Higgs Boson Machine Learning Challenge

Zhuoyue Wang, Danya Li, Yiyu Wang
EPFL Lausanne, Switzerland

***Abstract*—In this project we used machine learning methods to predict whether a given event's signature was the result of Higgs boson's decay or not, which is exactly a binary classification problem. We first completed a toolbox which can perform six basic machine learning algorithms. Then the given raw data was split into three groups according to one special feature called PRI_jet_num. After that we tested the performances of different methods on each data group with the help of cross validation. In the end, the optimal method and corresponding parameters were decided and we achieved 82.5% accuracy for the test data.**

I. Introduction

The Higgs boson is an elementary particle in the standard model of physics which decays rapidly and is hard to observe directly. As the Higgs boson's decay signatures look similar to others, some advanced machine learning algorithms are expected to make a good recognition of the preferred particle. The main task of this project is to process and clean the raw data and to find the best performed model to generate prediction. On the one hand, we attempted different feature engineering techniques and figured out some good ways that could improve prediction results. On the other hand, a 5-fold cross validation method was used to determine the optimal models and parameters.

II. Models and Methods

A. Data Prepocessing

Features are fundamental and essential elements in a machine learning model, which could significantly influence the model performance.

From the given data set, we can quickly observe that some columns contain -999 which seems to be not normal. After carefully reading the related physics background and the description of the features, we found some important clues to preprocess the data set. The unfriendly data -999 has a close connection with PRI_jet_num, in other words, if the number of jets is less than 2, then the value of some sensitive features would become undefined, namely -999. On the other hand, when the number of jets is equal or larger than 2, then most of features are meaningful. Hence, we divided our data into three groups as shown in the following.

jet_num	dropped features
2,3	22
1	4 5 6 12 26 27 28
0	2 3 4 5 6 12 23 24 25 26 27 28

However, the first feature DER_mass_MMC could sometimes be -999 because the topology of the event is too far from the expected. In order to deal with the unexpected value in this feature, we simply replaced -999 by median of the valid data in this column.

Having the goal to make raw data a better quality, a normalization step is also needed and the basic data processing is end up here. In the following part, other advanced methods which also helps would be introduced.

B. Cross Validation

For the purpose of testing our model, we implemented a 5 fold cross validation method, which helps us get a sense of the model performance. Basically, we split the data into 5 groups, and by turns, keep one group as test data and others for training. Here we consider the number of wrong prediction as the loss.

C. Feature Augmentation

As all the algorithms that we are supposed to carry out are linear models which might be not rich enough, we increased the representational power by adding a polynomial basis. Concerning about potential overfitting problem and the different importance of different feature, only features prefixed with DER which were selected by the physicists of ATLAS were augmented.

The best degree for each algorithm and each data group was obtained from drawing accuracy rates figure based on 5-fold cross validation, the following result with ridge regression method was taken as an example.

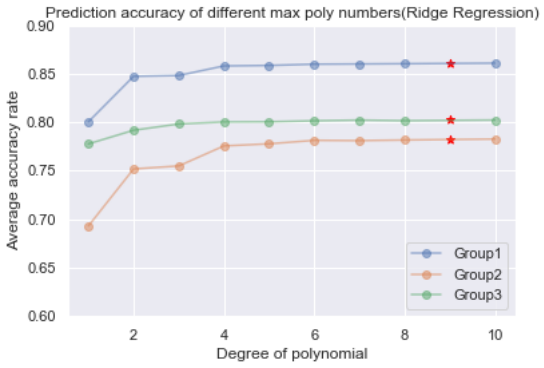

Fig. 1. it shows that when using ridge regression for each data group, nine is the best degree number which helps us achieve highest accuracy for each group.

D. Preliminary Test on Training data

From the ideas described above, we implemented all the required algorithms and got the following result.

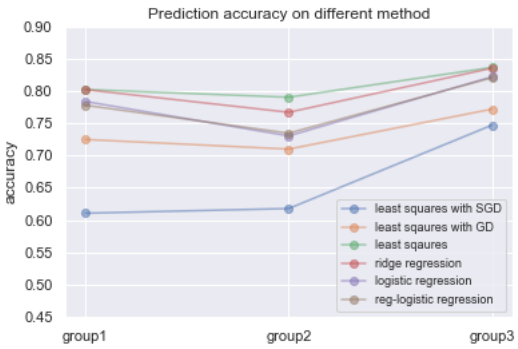

Fig. 2. after roughly analyzing and tuning different parameters for each algorithm and each data group, we fed the current best value and tested all of them by 5-fold cross validation. The result is shown in the figure.

As we can see in figure 2, least square method attains highest accuracy for each group in the training set, while ridge regression can almost achieve the same performance. The test accuracy of logistic regression and regularized logistic regression are close and a little bit lower than the least square method and ridge regression method. Least squares with gradient descent and stochastic gradient descent are the worst.

III. Analysis And Improvement

A. Result Analysis

We think the result is reasonable since the two best performing methods obtain the optimal weights vector w^* by directly solving matrix equation. This ensures that the optimal solution of the convex optimization problem is obtained on the training set while other methods depending on gradient descent which might not be able to achieve the same precision. To be explicit, the number of iteration and the choosing of step size would influence the performance of method with gradient descent a lot. And there is no guarantee that the gradient descent could reach the optimum.

Based on the result of preliminary test and our analysis, we tried some combinations of well-performed methods on the test data and got a best accuracy of 77.7% among them. In the best case, model ridge regression was used for group 1 and group 2 and logistic regression was used for group 3. Here we can see the least squares method didn't bring the highest accuracy for us. The reason is explained in the below.

In practice, the robustness of the algorithm is very important. Compared to least squares, ridge regression has a lower variance and it is less sensitive to outliers. And another point why ridge regression is preferred is that it's always solvable. In ridge regression, we get the optimal w^* from

$$\mathbf{w}^*_{ridge} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{1}$$

The matrix $(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{X})$ is always invertible. But there is no such guarantee for least squares method.

However, the performance of ridge regression is still worse than the result of preliminary test. We believe this is because of the existence of outliers.

B. Outliers Processing

The data samples were still not well-cleaned as there remains some outliers which might reduce the accuracy. We simply applied the standard deviation method, which is determined by feature distributions. After analyzing probability distribution of each feature, we found most of features follow an approximately Gaussian distribution. For this method, three standard deviations from the mean is a common cut-off in practice for identifying outliers in a Gaussian or Gaussian-like distribution.

In the project, for data column of each feature, we firstly computed the mean and standard deviation without outliers(for convenience we call it cleaned data), then we got the lower and upper bound according to standard deviation method. In order to better maintain the characteristics of the original data we replaced outliers bigger than upper bound the maximum of cleaned data and replaced outliers smaller than lower bound the minimum of cleaned data.

After applying the above method, the accuracy of our prediction results has been greatly improved.

IV. Final Result and Summary

In the end, with many times of parameters tuning and method selections, we achieved an overall categorical accuracy of 82.5% with F1-Score 0.736 on the public Leaderboard. In this case, ridge regression was applied to group one and group two and regularized logistic regression was used for group three.

In summary, we developed an algorithm to predict whether it is the signal of Higgs boson given some measurement data in this project. At the beginning, we tried a few ways to process the raw data and do the feature engineering. After that, different models were used to achieve the best performance and we spent a lot of time tuning the model parameters. In the end, with the help of our continuous exploration, the model has reached a quite good performance.

From this project, we learned that feature engineering is quite essential. Different ways to deal with unexpected values in the features made a big difference on the prediction result. Besides, we had a better understanding of all the implemented algorithms and we also believe some more advanced methods could be even more powerful. Hence we see the magic of machine learning!