Higgs Boson Machine Learning Project 1

Iris Toye, Alexei Ermochkine, Mathilde Morelli
EPFL

***Abstract*—The discovery of the Higgs boson allows to explain why particles have mass. However, the measures taken by physicists don't allow a direct identification of the boson. In order to classify a signature, as coming from a Higgs boson or from noise for example, machine learning is necessary. The algorithm described in this paper allowed accurate classification of 76.7 % of the signatures. Some applications of the discovery of Higgs boson allows to determine if some particle has a mass, or even to find signs of dark matter.**

I. Introduction

The goal of this project is to classify signatures from elementary particules to predict if they are Higgs boson or not. In order to do that, the dataset from the CERN has been preprocessed and standardized. Then, a classification algorithm using logistic regression has been run, in order to determine which signature from the test set came from a Higgs boson. The efficiency of this prediction has been assessed through different metrics : precision, recall, and AUC.

II. Models and Methods

A. Pre-processing

1) Missing data: At a first glance, one notices the dataset is missing several inputs, marked as '-999' in the .csv file.

features	0	23,24,25	4,5,6,12,26,27,28
n.of missing data	38114	99913	177457

For the training/testing of our model, we worked on data points with no data missing, therefore we removed the entire lines that had missing data.

Of course, that left out more than 50% of our points which was too considerable of a loss for our model. Note that the lines that had missing data were not entirely obsolete: only about a third of one line's data was missing. A solution was to replace missing data with the mean or median of its respective feature. For the features with more than 50% data missing (for columns 4,5,6,12,26,27,28), the median was used, while the mean was used in the opposite case (columns 0,23,24,25), in order to minimize the effect of outliers when there were no sufficient data.

2) Investigating correlations in the dataset: An easy way to get a better accuracy of our model as well as minimizing computational cost is to remove highly correlated data. This kind of data brings no additional information about the experiment (w.r.t the other features it is correlated to) and is therefore a computational burden. We noticed that some features showed a Pearson's correlation coefficient of over 0.85. After removal of said features, running a logistic regression never made a difference in our loss function or accuracy.

$\lvert C_{9,21}\rvert = 0.900$	$\lvert C_{9,29}\rvert = 0.955$
$\lvert C_{21,23}\rvert = 0.876$	$\lvert C_{4,6}\rvert = 0.836$

We chose to remove features 4, 21, 23 and 29. The effect was a better run-time of the regression, although the accuracy did not increase in a noticeable way.

3) Polynomial expansion: After the removal of unnecessary columns, we expanded the model with polynomes. We chose not to expand the columns corresponding to angles, as it did not make a lot of sense. Moreover, we made the hypothesis that the features were independents, and selected the best degree for each column independently from the others.

4) Standardizing the data: As a first attempt we normalized all the data. But after an examination of each feature's distribution, we noticed that most of them didn't follow a normal distribution. Therefore, we categorized each feature, and standardized them accordingly, depending on the distribution observed. The categories are : features that follow a normal law, feature of which the log follows a normal law, angles and non-normal features. We applied cosine function on the angles, normalized normal distributions with their mean and standard deviation, and standardize the other feature by subtracting the mean and dividing by (max - min).

5) Categorical data: We noticed that the feature number 22 (PRI jet num) was discrete, and indicated a way to measure the data (number of jet used). Therefore, we thought it did not make a lot of sense to include it as it was in the model. That is why we created dummy variables : four columns, each one corresponding to one value of PRI jet num (0,1,2,3). For example, if PRI jet data was 2, we put a 1 in the column corresponding to "2", and 0 in the other.

B. Regression and classification methods

- A: Gradient Descent with MSE
- B: Stochastic Gradient Descent with MSE
- C: Least Squares
- D: Ridge Regression with cross validation to find best lambda
- E: Logistic Regression
- F: Regularized Logistic Regression
- G: K-nearest neighbors classification

Unfortunately, the K-nearest neighbors classification only worked on small samples, and takes too long to execute on the a big subset of the data set. This confirmed that the KNN classification has better use for small dimension data sets or small data sets.

Figure 1 allowed to choose which classification method to use. Since linear gradient descent and least squares are

regression methods, we chose to use log_regression, and to try to improve the result with reg_log_regression.

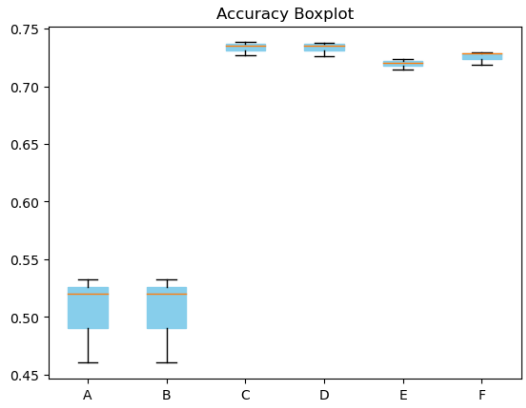

Fig. 1. Boxplot showing accuracy of methods A-F

C. Choice of processing

In order to choose the processing of the data to apply, we compared the mean accuracy of each method. We compared the following processing :

1) Logistic regression
2) Logistic regression + normalization
3) Logistic regression + normalization + w0 (offset)
4) Logistic regression + standardization depending on the distribution + w0
5) Logistic regression + standardization depending on distribution + w0 + high correlation features removed

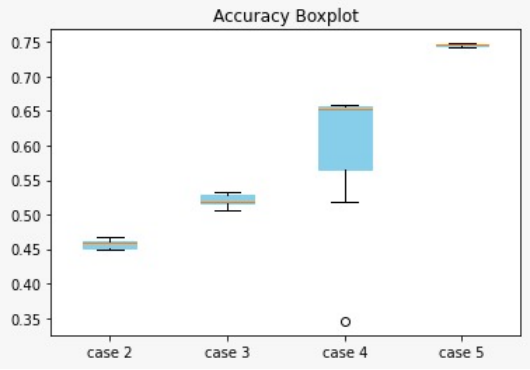

Fig. 2. Boxplot showing accuracy of methods 2-5

The accuracy of the data without normalization (case 1) could not be obtained, because of NaN problems.

We can clearly see the upgrades on the mean accuracy allowed by the different methods. The high variance of method 4 is not very satisfying, but did not show on other tests.

We decided to work with the method 5, with addition of polynomial expansion and creation of dummy variables instead of the categorical feature (PRI jet num) (these methods giving the highest accuracy on the whole dataset).

D. Evaluating the performance of a model

In order to evaluate how well our model can classify binary outcomes, we used a number of different metrics, in addition to accuracy: the precision $\frac{TruePositive}{TruePositive+FalsePositive}$, the recall $\frac{TruePositive}{TruePositive+FalseNegative}$, and the AUC of the receiver operating characteristic (ROC) curve.

III. Results

A. Comparison of the methods

Figure 3 shows the AUC score (of 0.80) for Logistic Regression, which we ended up using for the run.py.

We compared the accuracy of different methods earlier. The method with the highest accuracy (0.767 by testing the whole dataset) is the method with addition of the polynoms and dummy variables. Therefore, this is the one we chose to use.

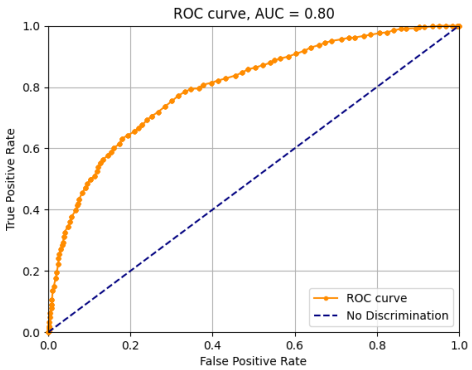

Fig. 3. AUC of ROC

IV. Discussion

The progression of the accuracy has been quite satisfying. One of the best improvement came from the adding of an onset. The innovation of our paper came from the different standardization for each type of data, that made the accuracy increase by more than 10 points compared to the normalization.

Still, the accuracy is not sufficient, especially compared to the best accuracy achieved by the other groups. It can be due to the fact that the regularized logistic regression did not provide a good improvement for us, or the fact that our computers could not support the run of a lot of iterations. Moreover, the processing of the data could have been more efficient.

V. Summary

Overall, we successfully classified the signatures from elementary particules and predicted with a O.767 accuracy whether they were a Higgs boson or not. Through preprocessing, classification using logistic regression and the use of various efficiency metrics, we were able to create a satisfyingly good model.

Acknowledgements

Some methods, especially for cross validation were inspired by labs from the Machine Learning course.