Project 1 Report: Higgs Boson Challenge

Tommaso Farnararo, Francesca Venturi, Girolamo Vurro
Department of Computer Science, EPFL, Switzerland

***Abstract*—The purpose of this report is to describe inspection and manipulation techniques of a copious dataset - simulated by the ATLAS experiment performed at CERN - aiming to perform data mining and, eventually, reliable predictions.**

I. Introduction

The Challenge is a classification problem for the decay signature of the Higgs Boson, related to the probability of a collision to generate this elementary particle [1]. To reach the highest predictive accuracy, an exploratory data analysis is performed (II), followed by a cleaning phase (III). We then proceed with feature engineering (IV) and, eventually, we adopt standard Machine Learning algorithms to predict decay signature labels (V). To conclude, results are presented (VI) and conclusions are drawn, opening prospects we would have explored if deeper research was possible (VII).

II. Exploratory Analysis

A. Exploration

The dataset is split into a *train* set and a *test* set. The former consists of 250,000 events with 30 features, whose output variable is binary and indicates the decay signature of the collision event: -1 for "background" and 1 for "signal". The latter consists of 568,238 events with the same 30 features. All the research is carried out considering the binary values of 0 and 1, to be coherent with logistic algorithms. Once predictions are available, the reverse conversion is applied.

B. Categorical features

After a rough theoretical study, it emerges that the 22^{nd} feature (*PRI jet num*), is categoric. Therefore, it is convenient to split the train and test sets into three classes according to their categoric values (0,1 and 2 or 3). The last two classes, i.e. 2 and 3, are kept together as they would be lacking a sufficient amount of observations if considered separately. The following studies are performed on every sub-dataset - train and test - separately. Consequently, a further data inspection is performed on every sub-train set, including plots of the empirical distributions of the features, given their binary output.

III. Data Diagnostics and Cleaning

A. Missing Values

Missing values are identified by *-999*. As soon as the splitting happens, it turns out that some features are constant among a given sub-dataset, hence they can be removed. This procedure allows to remove the features entirely made of *-999* in each sub-dataset, as well as the actually constant ones. After this substantial cleaning, all the remaining features do not have *-999* values, except for the first one, i.e. *DER mass MMC*. These missing values are then imputed employing of a Ridge Regression, i.e. predicting them after a learning process in which we regress the non-empty events of *DER mass MMC* over the corresponding observations of the remaining features.

B. Highly Correlated Features

The critical tolerance for correlation among features is set at 0.8, resulting in the removal of approximately 4 columns for every sub-dataset.

C. Outliers

Since there is no a generic definition of *outlier*, the following is adopted: x is an outlier if $|x - \mu| > 2\sigma$, where μ and σ are the mean and the standard deviation of the corresponding feature, respectively. Outliers are approximately 5% of all the observations. Once identified, these values are then capped accordingly.

D. Balancing

The train set is quite unbalanced ($66\% - 34\%$). However, the loss of information due to balancing has negative effects on predictive accuracy. Indeed, the robustness gained with a balanced dataset does not recover the information carried by the observations removed.

IV. Feature Engineering

A. Long Tails and Angles

From the empirical distributions plots, it emerges that almost all the non-negative features have a long tail to the right: this requires performing a *logarithmic* transform - $log(1 + x)$ - of these features. From a theoretical overview of the documentation, it emerges that some features represent angles, namely the ones whose label contains *phi*. They are expressed in radians in the interval $[-\pi, \pi]$ and their distribution is symmetrical around the mean, which is zero. Therefore, a *cosine* transform - $cos(x)$ - is applied: the choice of the *cosine* prevails on the *sine* because it follows the same symmetrical pattern of the aforementioned distributions, i.e. even symmetry.

B. Polynomial Expansion

As far as increasing accuracy is concerned, the most effective phase is polynomial expansion. For each sub-dataset, an intercept is added (constant *ones* column) to the model matrix, followed by the polynomial expansion of each feature: this is done up to an optimal degree in terms of maximization of the accuracy, tuned by Ridge Regression and cross-validation over the hyper-parameters λ and *degree*. As a consequence of this procedure it emerges that, for every sub-dataset, the degree chosen is the highest among the ones provided in the

grid search, i.e. 14: this result will be discussed later (Section VI). In addition, the square root of the model matrix is added, together with coupled cross-products of the feature matrix.

C. Standardization

Lastly, the obtained dataset is standardized, in order to allow the numerical models to converge to a finite solution.

Model	Accuracy
1: OLS on Original Dataset	0.739 $\pm$0.01
2: Split and handling NaN	0.750 $\pm$0.02
3: Transformations and outliers	0.752 $\pm$0.01
4: Polynomial Expansion (d=7)	0.823 $\pm$0.05
5: Add products and $\sqrt{x}$	0.831 $\pm$0.02

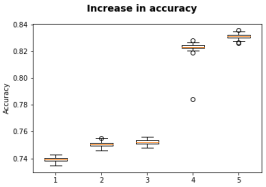

Fig. 1: Increase in train accuracy

The figure above shows the effect of data manipulation on train accuracy, computed through k-fold cross-validation on the train set. It is evident that every step progressively increases the predictive accuracy.

V. Methods

The approximate solution of a linear problem may be attained through numerical algorithms - Gradient Descent (GD) and Stochastic Gradient Descent (SGD) - as well as through analytical closed-form solutions - Least Squares (LS). The latter is preferred, since the numerical methods may not properly converge due to the size of the dataset. Moreover, to avoid *overfitting*, the penalized version of LS, i.e. Ridge Regression, is implemented. The choice of the optimal regularizing hyper-parameter λ is obtained through a grid search among a range of possible values.

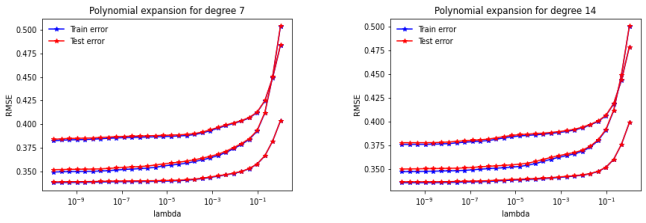

Fig. 2: Train and Test error varying λ

Moreover, in the linear regression case, it is necessary to determine a threshold to convert predictions from continuous to binary values. To this end, we implement cross-validation on the train set, selecting the hyper-parameter t that maximizes the accuracy of the predictions. Clearly, the optimal threshold is centered into the prediction interval, i.e. $t = 0.5$.

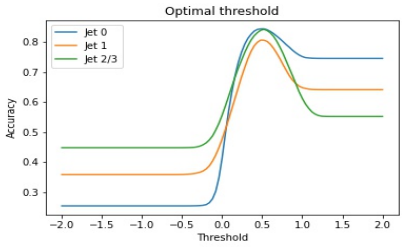

Fig. 3: Accuracy as a function of classification threshold

In addition to linear models, logistic regression by Gradient Descent and Stochastic Gradient Descent are implemented.

VI. Results and Discussion

We train each model on our data, apart from Penalized Logistic Regression due to limited computational power. Train and test accuracy are reported in the table below. Finally, we obtain the best accuracy to be 0.833 through Ridge Regression, with its hyper-parameters tuned as previously explained.

Method	Train Accuracy	Test Accuracy
Least Squares GD	0.765	0.763
Least Squares	0.831	0.830
Ridge Regression	0.830	**0.833**
Logistic Regression GD	0.794	0.786

TABLE I: Train and test accuracy

As expected, the algorithms that perform the best are those with analytical solution. On the train test, Ordinary Least Squares provide a slightly larger accuracy than Ridge, while on the test set the Ridge Regression performs the best. Nevertheless, their values are not too far from other, both on the train and the test sets. It is interesting to observe that penalization does not play a substantial role. Hence, this thesis is supported by two shreds of evidence:

- according to Fig. 2, the penalization parameter λ is almost null for each Jet class, leading to the conclusion that *overfitting* is avoided *a priori*, given the copious amount of data
- the best performing degree of polynomial expansion (Section IV) is the highest provided in the grid search array, meaning that the expansion would be larger if only more powerful computational means were available.

VII. Conclusions and Future Work

The project reveals the significance of pre-processing for classification techniques. In particular, data cleaning and feature engineering are fundamental to handling raw data and increasing prediction power. Different classification methods are analyzed; it's clear how methods with numerical algorithms fail to obtain the same accuracy as those with a closed-form solution. Finally, the result we obtain is satisfactory, also because all the algorithms are quite elementary and not exceptionally powerful. Nonetheless, we are aware that several limitations and challenges can be tackled in future work. First of all, more computational power would allow us to obtain combinations of new features to grow in accuracy. More precisely, it would be interesting to search the polynomial expansion degree such that penalization becomes effective, namely the number of features that make the penalization parameter λ not null. With more time available, other data manipulations such as PCA could have been tried; this would extract the primary information of the dataset while reducing the dimensionality and, as a consequence, decreasing in computational cost.

REFERENCES

[1] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kegl, and D. Rousseau, "Learning to discover: the higgs boson machine learning challenge," 05 2014.