# Project 1 Report - Higgs Boson Machine Learning Challenge

Manon Dorster, Alexandre Maillard, Antonin Faure

Date : 31/10/2022

**Abstract**

The aim of our project is to apply machine learning methods to CERN particle accelerator data to determine Higgs boson generation across multiple proton collision events. From a dataset consisting of feature vectors representing the decay signature of collision events, our goal was to predict whether the events consisted in a signal (a Higgs boson) or background.

## I Introduction

The project consisted in developing a model from a training data set in order to predict whether events described by feature vectors in a test set corresponded to a Higgs boson generation event or not. In the first place, we analyzed our datasets and cleaned our data by ridding it of insignificant features. The next step was to implement learning algorithms to generate models that could fit our data, based on regression and classification. After comparing the scores for each method, we concluded on a machine learning method that was most adequate to predict our data.

## II Data Analysis and Feature Engineering

### a. Data Set Description

The data we were provided with for this project consisted in:

- A training set of 250 000 collision events with 30 features and a label column (-1 or 1). The label -1 corresponds to a background event and the label 1 stands for a signal event.
- A test set of 568 238 events, organized in the same manner as the training set except for the empty label column. Our work consisted in accurately predicting the labels for the test set.

### b. Data Cleaning

For some entries, variables were meaningless and could not be computed, and were therefore set to -999.0. We replaced all meaningless values with the mean value of the corresponding feature column. We thought of deleting features columns with a majority of -999.0 values (>50%), but as shown in Figure 1, the accuracy of the predictions for the "dirty" original data is higher than the "clean" one obtained while removing features. Thus, we chose not to remove such features.

### c. Training Set Split Method

By looking at the correlation between the features with a high proportion of meaningless values, we found that the 22nd feature (which takes integer values in $\{0, 1, 2, 3\}$) was highly correlated to the meaningless -999.0 values. By splitting the data set into four sets according to the 22nd feature value, we observed that in the first and second sets there were respectively 10 and 7 features with 100% of -999.0, which could thus be fully removed. After splitting the rows according to the value of the 22nd feature, we obtained four sets which corresponded to 39.97%, 31.02%, 20.15% and 8.87% of the original training set, containing respectively 19, 22, 29 and 29 feature columns.

### d. Data Standardization

Standardization is an important step in machine learning and reduces the risks of feature matrix ill-conditioning. We first shuffled our data to reduce noise. We then implemented a function to standardize our data sets by subtracting the mean and standard deviation of the training data from the data. In fact, it is important that both data sets be standardized using the training mean and standard deviation in order to obtain relevant prediction results.

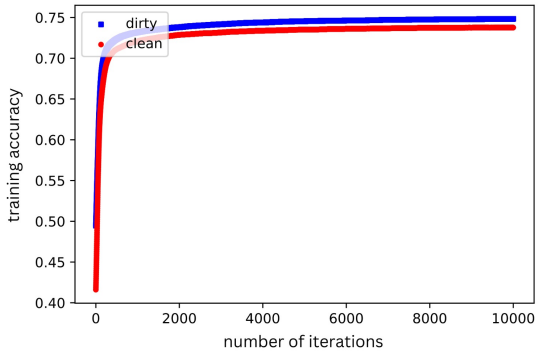


Figure 1: Logistic regression training accuracy as a function of the number of iterations using the original training data (blue) and the filtered data (red).

# III Model Prediction

We began by implementing the functions we were asked to implement and tested the following models on the given test set: mean square error with gradient descent, mean square error with stochastic gradient descent, least squares and ridge regression. The results are indicated in Table 1. In this section, we detail only the most relevant prediction models.

## a. Ridge Regression

For the ridge regression model, we extended our features to polynomial features and implemented a 4-fold cross validation algorithm to determine the most optimal degree for the polynomial basis (the degree that minimizes the root mean square error over the validation sets) and the most optimal ridge parameter $\lambda$. Extending to polynomial features reduces underfitting, and k-fold cross validation reduces the risk of overfitting. We found that degree 2 feature expansion seemed to be the best choice to fit our data without overfitting. The accuracy score for the ridge regression model with degree 2 feature expansion is displayed in Table 1.

## b. Logistic Regression

Since the labels take two values (-1 or 1), it was obvious that the data was classified into binary classes: one class corresponding to a Higgs boson generation signal and the other to background signal. Hence, the most adequate choice of prediction model seemed to be logistic regression.

We first implemented a logistic regression algorithm, which indeed produced more accurate results (as shown in Table 1). We performed some hyperparameter optimization on $\gamma$ and tested our model with varying numbers of iterations. We observed that $\geq$15'000 iterations did not significantly improve the results.

Then, we extended the feature matrix to a polynomial feature matrix, since we esteemed that our predictions were undergoing underfitting issues. We implemented a degree II optimization function similar to the one described in Subsection a.. We then found the most optimal values for $\gamma$ for the logistic regression predictions with degree II feature expansion and a 4-set data split (see II c.), and obtained that $\gamma = 0.1$ for the first and fourth sets and $\gamma = 0.01$ for the second and third ones were the best choices. With a training accuracy of 82%, our predictions with degree II feature expansion were clearly better and reduced underfitting of our model. We attempted a degree III expansion but it did not improve our results and was, in addition, far too computationally costly.

## c. Regularized Logistic Regression

Lastly, we thought of improving our prediction model by adding a regularization term to our logistic regression of degree II, in order to further reduce overfitting risks.

# IV Computational Costs

## a. Degree II Feature Expansion

For our degree II polynomial expansion, we computed every cross-term $\mathbf{x_i x_j}$ for $\forall i, j$ and $j \leq i$, which is costly compared to simply computing all $\mathbf{x_i}^2$. In addition, it increases the computational cost of the model fitting algorithms since it significantly increases the number of features.

## b. Logistic Regression

For each iteration, we computed the log loss function and we decided to break our function if the loss did not change enough according to a threshold. The breaking technique allowed for the algorithm to stop running through unnecessary iteration loops and thus to run quicker.

# V Results

We used several methods for our tests, abbreviated as follows : **D-II** : degree II polynomial feature expansion, **S4** : split in 4 sets according to 22nd feature value (see II c.), $\gamma_i$ : $\gamma$ value used on set $i$.

| Model | Training Accuracy | Test Accuracy | Remarks |
|---|---|---|---|
| Least squares | 0.806 | 0.803 | done with D-II and S4 |
| Ridge regression | 0.806 | 0.804 | done with D-II and S4 with $\lambda = 0.001$ |
| Mean square error GD | 0.806 | 0.774 | done with D-II and S4 with $\gamma_1 = 0.1$ and $\gamma_{2,3,4} = 0.01$ |
| Mean square error SGD | 0.807 | 0.781 | done with D-II and S4 with $\gamma_1 = 0.1$ and $\gamma_{2,3,4} = 0.01$ |
| Logistic regression | 0.754 | 0.751 | done with 15'000 iterations and $\gamma = 0.01$ |
| Logistic regression degree II | 0.827 | 0.825 | done with 15'000 iterations, D-II, S4 with $\gamma_1 = 0.1$ and $\gamma_{2,3,4} = 0.01$ |
| Regularized logistic regression degree II | 0.826 | 0.825 | done with 15'000 iterations, D-II and S4 with $\lambda = 0.001$, $\gamma_{1,4} = 0.1$ and $\gamma_{2,3} = 0.01$ |

Table 1: Training and test accuracy for each prediction algorithm

# VI Conclusion

To conclude, we implemented six different classification algorithms. We then performed some optimization on the logistic regression and managed to improve the training accuracy from 75.4% up to 82.7% by using degree 2 polynomial feature expansion and smart data splitting and cleaning. We obtained the best prediction accuracy with the degree II 4-set split logistic regression model.