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, given a raw data-set.**

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 removing 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. Considering the proportion of those invalid entries and the meaning behind the feature, a good option seemed to infer those values based on available valid entries. [3]

3 - Inter-feature Semantic Correlation for Data-set Splitting
Many features show invalid values (-999) which, according to the data documentation, means the value is undefined for those entries [2]. It appeared that the discreet feature `PRI_jet_num` was the decisive factor of those values definition. 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 the features heavily depending on this label.

4 - Polynomial Feature Expansion
As presented in the next section, only linear models were used in the scope of this paper, 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, only linear regression models were used.

Least Square
this model is the most straightforward and presents the least computational complexity of all models discussed in this paper. As such, it has been used as an indicator to compare the improvements brought by the work done on the data-set.

Least-Square Gradient Descent
and its stochastic variation.

Ridge Regression and **Logistic Regression**
using normal equation.

C. Method

1) Feature Engineering Contribution Assessment: To assess the improvements brought by the feature engineering, the *Least-Square Gradient Descent* was ran before and after applying the aforementioned procedures to the data-set. Then, the obtained `accuracy` and `F1-scores` were used to compare the results. 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) Fine-Tuning of Hyper-Parameter: For each proposed model, parameters such as the maximal number of iteration, the *learning rate* λ and γ were explored using *cross validation* [6] techniques. Using the same indicators mentioned above, optimal values for those parameters were determined. Note that in this step, computational complexity was taken into account, mostly for the number of iteration passed to each model as it quickly increases the time of computation and thus, was deemed a relevant factor in the scope of this paper.

3) Model Selection and Custom Model Definition: Using the same indicators mentioned in the previous paragraph, models are compared based only on the results they provide. No computational complexity variable was taken into account. Combining the observed best model and the most relevant operations on the input data(e.g. feature-wise splitting) , a custom model was then defined for optimal performances. Note that the parameters for each models were determined using *k-cross validation* and that the final comparison used the best results obtained for each model.

III. Results

A. Featuring-Engineering Results

As shown in table III-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.

	Raw	**Outliers, Standard.**	**Inference, Polyn.Exp.**	**Splitting**	**Balancing**
Acc.	0.74	0.75	0.79	0.831	0.79
F1-S.	0.73	0.737	0.791	0.808	0.79

B. Model Selection

As shown in table III-B, performances are better with *regression* models rather than *least-square* ones. Furthermore, it appears that the best performing model is *Logistic Ridge Regression*.

	LS-GD	**LS-SGD**	**RidgeReg.**	**Log.Reg.**	**Log.Rid.Reg.**
Acc.	0.74	0.75	0.79	0.831	0.79
F1-S.	0.73	0.737	0.791	0.808	0.79

C. Final Results

Using the previously obtained results, an optimized model was obtained to best infer the missing data (*cf II*). Results for this model are shown in the first row of table III-C. Then, combining the feature-engineering procedures and the models that showed the best performances, the final model and its parameter was defined as described in table III-C.

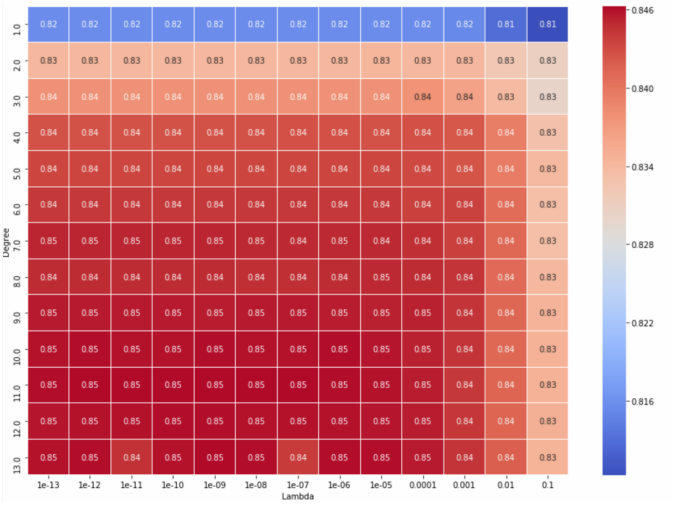

Fig. 1. Caption

Usage	**Name**	λ	γ	**max iter**	**Acc.**	**F1-S.**
Data Inference	Reg	0.1	0.1	10000	0.9	0.9
Final Model	Reg	0.1	0.1	10000	0.9	0.8

IV. Discussion

A. Data-set Analysis and Feature-Engineering

Results have shown that, although the feature-engineering procedures that were applied to the data-set always brought performance improvements, the most significant improvement was 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 performances.

C. Possible Improvements

The choice that was made for the global classification model was not *Regularized Logistic Regression* due to its enormous need in terms of computational resources for parameters fitting and cross-validation. One could then argue that performances could be further improved using this type of classification model.

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 27, 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.