Machine Learning to discover Higgs Boson

Chaffard Clement, Grobel Coralie, Roch David

***Abstract*—The Higgs Boson was first observed in March 2013 at the Large Hadron Collider at the CERN. It is an elementary particle which explains why other particles have mass. To identify Higgs Boson, physicists smash protons into one another at high speeds and analyse the decay signature of the particle. We used data from the collision as a training set to build models that predict if a Higgs Boson have been emitted or not. The purpose of this research is to investigate several machine learning optimization models and evaluate their effectiveness in predicting the presence of a Higgs Boson. The best model was the one that used ridge regression since it had the highest overall accuracy.**

I. Introduction

Physicists regard the Higgs boson to be the cornerstone of matter's basic structure. This particle is responsible for the mass of all other particles in our universe. The Higgs boson, on the other hand, is extremely difficult to witness directly. Indeed, it only emerges for a fraction of a second in high-speed particle collisions. Fortunately, the decay signature of the boson may be used to infer its identity. The Higgs Boson was discovered at the Large Hadron Collider (LHC) in 2013, over 50 years after its existence was postulated in 1964. Machine learning can assist distinguish whether a signature is due to a Higg Boson or another collision background event since a collision event produces many identical decay signatures. The performance of machine learning models trained on original CERN data in locating the Higg Boson in a collision event will be investigated and evaluated in this research.

II. Exploratory Data Analysis

A. A first glimpse at the data

The task may be seen as a binary classification issue between Higgs Boson and any other particle. The training set is made up of authentic CERN data that refers to the characteristics and forecasts of about 250000 occurrences. The raw data contains 30 features (d=30) that characterize events seen during experiments at CERN's LHC. The data is mostly continuous, except for the variable PRI_jet_number which characterizes the discrete number of jets (maxed at 3). We also observed an important number off missing values represented as the integer -999 in the dataset.

B. Data preprocessing

To address the problem of the missing values, we decided to drop columns where more than a third of values were missing. Among the 30 observed characteristics 10 were discarded this way. For the remaining missing values we chose to replace them with the median of the values of the corresponding feature. We chose the median as a more robust option to outliers than the mean. In a second step, we studied Pearson correlation between the remaining features, in order to remove highly correlated features and simplify the model. One column was eliminated this way. Finally we normalize the data so that each dimension get the same influence to the result. After that, we are now able to divide the data into different training and testing sets and start training our models. To make sure our study is reproducible we fixed a seed to value 1.

III. Model Training

We implemented six different models and one more that use all other models:

- Least Squares gradient descent (GD)
- Least Squares stochastic gradient descent (SGD)
- Least Squares normal equation
- Ridge Regression using normal equation
- Logistic Regression
- Regularized Logistic Regression
- Combination of all the above methods

To be able to train our model, we have first searched for optimal parameters specifics to each method. Our parameters are : the degree for polynomial expansion, the regularization factor λ and the waiting factor γ. We furthermore need to take care to have the maximum number of iteration enough big to allow the model to converge to the best solution.
To optimize these parameters, we have used two criteria in a k=4-fold cross validation function : the smallest loss or the highest accuracy. By training and testing the model 4 number of times on different subsets of the same training data we get a more accurate representation of how well our model might perform on data it has not seen before and we avoid training an over fitted model. We will explore further this criteria when talking of model evaluation in part V.
We also implemented a system of threshold for logistic regression which stops the training if the difference between 2 consecutive loss is under the threshold. We set threshold of 10^{-8}. This implementation proved to be very useful and saved a lot of time.

We then added a seventh method that will first predict independently the result for all other methods and then combine those results. To combine them, a system of voting was done. As all the prediction are either 1 or -1, we can just add the six different model predictions of a data together and if it is bigger than 0, the prediction is 1, if it is smaller than 0, the prediction is -1 and if it is equal 0, we use the prediction of our best model.

IV. Results

The following table presents the accuracy obtained for each model and their optimal parameters:

Model	**Hyperparameters**	**Accuracy**
Least squares GD	gamma = 0.39810717	0.706
Least squares SGD	gamma = 0.00316228	0.682
Least squares normal	degree = 12	0.811
Ridge Regression	lambda = 10^{-5} degree = 13	0.813
Logistic Regression	gamma = 10^{-5}	0.714
Regularized Logistic Regression	gamma = 10^{-2} lambda = 10^{-5}	0.715
Combination of all methods	optimal parameters of each methods	0.734

Fig. 1. Summary of our results for each model

As we can see in Table 1 , Least squares normal's equation and Ridge regression's model are the ones with the best accuracy. To be able to compare them we have plotted the accuracy in function of the degree of these two methods in the following figure.

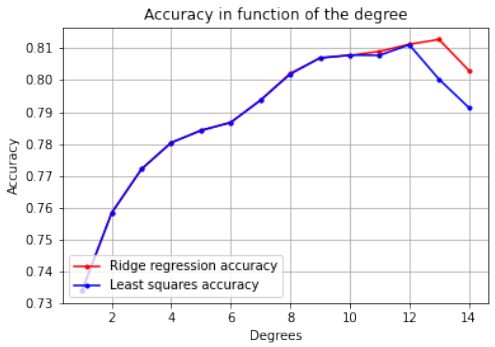

Fig. 2. Mean accuracy on 4 k-folds cross-validation for Least squares and Ridge regression method in function of the degree using the optimal lambda

We can see on this figure that the accuracy of Least squares methods and Ridge Regression are really close until degree 10. At this degree Least squares has his best accuracy but then his results falls down. For Ridge regression it stays stable for across three degrees.

V. Model Evaluation and Discussion

As mentioned in Part III, to assessed the model performance we used the error and the accuracy. For the Least squares models, the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) were selected. For Ridge Regression, we have selected the L2 regularized loss. Finally, for Logistic Regression models we used negative log likelihood. Because of these different computation of loss we have compared our model on their accuracy.

As we have seen in figure 2 and in Table 1, the two models with the best accuracy are Ridge Regression and Least Squares differing by only 0.2%.However, Least squares is only a special case of Ridge regression with a λ equals to zero. So using the ridge regression allow us to do regularize the update step of the weights. Which should give us less over fit.Knowing all of that we take **Ridge regression** as the best model with a lambda equals to 0.00001 and a degree of 12. With those parameters, Ridge regression performs with an accuracy of 0.813 on our testing set and 0.795 on AIcrowd.
The combination of all method by a voting system was not as good as we hope. To increase the precision of this method, we tried to add a weight to each model and compute those weight with a grid search for values $\epsilon[0;4]$. The result was either giving a weight of 1 to our best model and 0 to each others or giving a bigger weight to our best model and low weight to the other models so that they doesn't influence the prediction. So the best accuracy found was always exactly the same as our best model's one. (We will explain later that it might not be the ridge regression due to imprecision of the function numpy.linalg.solve). This might be due to two reasons. The first one is that we were not able to find the right weights and the second one is that there is 20% of the data that are wrongly classified by all our models.

Concerning our data pre-processing, we observed minor improvement in the accuracy across all our methods ($\approx 1\%$) mostly due to the handling of missing values, normalization and dropping correlated feature didn't significantly improved our models.

Additional characteristics to consider for model selection include the algorithm's execution time and space complexity, which might be critical depending on the resources available. We noticed some variations in the results of Least Squares and Ridge regression when running the code on different computers. The difference emerges using the numpy linalg solve method for very high degrees. We make the hypothesis that the machines make different floating point rounding for very big or very small numbers.

We also faced serious difficulties with large losses repeatedly creating overflow errors obstructing in some cases hyperparameters optimization.

VI. Conclusion

Based on those results, we see that the values reported of the event can allow us to predict if the particle is a Higgs boson or not. Our best model is the ridge regression using 13 degrees and the parameter $\lambda = 10^{-5}$. Doing that, we achieved a precision of 80%.
Those results are encouraging and promote the importance of data analysis in high energy physics. To go further, we could try to use a machine learning approach to combine all the methods, to better choose the weight given to each model when voting.

References

[1] John Ellis, Mary K Gaillard, and Dimitri V Nanopoulos. "A phenomenological profile of the Higgs boson". In: *Current Physics–Sources and Comments*. Vol. 8. Elsevier, 1991, pp. 24–72.

[2] Sotiris B Kotsiantis, Dimitris Kanellopoulos, and Panagiotis E Pintelas. "Data preprocessing for supervised leaning". In: *International journal of computer science* 1.2 (2006), pp. 111–117.