Machine Learning Project 1 - Higgs Boson

Yawen Hou, Peilin Kang, Yueran Liang
School of Computer Science and Communication System, EPFL, Switzerland

***Abstract*—Machine learning provides an opportunity for machine to learn from data and help people implement predictive analytic. However, to make good predictions, it is necessary to choose proper techniques in order to reach high accuracy. This project tackles the binary classification problem of the Higgs Boson dataset from CERN. This report will first introduce six different models with an overview of their performances. Optimization techniques used on the chosen model to attain higher accuracy will then be explained in the article.**

I. Introduction

The Higgs boson is an elementary particles which explains why other particles have mass. However, it dacays so rapidly that scientists cannot observe its presence directly. Our job is to estimate the likelihood of a given event to be Higgs Boson (signal) or other particles (background). We will first make comparison among six learning models and choose the most appropriate one. Then, we will explain our data preprocessing methods to clean the data-set and extract useful information. Finally, we will discuss shortly on our results.

II. Models and Methods

A. Model Analysis

Table I shows the performance of the six models implemented based on the lecture notes. To obtain these results, we first split the train data into two parts, 80% for training and 20% for validation and randomly chose the values of λ, γ and max_iters (except for ridge regression). We then evaluated each model's performance. Since ridge regression aims to mitigate overfitting and underfitting, we decided to combine this method with cross validation to choose the best degree between 1 to 5 and the best λ among 40 random λs within the range of log(-10) to 0 based on the best loss (lowest test loss). All models use the same loss function, RMSE, except logistic regression and regularized regression which use negative log-likelihood function.

	Parameters used					
Methods	λ	γ	max_iter	Deg	Loss	Acc.(%)
GD	/	0.1	500	/	0.3398	74.46
SGD	/	0.01	1000	/	0.3911	72.70
Least Squares	/	/	/	/	0.3393	74.48
Ridge Reg	0.5541	/	/	3	0.8238	74.66
Logistic Reg	/	10^{-7}	20000	/	$0.62171 \cdot 10^5$	75.6
Reg Logistic Reg	1	10^{-6}	2000	/	$0.62175 \cdot 10^5$	75.6

TABLE I: Performance of the six algorithms

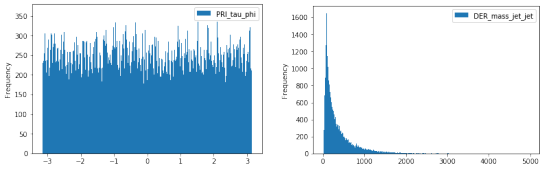
We noticed that the accuracy of each model is similar without any data preprocessing (around 75%). Due to the course schedule, we implemented the logistic regression in the last days before the project submission, so we used Ridge Regression for further tuning of the hyperparameters with cross-validation (see section C.).

As explained previously, we chose the degree and the λ for ridge regression by minimizing RMSE. However, it showed that this was not the best error function to use in order to improve our model's accuracy. In a later section C., we will introduce a customized cross validation process using a more robust error estimator.

B. Data Preprocessing

a. Data selection

We visualized the features using histograms and found 5 features subject to uniform distribution. They are `PRI_tau_phi`,`PRI_lep_phi`,`PRI_jet_leading_phi`, `PRI_jet_subleading_phi`. Uniform distributed variables tend to have less influence on the result, so we decided to drop these features from the data. Figure 1a shows the histogram plot of one the dropped features.



(a) Uniform distributed (b) Right-skewed variable

Fig. 1: Particular Variables Distribution

b. Data Cleaning

According to the official documentation [2], the `DER` (for DERived) features are computed from the `PRI` (for PRImitives) "raw" features measured by the detector. We noticed that the columns of some specific features are filled `-999` values (indicating that they are `undefined`). In the case of the derived features, it means that they cannot be calculated from the primitives features. Here are some examples of these data:

1) `DER_deltaeta_jet_jet`: The absolute value of the pseudorapidity separation between the two jets (undefined if `PRI_jet_num` ≤ 1).
2) `DER_sum_pt`: The sum of the moduli of the transverse momenta of the hadronic tau, the lepton, the leading jet (if `PRI_jet_num` ≥ 1) and the subleading jet (if `PRI_jet_num` = 2) and the other jets (if `PRI_jet_num` = 3).

Obviously, these kind of features strongly depend on the value PRI_jet_num, the number of jets created in the event. In addition, we noticed that the validity of the undefined PRI features also depends on PRI_jet_num.

Interestingly, PRI_jet_num only takes four integers: 0,1,2,3. After splitting the data into 4 groups according to the PRI_jet_num, we noticed that some columns are entirely filled with −999 due to the reasons explained above. We dropped all these columns.

Furthremore, we also found out that there are two outliers (values abnormally large) in the DER_pt_h feature of two sets, set PRI_jet_num = 0 (outlier 2834.999) and set PRI_jet_num = 2 (outlier 1053.807). To get rid of the influence of these outliers, we dropped this two rows from the data.

Finally, we noticed each set have several rows missing the value for DER_mass_MMC. Therefore, we separated again each set into two subsets according to the validity of DER_mass_MMC, and dropping this column in the subset where it is invalid. We also noticed that PRI_jet_all_pt is always zero when the PRI_jet_num is zero. As we know that the momentum of object is absolutely zero if this object doesn't exist, we also dropped this columns in the two subset where PRI_jet_num is zero.

c. Data augmentation

We read the post about angle transformation written by the host of the Higgs Boson competition on the forum [3]. It is mentioned that the features we dropped in the subsection b (angle ϕ) can be replaced by their difference, which also should be in a range from 0 to π. Moreover, the value of angle η should also be flipped if the tau's η is negative. In addition, according to the official document [2], the feature DER_deltaeta_jet_jet represents the absolute value of two ηs between two jets and DER_prodeta_jet_jet indicates the product of the ηs of the two jets. we think about it is meaningful to calculate these type of features with respect to new combinations of other particles. As a result, it improved the accuracy by 0.2%.

C. Cross Validation

As mentioned in section A., cross validation is used to tune the hyperparameters (polynomial degree and λ for penalty) in order to get the best prediction. In the mean time, we noticed that mean square error does not represent well the prediction error, since in some cases, the accuracy is relatively high, but the RMSE is abnormally large, as seen below. This causes our models to dismiss some of the best choices of the hyperparameters.

```
Degree: 12, test_loss: 193.2926,
lambda: 2.656e-06, accuracy: 0.79997
```

Therefore, instead of minimizing RMSE, we decided to maximize the percentage of rightly predicted labels. This change improved our accuracy by nearly 4% (see Table III).

Here are the best λ and degrees we obtained:

Sets	λ	degree
0 Jet with valid mass	$2.48 \cdot e^{-7}$	12
1 Jet with valid mass	$2.66 \cdot e^{-6}$	12
2 Jets with valid mass	$1.00 \cdot 10^{-2}$	12
3 Jets with valid mass	$6.73 \cdot 10^{-4}$	12
0 Jet with invalid mass	$3.05 \cdot e^{-4}$	10
1 Jet with invalid mass	$9.55 \cdot e^{-16}$	4
2 Jets with invalid mass	$3.20 \cdot e^{-9}$	4
3 Jets with invalid mass	$4.50 \cdot 10^{-4}$	4

TABLE II: Best hyperparameters obtained with the customized 10-fold Cross-Validation.

D. Data Transformation - Log

Later in data exploration, we discovered that multiple features have a right-skewed distribution. In the video we consulted [1], they suggested that applying a log transformation to right-skewed distributed features would improve the accuracy. Due to time constraint, we did not have the time to tune the hyperparameters with transformed data, but we did notice an improvement of 0.08% after applying logarithmic to the following features (after tuning): PRI_tau_pt, PRI_met, DER_pt_ratio_lep_tau, PRI_lep_pt, DER_mass_vis, DER_sum_pt, PRI_jet_subleading_pt, DER_mass_jet_jet, PRI_met_sumet. Figure 1b shows an example of the distribution of a feature that is treated with log.

III. Result and Conclusion

Method	Ridge R.	Max. Acc.%	Cleaning	Data Aug.	Log
Acc. (%)	74.66	79.58	83.17	83.357	83.365

TABLE III: Summary of the improvement in accuracy

Based on ridge regression, we kept trying different ways to optimize our prediction, as shown in Table III. The first step is to use maximum accuracy find the best degree and lambda instead of using minimum RMSE loss, by which the accuracy was improved by nearly 5%. Then, we cleaned the data by splitting and eliminating all NaN rows and columns and improved the accuracy by 3.6%. We discovered that augmenting the features using primitive features lead us to another improvement of 0.2%. Finally, we applied a log transformation to right-skewed distributed features and achieved our best accuracy 83.365%. The strength of our best model is that we use the inner relationship between features and it can achieve good accuracy. Due to time constraint, we did not have time to test each method individually and all their combinations, we just seek improvement based on our last best combination of process. Most importantly, we learnt that data preprocessing is an essential step in all machine learning process.

References

[1] Wanda Wang, Rob Castellano, Yannick Kimmel, and Ho Fai Wong (September 4, 2016). *Higgs Boson Kaggle Machine Learning*, [Video file]. Retrieved from: https://www.youtube.com/watch?v=Xv_tVVJvDfE

[2] Claire Adam-Bourdarios, Glen Cowan, Cecile Germain,Isabelle Guyon,Balazs Kgl and David Rousseau (July 21, 2014). *Learning to discover: the Higgs boson machine learning challenge*, [PDF file]. Retrieved from: https://higgsml.lal.in2p3.fr/files/2014/04/documentation_v1.8.pdf.

[3] David Rousseau (2014). *Reducing the feature space* . Retrieved from: https://www.kaggle.com/c/higgs-boson/discussion/9576*[Competition Forum]*