Higgs Boson Classification using Machine Learning

Lei Pang, Jingran Su, Xintong Kuang

***Abstract*—In this project, our team applied machine learning methods to CERN's Higgs Boson data set. We first process and clean the data, split them into different groups according to the PRI_jet_num feature, and then apply six regression models on each group to find an accurate classifier to generate prediction. We then try to make an improvement by using polynomial feature expansion. Finally, with the best model and corresponding parameters we achieved 83.0% accuracy on the test data.**

I. Introduction

The Higgs boson is the particle associated with an energy field that transmits mass to the things that travel through it. The aim of this project is to overcome the problem of binary classification on the Higgs Boson data set. This report presents the whole procedure on solving such problem by finding out the best model among six machine learning algorithms with feature expansion improvement.

II. Models and Methods

A. Data Preprocessing

It is essential to do preprocessing at the beginning to find out feature correlation, apply zero variance analysis, and fix error values. On this data set, we perform the following steps in order:

1) *PRI_jet_num has 0,1,2,3 categories*
   Features could influence model performance, we first performed feature digging to check the correlations between each feature. Each feature seems not having single value. For features having strong correlations, they have some values around -999 and 0.
   We found that the 23rd feature(PRI_jet_num) has only four values (0,1,2,3). According to its physical definition, this is a categorical feature. Hence, we decided to separate the date into four groups based on the jet number 0,1,2,3, and then check the characteristics for each category.
2) *Processing the zero-variance features*
   We now have 4 groups each associated with 0,1,2,3. We performed variance analysis of columns for each group. Zero variance has no contribution to the model training, since the values of these columns are same. We detected the zero variance features and deleted them from the data set since they are not informative.

jet_num	zero variance features
0	4, 5, 6, 12, 22, 23, 24, 25, 26, 27, 28, 29
1	4, 5, 6, 12, 22, 26, 27, 28
2	22
3	22

Table I: Zero variance features for each jet_num group: The 22 means the jet_num categorical feature.

   After deleting the zero-variance features, each group has different amounts of features.
3) *Fixing NULL value and outliers*
   We observed that there are many columns containing NULL value (-999). They are missing values. We detected these abnormal values and replaced them with the median of non-NULL values.
   There are many outliers in the features. Outliers in this project refer to data that exceed the range $[\mu - 2\delta, \mu + 2\delta]$, where μ is average value and δ is standard deviation. We replaced outliers with the nearest boundary depicted before.
4) *Normalization*
   Since the model convergence is sensitive to the feature ranges and different feature ranges could cause divergence of the model, we perform a normalization step as our final step. What this project choose is using Max-Min scale to normalize all features within 0 to 1.

With the whole procedure described above, we now obtain four data sets according to their jet number. There are no zero variance features, all NULL values and outliers have been fixed by replacing with suitable values.

B. Implement of Regression Techniques

Six regression techniques are chosen as the machine training model for our data set. At the beginning, we need to initially select hyper-parameters which are not directly learnt within estimators. Thus we implemented cross validation to obtain the optimal hyper-parameter lambda in [0.001, 1] for ridge regression and [0.1, 1] for reg-logistic regression respectively in order to minimizes losses. From the results of cross validation, it seems that the smaller the lambda, the smaller the losses. However, the penalty part will be useless with too small lambda. Hence we choose lambda = 0.001 for ridge regression and lambda = 0.15 for reg-logistic regression, which is the average value of the optimal lambdas for four groups.

We split the train data into a training (80%) and a test set (20%) to train and test models. We standardized the data for iteration, which helps maximize the operating efficiency.

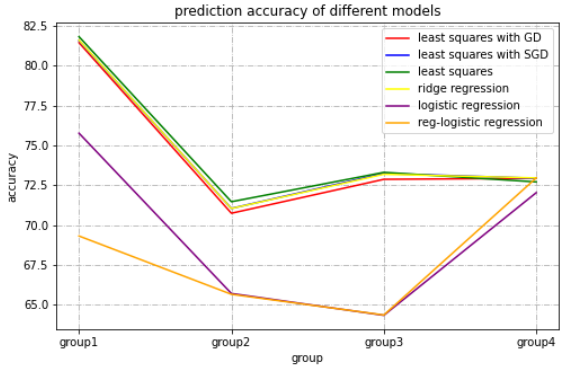

Figure 1: Prediction accuracy of the six models: the hyperparameter lambda are roughly tuned and chosen as the best one.
Notes: accuracy denotes the correct prediction probability in %, group denotes the category of the data set divided by the feature PRI_jet_num, label denotes the tested regression model.

Then, we tested the model for each group and reported the average correct prediction probability. The prediction performance for each model are presented in Figure 1.

III. Analysis and Improvement

A. Feature Expansion

In real-world case, simple models such as linear regression do not incorporate complex relations between features. In order to further improve the performance of our models, we used polynomial expansion to create new features. To determine the degree of polynomialization, we used cross validation again. Since the required running time for logistic regression is much long, we used the ridge regression as the base model to estimate the best degree in [3, 9]. We set the lambda = 0.001 and k-fold = 5 and observed that the rmse decreases slightly as the degree increases.Finally, we retrained the data and plot the performance of the six models after polynomial expansion with degree = 9 in Figure 2.

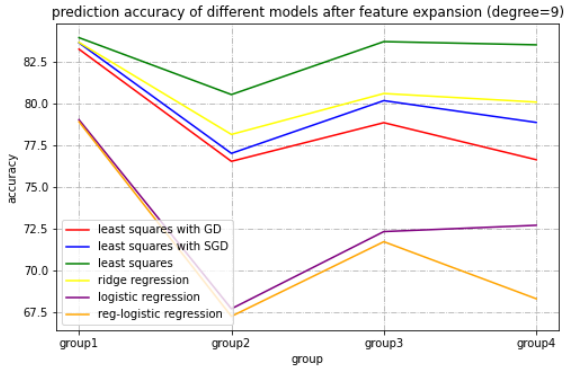

Figure 2: Prediction accuracy of the six models after feature expansion: the hyperparameter degree are roughly tuned and chosen as the best one

B. Tuning Hyperparameter

Although the best prediction is obtained with least squares, we want to explore how much the prediction performance of the ridge model can be improved with the optimization of the hyperparameters. We set up a grid search to compute the optimal degree of polynomial expansion in [6, 9] and lambda for ridge regression among [0.0001, 1]. The relation between rmse and pairs of degree and lambda are plotted in Figure 3.

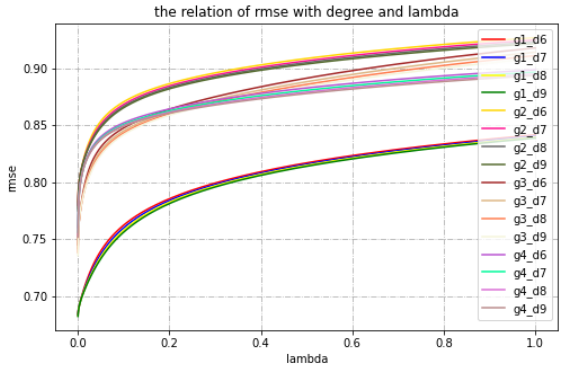

Figure 3: Relation of rmse with degree and lambda
Notes: lambda is one hyperparameter to be optimized, rmse is the root mean square error tested from the cross validation, label denotes the combination of different group and degree with a specific color.

As the Figure 3 shows, the smaller lambda, the smaller error. However, if the lambda = 0, the ridge regression is identical to least squares model. Hence, we still choose lambda=0.001 to distinguish slightly between ridge regression and least squares model. We also estimated the rmse with degree of 6-9, and found that rmse is smallest with degree 9.

IV. Results and Discussion

Our work tells us that one can significantly improve the future prediction performance by iteratively optimizing and testing the models. In the initial forecast, only the prediction accuracy of the first group could reach more than 80%, while that of the remaining groups are all below 75%, and may even be less than 65% (group 2). After polynomial feature expansion with degree 9, the performance of all models has improved to varying degrees.

We expect two logistic regressions to have better performance, whereas in our case both of them do not produce a good predictive result. Therefore, we choose the least squares and ridge regression as our final models to generate predictions on the test data set with the tuned hyperparameters (lambda = 0.001, poly-degree = 9), and obtain the prediction accuracy 83.0% and 81.3% respectively. The best result for AIcrowd competition leadboard we used is 83.0% generated by least squares model.