Higgs boson : Machine Learning project

Rhita Mamou, Zied Mustapha, Kepler Hugo Warrington-Arroyo
CS 433 Machine Learning, EPFL , Switzerland

***Abstract*—This paper is the result of a Machine Learning project based on a real-world data set : the decay signature of a collision event of particles. In this report, we show all our attempt in order to predict whether the observations we have are caused by the decay of Higgs boson (”signal” : s) or by other particles (”background” : b).**

I. Introduction

The Higgs boson is an elementary particle that explains (among other things) why other particles have mass. Hence, the data set we have is a set of observation from the ATLAS experiment of CERN particle accelerator data. In this experiment, protons are smashed together in the hope to produce Higgs bosons. Because of the rapid Boson decays, scientific cannot observe them directly. That's why we have to measure the ”decay signature” in order to predict whether the collision was from Higgs boson or not. In order to make the best predictions, we first cleaned the data and did some feature processing and engineering. After that ,we used many machine learning algorithms for binary classification and manage to pick the model that maximized the accuracy on the test set.

II. Exploratory data analysis and data cleaning

With a training data set of 250 000 data points and prediction labels (s or b) and 30 features, we can do various methods to explore and understand our features and their behaviour.

- We started by plotting histograms of all the features in order to have an idea of the distribution of each feature. We noticed that the data set have many elements with the value '-999.0'. This corresponds to the in data set NaN value, i.e. missing values. Also, we detected the only discrete feature : ”PRI_jet_num” (see 1), meanwhile all the others are continuous. Hence, this discrete variable (that has only 4 values) can help us categorize the data by splitting it into 4 categories.
- Thus, we decided to split the data into 4 different data sets to help us analyse and clean the data set. But also, it could be useful for classifying signal vs background datapoints more accurately. We first plot the feature graph for each category to visualize the relevance of each feature for predicting the particle. For every category, we select the relevant from the irrelevant features (for which their value do not change) that we discarded from the data set.
- Still having the missing value (-999.0), we need to clean the data by replacing this NaN value. After wondering which would be the best replacing value, we decided to replace by the mean value, since we were able to obtain slightly better results with it.
- We then plotted a correlation matrix to better understand the relations and dependence between features of each category (see 2). This matrix showed us that some features had low correlation with other features, we could thus multiply them to engineer new attributes for the model to perform predictions on.

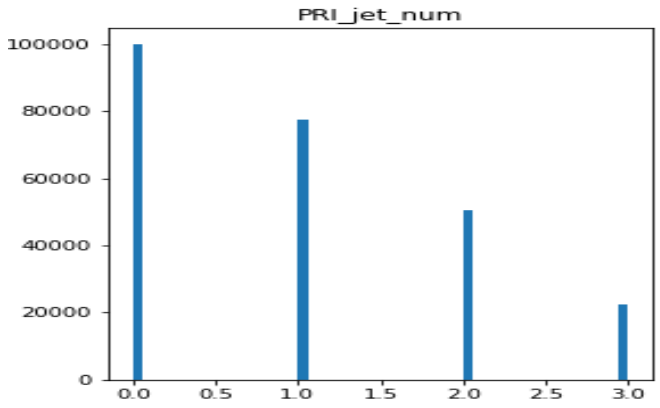

Figure 1. Distribution of the feature PRI_jet_num.

III. Feature Processing

A. Data standardization

We used standardization of our data to re-scale it and have a mean of 0 and a standard deviation of 1, to obtain a distribution closer to the normal distribution which performs better and to avoid overflow errors.

B. Interaction between features

From the correlation heatmaps, we notice that some features have low correlation with all other features (i.e. the line or column is uniformly colored and the color indicates a value close to 0). These low-correlation features can be pairwise multiplied with all other features to engineer new attributes for the model to perform predictions on.

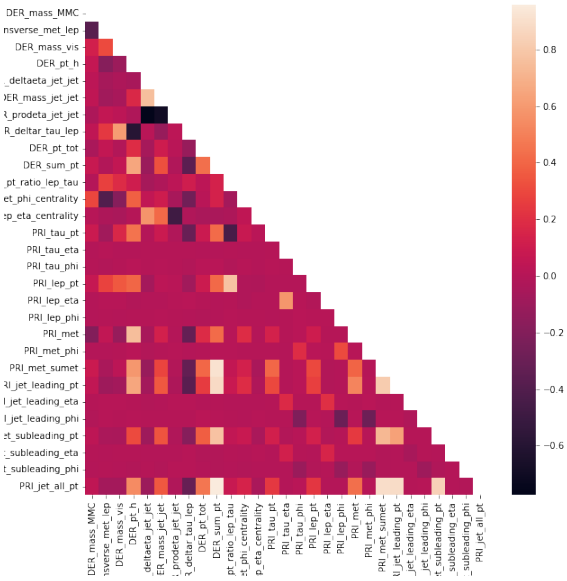

Figure 2. Correlation matrix of all features in 4th category

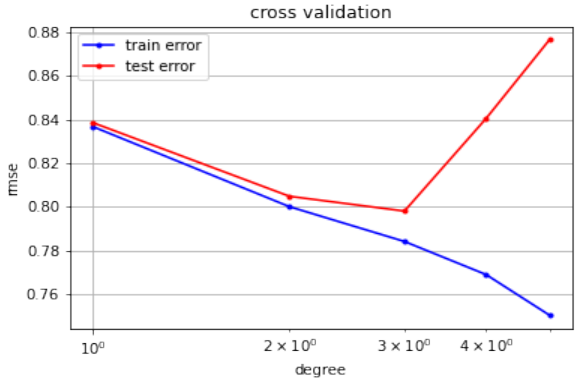

Figure 3. Cross validation for degrees in 3rd category.

C. Polynomial Expansion

We wanted to introduce non-linear terms in this project, so we tried to expand our features using polynomial expansion and applying cross-validation on our different categories, and then to select the best lambda (between 10e-6 and 10e-2) and the best degree to minimize our test and training error.

D. Cross validation

To select the best parameters (the degree of expansion and the lambda) we use the cross validation k-fold method for each of the four categories. Here is the results:

- 1st category: best degree = 1 and best lambda = 10e-6
- 2nd, 3rd and 4th category: best degree = 3 and best lambda = 10e-6

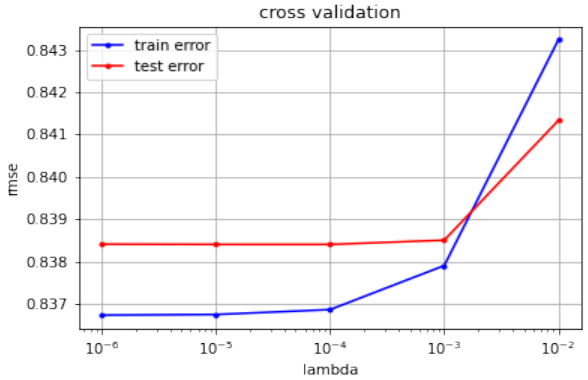

Figure 4. Cross validation for lambdas in 3rd category.

IV. Models and Choice of Implementation

We applied the methods seen in class, with a 90-10 train-validation split on each category of the train set. We first started to use least square model with a gradient descent, then a stochastic gradient descent and finally with normal equations. But after cross-validating , we found that the best lambda is 10e-6, and not 0, that's why we used ridge regression. After doing the cross validation for degrees, we discovered that the best degree was 3 and not 1, so we did a polynomial expansion on the data.

V. Results and Discussion

After using cross validation with ridge regression on our features engineered, we obtained an accuracy of 0.797 on aircrowd. We tried to do the logistic but it took too much time, but we believe it would have improved our model.

VI. Conclusion

After testing multiple models and methods we got the best result using ridge regression with specific parameters stated above. Logarithm transformation would be the next step to maybe have a more precise prediction. We can also conclude that it is possible to predict if a particle is a Boson or not using non-linear ridge regression with an accuracy around 0.8.