Advanced Machine Learning
Project 1

Zuzanna Glinka, Nikola Miszalska, Malwina Wojewoda

April, 2024

Contents

1 Methodology

1.1 Task 1

For the task, we selected 9 datasets and their specifics are outlined in the Table 1. The first three datasets, as recommended, are small, each containing fewer than 10 variables and subsequent datasets have more variables. The count of variables refers only to predictor variables and does not incorporate the target variable.

The datasets we selected do not have any missing data, so they do not require imputation. We conduct preliminary analysis on each dataset using correlation matrix, as shown in the Figure 1. We observed that in some datasets features might be collinear. To address this problem, we used Variance Inflation Factor (VIF) coefficients, which quantify the degree of multicollinearity and the formula for this is:

$$\mathrm{VIF}_i = \frac{1}{1 - R_i^2}.$$

In this formula VIF_i represents the VIF value for the i-th feature and R_i^2 represents the coefficient of determination from regressing the i-th feature on remaining ones.

A VIF of 1 means no multicollinearity, and higher values mean stronger multicollinearity. Usually VIF values greater than 5 or 10 are considered high. In our case, we use value of 10 as the threshold and we iteratively removed variables with the highest VIF values, recalculated the VIF for the remaining features, and repeated this process until no VIF value exceeded the threshold. Numbers of variables that remain after this process are also included in the Table 1.

1.2 Task 3.1

In our solution we have implemented early stopping, which can be set as boolean value when defining an instance of the classifier class, by default set to `True`. The idea is that the algorithm is stopped if the log loss calculated on the validation set does not decrease after a certain number of iterations. This number is `patience` parameter in the `fit()` function and by default is set to 10 iterations.

In addition, the parameters of this function are (`X_valid` and `y_valid`), which represent the validation set to be used for controlling the loss for early stopping. If it is not provided, the function automatically splits the specified training data into 80% for training and 20% for validation. However, disadvantage of this method is that in such case training is performed on less data than initially specified by the user.

When early stopping is triggered, the model return the best-performing result obtained during training, which ensures equal or better results than if it were the results from the last iteration.

1.3 Task 3.2

We used balanced accuracy to measure performance of our classification algorithms. This metric is particularly used when dealing with imbalanced datasets. It is calculates as the arithmetic mean of sensitivity and specificity, so the formula is as follows:

$$\text{balanced accuracy} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right).$$

For each algorithm and dataset combination, we split the data into training and test sets 10 times. For each split, we trained the model on the training set and measured the balanced accuracy on the test set. Distributions of results for each method and dataset are demonstrated in the Figure 2. In each case, the algorithms were fitted with early stopping enabled with patience of 50 iterations.

Table 1: Summary of datasets.

dataset name	source	total instances	total variables	non-linear variables	missing values
Transplant	OpenML, ID: 544	131	3	3	No
Phoneme	OpenML, ID: 1489	5404	5	5	No
Pima Indians Diabetes	OpenML, ID: 37	768	8	5	No
EEG Eye State	OpenML, ID: 1471	14980	14	4	No
Breast Cancer Wisconsin	OpenML, ID: 1510	569	30	7	No
Steel Plates Faults	OpenML, ID: 1050	1941	33	19	No
QSAR Biodegradation	OpenML, ID: 1494	1055	41	28	No
SPAM E-mail	OpenML, ID: 44	4601	57	56	No
Ozone Level	OpenML, ID: 148	2534	72	17	No

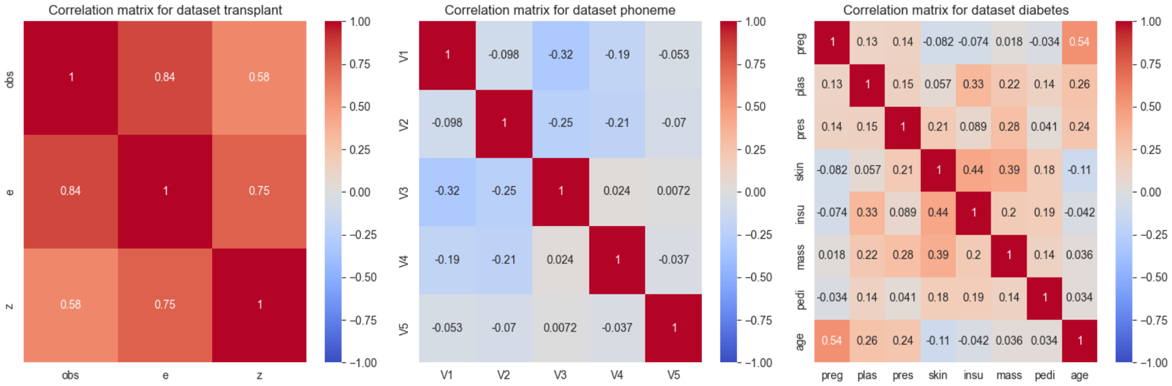


Figure 1: Correlation matrix for small datasets before removing collinear features.

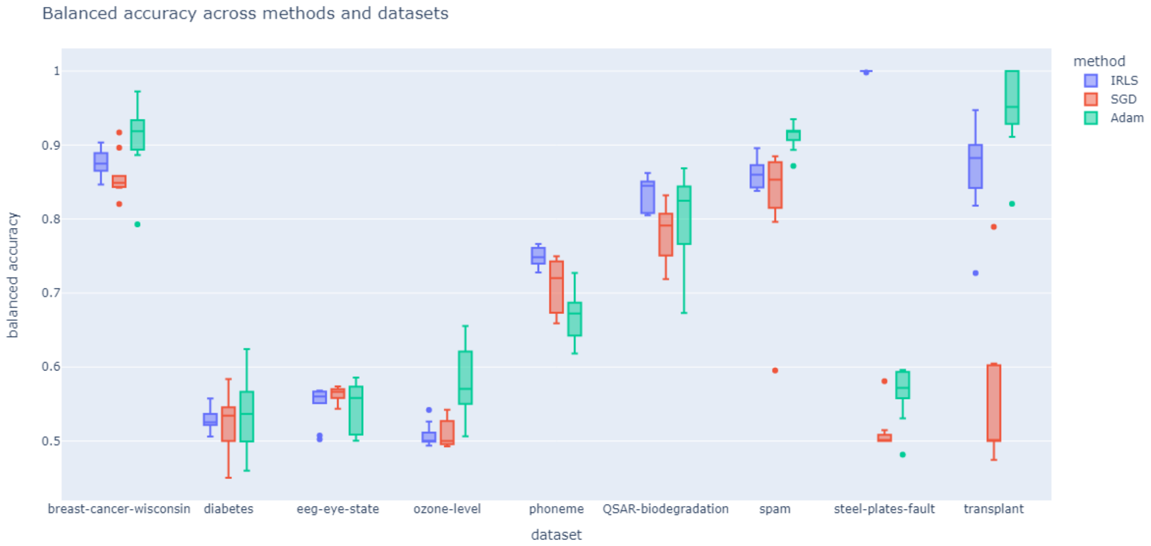


Figure 2: Distribution of balanced accuracy for each dataset, categorized by method.

2 Convergence analysis

We investigated the convergence of the implemented algorithms by analyzing how the log-likelihood function changes with the number of iterations. To obtain results, we included saving value of log loss at each iteration within the training function. Then, we converted these log loss values to log-likelihood by multiplying them by the negative of the number of samples. We plot the results for each dataset, to visualize the convergence of the algorithms, which is shown on the Figure 3. It is worth noting, that the y-axes in this case have different ranges, so the plots should not be directly compared to each other. They are only intended for comparing methods within the same dataset.

In the Figure 3a the log-likelihood plot for the Adam method starts at the lowest point and significantly increases in the initial iterations, surpassing the other two methods. The SGD method exhibits a stable increase in likelihood in each iteration, reaching values lower than those of the Adam method. IRLS stabilizes within the first few iterations, reaching a value lower than the other two methods.

As we can see in the Figure 3b, for the Phoneme dataset all three methods stabilize very quickly. The highest log-likelihood value is achieved by the Adam method, slightly lower for SGD, and IRLS performs the worst.

On the plot depicting the Pima Indians Diabetes dataset, all three methods stabilize quickly. Interestingly, in this case, the Figure 3c looks a bit different from the previous two datasets, since SGD achieved good results in the first few iterations, but then the values of log likelihood dropped significantly, to fluctuate in upwards trend. The highest log-likelihood values are achieved using the Adam method.

In the Figure 3d for the EEG Eye State dataset, all three methods exhibit significant instability and do not converge. This indicates that the optimization process fails to identify meaningful patterns or fit the model to the data effectively.

From the Figure 3e, which displays the results for the Breast Cancer Wisconsin dataset, it can be found that all of three methods converges quickly within the first fifty iterations. Adam method performs the best and SGD achieved its best result within around 100 iterations, after which it slightly drops and maintained a similar level.

For the Steel Plates Faults dataset in the Figure 3f, the IRLS method performs the best, achieving a likelihood close to 0 within the first few iterations. The other two methods diverge significantly in results. Although the likelihood for the Adam method is slightly higher than for SGD and shows a gentle increasing trend on the plot, both the Adam and SGD methods are highly unstable and do not converge.

For the QSAR Biodegradation dataset, from the Figure 3g, it is evident that the IRLS and Adam methods stabilize very quickly, achieving similar log-likelihood values, although lower than for the third method - IRLS, where this value surpasses them, showing an increasing trend in subsequent iterations.

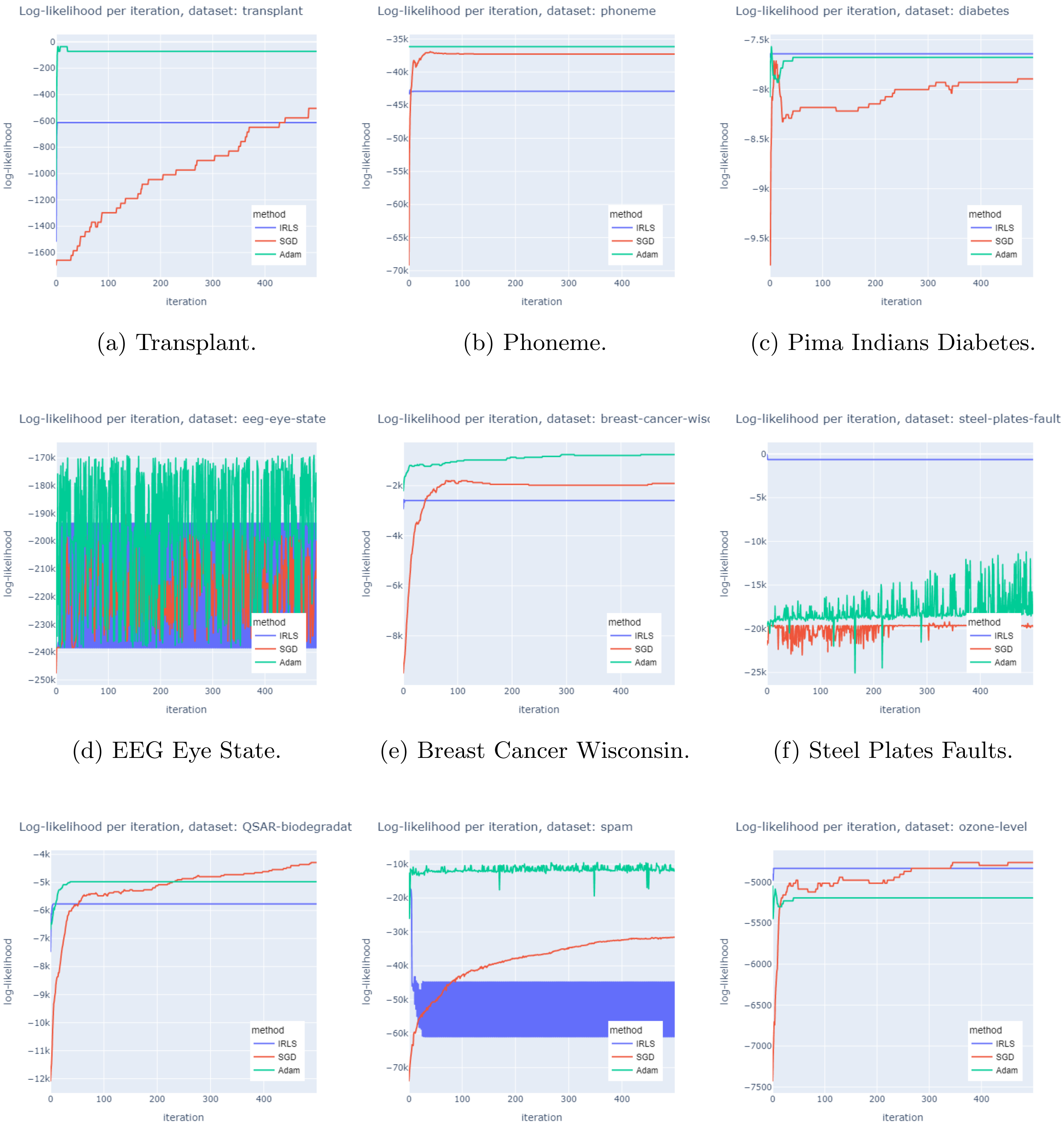
Looking at the Figure 3h, we observe that the Adam optimizer best fits the data, although it is slightly unstable. SGD converges with each successive iteration resembling a parabola, but ultimately achieves worse log likelihood than ADAM. In this case IRLS performs the worst, showing high instability for this dataset and do not converging.

For the last dataset, Ozone Level in the Figure 3i, the SGD achieves its best result, around -4500, after approximately 350 iterations. However, it is only slightly better than IRLS, which achieves a similar result but much faster. Adam also converges quickly, but then its log likelihood fall and stabilizes, resulting in the worst performance overall.

The following Figure 3 lead to the conclusion that all three methods exhibit different behaviors depending on the specifics of the dataset, and it cannot be unequivocally stated which method performs best. However, several conclusions can be drawn from the plots:

- The SGD method typically requires significantly more iterations than the other two for the log-likelihood value to begin converging, often up to 400 iterations,

- The IRLS method starts converging in almost all cases after just a few initial iterations.



(a) Transplant.

(b) Phoneme.

(c) Pima Indians Diabetes.

(d) EEG Eye State.

(e) Breast Cancer Wisconsin.

(f) Steel Plates Faults.

(g) QSAR biodegradation.

(h) SPAM E-mail.

(i) Ozone Level.

Figure 3: Log-likelihood per iteration across methods for each dataset.

3 Comparison of classification performance of methods and popular classifiers

The next task was to compare the results obtained using our implemented methods: IRLS (Iterative Reweighted Least Squares), SGD (Stochastic Gradient Descent) and Adam (Adaptive Moment estimation), with other popular classification algorithms, namely: LDA (Linear Discriminant analysis), QDA (Quadratic Discriminant Analysis), Decision Tree and Random Forest.

To make the comparison relevant, the predictions were made 10 times per model, on each dataset. The results of this analysis are shown in boxplots on the Figure 4.

As can be seen in the Figure 4a, showing the distributions of the Transplant dataset, the balanced accuracy results for our methods are much worse than for the other popular classifiers. For the decision tree and random forest, they are even as high as 1, when for our methods Adam has the best result, with a maximum of about 0.8. It is also worth noting how for different splits of the data the results varied when applying the SGD method: from 0.27 to almost 0.8 with an average around 0.5.

As for the previous one, for the Phoneme dataset in Figure 4b, popular classifiers outperformed the methods we implemented, except for LDA, which proved to get worse result. However, our results do not deviate significantly, remaining at around 0.7.

In the case of dataset Pima Indians Diabetes in the Figure 4c, our methods, similarly as before, have a little worse results than the other 4 popular classifiers. In the case of this dataset, however, none of the methods are good, as no results are better than 0.65. Adam's method predicts even worse than random guessing.

For the EEG Eye State dataset, shown in the Figure 4d, Decision Tree and Random forest stand out as better results, while the others remain at a similar level, slightly better than 0.5.

For dataset Breast Cancer Wisconsin, depicted in the Figure 4e, the outcomes of all classifiers are consistently close to a value of 0.9. Adam's results are quite stable and simmilar to those of Random Forest. Conversely, SGD has the worst and the most dispersed outcomes.

The results in the Figure 4f, so for dataset Steel Plates Faults, look really interesting. Here, the IRLS, Decision Tree and Random Forest are as high as 1, while the others have a balanced accuracy of only slightly more than 0.5, and QDA even exactly 0.5.

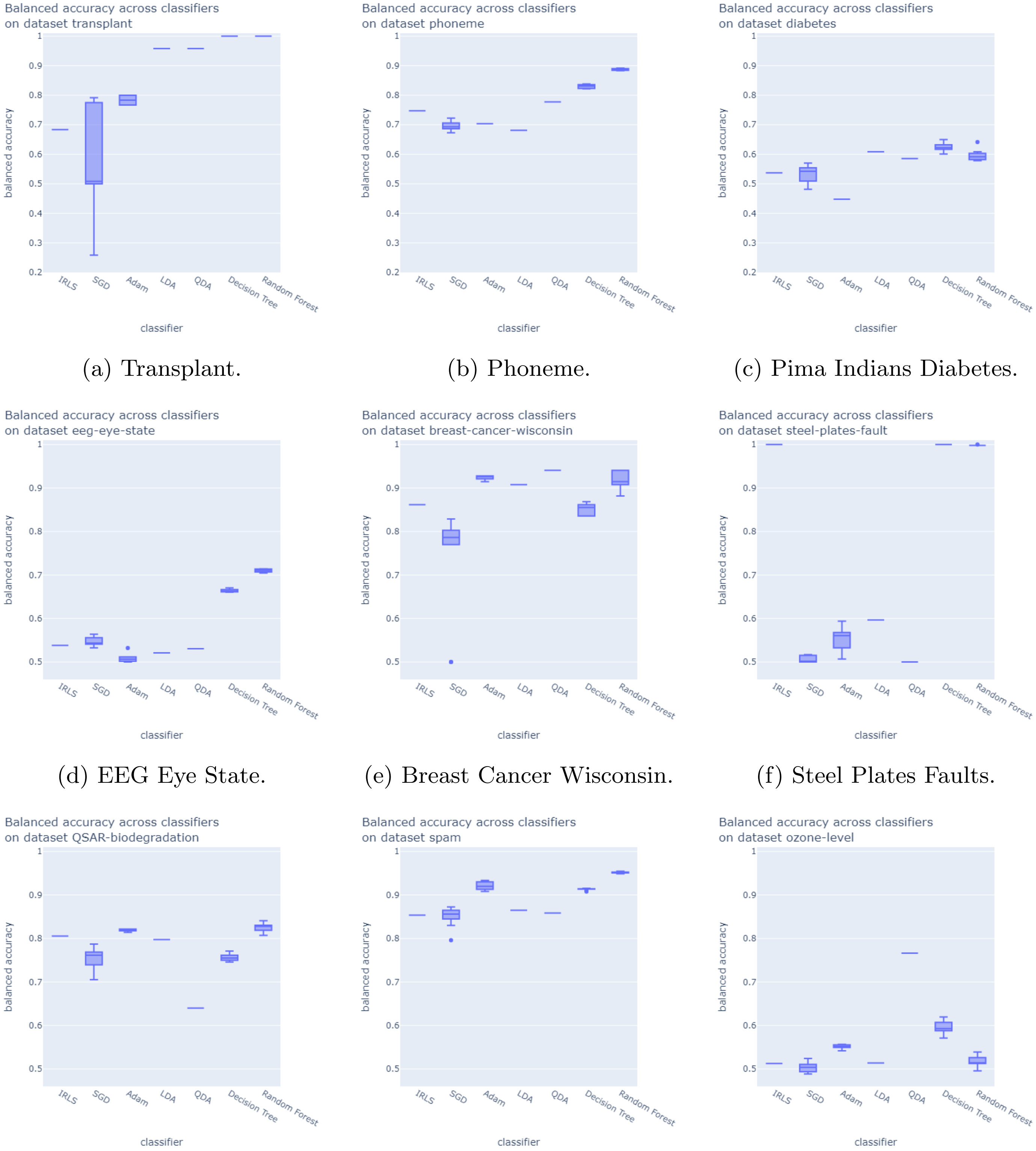
For the QSAR biodegradation dataset, in the Figure 4g, the best results of the balanced accuracy metric are for Random Forest and only a bit worse for Adam, both just over 0.8. IRLS and LDA also attained results around this value and Decision Tree and SGD performed slightly worse but still were comparable. However, QDA stood out as the worst performer, with a balanced accuracy of about 0.64, significantly lower than the others.

In the Spam E-mail dataset, in the Figure 4h, the results are generally satisfactory, with all models achieving balanced accuracy scores between 0.8 and 0.96. Random Forest performs best, followed closely by Adam, and comparably with Decision Tree. Conversely, IRLS, SGD, LDA as well as QDA, achieved nearly identical balanced accuracy scores, hovering around 0.86, with SGD showing the highest variability.

In the Figure labeled as number 4i, the results for the Ozone Level dataset are shown. In this case, all models, except for QDA, produced unsatisfactory results, so balanced accuracy below 0.63. SGD, in some cases, even performed worse than random guessing. Slightly better than others but still poorly performing is the Decision Tree with a result of 0.6. The lower score of the Random Forest may indicate overfitting. QDA stands out with significantly better results, achieving a balanced accuracy of approximately 0.77.

In this experiment, models for each dataset take the same parameters. For most of the datasets parameters tuning improve the results. The greatest impact has learning rate. Default values used in this experiment is 0.0001 for SGD and 0.02 for Adam.

In summary, there is no one standout model among. Depending on the dataset different ones performed the best. Overall, in compare to other implementations IRLS, SGD and Adam had similar or quite worse results.



(a) Transplant.

(b) Phoneme.

(c) Pima Indians Diabetes.

(d) EEG Eye State.

(e) Breast Cancer Wisconsin.

(f) Steel Plates Faults.

(g) QSAR biodegradation.

(h) SPAM E-mail.

(i) Ozone Level.

Figure 4: Balanced accuracy across classifiers for each dataset.

4 Comparison of classification performance of models with and without interactions

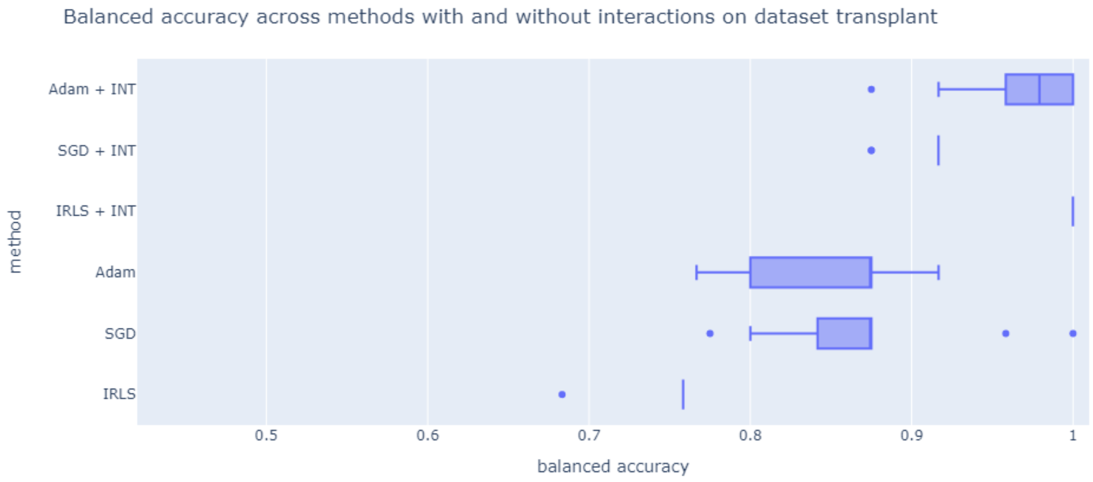
Implemented methods have option to add interactions between variables. This involves adding to the model not only the individual variables but also the product of each pair of variables. For example, as specified in the task: having variables X_1, X_2, X_3, the model with interactions is based on the variables: X_1, X_2, X_3, $X_1 \cdot X_2$, $X_1 \cdot X_3$, $X_2 \cdot X_3$. In this section, let's compare models with interactions to those without on sets with a small number of variables, i.e. on Transplant, Phoneme and Pima Indians Diabetes.

In the case of the former, shown in the Figure 5a, it can be seen that for each of the algorithms, the model with the interactions used performs better on average. Particularly noticeable is the difference in the case of IRLS, where the value of balanced accuracy before the addition of the interactions was about 0.76 and after their addition it is 1.

For Phoneme dataset, in the Figure 5b the results are in the range of 0.65 to 0.8, and also in this case for each method, its version with added interactions performs better o than the basic version by about 0.04. It is also interesting to note the low variability in the results for all algorithms.

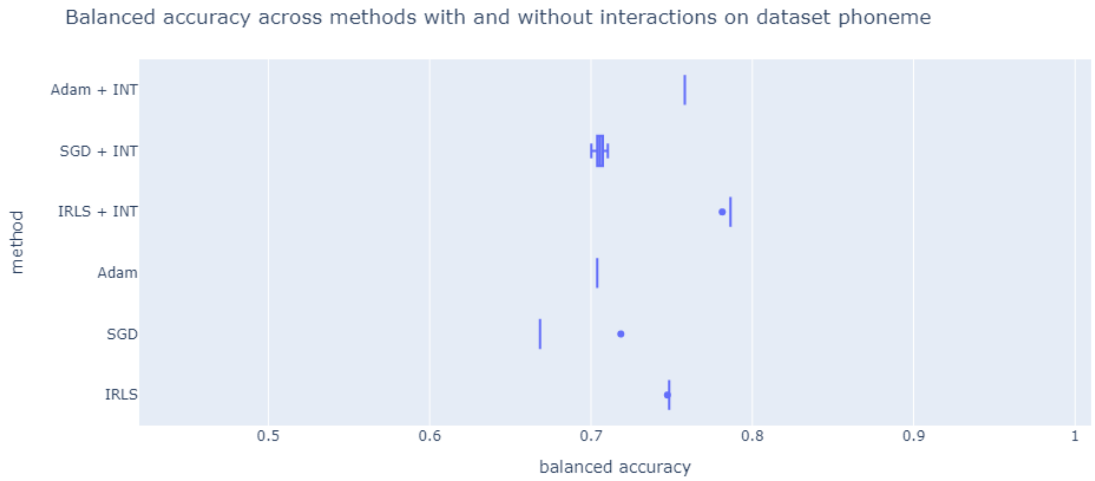
On the Pima Indians Diabetes dataset, shown in the Figure 5c, none of the methods managed to score better than 0.62, resulting in slightly worse outcomes overall. Specifically, the IRLS method and Adam show improved results when interactions are considered, similar to previous findings. However, it's important to note that Adam's improved results come with increased variability, unlike the consistent outcomes observed without interactions. Similarly, SGD also exhibits increased result variability, but unlike the other methods, its performance worsens when interactions are introduced.

Overall, in case of analysed small datasets models with interaction usually perform better. It is a bit surprising, because they introduce collinearity into the model. However, it also adds some desired features that ultimately improve the results. Potential reason for this improvement could be adding more complex and non-lienear relationships in the data.

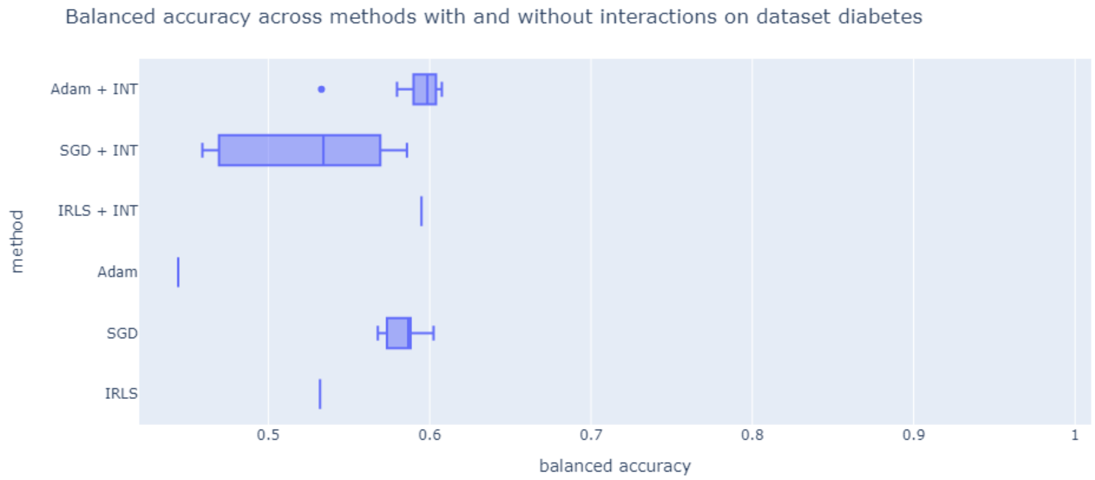


(a) Transplant.

Figure 5: Log-likelihood per iteration across methods for each dataset.



(b) Phoneme.



(c) Pima Indians Diabetes.

Figure 5: Log-likelihood per iteration across methods for each dataset.