Master Semester 1 - Fall 2021
CS-433 - Machine Learning

Project 1 - Higgs Boson

Students: SANCHEZ NDOYE Miguel-Angel
FERCHIOU Sami
BRUNO Etienne

Date: *Monday, November 1, 2021*

Abstract

The aim of this project is to find a model able to predict whether a given set of measurements (original data from the CERN) represents the emission of a Higgs boson particle or can be considered as background noise (i.e. is due to physical phenomena we are not interested in). In this project, we implement six different machine learning algorithms on data whose features are carefully chosen, cleaned and eventually improved. Using grid-search and cross-validation, we are able to optimize the hyper parameters. With an accuracy of 80.6%, ridge regression was found to be the best performing algorithm for this problem.

1 Introduction

The goal of this project is, given a set of measurements describing a particular physical experiment, to find a model able to determine if a specific measurement corresponds to the emission of a Higgs boson particle. The aim is thus to emulate, loosely, part of the process that was followed by scientists at CERN and that led to the discovery, the 4 July 2012, of the Higgs boson particle. This goal will be approached using standard machine learning algorithms.

In terms of machine learning, determining if a data point can be attributed to the emission of a Higgs boson particle (signal) or not (background noise) can be seen as a classification problem.

In this paper, we will describe how, using training and testing data sets from the ATLAS experiment, we selected, implemented and optimized the algorithm that made that classification task possible. The performance of the algorithms we implemented was judged, on a test data set via the website AIcrowd which allowed us to assess our accuracy as well as our F1 score.

2 Models and Methods

2.1 Data Analysis

The training set we worked with is composed of 250'000 samples, each described via 30 different features (represented as columns). The training set also has a label stating if the data point is a Higgs boson signal or a background noise. The testing set is made of 500'000 samples with the same number of features. The only exception is that the test set does not contain values for the prediction (this field will be estimated by our model).

By exploring the data, we noticed that 72.75% of the samples (181'886 among the total of 25'000) had at least one undefined feature. From another perspective, we can state that 1'580'052 features among the total 7'500'000 features are undefined. Moreover, we discovered that the distribution of the undefined values is closely related to one attribute, mainly the PRI_jet_number which takes integer values between 0 and 3.

On the following figure, we plotted the correlation matrix of the initial training data set to get an understanding of the linear dependence between features. As one can see on Figure 2.1, some features are highly correlated (>.8).

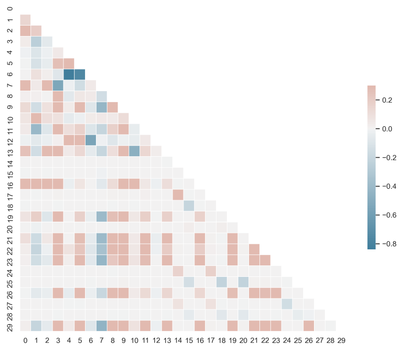

Figure 1: Correlation matrix of the features

2.2 Data Cleaning & Features Engineering

To apply our machine learning models, we need to work with clean data. Following the aforementioned observations, we decided to split the data set into four subsets depending on the PRI_jet_number value. Having these four new data sets, we then removed from each subset the columns which had null standard deviation (constant attribute for the whole group). In other words, we removed the columns which are constant and do not bring additional information to the model. This process greatly reduced the remaining number of undefined values encountered in each data set. For these unresolved values, we decided to replace them by the median value of the corresponding column.

After that, we standardized the data to obtain coherent values and prevent features with wider ranges from dominating the distance metric. To do this, we removed the mean value of each column (attribute) to the according data point and divided the result by the standard deviation of that attribute.

Another method that we used to prevent wide range feature is logarithmic transformation. We computed the log on each data point respectively to its sign (by restoring its sign after applying the logarithm over the absolute value of the data point). Finally we implemented a polynomial features expansion over the whole data set. This process consist of raising the value of each data point attribute to multiple degrees.

From Figure 2.2, we note that with our data separation into 4 subsets according to the jet_number, we substantially reduced the number of features that are highly correlated. However as two columns still have a correlation of .4, some improvements can still be reached.

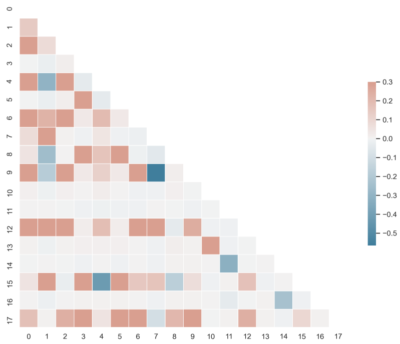

Figure 2: Correlation matrix of the features for the group of samples with PRI_jet_number 0

2.3 Machine Learning Models

Among the six machine learning models, four of them are used for regression issues (least squares, least squares with gradient descent, least squares with stochastic gradient descent and ridge regression). The two other models (logistic regression and regularized logistic regression) are used for classification problems. Each one of these models (apart from least squares) take as input multiple hyper parameters such as:

- max_iters : The maximum number of iterations of the algorithm
- initial_w : An initial vector of weights (initiliazed to $\vec{0}$ in our case)
- degree : The degree used for the polynomial expansion
- gamma : The incremental step or the learning rate of the algorithm
- lambda : The regularization parameter that add a penalty to big weights

The performance of the algorithm is related to the choice of these hyper-parameters. To optimize these values, we have implemented a grid search algorithm combined with a 4-fold cross validation on each of the four subsets. This algorithm iterates over a list of possible values for each hyper-parameter (grid search), train the model on 75% of the training set and test it on the remaining 25% (cross validation). The algorithm then selects the parameters that lead to the average minimum loss on the test set.
We have implemented this algorithm on all the models that rely on hyper-parameters (least square gradient descent, least square stochastic gradient descent, ridge regression, logistic regression and regularized logistic regression) in order to find the best parameters and test them.

3 Results

3.1 Performance of models

We evaluate the performance of each model by uploading on AIcrowd website the predicted values for the test set. The server output the percentage of data points that were properly classified between Higgs boson and background noise.
In the table below, we listed the actual model used for prediction and their respective accuracy.

Table 1: Performance of the different ML models

Models	Accuracy
Least Squares	74.5%
Least Squares Gradient Descent	71%
Least Squares Stochastic Gradient Descent	68.5%
Ridge Regression	80.6%
Logistic Regression	73.6%
Regularized Logistic Regression	73.6%

We can easily see that the best predictions were computed thanks to Ridge regression model.

3.2 Further discussion

Initially, we were expecting that a logistic regression (regularized or not) would lead to the more accurate predictions of our test data as it is a classification problem (since logistic regression is a classification algorithm). But after analysing the results we clearly noted that the most suitable model for this problem was the Ridge regression, giving us an accuracy of 80.6%. Similarly we also discovered that standardizing our model reduced our accuracy of about 10%. This decrease could be explained by the fact that some features may be more important than others. By standardising the data we would then loose this information and indeed decrease our accuracy.

4 Conclusion

Throughout this project, we have developed and tested six important machine learning models in order to make predictions for the detection of the emission of Higgs bosons. We also gained practical experience in the implementation of such methods, abilities to compare various methods and, tools to improve models such as features processing, polynomial features expansions, logarithmic transformation, cross validation and grid-search for the selection of hyper-parameters. As a result of the improvement methods, we were able to reach 80.6% of accuracy in for the prediction with our best model.