Machine Learning (CS-433) - Project 1 Report

Titouan Brossy, Francesco Sala, Nicolò Viscusi
EPFL, Lausanne, Switzerland

***Abstract*—Supervised machine learning techniques can be used in binary classification tasks such as the Higgs boson machine learning challenge of CERN, which aims to identify possible Higgs boson events from indirect measurements. Feature engineering as well as linear and logistic regression models were used to predict with the highest possible accuracy the presence or absence of these particles. Furthermore, cross-validation was used on regularized models to find the optimal regularization constant and 3^{rd} degree polynomial augmentation was adopted. The implementation of such models attained an accuracy varying from 70 to 80%, and the highest accuracy was obtained through ridge regression.**

I. Introduction

In this paper, machine learning techniques were applied to a data-set of physical values obtained by scientists at CERN, to learn a model that could predict the presence of the Higgs boson during an experiment. In particular, the models used are the least squares regression with gradient descent, stochastic gradient descent and normal equations, as well as ridge regression. Finally, two proper methods for classification were implemented: logistic regression and its regularized version.

II. Models and methods

A. Data loading

The training and test datasets, available in CSV format, were downloaded from [4]. The former was made up of 250000 events, the latter of 568238. This set was used only as further validation through submissions of the predictions on aicrowd.com. Data were then loaded into a `pandas.Dataframe`, suitable for data visualization, and 3 `numpy.array` variables, one for the ids, one for the predictions and the last one for the features.

B. Data visualization

In order to visualize the features and their respective distribution, histograms and boxplots of the variables were plotted. Boxplots are of particular interest because they allow to identify potential outliers appearing in the distribution. The dataset was further explored by creating scatter plots of 2 given features and selecting the color of the dots according to the type of signal.

C. Data processing

Steps of data processing were applied after visualization of the 30 features. The first step of data processing involved the identification of potential categorical or discrete features and the application of one-hot encoding [2] (in the dataset, only the feature PRI-JET-NUM is discrete). Subsequently, outliers values were identified[1] and replaced with the median of the feature distribution [3], since outliers might increase the variance of the data and thus reduce the statistical power of the models; additionally, a binary variable representing samples containing the outliers for each feature was created, under the hypothesis that outliers might also be significant for the type of event. Likewise, undefined values (equal to -999) were not removed but replaced with the median, since undefined values in the dataset are of type MNAR (Missing data Not At Random) [1]; the undefinedness depends on the value of PRI-JET-NUM (see [5] for more information). Before normalizing the continuous features, the processed data underwent polynomial of degree 3 feature expansion, under the assumption that the decision boundaries of this problem were not linear.

D. Model selection

Before training the 6 models, the 250000 events dataset `train.csv` was split into a training and a test set. The test dataset was used for estimating the generalization error defined by the cost function used by the model: accuracy[2] of the predictions on the test dataset was also assessed. For models requiring hyperparameter tuning (ridge regression and regularized logistic regression), before training each model, 4-fold cross-validation was performed on the training dataset to find the best hyperparameter.

1) Least Squares with Gradient Descent: the first method taken into consideration tried to minimize the loss function (i.e. the mean-squared error in the case under consideration) via gradient descent algorithm. At each step, the gradient is computed and one step is taken in the opposite direction of the gradient: $\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \gamma \boldsymbol{\nabla} \mathcal{L}(\boldsymbol{w}^{(t)})$.

2) Least Squares with Stochastic Gradient Descent: the basic idea of the algorithm is the same as the previous one. However, the gradient of only one component of $\mathcal{L}$, $\boldsymbol{\nabla} \mathcal{L}_n(\boldsymbol{w}^{(t)})$ (*mini-batch-size* = 1), is computed. The update rule is the same.

3) Least Squares with normal equations: normal equations defined by the gradient of the mean-squared loss were solved to find the optimal weights for the regression model. Since the Gram's matrix could be singular, and hence not invertible,

[1] Outliers values were defined outside the intervall $[Q_1 - 3 \cdot IQR; Q_3 + 3 \cdot IQR]$, where Q_1, Q_3, IQR are the first quartile, the third quartile and the interquantile range respectively.

[2] Accuracy was computed as follows: $E_{x,y \in D}(1_{\eta(x) \neq y})$, where D is the test dataset and $\eta(x)$ is the model prediction: $\eta(x_n) = 1$ if $\boldsymbol{x}_n^T \boldsymbol{w} \geq 0.5$ for linear models and $\eta(x_n) = 1$ if $\sigma(x_n^T w) \geq 0.5$ for logistic models (w is the weight computed with the models) .

the command `numpy.linalg.lstsq` was used to find the least-squares solution to the linear system.

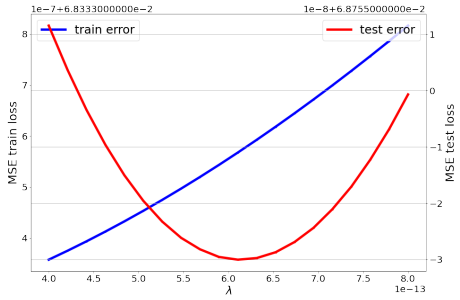
4) Ridge regression: in this case, a regularization term $\lambda||\boldsymbol{w}||_2^2$ was added to the normal equations, and the optimal weights were computed as a closed-form solution of the linear system (the regularization prevents the system matrix from being singular).

5) Logistic regression: this model tries to minimize the negative log likelihood defined by the sigmoid function via gradient descent: $P(y = 1 \,|\, \boldsymbol{x}_n, \boldsymbol{w}) = \sigma(\boldsymbol{x}_n^T \boldsymbol{w}) = (1 + \exp\left[-(\boldsymbol{x}_n^T \boldsymbol{w})\right])^{-1}$.

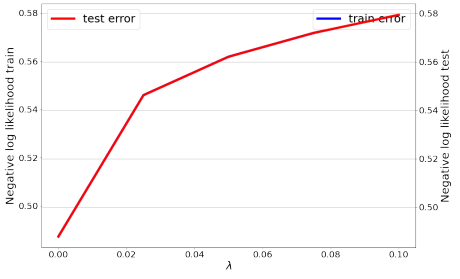
6) Regularized logistic regression: this model as well tries to minimize the loss or negative log likelihood defined by the sigmoid function via gradient descent, but in this case the loss is penalized by a regularization term $\lambda||\boldsymbol{w}||_2^2$.

III. Results

Figure 1 shows the cross validation test and training error for the hyperparameter λ for the two regularized models. Histograms and boxplots of the features are shown in figure 2 in Appendix A. Table I summarizes the selected parameters (λ_{opt} for regularization, γ step-size for gradient descent, `max iters` the maximum number of iterations allowed), the train and test loss, and the accuracy on the test set for each model.



(a) Ridge regression with normal equations. There exists a (local) minimum for the test error: the corresponding value of λ has been used to find the optimal weights



(b) Regularized logistic regression via gradient descent. Train and test error show the same pattern and differ only minimally in their values. The minimum is reached when there is no regularization (λ = 0)

Fig. 1: Average train and test error as a function of λ during cross-validation.

Model	Degrees of freedom	Training loss	Test loss	Accuracy
Least squares GD	`max iters` $= 1e4$, $\gamma = 1e-2$	0.0807	0.0813	0.7606
Least squares SGD	`max iters` $= 1e3$, $\gamma = 1e-4$	0.171	0.173	0.654
Least squares NE	-	0.0684	0.0695	0.815
Ridge regression	$\lambda_{opt} = 6.1e-13$	0.0684	0.0695	0.815
Logistic regression	`max iters` $= 1e4$, γ = 0.2	0.464	0.4673	0.774
Reg log regression	`max iters` $= 1e3$, $\gamma = 0.2$ = , $\lambda_{opt} = 0$	0.487	0.491	0.757

TABLE I: Training loss, test loss and accuracy for each method, and values of parameters used

IV. Discussion

First of all, it must be highlighted that the given problem is a classification problem: the possible values of the output are discrete (either "*background noise*" or "*signal*"). Consequently, the regression methods (methods in sections II-D1, II-D2, II-D3, II-D4) are not suitable for the given task. This is due to multiple reasons: first of all, the training set has an unbalanced number of points in each class (only 34.3% ca of data points are associated with a *signal* output). Additionally, while the MSE[3] cost function tries to minimize for each data point the distance between the model and the data, the aim of a classifier should be to minimize the number of wrong predictions.

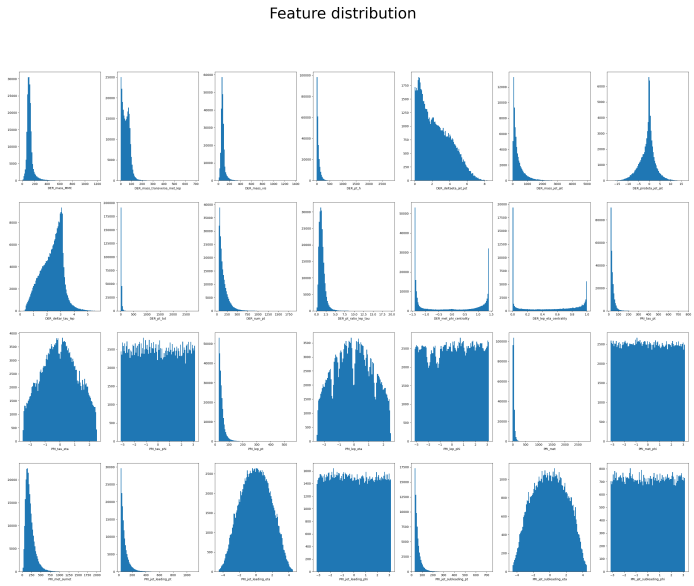
Table I shows the value of training and test loss for each method, and their accuracy computed on the part of the `train.csv` data not used for training. Interestingly, it can be noted that, despite being a regressor, the method with the highest accuracy is the closed form solution of the least squares problem. Its regularized version performs equally well: the optimal λ value is negligible (comparable to the $\varepsilon_{machine}$ of the floating point representation used), and hence the $\boldsymbol{w}$ found is the same as the one of the previous method. The models with the highest losses are the logistic regression and its regularized version: this is due to the fact that the loss of these method is different from the MSE loss, therefore the values obtained for the first four methods are not comparable with the last two. As in the case of the ridge regression, even the logistic regression failed to yield a more accurate solution than its non-regularized version: $\lambda = 0$ was the best hyperparameter found.

V. Summary

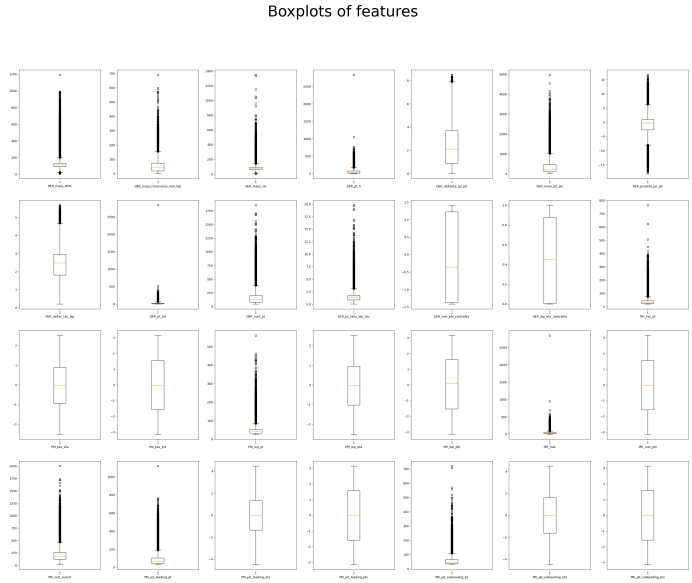
The model with the highest accuracy is the closed form solution of the least squares problem. That said, with a higher computational power, the logistic regression could be run again allowing a higher number of maximum iterations: this may yield a better accuracy than the one attained by the least squares models. We arbitrarily attempted a 3^{rd} degree feature expansion: further work could focus on tuning the hyperparameter d, the polynomial degree, for each method. Additionally, regularization did not contribute to reduce the test error for the penalized logistic regression; this is probably due to a too simple model. Further data processing or feature expansion could have been performed to make the model more complex. On the whole, the accuracy values obtained can validate the adopted procedure.

[3]Mean Square Error.

APPENDIX A
FIGURES FOR DATA PROCESSING



(a) Histograms of features with bins of 100 samples. It is worth mentioning that pseudorapidity variables (`eta`) among the primitive features have a very symmetric distribution. On the other hand, azimuth angles (`phi`) and momenta (`pt`) show a uniform and very sharp left skewed distribution respectively. The derived features can show symmetric distribution such as the centrality features, but most of them follow a left skewed distribution.



(b) Boxplots of features. It is interesting to see that features with symmetric and uniform distribution show no outlier. On the contrary, all other variables show some outliers in their distribution.

Fig. 2: Visualization of continuous primitive and derived features. Undefined values were not included.

References

[1] T. Makaba and E. Dogo, "A comparison of strategies for missing values in data on machine learning classification algorithms," in *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, Nov. 2019, pp. 1–7. DOI: 10.1109/IMITEC45504.2019.9015889.

[2] J. Brownlee. "Ordinal and one-hot encodings for categorical data," Machine Learning Mastery. (Jun. 11, 2020), [Online]. Available: https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/ (visited on 10/22/2022).

[3] A. Sahu. "How to handle outliers in machine learning," Analytics Vidhya. (Apr. 8, 2021), [Online]. Available: https://medium.com/analytics-vidhya/how-to-handle-outliers-in-machine-learning-5d8105c708e5 (visited on 10/22/2022).

[4] *Epfl machine learning higgs: Challenges*. [Online]. Available: https://www.aicrowd.com/challenges/epfl-machine-learning-higgs.

[5] *Learning to discover: The higgs boson machine learning challenge*. [Online]. Available: https://higgsml.lal.in2p3.fr/files/2014/04/documentation_v1.8.pdf.