Finding the Higgs Boson
Machine Learning Project 1

Carlos Megías, Jorge Sánchez and Roser Viñals
École polytechnique fédérale de Lausanne, Switzerland

***Abstract*—Machine learning (ML) has become a key enabler for the analysis and classification of data. In this project, a CERN's database has been used to create a model for predicting whether a decay signature of a collision event represented by a vector of features was signal (a Higgs boson) or background (something else).**

I. Introduction

This report has been divided in five sections. Section II presents the different techniques considered for feature preprocessing in order to deal with the CERN's data set particularities. Concerning the predictive model, Section III justifies how the hyperparameters and the cost function have been selected. Section IV presents and analyses the results obtained, and Section V concludes the report.

II. Data Preprocessing

In this section, we describe the feature processing techniques implemented. First, the main challenge faced is how to deal with -999 values, which correspond to features of some data points that are meaningless or cannot be computed. We have studied different techniques: a) standardizing each feature (considering the -999 values); b) changing the -999 values by the mean of the others values of each feature (each column); c) changing these values by zeros; d) removing every entry (row) which have a -999 value in at least one of its features; e) and removing each feature which has at least one -999 value in a entry. For choosing the best technique, we have measured the MSE (Mean Squared Error) and achieved accuracy, defined as the percentage of success, using the Least Squares method as shown in Figure 1. Thus, we checked that the best results are obtained when using technique b), as in terms of MSE as of accuracy (which are obviously strongly correlated since when decreasing the error the accuracy should increase). As the -999 are missing values, it seems more suitable replacing these values by the means than removing a huge amount of data (as done in d) and e)), since this would imply the discard of data that could be useful for creating a more reliable model. From these results, we decided to use technique b). However. once the final model was selected we improve the feature processing. As explained in section IV, the best result was obtained when combining option b) with standardization and outliers removal.

III. Model selection

In this section we describe the criteria and final choices that we have used for obtaining our results.

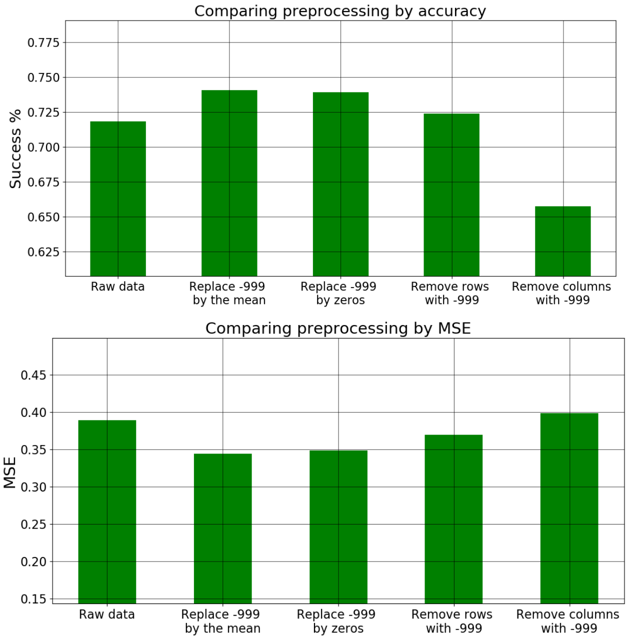

Fig. 1: Feature processing techniques considered

A. Method selection

We have considered two cost functions: the Mean Absolute Error (MAE) and MSE. We have selected the MSE since penalizes large errors. On behalf of the method, after comparing the different techniques implemented such as Least squares (with or without gradient descent), ridge regression and logistic regression, we could observe that the method which was achieving higher performance was ridge regression. This performance has been quantified using the criteria described in subsection III-B.

B. Selection criteria

1) **Minimum error or maximum accuracy selection**: When searching the optimum weights for building our predictive model we aim minimizing the cost function (as explained in III-A, the MSE). However, it may not be the best idea using this criteria for comparing models which are already built; in this case what makes the most sense is to decide which model is better checking the accuracy that the models have achieved, since this indicates its predictions quality, rather than checking the MSE. After trying both options, we concluded that

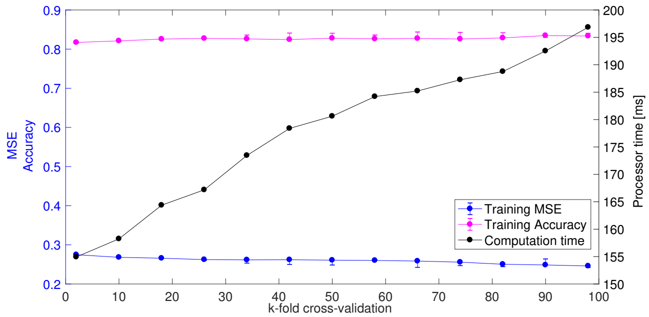

Fig. 2: Cross-validation estimation of MSE, accuracy (mean, maximum and minimum) and computation time for ridge regression with $degree = 10$ and $\lambda = 10^{-6}$.

taking the maximum accuracy was leading us to better results.

2) **Cross-validation**: In order to estimate the error for different models and select the best, we have used one k-fold cross validation. This technique achieves a good compromise between having an unbiased estimate of the generalization error and variance, and a reasonable computation time. Figure 2 shows for different number of partitions, the estimated performance in terms of accuracy and MSE and the computation time required for a fixed model. Since the error estimations does not fluctuate a lot when changing the number of partitions, we have selected a not very high K value, K=6.

C. Hyperparameters selection

First, we have included the polynomial expansion technique. Different implementations for the polynomial expansion have been developed and compared (included in the code) in order to get a different representation of the data that could lead us to better results. The result of applying a determined *degree* polynomial to each feature is a set of new entries with $degree \times NumOriginalFeatures$ features. In order to perform this proceadure, the polynomial *degree*, which is our first hyperparameter, needs to be determined in order to get the best model.

Second, for ridge regression, another hyperparameter is required: λ. This hyperparameter adjusts the regularization for preventing overfitting. The regularization used is L2-Regularization which takes the euclidean norm of our weights vector. For the selection of the two best hyperparameters, we first explored a wide range of possibilities and then, we have narrowed down this range until the final values were selected. Using k-fold cross validation (see 2), we have iterated over the different ranges and we have obtained the results in Figure 3. We see that the accuracy reaches the maximum when $degree = 11$ and $\lambda = 3.73 \times 10^{-5}$, the same values than the ones used for the Kaggle's submission.

IV. Results

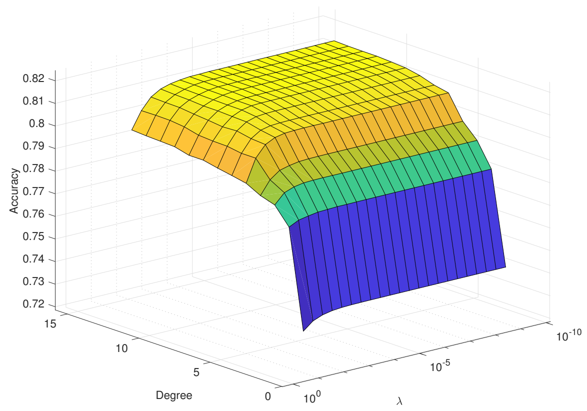

Fig. 3: Hyperparameter selection: $degree = 11$ and $\lambda = 3.73 \times 10^{-5}$

The final results have been obtained after applying the preprocessing techniques described in Section II. We have first removed the -999 values and replaced them by the mean of the corresponding features. Afterwards, the data has been standardized. After the standardization, the outliers have been removed by setting a threshold in -5 and 5, i.e. the features whose values are bellow -5 or above 5, have been replaced by 0. This threshold could have been optimally selected by computing the best one for each feature. However, for simplification, we considered the same threshold for all the features and we just tried few values in the range between 1 and 7. The better performance was achieved when taking 5.

By iterating over a wide range of values of the two hyperparameters considered, and taking the ridge regression with the MSE function, the model has been selected and tested. The achieved accuracy is 0.8173 in the Kaggle's submission. It is important mentioning that the best results have been obtained when:

1) Replacing the unmeasured values (-999) by the corresponding means, standardizing the data and removing the outliers by setting a threshold.
2) Using k-cross validation. For computation reasons, we have used 6 partitions.
3) Using ridge regression to minimize the MSE has led us to better performance than other techniques such Least Squares or Logistic regression.
4) The hyperparameters selection has been critical. A wide range was considered initially and has been narrowed until the optimal values were selected.

V. Summary

In this project, we have developed a classifier that predicts whether a collision event has occurred or not. We have shown that the choice of the feature preprocessing technique and the hyperparameter selection is critical. The best performance was achieved when using 6-cross validation and ridge regression technique for computing the weights with $degree = 11$ and $\lambda = 3.73 \times 10^{-5}$, achieving a final accuracy of 81.784%.