Learning to discover: the Higgs boson machine learning challenge

Nino Hervé, Florent Jeanpetit, Margaux Roulet
Machine Learning, EPFL, Autumn 2019

***Abstract*—This project focuses on how Machine Learning techniques can be used to improve analysis of Higgs data. Regressions techniques are implemented to predict the data pattern of Higgs boson decay signature. Binary classification will be used to classify whether an event is related to Higgs boson signal or background noise. This report describes a working model optimized from basic machine learning methods. It includes exploratory data analysis, feature processing and engineering. Upon various implementations the chosen model is the ridge-regression method combined with a strong processing of the data discussed below.**

I. Introduction

The Higgs boson is an elementary particle in physics which explains why other particles have mass. Scientists proceed to collisions between protons at high speed to extract even smaller composants. The Higgs boson is potentially one of these by-products. However its life-span is so short that no experience succeed in measuring its activity. Thus, scientists rather study its "decay signature". Indeed particles collision delivers a huge amount of data that can be processed to find the nature of the by-product. Since it is difficult to distinguish specific decay signatures, a strong likelihood model is required. The aim of this project is to identify Higgs boson signature from backgrounds' data, i.e create a binary classifier based on actual CERN accelerator data.

II. Models and Methods

A. Basic Implementation

For this project, six regression functions were implemented (see Table 1). The six regression methods were used on the training dataset to figure out which method is likely to make the most accurate predictions. To this end, training set was divided into two sub datasets as if they were generated independently. 80% of the training dataset was used for learning and the remaining 20% of the training dataset was used as a validation set to test the accuracy of the predictions made.

Then, data was processed and for gradient descent and logistic methods standardized since they are sensitive to ill conditioning. For each method, optimal parameters were found by evaluating different values. The performance of each method was estimated by comparing the resulting accuracy scores (see Table I).

As expected, gradient descent and stochastic gradient descent linear regressions gave the lowest accuracy scores. These models are too limited to find a good fit of the data. Least square and ridge regression gave the second best scores. Adding the regularizer λdo not significantly improve the results here. Logistic regressions methods gave the highest accuracy score.

Methods	γ	λ	nb iter	score
Gradient descent	7e-04	-	1000	70.75%
Stoch gradient descent	0.003	-	3000	69.45%
Least squares	-	-	-	74.69%
Ridge regression	-	10e-05	-	74.69%
Logistic regression	10e-7	-	6000	75.13%
Reg logistic regression	10e-6	-	4000	75.09%

Table I: Methods implemented and their validation set accuracy score. Splitting ratio of training dataset set to 0.8

Based on the accuracy score, both logistic regression and ridge regression methods were selected. Difficulties were encountered when optimizing logistic regression using polynomial basis function. It took really too much time to run through all the data set. Furthermore, the score obtained with polynomial was significantly better than logistic (see Table III). This is why ridge regression method was finally chosen.

B. Data Pre-processing

Before testing any regression methods, it is imperative to carry out a pre-process of the data. The first part of this project consisted indeed in the exploration of the dataset in order to extract more relevant information from the data. Three major observations were made:

It can happen that for some entries, some variables are meaningless or cannot be computed; in this case, their value is -999.0, which is outside the normal range of all variables. Those nan values were replaced by the median computed over all datapoint of the specific features, or can be deleted. Better accuracy results are shown by replacing nan values with the median.

Standardization of data have been made since it increases the accuracy score.

There are many outliers in the dataset. These outliers are data examples that are far away from most of the other datapoints. They are therefore non-informative values and thus, brought back to the values of the upper or lower limits.

All variables are floating point, except PRI-jet-num which is integer. Jet is the number of pseudo particle that may appear in the detector when compounds collide. It comes out that certain floating variables are actually impacted by

this jet number. It is thus a categorical variable that allows the separation of the data into four sub datasets. There is now four subsets as shown in Table 2.

Following the splitting of the data according to the number of jet, it appears that some features have no variance. There are consequently removed since they do not give any valuable information.

Jets	nb train data	%	nb test data	%	nb features
jet 0	99913	40	227458	40	18
jet 1	77544	30	175338	30	22
jet 2	50379	20	114648	20	29
jet 3	22164	10	50794	10	29

Table II: Jet subsets and their respective number of features.

C. Feature Engineering

In order to increase the representational power of ridge regression model, the input dataset is augmented adding a polynomial basis so that one ends up with an extended feature vector. In fact, in quantum physics, features often follow non-linear and stochastic distributions.

Considering that interaction between features might occur, first order cross-terms of features is computed xi*xj. If two features are dependent, their interaction might indeed vary the predictions after considering each individual feature effects

The results obtained with different combinations of feature engineering are shown in Table III, using a tuned polynomial degree of 6.

poly 6	cross-terms	score
no	no	74.73%
yes	no	82.1%
yes	yes	83.1%

Table III: Ridge regression with poly 6 and cross-terms implemented on processed dataset.

Building the polynomial basis function considerably increased the accuracy score. This results confirms that the dataset follow a non-linear distribution. Adding the cross terms also considerably improves the accuracy score, meaning that features have significant interactions.

D. Model Selection

Optimal hyper parameters for ridge regression were found by grid search. The data were randomly splitted using 10-fold cross-validation to return an unbiased estimate of the loss. Using polynomial expansion, both the degree of the polynomial and the regularization parameter lambda were tuned. The cost was computed over all values of lambda and degree in a well defined ranged and the best hyper parameters which gave the smallest cost using cross-validation were chosen.

The degree of the polynomial of the subsets were limited to degree 6 even though the root means square error (rmse) was smaller for higher degrees. Indeed, the accuracy seems to converge. As shown in Figure I, after degree 6 the accuracy score's variance is negligible. Only jet1 showed a significant smaller rmse with a higher degree. Hence, degree 7 was chosen for this particular jet.

The hyper parameters found are shown in Table IV. These lambda tend to 0, hence accuracy results of least-squares methods should be similar. This is verified in Table IV.

Jets	degree	λ	rmse	Ridge score	LS score
jet 0	6	1e-12	0.0.679	85.55%	85.5%
jet 1	7	7.443e-10	0.751	81.44%	81.1%
jet 2	6	1.125e-10	0.706	83.76%	84.0%
jet 3	6	1.701e-11	0.707	84.31%	84.9%

Table IV: Ridge regression optimal hyperparameters and respective loss (rmse) of each jet.

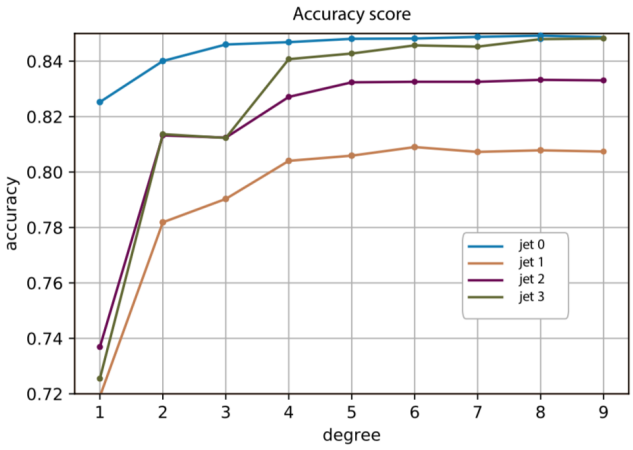

Figure 1: Jets accuracy score with respect to polynomial degree. Each jet feature was previously processed and optimal λwas found for each degree.

III. Results

A final score of 83.7% was obtained on Aicrowd. The associated python code is visible on run.py.

IV. Discussion

This project showed that any Machine Learning issue should be considered under various aspects. The data preprocessing plays a surprising and key-role and further explorations should be made to improve the proposed model. Especially, a better understanding on how DER variables were derived from PRI raw variables should be a start. Then, it would be possible to add functions that seem to better describe the model behavior and to add meaningful higher order cross-terms for specific feature.