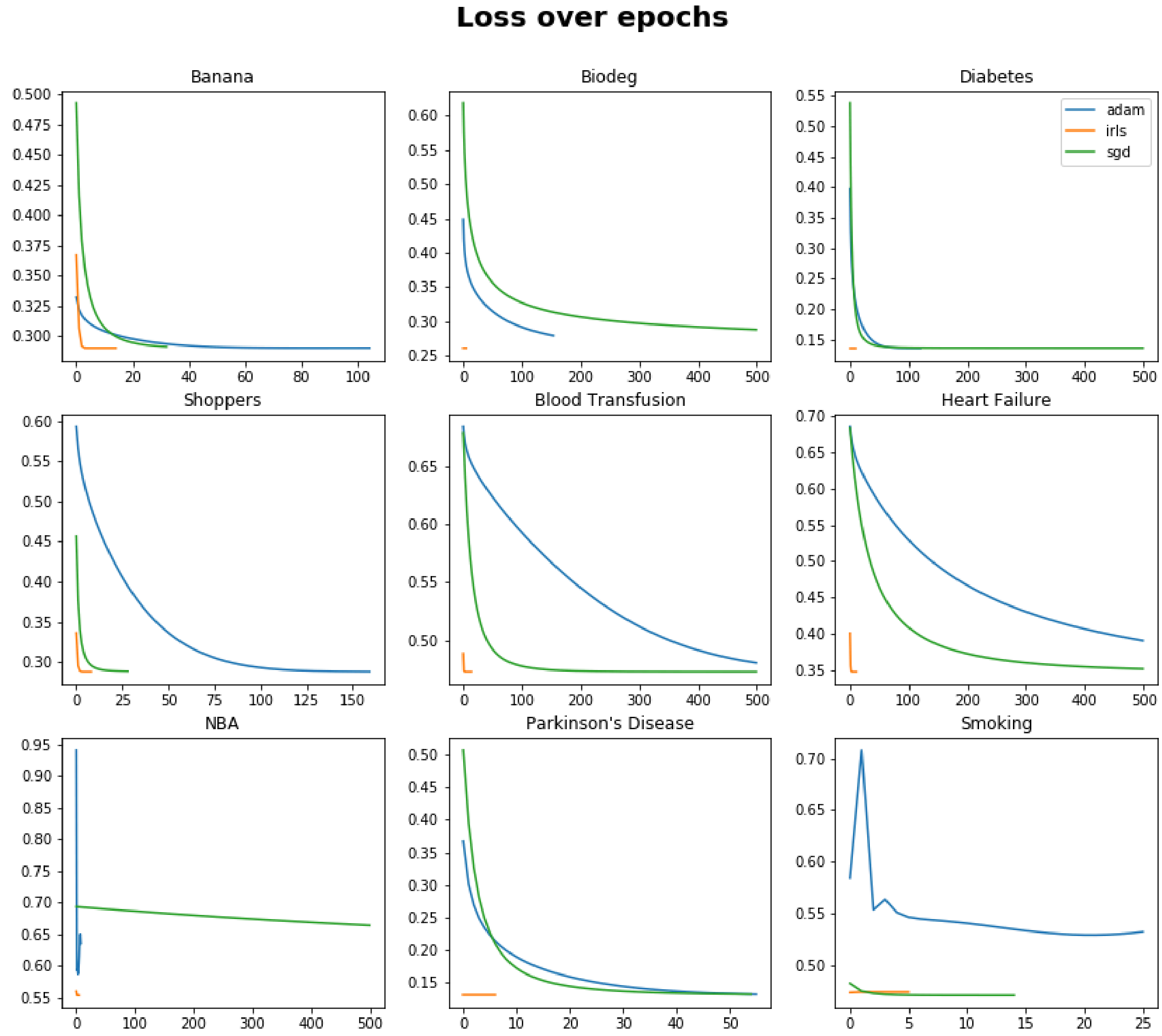

Figure 1: Value of loss function for each optimizer on different datasets.

Comparison of classification performance

The classification performance is measured as balanced accuracy on the test set that is not seen during the training phase and comprises 20% of the data.

Out of the different methods of training logistic regression, Adam typically is outperformed by SGD and IRLS. However, that might result from a lack of an extensive hyperparameter search. This is also the most parameterized optimizing method. The results of the IRLS and SGD are generally quite similar. We did not notice any clear distinctive behavior regarding the standard deviation.

There are 3 datasets in which the decision tree and random forest outperformed logistic regression significantly, 3 in which the tree models were outperformed and on the remaining 3 the results were similar. The LDA, QDA, and were typically similar to the ones of logistic regression however, there was not any case in which they outperformed random forest, they slightly overperformed decision trees in a few cases.

We do not see any clear pattern related to the size of the dataset and the performance of the method. However, there was a set of datasets that clearly displayed features that enabled easier division based on the decision tree methods.

Based on the results, a good rule of thumb might be to try any of the logistic regression methods and compare them with random forest to obtain the best results.

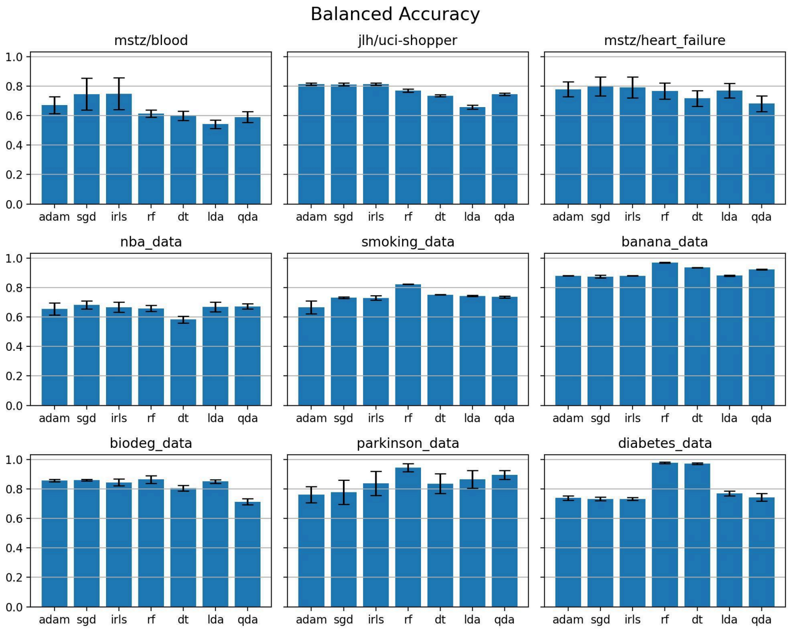

Figure 2: Comparison of balanced accuracy between the optimizers/models on all datasets

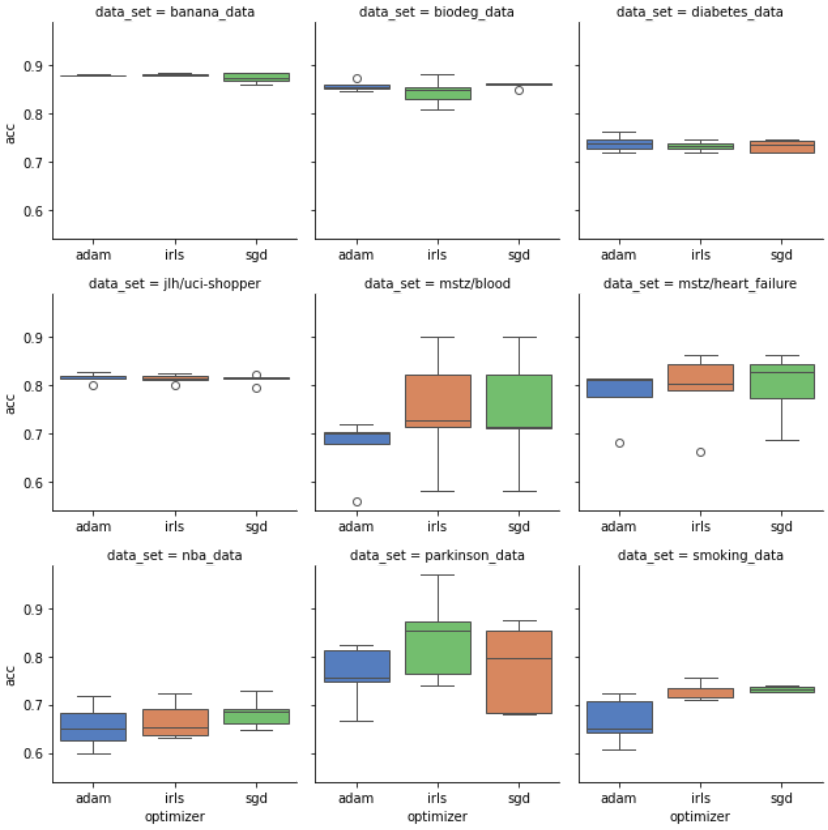

Figure 3: Comparison of balanced accuracy distribution between the optimizers/models on all datasets

Comparison of classification performance of models with and without interactions

All the small datasets (having under 10 features) were tested additionally (according to the data division rule) with feature interactions. Each pair of variables were multiplied by each other.

The biggest benefit of this operation was LDA and QDA. For both of these models, the mean accuracy increased, and the standard deviation stayed relatively similar. In all datasets, the tree-based models achieved similar results (no improvement was noticed). In two of of three datasets, the method was also beneficial to the logic that benefited from the feature interactions. The increase of the mean balanced accuracy was observed with relatively the same standard deviation. However, for one dataset, the feature interaction had negative consequences. The accuracy decreased along with the greeted standard deviation. This dataset (mstz/blood) was the smallest, in terms of the number of samples and in terms of the number of features (4). Therefore it is not the case that the feature interaction is beneficial in all cases.

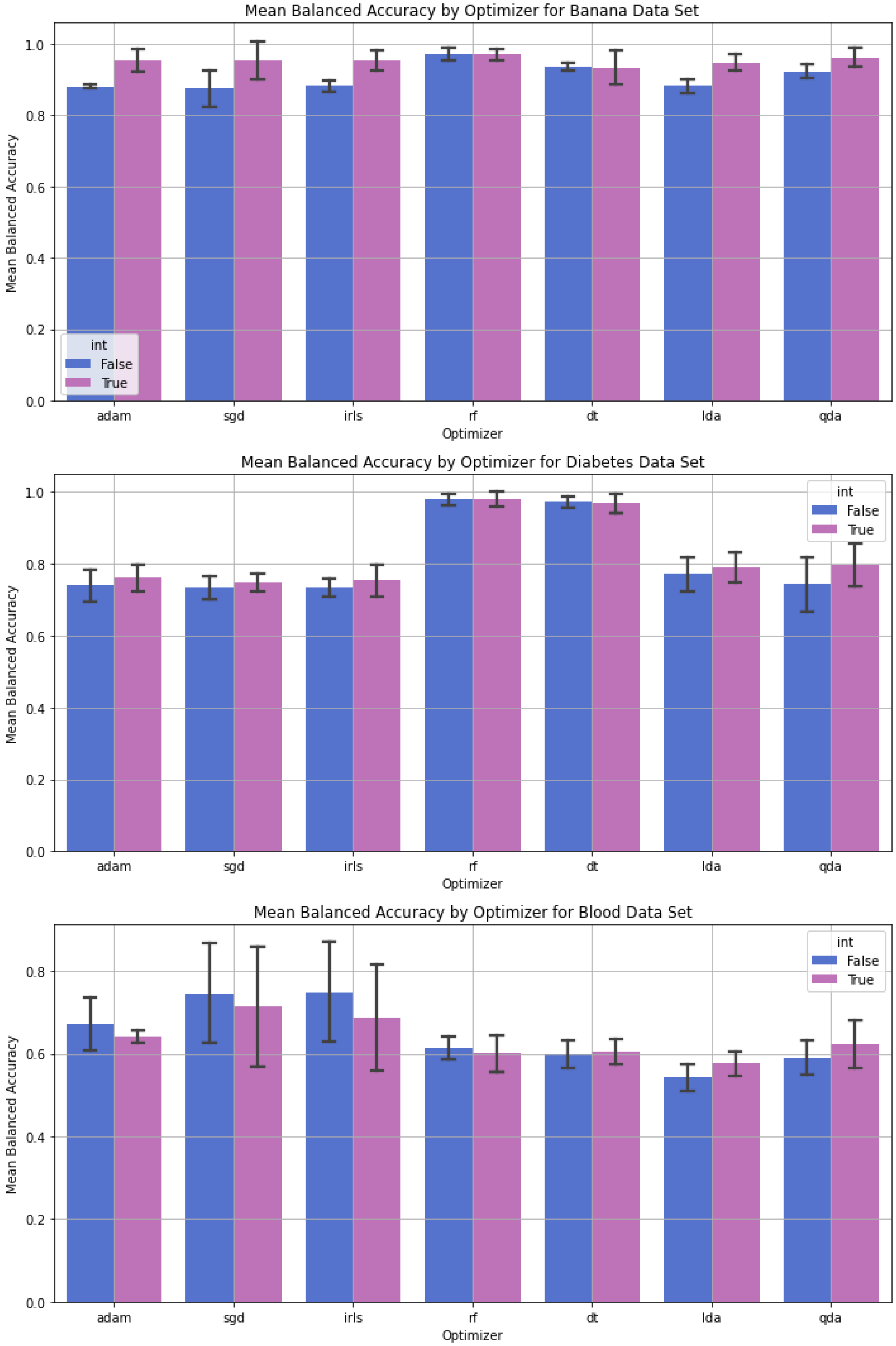

Figure 4: Comparison between the lack of feature interaction and feature interactions of the optimizers/models on small datasets