Machine Learning: Project 1

Olivier Stähli
olivier.staehli@epfl.ch

Ivan Bioli
ivan.bioli@epfl.ch

Fabio Matti
fabio.matti@epfl.ch

Abstract

We implement six standard regressors. Subsequently, the Higgs boson data set for binary classification from 30 numerical features is used for testing our implementations. We perform various preprocessing techniques and tune the parameters of our regressors to maximize accuracy, as first priority, and F1-score of the predictions, as second priority. Additionally, an optimized regressor for this specific task is developed.

1 Introduction

The features in the data set refer to numerical measurements coinciding with the observation of a decay signature, either caused by an event involving a Higgs boson ('s' for signal) or not related to Higgs bosons ('b' for background) [1]. Based on 250'000 training samples, our goal is to predict the unknown labels for the test data.

2 Models and Methods

2.1 Preprocessing

In a first step, we visualized and analyzed the data. We modified the features that showed undefined values, which were set by default to -999 [1]. We tried replacing these undefined values with zero, the mean value or the median value of the feature (computed ignoring the undefined values). Among these approaches, the latter proved to be the most effective solution. We also tried adding a binary variable indicating that the corresponding entry of the feature was undefined, which did not prove to be effective.

Subsequently, we replaced outliers, i.e. values that deviate from the mean value more than three times the standard deviation according to the Grubbs' definition [4], by the mean value of the feature.

The feature "PRI_jet_num" is a discrete variable. Therefore, for the standard regressors, we tried adding a binary feature for each number of jets. However, this approach did not lead to any improvement in accuracy. Our optimized regressor exploits the "PRI_jet_num" feature to train a model for each number of jets. For a more detailed description of our approach, see Section 2.2.1.

As a final preprocessing step, we augmented the features using a polynomial basis of degree D with or without offset, and standardized the features by subtracting the mean value and dividing by the standard deviation.

The improvements to which our preprocessing leads are shown in Figure 1.

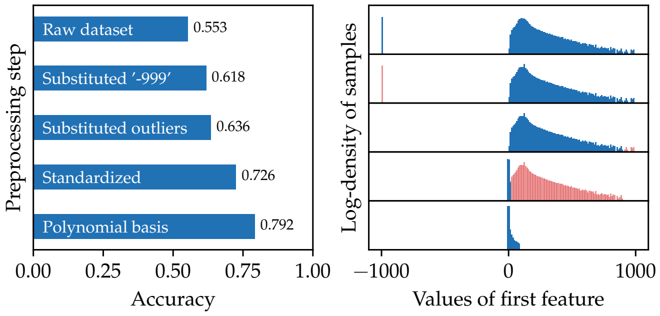


Figure 1 – On the right, the effect every preprocessing step has on the distribution of the values (here for the first feature). On the left, the progression of the mean accuracy in a 4-fold cross validation on the training set with regularized logistic regression.

2.2 Implementations

We implemented the six proposed regressors in Python, based on their description in [6]. Our implementations solely rely on the NumPy 1.21.1 library. The source code is available on our Git repository.

We paid particular attention to the following aspects:

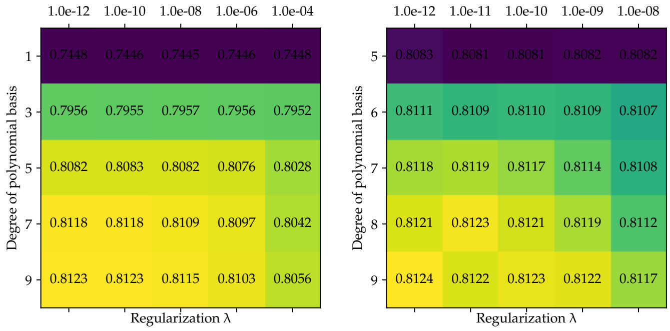
- Robustness: We implemented the regressors so that they can process any NumPy ndarray object as data matrix by diligent reshaping of arrays. Furthermore, we included "exp-guards" that prevent exponential functions from overflowing.
- Simple minimal working example: Each of our regressors may be fitted by just processing the training data without any other parameters.

2.2.1 Optimized regressor

The detailed description of the features in the dataset documentation [1], highlights that some features are systematically undefined for some values of the "PRI_jet_num" feature. For our optimized regressor, we therefore split our training set according to the four different jet numbers and trained four separate models. After splitting, the following operations were separately performed on each of the reduced training sets. Constant columns in the reduced sets, like "PRI_jet_num" and those with only undefined values, were removed. The preprocessing was then carried out in the same way as described in Section 2.1. To train the four separate models, we used as a base regressor one of the regressors proposed in the project description. Finally, we generated predictions for each number of jets separately.

2.3 Parameter tuning

We tuned our regressors through a 4-fold cross validation on the training set. Since we were dealing with a classification problem, instead of using a loss function, we used the accuracy as an indicator of how good our predictions are. The achieved accuracy is visualized through a grid-plot (see Figure 2). This type of visualization allows us extracting suitable parameters for our regressors. It also serves as an indicator of how stable the parameters are. If next to our ideal parameter, the accuracy falls off immediately, we avoid choosing this one, because of the uncertainty of whether this parameter will fail on the test set or not.



(a) Coarse grid search.

(b) Finer grid search.

Figure 2 – Visualization used for tuning the parameters. Here, the ridge regressor was used in a 4-fold cross-validation. The accuracy is plotted on a grid for various choices of the regularization parameter λ and the degree of the polynomial basis. Once a coarse grid search is done, a finer grid search is performed to find the parameters.

The best set of parameters for each regressor can be found in Table 1.

3 Results

The F1-score and accuracy our regressors were able to achieve are provided in Table 2. For our optimized regressor, as a base regressor we use the best linear regressor, i.e. ridge regression, and both logistic regressors.

Table 1 – Ideal parameters for the regressors found by means of grid-search and 4-fold cross-validation. D is the degree of the polynomial basis, λ is the regularization parameter, N is the maximum number of iterations, and γ is the learning rate. If used, the initial parameter vector $\boldsymbol{w}_0$ was always set to zero.

Regressor	D	λ	N	γ
least_squares_GD	4	-	200	0.07
least_squares_SGD	4	-	1e+5	5e-4
least_squares	8	-	-	-
ridge_regression	8	1e-11	-	-
logistic_regression	4	-	500	2e-6
reg_logistic_regression	4	1e-4	500	2.5e-6
optimized_ridge	9	1e-13	-	-
optimized_logistic	5	-	500	1e-5
optimized_reg_logistic	8	1e-9	500	6.3e-6

Table 2 – Prediction performances achieved with the regressors on the parameters specified in 1. The regressor's names are clickable links to the respective submission.

Regressor	F1-score	Accuracy
least_squares_GD	0.666	0.789
least_squares_SGD	0.626	0.773
least_squares	0.711	0.814
ridge_regression	0.711	0.814
logistic_regression	0.679	0.805
reg_logistic_regression	0.699	0.806
optimized_ridge	0.731	0.826
optimized_logistic	0.711	0.812
optimized_reg_logistic	0.718	0.817

4 Discussion

Following the discussion in [6], we expected linear models not to be particularly suitable to classification problems, such as the present one. However, among the proposed regressors, the ridge regressor reaches a higher accuracy with respect to the logistic and the regularized logistic regressors. This might be due to the fact that the dataset is not too unbalanced and we remove extreme values through preprocessing. Furthermore, our optimized regressor improved the accuracy and the F1-score by more than 1% and, once again, the adoption of the ridge regressor as a base regressor leads to better results with respect to the use of logistic regressors.

References

[1] C. Adam-Bourdariosa et al. *Learning to discover: the Higgs boson machine learning challenge*. 2014. URL: https://higgsml.lal.in2p3.fr/files/2014/04/documentation_v1.8.pdf.

[2] J. Brownlee. *Discover Feature Engineering, How to Engineer Features and How to Get Good at It*. 2015. URL: https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/.

[3] P. Domingos. "A Few Useful Things to Know about Machine Learning". In: *Commun. ACM* 55.10 (Oct. 2012), pp. 78–87. ISSN: 0001-0782. DOI: 10.1145/2347736.2347755. URL: https://doi.org/10.1145/2347736.2347755.

[4] F. E. Grubbs. "Procedures for detecting outlying observations in samples". In: *Technometrics* 11 (1969), pp. 1–21.

[5] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer, 2009. URL: http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

[6] M. Jaggi, R. Urbanke, and M. E. Khan. *Machine Learning (CS-433): Lecture notes*. 2021. URL: https://github.com/epfml/ML_course.

[7] The Free Encyclopedia Wikipedia. *Exploratory data analysis*. 2021. URL: https://en.wikipedia.org/wiki/Exploratory_data_analysis.