Higgs Classification Using ML methods

Yiyang Feng, Naisong Zhou, Yuheng Lu
Project 1, CS433 Machine Learning, EPFL

***Abstract*—Machine learning has gain more application as new models are proposed. In this essay, we applied machine learning techniques on CERN's Higgs Boson Dataset. With noise distributed in data, we took different pre-processing methods for better performance and compared several models on this task.**

I. Introduction

The Higgs boson is an elementary particle which is produced by the quantum excitation of the Higgs field. It is extremely unstable, decaying into other particles almost immediately. The dataset we use is built from official ATLAS full-detector simulation on particle physics. The measurement follows basic laws in quantum physics yet involve background noise. The aim of this essay is to eliminate background noise of raw measurement data and then construct a reliable classifier. To do this, we applied different processing and regression techniques on the dataset and examined their efficiency.

II. Data Pre-processing

Several observations lead us to think about corresponding processing directions. With pre-processing, our method manage to eliminate prominent errors in the raw data, improving accuracy.

A. Categorical feature 'PRI_jet_num'

We observed that, despite all the other features which are in float type, there is a feature 'PRI_jet_num' which in integer type, with values restricted to 0,1,2,3. A jet is a hadron which will cluster together being produced in a particle collision.'PRI_jet_num' represents the number of jets in a collision event. According to experimental settings, some other features are describing features of the jets like jets traverse momentum, etc. It is also stated that possible larger values have been capped at events with 'PRI_jet_num' 3. This has made us think about grouping data by this feature into 4 groups. Each group have size 99913, 77544, 50379, 22164 respectively. Due to the sum of group 2,3 size is similar to group 0,1 each and feature dimension in these two are the same, we see them as a whole. In summary we learned 3 submodels.

B. Null values and outliers

There is 11 columns of raw data who have null values. Ten of them are related to jets and one is feature 'DER_mass_MMC'. We take different approaches on these two kinds of null values.

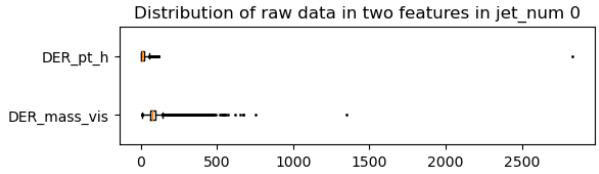

Figure 1. The distribution of raw data in 'DER_pt_h' and 'DER_mass_vis', dots denotes outliers.

For feature 'DER_mass_MMC', there is 38114 null values. We filled null spaces with the medium of the rest for coherence. For jet related features, we observed that the events with these null features are identical to events where 'PRI_jet_num' = 0 or 1. Naturally, we customized feature shape of input to model for each group: for 'PRI_jet_num' = 0 and 1, we only use features without null values and 'DER_mass_MMC'. Numbers of features used for each submodel are:18, 22 and 29.

The rest data have many outliers, as shown in Figure 1. We defined a valid interval $[lb, ub]$for each feature using mean and stardard deviation:$lb = \bar{x} - 2 \times \sigma(x), ub = \bar{x} + 2 \times \sigma(x)$, where $\bar{x}$ is the mean of x and σ is the standard deviation of x. Then we clipped values outside valid intervals and assigned these them to the closest boundary.

C. Normalization or standardization

Values of features have different dimensions, which will lead to oscillation in gradient descent when optimizing parameters. As a result, we did scaling to features to ensure the performance of finding minimum. As for which specific method to choose, we did experiments between normalization and standardization. Normalization is to scale values in a range of [0,1], while standardization is not bounded by range but assumes that data follows Gaussian distribution. Our ablation experiments analyzed accuracy of using normalization and standardization,

D. Unequal distribution in binary labels

Among all 250000 training event samples, 85667 have label s and 164333 have label b. Due to this unequal distribution, we re-sampled the dataset using either over-sampling or under-sampling. Over-sampling is to re-sample both sets with the size of larger set, under-sampling is to

re-sample both sets with the size of larger set. This is to reduce bias in the model in different labels. However, our experiments show that re-sampling in this task is not a good approach. Reasons may be the distribution gap between training set and validation set.

E. Ablation study

Table below shows the ablation study results we used in pre-processing. FFE refers to full feature engineering techniques in final submission, i.e. standardization, clipping of outliers, grouping by 'PRI_jet_num' and reassigning the null values to mediums. N is normalization.S refers to standization. RS refers to resampling. O refers to clipping of outliers.FC means using the first column which contains null values. $-ALLNAN$ means removing all columns with null values. PE refers to polynomial expansion of features. Tr_Acc and Val_Acc are training set accuracy and validation set accuracy.

The ridge regression experiments here apply hyper-parameters of λ = 0, degree = 9, learning_rate = 0.1, max_iterations = 2000, k_fold = 5, random_seed = 20221031, batch_size = 1.

Pre-processing methods	Tr_Acc(%)	Val_Acc(%)
FFE+N	61.22	61.39
FFE+S	83.08	**83.00**
FFE+S+RS	**90.55**	80.48
FFE+N-O	80.46	80.52
FFE+N-O-PE	75.86	76.01
FFE+N-O-FC-ALLNAN-PE	74.21	74.14

Table I
ABLATION STUDY ON PRE-PROCESSING TECHNIQUES.

III. METHODOLOGY

In this part, the effects of six foundational models will be compared and the best one will be given in this task. We first implement all the six regression functions and then adjust the hyper parameters to improve the accuracy of the models. Finally, ridge regression is selected as the best model in this project.

A. Hyper parameter tuning

First select the learning rate γ = 0.1 and k = 5 in K-fold cross validation. Then changing degree number to select the best degree of the polynomial basis and degree = 9 can provide the highest accuracy. In consistence with the former analysis on the data set, three different λs are needed for different models with 'PRI_jet_num' as 0, 1, and 2&3 respectively. The chosen lambdas are displayed in the Table II.

B. Model selection

Using the pre-processed data set, we compare 6 models' accuracy to get the best model in the task. Using Tr_Acc to denote accuracy on the training set and Val_Acc to denote accuracy on the validation set, the table III below displays the performance of every model in both training set and validation set.

Model	λ_0	λ_1	λ_2
ridge regression	10^{-6}	10^{-10}	10^{-10}
reg_logistic regression GD	10^{-9}	0.0001	10^{-8}
reg_logistic regression SGD	10^{-9}	10^{-7}	10^{-7}

Table II
HYPER PARAMETERS OF SIX REGRESSION MODELS

C. Results

From Table III, we can see that the least square regression model performed the best in all aspects of performance, which has the highest accuracy(83.07% on the training set and 83.00% on the validation set).Then select least square regression model and test it in the AIcrowd, we can a good figure accuracy(83.00%).

Model	Tr_Acc(%)	Val_Acc(%)
linear regression GD	73.57	73.47
linear regression SGD	46.16	46.17
least square regression	83.08	82.94
ridge regression	**83.08**	**83.00**
logistic regression GD	80.10	80.14
logistic regression SGD	78.11	78.10
reg_logistic regression GD	80.16	80.14
reg_logistic regression SGD	78.50	78.55

Table III
PERFORMANCE OF SIX REGRESSION MODELS

IV. CONCLUSION

Our work shows that no good performance is without carefully feature engineering, model selection, and hyperparameter tuning. We conducted an ablation study and proved that all our efforts of feature engineering made a difference in improving the model's performance. Then we tuned the hyper parameters to make sure every model can work in their best condition. We assumed that regularized logistic regression with SGD would perform the best for its widespread use in classification. However, experimental results showed that ridge regression was the best, contradicting our intuition. This taught us a lesson that a machine learning problem cannot always be solved by the method that is mostly used. We need to implement different methods and then compare their performance to get the best one. Last but not least, we should admit that our work was not perfect and perfectionism is not reachable at any time. We tried our best to refine our model and came to several meaningful conclusions through hands-on experiments. Therefore, we believe that our efforts pay off and all we have done are valuable.