Machine Learning Methods For Higgs Boson Signature

Antoine Bedanian
Master in financial engineering
EPFL
Lausanne, Switzerland
antoine.bedanian@epfl.ch

Nicolas de Lestable
Master in financial engineering
EPFL
Lausanne, Switzerland
nicolas.delestable@epfl.ch

Ikram El Mouden
Master in financial engineering
EPFL
Lausanne, Switzerland
ikram.elmouden@epfl.ch

***Abstract*—In this paper we will present a module with which the problem of detecting the creation of Higgs Boson will be solved. To do so we apply machine learning methods to original data of the CERN particle accelerator. In fact machine learning methods has become prevalent recently to solve a lot of different problems and in different fields. One of the most used machine learning methods is regression, we will use different type of this method in order to solve our problem.**

I. Introduction

The Higgs Boson is an elementary particle in the Standard Model of particle physics, produced by the quantum excitation of the Higgs field. It's mechanism has been proposed in 1964 by Peter Higgs and it's existence was confirmed by the Large Hadron Collider at CERN in March 2013 [1]. Our goal is, given a decay signature, predict if the signal corresponds to a decay of a Higgs boson or is just noise. To do so we worked with a dataset with 30 features that represent the signature event from which we want to infer if a Higgs Boson was created. Using data cleaning, feature engineering we implement machine learning moethods to solve this classification problem.

II. Models implementation

The first task of this project was to implement the machine learning methods we saw in the lecture note. We code the 6 required functions: Linear Regression using Gradient Descent, Linear Regression using Stochastic Gradient Descent, Least Squares Regression, Ridge Regression, Logistic Regression using Gradient Descent and Regularized Logistic Regression using Gradient Descent. In order to compare the methods, we apply them to our dataset before we analyze them, we just remove the features with more than 70% of NaN and replace the rest of the NaN value by 0. We obtain the result described in Table 1:

Methods	Prediction	λ	γ	Max iter
OLS	0.74			
Gradient Descent	0.64		10^{-18}	1000
Stochastic Gradient Descent	0.64		10^{-18}	1000
Ridge Regression	0.74	10^{-18}		
Logistic Regression	0.75	10^{-18}	10^{-18}	1000
Reg Logistic regression	0.75	10^{-18}	10^{-18}	1000

Table 1. Summary of the methods comparison

We can see from this table that our best functions are the Logistic Regressions methods followed by the Ridge Regression method and the OLS method, but we need to analyse and clean the dataset to have better results.

III. Dataset analysis and feature engineering

A. Data analysis

When we first implement our machine learning methods, we notice the presence of some features with a lot of NaN values (code by -999). In our first implementation we remove them or put value 0. Now we decide to investigate more closely the dataset in order to do appropriate dataset cleaning and feature engineering before running our algorithms. We notice that some of the features where highly correlated like we can see in the correlation matrix:

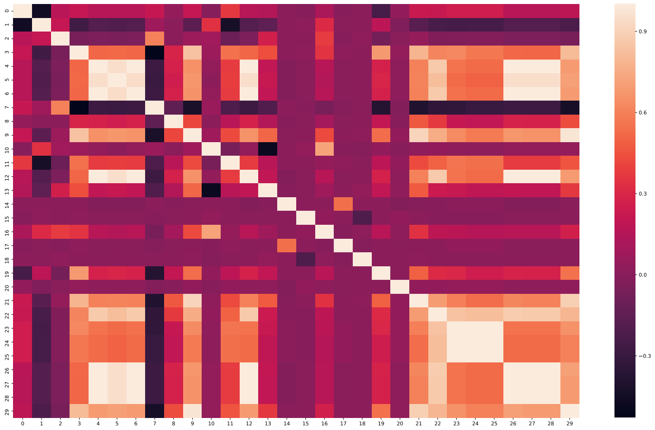

Fig. 1. Correlation between different features

We also observed, that feature 22 can be used to subset our dataset using the fact that it take only 4 different values $(0, 1, 2, 3)$.

B. Feature engineering and data processing

Using the value in feature 22, we divide our datset into 4 subsets. After this decomposition, we highlight that the subsets 0 and 1 have features with only NaN or unique value so we decide to remove them. We choose after that to replace the rest of the NaN values by the mean of the feature. [2] Then we process our data, we transform some future by applying functions on it, we standardize all the features. We apply our feature engineering process on the train and the test set.

IV. Cross Validation

To compute our prediction we decided to implement a Ridge Regression method. In order to found the best parameter λ we run 4 Cross Validation test, one for each subset of our data. For each Cross Validation, we split the train set randomly in 2 subsets 80% for the train set and 20% for the test set. We also apply polynomials functions with different degrees for each subset on it and try to minimize the RMSE. The following plots illustrate one example of the Cross Validation results, we plot the evolution of train and test error as a function of lambda:

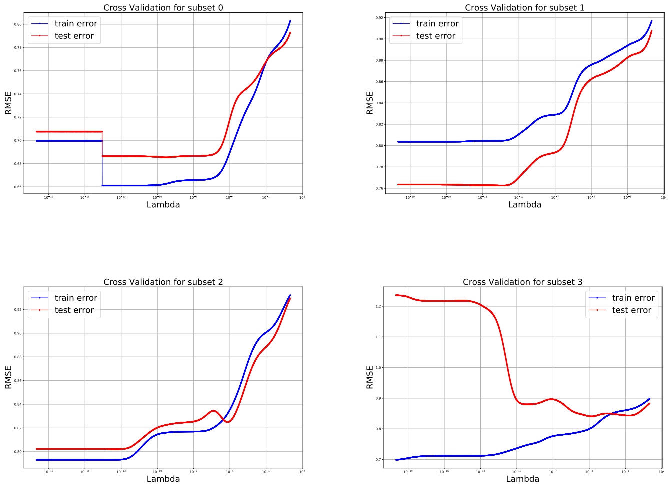

Fig. 2. Cross Validation for the 4 subsets trying to find the best λ

The parameters that we obtain from our Cross Validation of each subset are summarized in the following table Table 2.

Subset	λ	Degree
Set_0	$1{,}0 \times 10^{-12}$	2
Set_1	$4{,}7 \times 10^{-12}$	2
Set_2	$4{,}2 \times 10^{-14}$	2
Set_3	$1{,}6 \times 10^{-6}$	3

Table 2. Subset best parameters

The Cross validation method gives us the best degree and λ for each subset without over-fitting. [3]

V. Implementation and Predictions

Using the previous parameters found, we implement the 4 Ridge Regression to find the best value for W. Then we use test data to obtain our prediction. We manage our output using a sigmoid function and by changing the 0 output to -1. The previous transformations helped us to clean the dataset but they were not enough, we found a prediction of 77%. So we decide to add some features to have better predictions: we apply the function sin. The accuracy of our prediction found after this change around 79%. To make sure that our final way to process the data gives us the best model we made 3 different submissions, to compare their results. The first one without dividing the dataset into subsets, the second one with our 4 subsets and with the polynomial function and the last one where we add the sinus function. the accuracy on the AIcrowd leaderboard obtained with our three model using the Ridge Regression algotithm are summarized in the following table.

1st model	2ed model	3rd model
75%	77%	79%

Table 3. Models accuracy based on Alcrowd

VI. Result and comments

The predictability of each model was in accordance with what we studied during the machine learning courses. Our results show well how the feature engineering and data cleaning improve our prediction. However, as we are facing a binary classification problem, Logistic Regression would have been a more suitable technique but we preferred to concentrate our effort on data exploration and processing and improving Ridge Regression method, because when we fit the data with a Logistic Regression method we did not get a better prediction. The results that we managed to get show that Ridge Regression worked good in this case.

VII. Conclusion

In this project, we learned how to implement 6 methods of machine learning and experienced the limits of each one of them. We also learned how to explore and process the data, how to find creative ways to clean it and how to deal with missing value and exploit each features.

References

[1] Wikipedia. Higgs boson — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Higgs%20boson&oldid=922120045, 2019. [Online; accessed 22-October-2019].

[2] Jason Brownlee. Discover feature engineering, how to engineer features and how to get good at it. https://machinelearningmastery, 2014.

[3] Pedro Domingos. A few useful things to know about machine learning. 2012.