ML Project 1 - Higgs Boson

Borden Sven, Jesslen Artur & Viennot Valentin

***Abstract*—The purpose of this project is to explore data analysis to understand a dataset and its associated features. We learn to do feature processing to clean our dataset and then extract the most valuable information. At the end we implement machine learning to train and predict if data match to Higgs boson.**

I. Data Cleaning

Before doing any process to determine a model, we need to be sure the data we are using are consistent and well known. We are first cleaning the data in different ways. We can see in Figure 1 an example with a feature (DER_mass_MMC), the comparison visually shows how important it is to clean our data.

A. Missing values

For each column, we replace all missing values (-999) with the median of the column computed with all the non-missing values.

B. Outliers

For each column, we replace all the outlier values. We consider as outlier values out of the 1^{st} percentile and the 99^{th} percentile range. We replace those values with the median calculated previously.

C. Normalization

We normalize each column independently to have a common scale without distorting differences in the ranges of values.

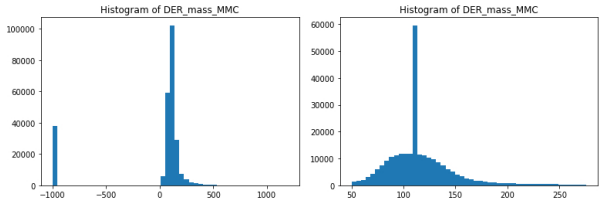

Figure 1: Distribution visualization of the feature DER_mass_MMC before and after data cleaning (without separation of jets)

D. Jet Separation

The first step to understand our data is to analyze the given features. Most of the features are floating points data. When we take a closer look, we can see that a feature is different from the others. The feature PRI_jet_num is indeed very interesting. Depending on its value (0, 1, 2 or 3) some of the other features can be undefined and are therefore meaningless. To avoid a bad influence on the results of these features, we separate our data into 4 categories that will be processed separately. For each jet, we will remove the undefined features.

II. Split Test & Train datasets using cross-validation

In order to avoid over-fitting, we split the dataset into two parts: a training dataset and a testing dataset. The first one is used to generate and optimize different weights. The second one is used to evaluate how our model performed as we know the correct answer related. We pay attention to have the same distribution of the variable in the two datasets. The ratio between the two datasets is also important, we chose 80% for the training data-set and 20% for the testing dataset by using the cross-validation method with a *K-Fold = 5*. The cross-validation allows us to minimize the influence in the choice of the sub-datasets and therefore avoid to over-fit the data.

III. Feature extension

After having removed meaningless features, we now have only some linear features, but they might be too simple to correctly fit the real model. A good way to remedy this problem is to extend our features.

A. Apply general functions

First, we apply the logarithm function on some features. These features (DER_mass_MMC, $DER\_mass\_transverse\_met\_lep$, DER_mass_vis, DER_pt_h, $DER\_mass\_jet\_jet$, DER_sum_pt, $DER\_pt\_ratio\_lep\_tau$) are chosen manually. They have all exponential growth and can, therefore, have a lot of influence on the results. We also apply the hyperbolic arc sinus function on one feature ($DER\_prodeta\_jet\_jet$) to linearize this feature.

B. Calculate the momentum

Then, after some research, we find that the momentum (eq. 1) of different particles such as the hadronic tau, the lepton, the leading jet and the sub-leading jet can be calculated with some given features.

$$P = \frac{P_t}{sin(tan^{-1}(2e^{-\eta})} \tag{1}$$

C. Add polynomial basis

Finally, we extend our input data by adding a polynomial basis on each feature. The degree of the polynomial extension is chosen by comparing the loss, accuracy and F1 score for each degree on the test subset of data with cross-validation.

IV. Feature selection

As we have seen in section III, we now have lots of features in our custom dataset. We now will select features that contribute most to our prediction variable. Indeed, having too many variables can decrease our prediction accuracy.

A. Stepwise regression

We choose the stepwise regression as a method of choice of variable. As we can see in Figure 2, this method considers a variable for addition or subtraction at each step based on a specific criterion (adjusted R^2 in our case) [1].
Specifically, we selected the forward selection which starts with no variables and, step by step, adds the one which improve the model the most. We stop the regression when none of the unadded variables could improve statistically the model.
The choice of this method appears to be the best choice as we have a large number of potential explanatory variables.

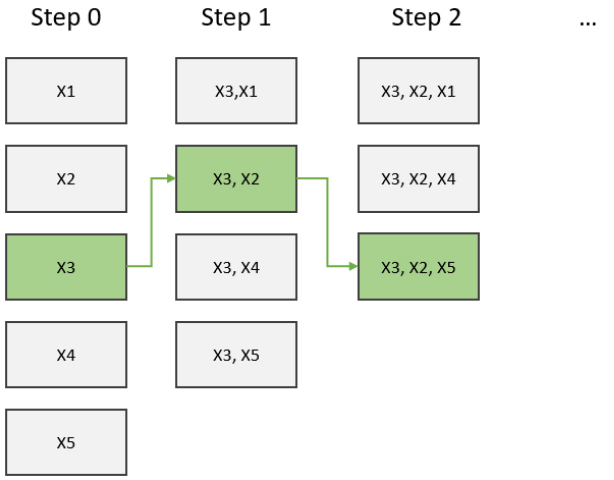

Figure 2: Forward step-wise. This method starts with no variable and step by step adds the one who improve the most the model

Adjusted R^2: We use adjusted R^2 as a criterion to add a variable to our model as written in equation 2, with n the number of points in the data sample, k the number of independent regressors.

$$R^2_{adj} = 1 - \left[\frac{(1-R^2)(n-1)}{n-k-1}\right] \tag{2}$$

R^2 shows how well data points fit a curve. The adjusted version also indicates how well data points fit a curve but adjusts for the number of terms in a model. If we add more irrelevant variables to the model, the adjusted r-square will decrease. We use the adjusted version as we only have data from samples and not the entire possibilities. When we add an independent variable from a step-wise step, the adjusted R^2 will penalize it.

V. Methods

A. Choice of method

The method used to find the best vector of weights is the **ridge regression**. Some of our features are highly correlated, this method adds a small bias factor to the variables to alleviate this problem. We didn't use the logistic regression because of the failure of the likelihood maximization algorithm to converge for our data.

B. Define hyper-parameters

To define our hyper-parameters, we use the following simple process:

- First we build a new polynomial basis of arbitrary chosen degrees (e.g. 1 to 12)
- Then, we compute the ridge regression with cross validation with different lambda included in an arbitrary interval (e.g. between 1e-4 and 1e3) to find the lowest error and thus, the corresponding lambda for each polynomial basis.
- Finally we choose the best polynomial basis based on the computed accuracy, and F1 score and the loss on the test data set.

VI. Results

A. Best result

We produce our best result using a Ridge Regression with specific lambdas, and with the jets separation. We obtained an accuracy of 82.4% and a F1 score of 72.9% on submit 23462. As we don't split the data in the file "run.py" we can always reproduce the same result.
As we work on four sub-datasets regarding the value of PRI_jet_num, we train four different models with predefined hyper-parameters before merging them together at the end. We compute relative model accuracy with cross-validation as mentioned in section II. Also, the data is cleaned before any use.

B. Feature selection result

As we mentioned in section IV, we think it is possible to increase our performance to predict Higgs Boson with a feature selection. Unhappily, the results drop when we select features with our forward step-wise function, which includes ridge regression (we also tried with the logistic regression this method was not able to converge to a solution). This convergence failure could be a consequence of data patterns known as complete or quasi-complete separation [2].

C. Ways of improvement

Our method could be improved, first of all, our features extension and their selection with the forward step-wise algorithm could be a way to improve our predictions. A second way we thought of was to use the logistic regression. The logistic regression is indeed better adapted for classification problems. The fact that an event is over-represented in a training dataset (e.g. the background events) can induce a bias on our predictions.

References

[1] S. Andale, "Adjusted r2," 2013. [Online]. Available: https://www.statisticshowto.datasciencecentral.com/adjusted-r2/

[2] P. D. Allison, "Convergence failures in logistic regression," 2008. [Online]. Available: https://www.researchgate.net/publication/228813245_Convergence_Failures_in_Logistic_Regression