Higgs Boson: EPFL Machine Learning Challenge

Luca Rossi, Federico Betti, Ioannis Mavrothalassitis
CS-433 Machine Learning, EPFL Lausanne, Switzerland

***Abstract*—In this paper, we present our approach to the Higgs Boson Machine Learning Challenge which has been carried out by CERN in Geneve. The challenge consists in finding the best model to predict the signatures of the decay of a Higgs boson resulting from protons-protons collisions and to distinguish the latter from background events which are considered irrelevant. Hence this is a classification problem as the output is categorical $\big(y \in \{-1,1\}\big)$. We found the best predicting model to be an ensemble model made of ridge and regularized logistic regression.**

I. Introduction

The physical problem of interest is the detection of the decay signatures of the Higgs Boson after protons-protons collision as some of the kinetic energy is converted into new particles. The challenge consists in distinguishing with the best accuracy possible the signal events from the background analysis to then optimize the selection of the region of the space where the Higgs Boson is more likely to be observed.

II. Feature Pre-processing

The given training set had 250.000 training samples and 30 features which were all representing physical quantities measured directly by a detector or derived from them. The training outputs were then denoted as *s* for a signal event (i.e. Higgs boson found) or *b* for background events. If the event is marked as background, the probability of observing the Higgs Boson is very small.

A. Splitting the data:

First of all, we performed some **exploratory data analysis** in order to understand the distribution of each feature for each of the two labels. The value $-999.$ was a common value in the dataset which was always out of the range of all other calculated quantities; therefore, this value was considered as a NaN value. After reading carefully the paper, we noticed that entire columns of features were missing if the number of Pri-Jet-Num (the 23-th column of the dataset) was equal to zero, some of them were missing if this number was equal to one and none of them were missing if the Pri-Jet-Num was equal to two or three. Therefore we decided to **split the training set** (and then the test set) into 3 clusters **depending on the value of the number of jets** (0, 1, and 2 or 3). This was convenient not only because entire columns of null values could be dropped in the 0 and the 1 cluster (see next section) but this allowed us to create three different and more specific models which could be trained separately driving us to improve our final accuracy on the test set. In the end we decided to further split also the 2 and 3 cases. Indeed the optimal models for these two data subsets were not necessarily similar and the percentages of signal events corresponding to Pri-Jet-Num2 and Pri-Jet-Num3 was really different in the training outputs; this further split increased the accuracy by an average of 2%.

B. Null values:

As we described in the previous section, the training and test set contained a large number of null values denoted by -999. Since keeping these values was clearly compromising the accuracy of our model we decided to treat them opportunely. For each cluster, if an entire column of values was missing, we dropped such a column since it would not have given the model any valuable information; if not all the values in a column were -999., **we replaced the missing values with the median** of the real physical values of the column of interest, as the median is more robust to outliers with respect to the mean. Clearly the same process of feature engineering was done in the test set, namely we dropped the same columns as in the training sub-matrices and substituted the null values with the median of the column of interest (obviously the median of the corresponding column in the training matrix).

C. Outliers:

We decided to treat the outliers as follows: for each feature in each cluster we considered as outliers the values of that column which were $\leq \mathbf{Q15} - \mathbf{1.5IQR}$ and $\geq \mathbf{Q85} + \mathbf{1.5IQR}$, where $Q15$ denotes the 15% quartile, $Q85$ denotes the 85% quartile and IQR is the 75-25 interquartile. We then **replaced them with the median** of the column vector calculated on the physical values. This manipulation of the outliers increased our accuracy in the cross-validation testing by another 2% in average over the four clusters.

D. Mass detection:

When the mass of the Higgs Boson candidate was not calculated by the detector, only 1.134% of the events were then declared in the output to be coming from the decay signatures of a Higgs Boson. This led us to add an **additional feature** which was 1 **if the mass was detected and** 0 **otherwise**. This improved our performances by an average of 7% over the four clusters.

E. Standardization and bias term

As in typical fashion of the feature Pre-processing, we then continued by **standardizing the training data and the testing data "cluster-wise"**, where obviously for the latter we used the mean and standard deviation of the training, by a linear transformation of the form $X = \frac{X-\mu}{\sigma}$. We then added both in

the training and in the testing matrices a column for the **bias term** in order not to restrict ourselves to optimal decision-boundaries passing necessarily through the origin.

III. Model selection and validation

We tried training our model on the usual regression and classification models which were seen in class, namely:

- (Stochastic) gradient descent and least squares
- Ridge regression
- Logistic regression and regularized logistic regression

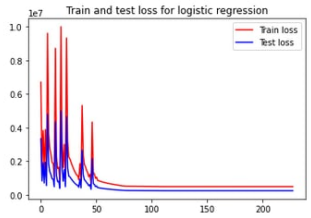
For the regularized logistic regression algorithm, we decided to **decrease the learning rate** every *iter-drop* iterations by means of $\gamma_{iter} = \gamma_0 * \text{drop}^{\frac{1+\text{iter}}{\text{iter}-\text{drop}}}$ where γ_0 is the initial step size and $drop = 0.5$. This made the descent algorithms converge way faster.
In order to introduce non-linearity in the model we decided to perform **polynomial feature expansion** and to choose the best degree by means of 4-fold **cross validation** on the degree itself and on λ when there was a regularization term, by then picking the values that were maximizing the accuracy on the testing folds.
It was clear from the beginning that it was better to **focus our attention on ridge regression and regularized logistic regression** as the other models were in general performing worse. The following table shows the best results (calculated in cross validation) for the other models, where the accuracy value is a weighted average of the accuracies over the four clusters:

	Accuracy
Gradient descent	0.8144
Stoch. Gradient descent	0.7562
Least squares	0.7742
Logistic regression	0.8163

At least initially we performed 4-fold cross validation on λ and on the degree also for the other two models. This was done using **randomized grid search on** λ (which usually performs better than the grid search). However, a further increase of the accuracy by an average of 1.5% over the four clusters was obtained by restricting our attention to degree 2 (which was always the optimal one in the previous tries) and performing **feature crosses** in addition to the standard polynomial expansion, while always doing cross-validation on λ. The results obtained in cross-validation accuracy are shown below:

	Accuracy
Ridge regression	0.8236
Reg. Logistic regression	0.8188

As this last two models were both performing well we decided to create an **ensemble predictor**: we generated three different predictions using three different models and we used a hard voting scheme to combine them in a unique prediction.



Above we display a plot comparing the train and the test loss for the regularized logistic regression for one of the clusters and for the optimal parameters. The behaviour of the two functions is similar so we are not overfitting and the regularization term works effectively. Unfortunately we cannot display the same plot for the ridge regression since we used the closed form solution (not an iterative algorithm) to compute the optimal weights.

IV. Results

The choice of the three predictors was tried on some combinations of regularized logistic and ridge regression models. The best result, which was obtained by doing an ensemble with two ridge regression models tuned with parameters found on different ranges (to assure diversity) and a regularized logistic regression model, resulted in 81.8% accuracy on the online test. The optimal parameter of λ for the three predictions are listed below:

Cluster	Ridge 1	Ridge 2	Logistic
0	1.0e-05	5.58e-06	7.6129
1	0.0045	3.56e-05	9.3815
2	0.0046	7.25e-05	6.4474
3	2.78e-05	3.97e-05	5.0851

V. Discussion and final remarks

In the feature pre-processing, the thing that improved majorly our models was the addition of the mass detection feature. Moreover, splitting the initial training data into clusters depending on the number of jets gave us the opportunity to train four different models more specific to each subset. A possible improvement, since many distributions of the features were heavy-tailed, could have been to try shrinking their support by applying feature-wise a logarithmic scaling of the features. Better results could have been achieved also by removing the outliers found by the criterions explained above instead of replacing them with the median, but we haven't considered this possibility as deleting outliers is normally a bad practice when they are frequent.
Concerning the choice of the models, contrarily to our initial belief, the performances of the Ridge Regression and the Regularized Logistic Regression were comparable and this led us to combine them in a unique predictor.
The cluster on which we had the most trouble was the one with Pri-Jet-Num = 1 as no model was increasing majorly the accuracy on this subset (barely 80%), while the accuracies on the other three clusters were around $83-84\%$. Probably a better insight about this cluster would have suggested a separate treatment to achieve a better performance.

REFERENCES

[1] Learning to discover: the Higgs boson Machine Learning challenge Claire Adam-Bourdarios ,Glen Cowan, Cécile Germain , Isabelle Guyon, Balàzs Kégl, David Rousseau, 2014

[2] James Bergstra, Yoshua Bengio Random Search for Hyper-Parameter Optimization
https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

[3] J.Brownlee "Discover feature engineering, how to engineer features and how to get good at it.", 2020,
https://machinelearningmastery.com/discover-featureengineering-how-to-engineer-features-and-how-to-get-good-atit/

[4] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, Applied logistic regression. John Wiley Sons, 2013, vol. 398.

[5] Shai Shalev-Shwartz, Shai Ben-David, Understanding Machine Learning, 2014 Cambridge University Press