ML Project 1: spot the Higgs boson

Manuel Leone, Gabriele Macchi, Marco Vicentini
Department of Computer Science, EPFL Lausanne, Switzerland

***Abstract*—Machine Learning techniques are becoming increasingly popular nowadays and they are used in several and different fields of science. These are useful when we deal with complex and high dimensional data sets which, sometimes, represented output of scientific experiments. Thanks to this project we are trying to implement some of these tools in order to work on a problem of binary classification on a data-set which come directly from CERN.**

I. Introduction

The aim of this project is to find the best method to distinguish signals of the High Boson from the background noise thanks to binary classification, we did this by using machine learning methods. We used the CERN's dataset which summarizes collision experiments. The dataset includes 250.000 categorized events with 30 features for each column. The huge amount of data made us face the problem of selection of features needed and how to approach null values. The data processing is a fundamental step of the project, followed by machine learning methods implementations. We start with the implementation of basic algorithm as Linear Regression, Ridge Regression and Logistic Regression. Implementation and discussion of these algorithms are in Sections II-A and II-C.

II. Models and methods

Implementing good machine learning methods requires skills in the fields of data analysis and algorithm design. Understanding the story behind the data is fundamental to choose the right direction in the following implementations. On the other hand, the broad panorama of ML algorithms in literature provides several possible implementations, able to change dramatically the performance of the algorithms. Our analysis will develop in depth inside this two aspects. Before to getting inside the developing process, the reader is advised that the preliminary step performed is a **dataset division**. Indeed, in order to have a idea of the goodness of fit and to compare the methods, the algorithms were trained on 80% of the available dataset and tested on the remaining 20%.

A. Basic ML implementations

The implementation of the base methods consists of six well known Machine Learning algorithm such as Gradient Descend, Stochastic Gradient Descend, Least Squares, Ridge Regression, Logistic Regression and Regularized Logistic Regression. The null values of the dataset were initially filled with the median of the column. Firstly, the computation concerned the entire dataset. In the table below different hyperparameters and results of the methods are shown. The accuracy achieved is computed on **our test set**. These results were not satisfactory, so we decided to move on with data processing, as described in the next section.

	Hyperparameters Used				
Methods	λ	γ	Degree	Max Iter	Acc.
Gradient Descent	/	10^{-6}	1	2000	0.718
Stochastic GD	/	10^{-7}	1	1000	0.704
Least Squares	/	/	1	/	0.741
Ridge Regression	10^{-8}	/	6	/	0.766
Logistic Regression	/	10^{-10}	1	10000	0.735
Reg. Logistic Regression	10^{-8}	10^{-10}	1	10000	0.735

TABLE I
Accuracies of the six algorithms without feature processing

B. Dataset analysis and preprocessing

The analysis of the dataset has been performed in different steps and each of them led to gradually improve the methods performances. The examinations we carried out are the following:

1) NaN analysis

The official documentation [1] reports the dataset variables as primitive (`PRI_`), raw quantities directly measured from the collisions detector, or derived (`DER_`), which are computed by the ATLAS physicist using the primitives. This information is useful to understand the physics means behind another fundamental property: the dataset presents many -999.0 as placeholders for missing values. So, our first aim was to understand how we could impute these *NaN*.
The analysis of the documentation shows a strict correlation between *NaN* and the `PRI_jet_num` quantity. The jets are, from our understanding of the physic background, the sub particles produced in a collision. The main point to understand here is that some quantities are intentionally undefined because impossible to compute or no-sense with a number of jet equal to 0 or 1. With this cardinal concept in the head, we proceeded to a dataset categorization in four different subset based on the `PRI_jet_num`. As a result, we produced only columns totally full or totally empty from *NaN* values, with the only exception of the `DER_mass_MMC` one.
At this point, we dropped all the columns taotally filled with *NaN* values. For the remaining missing elements we tried to use another time classical imputation methods such are using mean or average of the other values in the column. But this procedures never led us to a drastic gain in performance. Skilled by the experience of the previous division, we decided to carry out a second split based on *NaN* remainings in `DER_mass_MMC`, producing a total of eight subset, totally cleaned from all missing values.

2) Correlation analysis

To better analyze each of the features we move further to plot the scatter matrix between all of them in all the subsets,

in order to understand which of them have an high correlation. We discovered that at this point of the preprocessing pipeline some columns had the same values and a correlation of 1.00. We hypothesize that this features have the same physical meaning when considered only in the subset. To simplify our model and improve the prediction we decide do delete one of the two features.

3) Skewness removal

The scatter plot matrix visualization pointed out the presence of many right-skewness distributions features. We applied cube root transformation to this columns to transform them into centered distributions.

4) Adding 1s column and standardization

The last steps of the main preprocessing phase are data standardization followed by adding an all ones column. The former is performed to achieve better computation performance. The ones column precedes the feature matrix to be used as constant for the weights vector.

5) Polynomial features

We proceeded to polynomial expansion of each subset by taking powers and stacking them together. A note on our polynomial implementation is that we perform the raising only from degree 1, as we already add an all ones column. The exact degree chosen for each subset is presented below, in the context of using Cross Validation to increase Ridge Regression performances.

C. Model processing

With our processed data, we can move to the second improvement task we have to perform: tuning the hyper-parameters of our best models. We chose to dedicate our efforts only on Ridge Regression and Logistic Regression, as we saw at the very beginning of the project (TABLE I), that these two methods outperforms the others. For both of them we started by choosing the polynomial degree of all the subsets before moving on the λ and γ selections.

For the Logistic Regression, we noticed almost immediately that using polynomial does not increase our performance sufficiently, so we decided to use only the first order degree. For the Ridge Regression our first tests pointed out an opposite situation, with a decreasing test loss by increasing the polynomial degree.

After our previous tests, we performed a 12-fold cross-validation for the Ridge regression on our train set to choose the polynomial degree and the λ parameter. For the Logistic regression we could not do the same due to time constraints of the cross validation, so we choose the values for λ and γ manually. The hyper-parameters best choices for the Ridge regression are here reported.

Data-set	degree	λ
Jet 0 without mass	7	$2.4^* \ 10^{-9}$
Jet 0 with mass	8	10^{-10}
Jet 1 without mass	5	$4.89 * 10^{-10}$
Jet 1 with mass	8	$5.30 * 10^{-9}$
Jet 2 without mass	5	10^{-10}
Jet 2 with mass	8	10^{-10}
Jet 3 without mass	2	10^{-10}
Jet 3 with mass	8	$4.89 * 10^{10}$

TABLE II
RIDGE REGRESSION HYPER-PARAMETERS SELECTION RESULTS

III. RESULTS

The final results we achieved are computed using the tuned dataset only with Ridge Regression and Logistic Regression, using the hyper-parameters and degrees our tests and cross validation helped us to achieve.

	Hyperparameters		Results	
	λ	γ	Accurancy	F1-score
Ridge	\	\	0.824	0.733
Reg. Logistic Regression				

TABLE III
FINAL RESULTS FOR RIDGE AND REGULARIZED LOGISTIC REGRESSION

IV. SUMMARY

Overall, we would conclude we are sufficiently satisfied of the results we achieved with our implementation, although improving it is surely possible but have would require extra works and skills both in data analysis and hyper-parameters tuning.

We learnt how preprocessing is crucial when dealing with Machine Learning implementation. We could have increased the goodness of the result by using optimal methods to clean our dataset or by adding important features, as example by performing cross-terms expansions. We are convinced the presence of an expert in the topics we were dealing with would be helpful too. It this way we could have performed a better selection of the features based on deeper and insightful knowledge. It is also true, for sure, that the implementation of more advanced methods as Neural Network or Decision Tree would help us to reach high values of accuracy.

REFERENCES

[1] C. A.-B. et al., "Learning to discover: the higgs boson machine learning challenge," 2014. [Online]. Available: https://higgsml.lal.in2p3.fr/files/2014/04/documentation_v1.8.pdf