Machine Learning - Project 1

Arthur Passuello, François Quellec, Julien Muster
Department of Computer Science, EPF Lausanne, Switzerland

***Abstract*—When trying to define a statistical predictor on a data-set, Machine-Learning has proven to be quite a powerful tool. However, as for any statistical approach on a problem, a good assessment of the provided data, and its subsequent processing remains of capital importance to obtain meaningful and relevant results. Thus, the way towards optimized results passes through a thorough cleaning and preparation of the data - feature engineering, an exploration of different models to generate predictions and an optimization of its parameters. This paper aims to illustrate the steps of a Machine-Learning approach to obtain a class predictor as accurate as possible for Higgs Boson particles detection, on a dataset from the CERN.**

I. Introduction

First, this paper will describe the required processing of the data-set before it can be fed to any Machine learning model. Through a thorough analysis of the data, it aims to understand its meaning and make the best out of the information it contains. This process, called feature engineering, includes both cleaning the data of meaningless values but also and more importantly proceed to an enhancement of the data-set. This is done through statistical analysis, by identifying meaningful features with respect to the to-be-predicted label and enhance the quality of the data itself by handling meaningless entries, completing missing values or even emphasize features that were deemed relevant. Subsequently, this paper will present an exploration of different Machine learning models, discuss their performances and finally propose an optimized model based on prediction accuracy.

II. Feature Engineering, Models and Method

A. Feature Engineering

As a first step, the data-set has been inspected, semantically and statistically. From this, several issues and improvement opportunities have emerged. This section will describe the data-set and the modification that were applied onto it in order to optimize the information it provides.

Each entry is composed of 30 features, among those, it will be shown that several must be addressed so that the information they provide is better represented.

1 - Outliers and Data Standardization
A simple box-plot of each feature clearly shows that for most features, some entries seem to be outliers that would not fit well in any statistical modeling of the data. A simple solution to this is to winsorize [1] those features and substitute the extreme values with more meaningful ones. After this step, the mean for each feature can be considered more robust and thus, a more relevant standardization can be applied to each feature.

2 - Invalid/Missing Values Inference
The feature `DER_mass_MMC`, which according to the documentation [2] is an estimate derived from other measures, presents $38114 (\approx 15.2456\%)$ undefined values. By computing the correlation between this feature and all the others, we found that it was highly correlate with some of them, a good option seemed to infer those values based on available valid entries. [3]

3 - Inter-feature Semantic Correlation for Data-set Splitting
Many other features show invalid values (-999) which, according to the data documentation, means the value is undefined for those entries [2], depending on their discreet `PRI_jet_num` value. From this observation, a good solution seemed to split the data according to their `PRI_jet_num` values (integer label between 0 and 3) and discard, for each category, the undefined features (it does not make sens to infer this values).

4 - Polynomial Feature Expansion
As presented in the next section, the models used in the scope of this paper are mainly used for linear regression, which presents severe limitations when confronted to features that do not have a linear relationship. With this in mind, it seemed relevant to extend some features on a polynomial basis [4] in an attempt to create a linear representation of those relationships.

B. Models

In the scope of this paper, we will study 2 baseline regression methods and some of their derivations:

Least Square
this model is the most straightforward and presents the least computational complexity of all models discussed in this paper. It can be computed analytically or iteratively with **Gradient Descent** or **Stochastic Gradient Descent**. To avoid overfitting, we can add a regularization parameter to the equation, this technique is called **Ridge Regression**.

Logistic Regression
Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. As for Least square, we can use a **Regularization** parameter to avoid overfitting.

C. Method

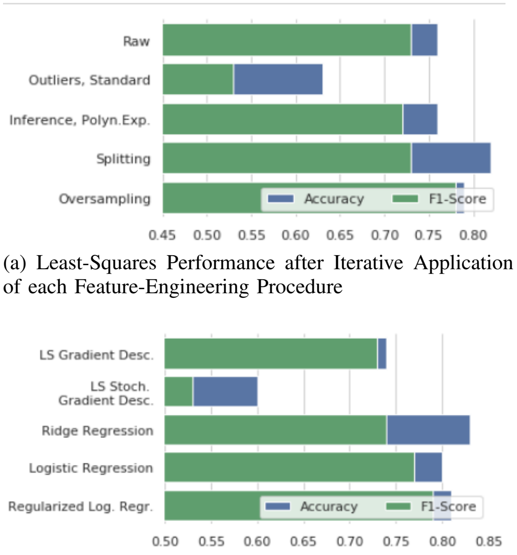
1) Feature Engineering: To assess the improvements brought by the feature engineering, the basic *Least-Squares* model was ran before and after applying each procedure to the data-set. Then, resulting prediction performances were compared. For the Missing Value inference described in the *Feature Engineering* part, we used Ridge Regression because it was a good trade-off between accuracy and complexity, we then chose the best λ with a 4-fold cross validation. Attention was also brought to the *imbalanced data set* issue [5] from the observation that, in the training set, one class had twice the number of entries than the other. An assessment of the impact of this issue and an attempt at resolving it by *oversampling*, i.e. random duplication of the entries from the class in minority until an equilibrium is reached, was done. The results used to assess the performance gain brought by each modification on the features were obtained using the *least-square* model.

2) Model Selection and Custom Model Definition: Once our features engineering done, we compared our different model performance on the same data samples (70% training, 30% testing).

Since the complexity of finding good parameters for a model is very high, we first performed a small random grid search for each model and then chose our best model, after what, we fully tuned all the parameters.

3) Fine-Tuning of Hyper-Parameter: For the proposed final model, parameters such as the regularization factor λ, the wisker size for capping outliers and the degree of expansion were explored using *4-fold cross validation* [6] techniques for each split. Using the same indicators mentioned above, optimal values for those parameters were determined.

III. Results



(a) Least-Squares Performance after Iterative Application of each Feature-Engineering Procedure

(b) Performances of each Method with Optimized Parameters

Fig. 1: Accuracy (blue) and F1-Score (green) Comparisons

A. Featuring-Engineering Results

As shown in figure 1(a), applying the feature-engineering procedures mentioned in II brought a significant improvement in the model accuracy. However, when trying to balance the training data-set using the *oversampling* method, performances worsened.

B. Model Selection

As shown in table 1(b), performances are better with *Ridge regression* rather than *Logistic Regression* and standard *least-square*. However, this result need to be taken with caution, our logistic regression models uses Gradient Descent to iteratively update the weights, but with a dataset of that size, the computation at each iteration is huge and we hadn't the computational power to let the model converge. See *Discussion* part for further details and possible improvements.

C. Final Results and Fine-Tuning of Hyper-Parameters

Using the best *feature-engineering* procedures (i.e. all exception *oversampling*) and the *ridge regression* model, the *fine-tuning* procedure was then started. The results for the λ and *degree* parameters are shown in figure 2. It appears that for $degree = 13$, $\lambda = 1e-4$ and $whis = 2.5$ we have the best stable results for each split, both on the competition platform (**83.1%**) and locally with our test sample (**84.2%**).

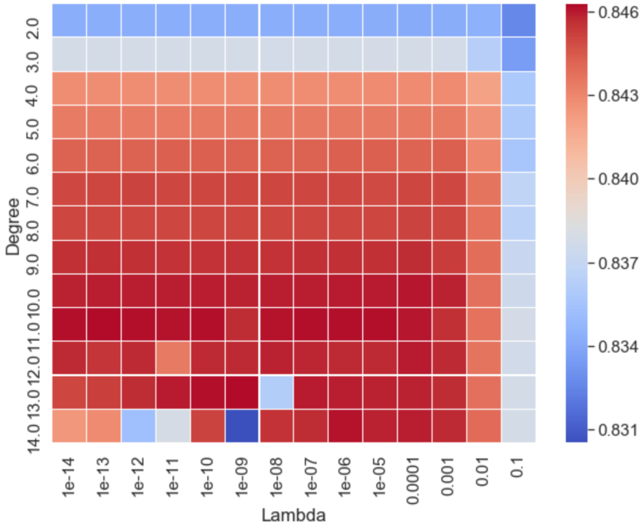


Fig. 2: Accuracy evaluation of Ridge Regression depending on the degree of polynomial expansion and the regularization factor

IV. Discussion

A. Data-set Analysis and Feature-Engineering

The graph in figure 1(a) shows that, using the *least-squares* method, accuracy decreases when the data is standardized and its outliers are processed. This observation remained specific to this method and the procedure was kept in use as it actually improved results with other methods.

Furthermore, results have shown that although the feature-engineering procedures that were applied to the data-set brought accuracy improvements in the predictions, a rather significant improvement was also brought by the realization that the feature `PRI_num_jet` could be used as a classification label for all entries. This shows the importance of studying the semantic behind each feature, the information it brings to the data and its potential impact on other features.

B. Oversampling

Results have shown that, in the case of linear models, balancing the training data-set using an *oversampling* method brought no performance improvement, rather the opposite. In light of this, one could deduct that linear classification models are more sensible to data tethering than label-wise unbalanced training data. This is also proven by the final results that were obtained, which are good enough to demonstrate that lack of balance had little to no impact on the overall prediction performances.

C. Possible Improvements

The choice was made that for the global classification model, the *Logistic Regression* methods would not be used due to its larger needs in terms of computational resources for hyper-parameters tuning and cross-validation. As this method could be considered the most sophisticated among those discussed in this paper, especially for binary classification, it could be expected to obtain better results with it using the same optimization procedures. Note that to reduce the computational needs of the Logistic Regression algorithm, a lot of "techniques" exist. We could for example reduce the complexity of each iteration by using *Stochastic Gradient Descent* or make the iterations converge faster with *Newton's Methods*

References

[1] D. Ruppert, “Trimming and winsorization,” *Encyclopedia of Statistical Sciences*, vol. 14, p. 8765, 2006.

[2] “CERN dataset from the atlas higgs boson machine learning challenge 2014,” http://http://opendata.cern.ch/record/328, accessed: October 28, 2019.

[3] MaytalSaar-Tsechansky and FosterProvost, “Handling missing values when applying classification models,” *Journal of Machine Learning Research*, vol. 8, 2007.

[4] M. Jaggi and M. E. Khan, “Optimization,” *CS-433 Machine Learning Course*, 2019.

[5] S. G. V. P. V. López, A. Fernandez and F. Herrera, “Classification with imbalanced datasets,” *Information Sciences*, 2013.

[6] M. Jaggi, “Generalization, model selection and validation,” *CS-433 Machine Learning Course*, 2019.