The Higgs challenge: A Machine Learning Practice Opportunity

Louis BETTENS, Louis GOUNOT, Xavier NAL

***Abstract*—In order to make sense of the increasing amounts of data produced by contemporary scientific advances, the assistance of computers and advances statistical techniques has become necessary. The field of machine learning consists of the study and use of algorithms that have the ability to find and recognize patterns hidden in vast amounts of data, and use them to make accurate predictions. This paper attempts to apply basic machine learning techniques to a dataset collected at CERN as part of a physics experiment.**

I. INTRODUCTION

Higgs Boson are elementary particle which explain why other particles have mass. When two protons collide at high speed, a new particle can be created as a result and rarely these particles are Higgs Boson. In the context of the Machine learning course, we worked on the CERN's datas in order to estimate if the decay signature comes from Higgs Boson or other particle. To do so, we will perform binary classification on those decay signatures and test different algorithms in order to have the best accuracy .

This report is organized as follow : Part 2 explains how we clean the datas and normalize them, Part 3 is the exploratory part of the datas, in part 4 we talk about feature processing,in part 5 we explain which algorithm we choose and in part 6 we talk about our resuts.

II. DATA CLEANING AND NORMALISATION

Since data in the real world comes in various formats and conventions, it makes sense to examine the representation of the input data to try to adapt it to the algorithms and programs we intend to use. The first clean-up we applied to the data is to make sure that missing data was represented using a not-a-number value rather that a sentinel value of -999 as was the convention in the original dataset. [1] While this convention is unambiguous in itself, it can hurt the accuracy of algorithm if they try to treat it as a meaningful datapoint.

The next step was to normalise the distribution of each feature by applying an affine transformation to bring the mean to 0 and the standard derivation to 1. This is common practice in machine learning since it allows to give equal weight to every feature in the eyes of distance functions.

To account for outliers, we then removed any data point that had any feature located more than 4 standard derivations away from the mean. This removes 5.01% of data points.

III. EXPLORATORY DATA ANALYSIS

In order to gather some insights as to how the data is structured, we first conducted a data exploration phase by plotting features together 2-by-2. We used the Bokeh data visualisation library in a Jupyter notebook that is provided alongside our code.

Our overall insight is that both classes have distinct but overlapping distribution. For example in figure 1, we can see that the orange class tends to be higher on feature 1 than the blue class, although both classes harbor counterexamples by this metric. Given that all pairs of features on their own show ambiguity, we expect that the logistic regression method might perform better since it can take that ambiguity into account. We also noticed that feature 22 only takes 4 distinct discrete values, for example on figure 2 Since perceptron-based classification algorithms might not have the ability to derive a good decision boundary in some cases with discrete, enumerated values, we decided to split this feature into 4 boolean features, each indicating one of the 4 distinct raw values of feature 22.

IV. FEATURE PROCESSING

After our exploratory data analysis discoveries, we can now do the next part of the project : the feature processing. We add a feature of constant nonzero value to create a homogenous coordinate systems that allows perceptron-based algorithm to avoid sticking to the center of the coordinate system.

As mentioned before, we remove the 22nd feature, and add 4 boolean features that encode each of its discrete values.

V. ALGORITHM SELECTION

We explored two algorithms: logistic regression, and ridge regression with polynomial feature extension. We split our training dataset into a test dataset proper (80%) and validation dataset (20%) to tune hyperparameters.
In the ridge regression, we test different degree for the polynomial feature and the degree 7 gives us the best result and the lowest cost function (0.734 for training and test). The parameter we use is a lambda of 0.0001. To compute the cost function, we use RMSE cost function which is calculated from the MSE cost function:

$$RMSE = \sqrt{2 * MSE} \tag{1}$$

In the logistic regression, we had some problems. We modify the numerical range of y between [0;1] using the sigmoid function. To compute the loss function we implemented the negative logarithm likelihood. We test the regular logistic regression with a lambda of 1, a gradient descent and different iteration between from 1000 to 10 000. But with method we did not get a good result compare to the linear regression.

VI. Results

Our best performing algorithm both on the validation dataset and in the competition was ridge regression with $\lambda = 10^{-4}$ and polynomial expansion up to degree 7. It achieved 80.6% categorical accuracy on the test regression. Logistic regression achieved at most 68%, which is only slightly better that the accuracy of the default submission which puts every test datapoint in the same category. (65.8%)

We can also confirm that, within the selected ridge regression algorithm, our feature processing decisions both improve accuracy on the validation dataset. Together, they give a 30% improvement. Our treatment of not-a-number values also helps by 1%.

VII. Conclusion

Overall, we find that perceptron-based machine learning algorithm are able, given the appropriate guidance, to recognize patterns in the data we were given and apply those to new data accurately. We also find that polynomial feature expansion can enable them to operate on non linearly separable data as well. We were not able however, to make good use of the logistic regression method. It is possible that this is due to a coding error on our part. Our results might be further improved by combining logistic regression or regularized logistic regression with polynomial feature expansion. Other methods, such as adaptative boosting, decision trees, or multi-layer perceptrons, migh also yield different and interesting results.

References

[1] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau, "Learning to discover: the higgs boson machine learning challenge," Tech. Rep., 2014. [Online]. Available: https://higgsml.lal.in2p3.fr/files/2014/04/documentation_v1.8.pdf

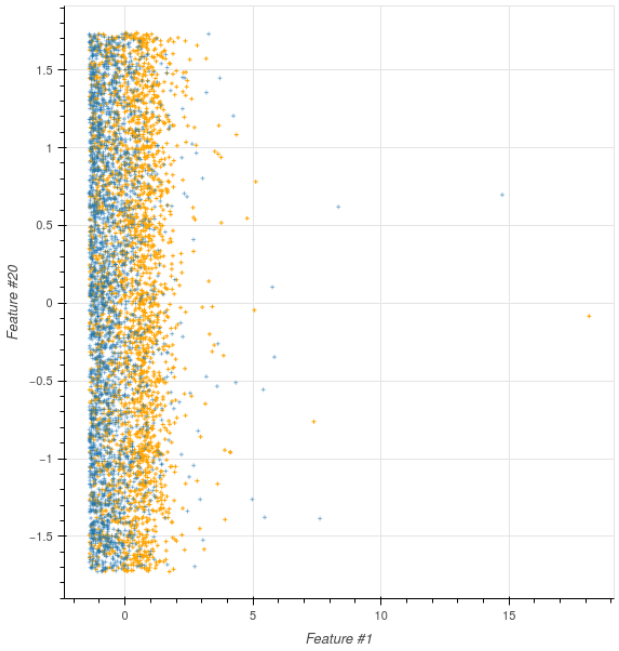

Figure 1. Example plot of Features 1 and 20

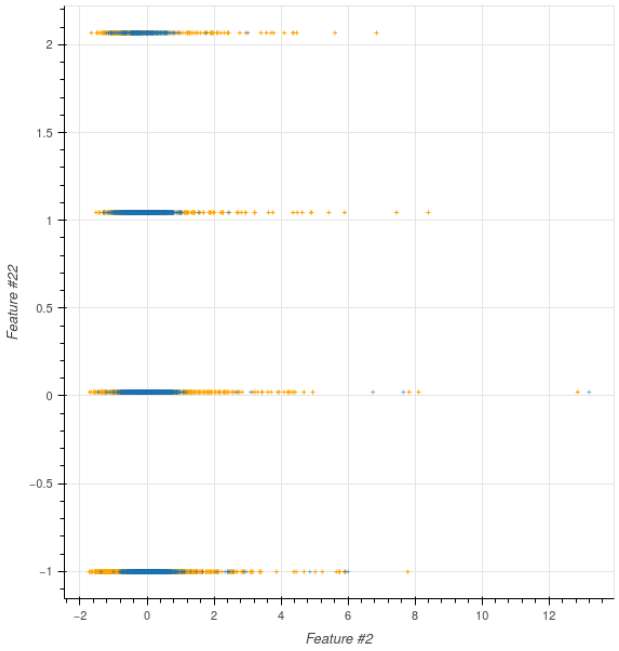

Figure 2. Example plot of Features 2 and 22