Machine Learning Project

The Higgs boson challenge

Ahmed Ezzo, Emna Fendri, Mohamed ELasfoury
Department of Computer Science, EPFL, Switzerland

ABSTRACT

We discuss in this paper different supervised machine learning models to detect the presence of the Higgs boson using the data from the CERN particle accelerator.

1 INTRODUCTION

The Higgs boson is an elementary particle in the Standard Model of physics and results from a proton-proton collision. Given the decay signature of a collision, our goal is to identify if it results from a Higgs boson (signal) or something else (background).
Nonetheless, this binary classification problem is very challenging due to the nature of the data. We discuss during the paper how we tackled this challenge by analysing and exploring data using several methods in order to find the best way to process the data and finally feed it to our models.

2 MODELS

For this task, we consider basic machine learning algorithms for regression, namely the *least squares*, *ridge regression* , *logistic regression* and *regularized logistic regression.*
After using a 4-Fold cross validation to get the hyper-parameters of these algorithms, we attempt a first run to get an idea on how the raw data behaves. We split the training data into 75% training set and 25% test set.After training the model on the 75%, we use the 25% to test the accuracies. The results are summarised in the following table.

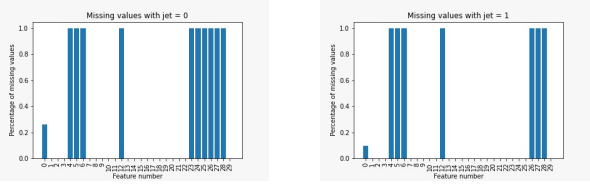
Method	Accuracy(%)	γ	λ
GD	69.528	10^-8	NA
SGD	70.606	10^-8	NA
Least Squares	74.402	NA	NA
Ridge	74.388	NA	10^-4
Logistic	68.796	10^-12	NA
Reg Logistic	68.796	10^-12	10^-8

Table 1: Results without cleaning or performing expansions

3 DATA PROCESSING

Feeding the machine learning model the best data possible is an important step. We control the numerical values that are used by the model and hence can have more predictable behaviours. This can lead to a big increase in our classification accuracy.

Machine Learning - EPFL,
2021.



(a) ***PRI_jet_num*** $= 0$ **(b)** ***PRI_jet_num*** $= 1$

Figure 1: Percentage of missing values for the two first partitions

3.1 Exploring Data

- The training set provides 250000 events of 30 features. We can distinguish two types of features. There are primitive variables that are "raw" quantities measured by the detector, and derived variables that are computed from the primitive values.
- All variables are floating point, except *PRI_jet_num* which takes 4 possible values : 0,1,2,3.
- Some variables take the value of -999 which indicates that they are meaningless or undefined. Note that this value is guaranteed to be outside the normal range of all variables.
- The data set is quite unbalanced. In fact, 66.6% of the data in the training set is labeled as a background event.
- An important observation is that according to the number of jets i.e. *PRI_jet_num* variable, there is a significant number of features that we can discard due to missing values. As we can see from **Figure 1**, for 0 and 1 jets, there are 10 and 7 features out of 30 respectively that show a missing value for 100% of the events. We also noticed missing values occurring for *DER_mass_MMC* feature (index 0) for roughly 6% of the events for all of the partitions.
  We will see how the result from learning when data is put into categories will give different (and better) results when the data is combined. This phenomena is known as the Simpson's Paradox.

3.2 Data Processing

3.2.1 *Data Cleaning.*

1.Partitioning and feature selection

The first step is to partition the data according to the number of jets. We will define the partitions as $J0$,$J1$,$J2$ and $J3$ for data points where $PRI\_jet\_num = 0$,$PRI\_jet\_num = 1$, $PRI\_jet\_num = 2$ and $PRI\_jet\_num = 3$ respectively. For $J0$ and $J1$ we discard the features indexed by $\{4, 5, 6, 12, 23, 24, 25, 26, 27, 28\}$ and $\{4, 5, 6, 12, 26, 27, 28\}$ respectively. For $J2$ and $J3$, all features are relevant so we keep the same set of features for both of these partitions. At this point, we

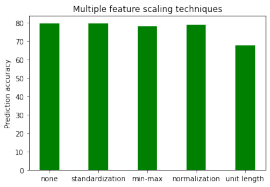


Figure 2: Accuracies of Least squares from multiple feature scaling techniques after grouping by jet number and handling missing values

considered training data by combining $J2$ and $J3$, but categorizing them gave us better results.

2.Handling missing values

After dropping some features according to the number of jet, there are still undefined values for few features. Multiple solutions are available to replace these faulty values. We could replace them by the feature-wise mean of the correct values, or even the median. We finally decide to replace these missing values with the feature-wise median since it gave us the most consistent results. Which is not very surprising as the median is not as influenced by outliers as the typical mean is. As we can see from **Figure2**, in the least squares case, the accuracy increased from 74.4% to 79.7%. This is thanks to grouping by the Jet number and handling missing values.

3.Handling outliers

We can detect and visualize outliers using box plots. We chose to map these values to the maximum and minimum values that we set at 6 standard deviations from the mean. In fact, this appeared to be a safe practice as we are not able to identify whether an extreme value is part of the population of interest or not.

3.2.2 Feature Transformation.

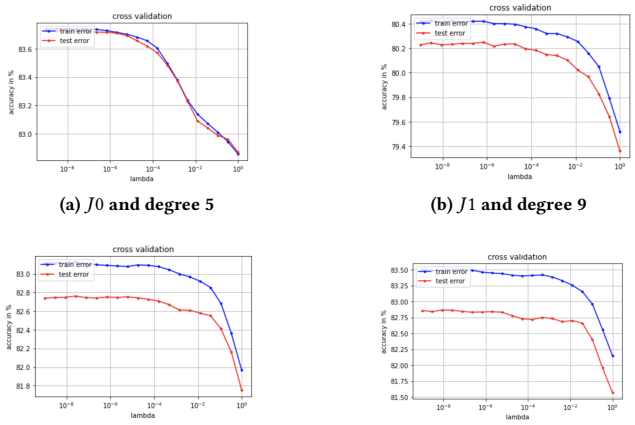
1.Standardization

As we can see, the different features often show different variances. We therefore standardize the data i.e. subtract the mean and divide by the standard deviation for each dimension. After this processing, each dimension has zero mean and unit variance. This is done to ensure that no feature dominates the other. For instance, the *DER_deltar_tau_lep* feature has small values (between 0 and 4) while the *DER_mass_MMC* feature is two orders of magnitude bigger. If we don't do this standardization, the former feature would be always dominated by the latter. From **Figure 2**, we can see that there is a slight improvement in accuracy thanks to standardization (from 79.7 % to 79.8%). When using feature expansions later, we notice that this standardization decreases the accuracy, therefore we don't use any feature scaling on the data.

2.Feature Expansion

For this task, expanding the features helped to better expose the important relationships between input variables and the target variable. For each partition of the data, We chose to perform a polynomial expansion on each feature up to a degree that has been selected by cross-validation and by tuning it with other hyper parameters.

We show in **Figure 2** the plots for a 4-fold cross validation using *ridge regression* to select the best λ. Note that we considered



(a) $J0$ and degree 5 **(b) $J1$ and degree 9**

(c) $J2$ and degree 8 **(d) $J3$ and degree 9**

Figure 3: 4-Fold Cross Validation on λ for *ridge regression* with the optimum degree for each partition

to evaluate the accuracy rather than the root mean square error (RMSE).

4 RESULTS

We summarise in the following table the final results after training on the processed data. Note that the optimum values for hyper parameters γ and λ are the same for the 4 partitions. Nevertheless, as mentioned in **Figure 2**, the optimum degrees differ according to the number of jets.

We get the best result from Least Squares with 82.89% of accuracy.

Method	Accuracy(%)	γ	λ
GD	76.66	10^-8	NA
SGD	74.51	10^-8	NA
Least Squares	82.89	NA	NA
Ridge	82.79	NA	10^-8
Logistic	71.87	10^-11	NA
Reg Logistic	71.87	10^-11	10^-4

Table 2: Results after data processing

5 SUMMARY AND DISCUSSION

For this binary classification problem, we expected *logistic regression* to perform better than *Least Squares* since the latter aims to minimize the mean-squared error, which counts positive and negative deviations from the class label i.e. $\{-1, 1\}$ equally bad, when only one of them lead to a missclassification. Although the results are not too far from each other, this can be due to the choice of the hyperparameter λ via cross validation for *ridge regression* according to the accuracy and not the RMSE.

The results also show that partitioning and training the data according to the jet number boosted significantly the performance for all of the considered algorithms.