CS-433 Machine Learning - Detecting the Higgs Boson by analyzing proton collisions

Eugenie Chabenat, Sara Djambazovska, Sepideh Mamooler
Department of Computer Science, EPFL, Switzerland

***Abstract*—Machine learning techniques have become a vital part of many research fields, allowing to make sense of complex multidimensional datasets. Our work explained in this report focuses on developing an algorithm for detecting the Higgs boson. A crucial part in finding traces of this particle lays in classifying and analyzing decay signatures of proton collisions provided by CERN. As part of this project, we trained six different regression and classification models and evaluated their efficacy of detecting the Higgs boson among the background noise.**

I. Introduction

The aim of this report is to compare several well-known machine learning methods for a recent machine learning challenge - finding the Higgs boson - using original data from CERN. Firstly, we explain how we pre-process the data. Subsequently, we explain the results we obtained by using the different regression and classification methods on the expanded dataset. Further on, we present our best results which were using Ridge regression for solving a classification problem. Finally, we conclude that we would obtain better results by using classification methods if we had higher computational power for fine-tuning and hybrid approaches.

II. Data Pre-Processing

A. Observations

The dataset is a subset of the simulated data used by CERN scientists to optimize the analysis of Higgs Boson. It is composed of 250 000 events and 30 features, which are kinematic properties measured by the detector and outcomes of functions applied on them. While studying the provided training data we observed that all features are continuous except for the PRI_jet_num which contains discrete values in $\{0, 1, 2, 3\}$. Moreover, data points corresponding to each of these categories share some characteristics. For instance, for all data points with $PRI\_jet\_num = 0$ $PRI\_jet\_leading\_phi$ is not provided, i.e are set to -999. We have summarized such observations in table II. As a result we decided to split the data into 4 datasets each corresponding to data points sharing the same PRI_jet_num, and model each of them separately. Table I shows the distribution of these categories in the data set.

PRI_jet_num	distribution
0	39.9615%
1	30.9925%
2	20.1665%
3	8.8795%

Table I

	jet 0	jet 1	jet 2	jet 3
PRI_jet_all	-	x	x	x
$PRI\_jet\_leading\_phi$	-	x	x	x
$PRI\_jet\_leading\_eta$	-	x	x	x
$PRI\_jet\_leading\_pt$	-	x	x	x
$PRI\_jet\_subleading\_phi$	-	-	x	x
$PRI\_jet\_subleading\_eta$	-	-	x	x
$PRI\_jet\_subleading\_pt$	-	-	x	x
$DER\_lep\_eta\_centrality$	-	-	x	x
$DER\_prodata\_jet\_jet$	-	-	x	x
$DER\_mass\_jet\_jet$	-	-	x	x
$DER\_deltaeta\_jet\_jet$	-	-	x	x

Table II
Common characteristics of data points with the same PRI_jet_num. For each features ’-’ means no data is provided and ’x’ represents the presence of data.

B. Data Cleaning

The dataset contains numerous meaningless values, which are signalled by -999. These correspond to feature values that are either missing from the data and we replace them by the median of their column, or do not make any sense for a specific jet category as shown in table II. These vanish by removing columns with zero variance. Moreover, we remove outliers by setting values out of range [$mean$-3std, $mean$+3std] to their closest bound. Another attempt was to remove co-linear columns, which was needed to be able to apply the least squares method, but did not improve the accuracy on our final submission with Ridge regression. We also standardize our data to have zero mean and unit variance for each feature.

C. Feature Expansion

We expand the features by adding the following transformations as well as polynomial expansions to the original features. The degree of the polynomial expansion is fine-tuned for each learning method using 5-fold cross-validation.

$$x \rightarrow cos(x)$$
$$x \rightarrow sin(x)$$
$$x_i, x_j \rightarrow x_i x_j$$

III. Applying all methods to original dataset

A. Least Squares

This model outperformed the others on the original non-split by PRI jet number dataset. Namely, after doing the data pre-processing mentioned above on the whole train and test dataset, we achieved a score of 80.3% on AIcrowd. However, using this method on the split dataset yielded worse results than Ridge Regression, leading us to the decision to choose that model for our final submission.

B. Least Squares GD and SGD

The best results we obtained for least squares GD adn SGD is summarized in table III.

method	gamma	max_iters	degree	accuracy
least squares GD	0.005	2000	1	77.05%
least squares SGD	0.1	1000	1	72.97%

Table III

We performed Least Squares SGD using a 4-fold cross validation in order to find the optimal degree and value of gamma to use for the accuracy to be the highest. We then tried different values of folds for cross validations to see if it could further improve the accuracy (up to 10 folds). But the highest accuracy was obtained with a 4-fold cross validation.

C. Logistic Regression

As this is a classification method, which predicts probabilities, our expectations were met when it's accuracy exceeded over the previous regression techniques, except for least squares (which does not need hyperparameter tuning). We performed 4-fold cross validation in order to find the optimal degree and gamma to increase accuracy. The training results with the optimal values, on the pre-processed dataset are shown on the table below.

method	gamma	lambda	max_iters	degree	accuracy
logistic regression	0.01	-	3000	2	77.59%
regularized LR	0.7	0.01	1000	2	77.26%

Table IV

However, after pre-processing the dataset by splitting it according to the different jets and using the optimized hyper-parameters obtained by cross validation, the yielded accuracy decreased to 67.9%. We realized that the computational power and time required to obtain better results using the jet separation method were not within our limits, encouraging us to chose Ridge Regression for our final submission.

D. Regularized Logistic Regression

This model was expected to outperform the others, as it predicts the probabilities of the labels and penalizes over fitting. In order to find the best hyper-parameters, we used 10 fold cross validation, and obtained the optimal values shown in table IV, However, upon using this model on each of the four split datasets, we obtained a lower accuracy than with Ridge Regression. While testing, we faced some overflow warnings due to the exponent calculation in the sigmoid function and the loss computation. We overcame them by allowing calculations with very small numbers exceeding the range of floating point numbers, by using the numpy $logaddexp$ function.

IV. Optimal method : Ridge Regression

Theoretically, as this is a classification task, we expect the classifications methods like logistic regression and regularized logistic regression to outperform other methods. However, due to our limited computation power and the iterative nature of these methods, we were not able to tune them for our partitioned data. Thus, we obtained the best accuracy by using ridge regression for the partitioned data and tuning the regularization parameter λ and the the polynomial expansion degree using 5-fold cross-validation. Table V shows these tuned hyper-parameters. We obtained 83.5% accuracy in the AIcrowd platform.

PRI_jet_num	lambda	degree
0	4.17×10^{-12}	6
1	5.73×10^{-8}	6
2	1.74×10^{-8}	6
3	1.37×10^{-11}	6

Table V

V. Conclusion

With our work on this project, we were able to practically apply the methods learned in our Machine Learning course and see the large impact that data prepossessing and correct feature engineering has on the final prediction accuracy. We believe that with with more computational power we would be able to use our partitioned dataset for iterative methods like regularized ridge regression and obtain more coherent results with the expectations we had based on our theoretical knowledge. Furthermore, higher computational power would allow for finer hyper-parameter tuning and thus, better performance. A future work can contain a hybrid approach which finds the methods resulting in highest accuracy for each partition.