Machine Learning project 1

Naomie Pont, Elsa Mauguin, Alexiane Guerry

October 2021

1 Introduction

In 2013, the Higgs boson elementary particle was discovered at CERN. The goal here is to recreate the discovery process, by applying machine learning techniques to actual CERN particle accelerator data. Data regroup background signals (noise) and actual decay signals (representing collision processes). To determine if an event resulted in background noise or in real Higgs boson, we need a robust model, that we will try to create here.

First we will pre-process the raw data. Then we will apply cross-validation to get the best hyper-parameter lambda, that we will use to test models such as ridge regression. After each model design, we will use the test data on aicrowd to end up with the best model, with the best weights and minimized loss.

2 Pre-processing of the data

The model we fit has to be trained on the data set which combines 30 features. The model is $y_m = X_m^T$ w where the matrix X has dimensions $[250'000, 30]$ before preprocessing.

We have data for the training and the tests, respectively stored in *xtrain* and *xtest* with sizes $(250000, 30)$ and $(568238, 30)$. We have to preprocess the data, with the hope to reduce their sizes and get shorter running times at cheaper computational costs. First, we visualize the data: Fig[1]. We will compare later with Fig[2] which will correspond to the visualization of the training data after the data pre-processing steps.

The pipeline is the following:
- remove missing features: as we observed many values of -999.0, we have to take care of the corresponding features. If a column has more than 25 percent (threshold that we chose) of these undefined values, we remove it. It leads us to 20 features instead of 30.
- standardization: we standardize with the mean and standard deviation, to have 0 mean and variance 1 distributions.
- visualization after the data pre-processing step

We tried to create dummies variables for the feature 22 that becomes the feature 18 after the removing of the -999.O data as it clearly appeared to be a categorical data, taking 0, 1, 2, 3 value. We were able to visualize the 23 features but unfortunately we did not manage to make it work with our model as it gave a best lambda of 2.00 for ridge regression and 4.00 for logistic regression.

3 Model

After preprocessing the data, we tried different models to fit our data and compared them with their losses, the accuracy and the F1 score accessed with submissions on the aicrowd platform. We began with a simple least squares model. We tested this model with AIcrowd and obtained 0,713 for the accuracy. In fact linear least squares gave a quite good accuracy even if it is a rigid classification method.
The linear regression using stochastic gradient descent gave an accuracy of 0,638 which is significantly lower than simple least square model.

We then wanted to try ridge regression and performed a 3-fold cross validation to find the best lambda. We tried to do a cross-validation in order to find the best degree for it but it was too long to run and to computationally expensive so we tried manually each degree from 1 to 10 and found that 8 was the best. We found a lambda of 2.205. This model gave us an accuracy of 0,684.

We then wanted to try on augmented features with different degree. We tried a cross-validation to find the best degree for it. It gave us 13. With this augmented features version, we got an accuracy of 0,738 which shows an amelioration in the model.

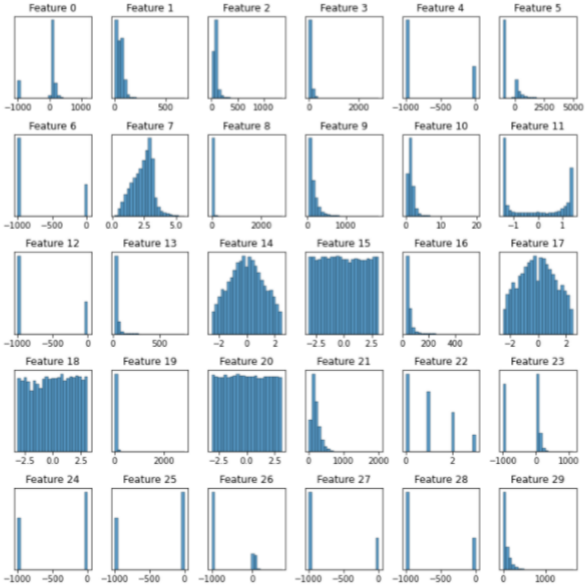

Figure 1: Pre-processed training data histograms in function of the features

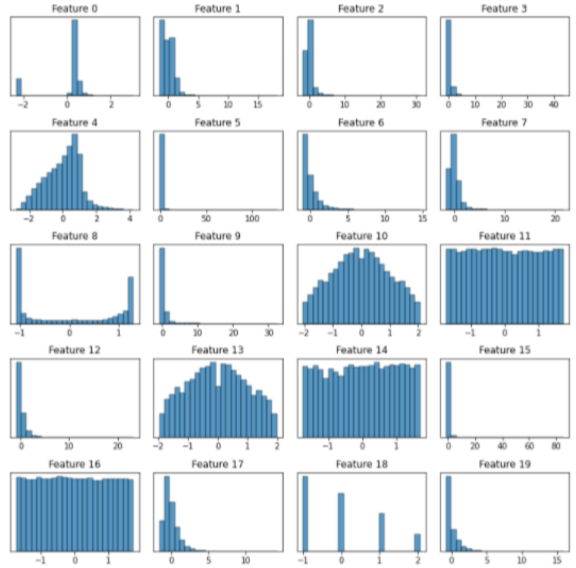

Figure 2: Processed training data histograms in function of the features

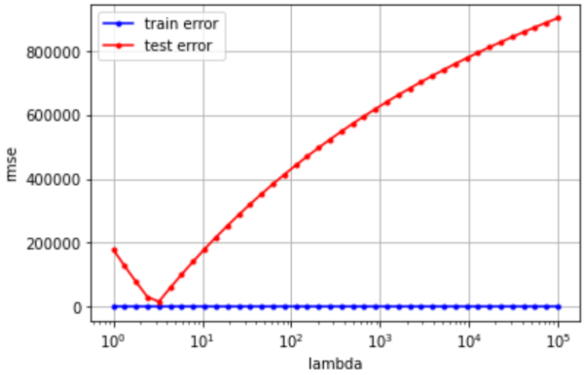

Figure 3: Train and test error in function of lambda for ridge regression

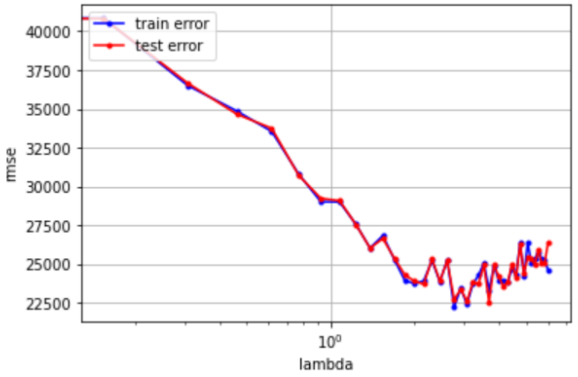

Figure 4: Train and test error in function of lambda for logistic regression

We also tried logistic regression. The result is not very concluding as we can see on the visualization curve of the cross-validation, our training error and test error follow the same curve which means that our model is underfitting. The regularized logistic regression gave an accuracy of 0,705. Ridge regression solves the multicollinearity problem through shrinkage parameter lambda and that's why it is better. We reused the optimal lambda that gives the optimal train and test error.

4 Results

We can observe the data histograms in function of features before and after the data preprocessing on Figure 1 and Figure 2.

The train and test error for ridge regression and logistic regression are displayed on Figure 3 and Figure 4. The best lambda looks obvious for ridge regression and the curves overlaps for the logistic regression as discussed in the previous section.

5 Summary

To conclude, we were given a data set, explored it and cleaned it in order to keep the good data and remove features that would false our model. We believe there is many improvements possible for the prepossessing of the data as for example remove the correlated data.

Among the models we tested, the more convincing and accurate one was the ridge regression performed on the augmented . Surprisingly, the simple least square gave us a pretty good score and that is were we can say there is a world between theory and practice.