Find Higgs Boson

Julien Berger, Marouane Jaakik, Yassine Khalfi

***Abstract*—In this paper, we will detail our contribution to project 1 of the machine learning course at EPFL. Using CERN data, we will create a classifier that allows us to predict whether a particle is a Higgs boson or not, based on certain information. To do this we will perform several data processing and modelling operations that we will describe and give the result.**

I. Introduction

To answer the classification problem, our work will be divided into several steps. First we will do data exploration, once our data is better understood this will allow us to focus on data processing (management of missing values, scaling and expansion of features ...). Finally, we will finish by training and validating our models.

II. Data Exploration

We started our exploration with histograms to see the distribution of our different features and also the difference in the distribution according to the target variable. This allowed us to see that in our data, we had some inconsistent values, in particular the -999 value that was found in some features. We considered the -999 values as missing values. So we find that we have missing values for many features and we even have 7 features where more than 70% of the values are missing. These inconsistent data are linked to the variable PRI Jet num. Indeed according to this variable some other variables are undefined and this explains the values -999, we will consider a specific model for each Pri jet num value and we will detail this later.

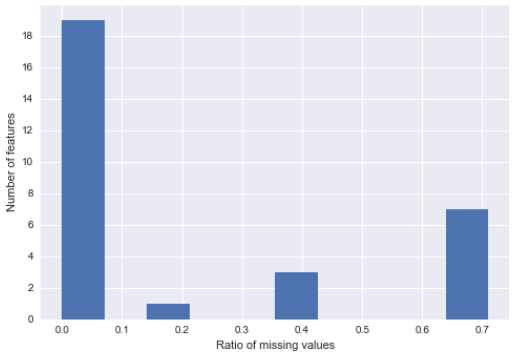

Figure 1: Distribution of missing values in features

During the data exploration we also explored the correlation between the features and the target variable and also the mutual information between the different features to allow us to better understand our data and to have a better idea of the interaction between the different variables. We used pearson correlation for our computing.

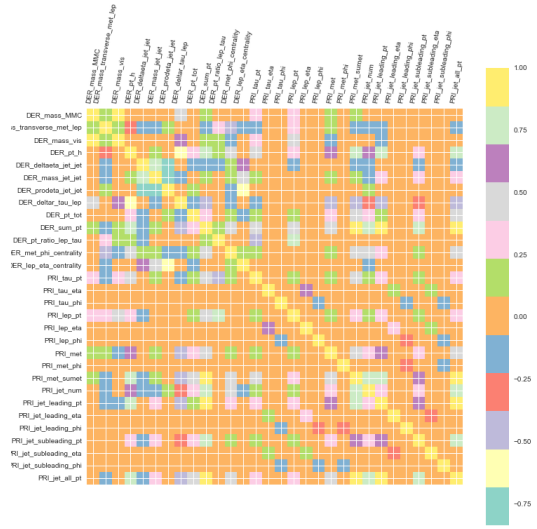

Figure 2: Mutual Features correlation

III. Feature processing

A. Missing values

After splitting the data according to PRI Jet Num, we replace the remaining missing values by the median of the other points. The calculation of the median values depends on the type of particle (Boson or not). We use median since it's a robust statistic.

B. Feature Scaling

To avoid to effect of scale in the different features we rescale the different continuous variables. We tried different strategies of rescaling (Min-Max scaler, Normalization, standardization). Different strategies give similar results, we chose normalization for simplicity.

C. PCA

Polynomial expansion and feature interactions explode the number of features in our dataset. To try to keep as

much information as possible while reducing the number of features we added a PCA. The idea behind PCA is to find lower dimensional representations of data that retain as much information as possible.

D. Data augmentation

To improve the capability of our model, we add new features to fit a non-linear function.

1) Polynomial expansion: We performed a polynomial expansion to add an intercept to each feature and also the polynomials of the different features. We also added interaction features between each pair of features (feature products). We tried several combinations to keep the one with the best results.

2) Bias: We experimented with adding a Bias term, which is a parameter that allows models to represent patterns that do not necessarily pass through the origin.

IV. Models

After all our pre-processing and data cleaning operations, we can move on to modelling. We considered several models: Least Squares SGD, Regularised Least Squares, Regularised Logistic Reg. Thus we have some models with a numerical resolution with gradients (complete or stochastic) as well as models with an analytical resolution like Least Squares. Some of our methods have hyperparameters such as term regularisation which we implement at the level of linear and logistic regression to fight against overfitting. To evaluate our different models we run a k fold cross validation at each step to be able to measure the performance of a model with a certain combination of parameters.

To evaluate the performance of our models in order to determine which models are the best, we rely on one main metric: accuracy. Thus our evaluation pipeline consists of a K fold to measure the accuracy associated with a grid search of our different parameters and pre-processing combinations (lambda for ridge regression, missing value replacement methods, degrees of polynomial expansion, addition of interaction features, etc). Our objective during the training is also to choose a model that will not overfit, that is why we rely for the evaluation on the result of the train accuracy and also the test accuracy. In addition, cross validation allows us to see if our models have a high variance in the result, which is also a sign of overfitting.

V. Models

A. First Results

First we tried to see what results the different algorithms could give us to see which one to focus on to maximise performance. The accuracy results for the train and the test set are available in the table below. The main takeaway from our experiments was that a regularized model performed much better on the test data and avoided some extreme cases of over fitting. In General adding a regularizer term performs significantly better than early stopping. Ridge regression is giving the best trade off between test and train accuracy.

Model	Train Acc	Test Acc
Least Squares	0.93	0.67
Regularised Least Squares	0.83	**0.813**
Linear Regression	0.87	0.72
Regularised Logistic Reg.	0.85	0.79

Table I: Best results with the different algorithms

B. Grid search for best model

We have performed a general grid search, to look for the optimal feature processing steps performed before training as well as the finding the best model with fine tuned hyper-parameters. Subsequently standardization and polynomial expansion followed by a ridge regression performed the best results after our first tests, whereas applying a PCA didn't affect much the accuracy, which at best could be used to speed up training using fewer features. For each jet class, we searched for the optimal lambda, the optimal degree for the polynomial expansion and the optimal values for other parameters.

For polynomial expansion, we explored the range of degree going from 2 to 10 (sometimes including the interaction terms of degree 2). For the parameter lambda, we tried different values following a logarithm range going from 0.1 to 1e-10. This was done for each class and in the end the value of the optimal parameters can be different depending on the class. Optimal results for each class of pri jet can be found below.

Pri Jet	Degree	Lambda	Test accuracy
0	6	1e-09	0.836
1	6	1e-09	0.787
2	6	0.04	0.817
3	6	0.04	0.808

Table II: Optimal parameters for each class with Ridge regression

VI. Conclusion

By using a Ridge regression that we will parameterise for each class of pri jet num we manage to create a model capable of detecting Higgs bosons with an accuracy greater than 0.8. This data split allows us to improve the performance of our model and to reduce the overfitting. It is also necessary to note that depending on the value of Pri jet num the accuracy can vary enormously. In the case of improvement it may be interesting to understand the origin of this disparity and to see how we can align the results of the different classes. This project shows the importance of combining a rigorous machine learning operations pipeline with good interpretation through our insights and knowledge of the research topic.