Machine Learning Project: Higgs boson particles classification

Gioele Luca Monopoli, Simone Roznowicz, Stefan Igescu
School of Computer and Communication Sciences, EPF Lausanne, Switzerland

I. Introduction

The project has the purpose to provide a valuable machine learning model which best suits the given problem in the field of particle physics. Given a vector of features representing the decay signature of a collision event, the goal is to predict whether this event was signal (a Higgs boson) or background (something else). The process, followed along the whole project, consisted of these steps:

1) Understanding which models could provide the best solution (i.e. the most accurate predictions) for the underlying problem: in particular Logistic Regression and Ridge Regression were developed, tested and compared using different hyperparameters.
2) Cleaning and preprocessing the dataset, namely performing feature scaling, in particular standardization of the data, and handling both invalid values and outliers.
3) Pondering the suitable hypervalues as the learning rate for the logistic regression and the penalty term for the ridge case, as well as finding the optimal polynomial expansion and interaction for the dataset. The whole is accomplished through cross-validation to obtain an estimate of the precision of the model.
4) Submitting the output file, which stores the predicted values, to AIcrowd and gaining the accuracy estimate from the platform.

II. Methods Used

In this section we are going to discuss the method used in the machine learning process.

A. Choosing the right model

After having developed all the methods, we considered in particular two of them: Logistic regression and Ridge Regression. The former, as the problem consists of a binary distinction, is theoretically considered most appropriate for a classification problem: given n features and dimensions, it defines a hyperplane that serves as the edge for the final decision. The second one was decided since its flexibility and speed was suitable for linear regression but, thanks to the additional penalizing term, optimized for dataset suffering from multicollinearity.

B. Data preprocessing

The methods were implemented so that it was possible to take into consideration the singular peculiarities of the train and test dataset. For this reason we started by performing data preprocessing. To start with, we noticed that some features appeared to have a considerable amount of invalid values, therefore, we considered to remove columns presenting at least 70% of invalid values to simplify the model. However, we observed that the results did not improve, then we abbonded this approach. Secondly, some outliers were identified and then removed, as they may have negatively impacted on the weigths and the hyperparameters of the models. The approach used to identify an outlier was through its statistical definition: a data point is defined outlier if it lies outside the interval $[Q_1 - k(Q_3 - Q_1), Q_3 + k(Q_3 - Q_1)]$, where Q_1 and Q_3 are the respective $25th$ and $75th$ percentile and k regulates the width of the interval. Thirdly, to ensure that all the features had the same magnitude and could be compared, we performed feature scaling on them, specifically the technique of standardization: to each entry, the mean of the corresponding feature was subtracted; then, the result was divided by the standard deviation of the same feature. Furthermore, since the the Logistic Regression algorithm is developed to work with y binary values [0,1] we mapped the given values [-1,1] to [0,1].

C. Feature Engineering and Expansions, Cross-validation

As part of our modelling, we focused on performing feature engineering over our dataset. This was done through polynomial expansion of different degrees and by considering the interaction between features. It is important to note that, for the polynomial expansion, we limited the degree to two because of both performance and timing reason.

4-fold cross-validation has been performed over the dataset in order to select the best hyperparameters. After each cross-validation, we also calculated the accuracy on the train data given by our models. This step was beneficial for gaining a personal accuracy in addition to the one provided on Aicrowd.

D. Sub datasets

In the end, to resolve the problem of the NULL values (-999) and to obtain a more precise prediction, we split the dataset into 4 parts and thus created 4 different models. The splitting was made based on a specific feature, named PRI_jet_num, which was noticed to be categorical with 4 different values: 0, 1, 2, 3. The value of this feature appears to be significantly correlated to the one of some other variables: for instance, every line presenting value 0 for

PRI_jet_num has also invalid values (-999) for eleven specific features. Therefore, the presence of invalid values was not random but often associated to this particular feature. Thanks to the documentation of HiggsML we confirmed in fact that many features do depend on this PRI_jet_num feature. After having split the datasets, we additionally decided to remove all features in which only null values were present because of the relation with PRI_jet_num.

III. Results

In this section we are going to expose some considered approaches and display their respective performances: in fact, cleaning and feature engineering were applied in different ways in order to obtain better results. The whole is applied to two regression models, namely Logistic and Ridge Regression, and with the 4-fold cross-validation technique. Note that we used a penalized term lambda ranging from $1\text{e}{-10}$ to 1 for the Ridge, and 100 iterations and learning rate γ ranging from $1\text{e}{-10}$ to 1 for Logistic.

A. Raw Data

We trained the model with the raw data to get an insight on the accuracy without any data preprocessing or feature engineering. As expected, such a model poorly performed. We tested our trained model using the given labels, and received an accuracy of 0.744 with penalized term $\lambda = 3.59\text{e}{-05}$ for ridge, and a worse accuracy of 0.463 with learning rate $\gamma = 1\text{e}{-10}$ for logistic. The cross validation both gave us 0 as the best degree for the polynomial expansion.

B. Removing outliers, standardization and feature expansions

In this approach we firstly removed the outliers from the dataset, performed standardization, and applied the feature expansion of degree n to the preprocessed dataset. We then applied cross-validation to select the best degree and hyperparameters for each method. This model performed better. We tested our trained model using the given labels and obtained an accuracy of 0.792 with penalized term $\lambda = 1\text{e}{-10}$ and second degree of polynomial expansion for Ridge, and a slightly better accuracy of 0.781 with learning rate $\gamma = 2.15\text{e}{-07}$ and second degree of polynomial expansion for Logistic.

C. Adding mixed products

We repeated the last mentioned approach by additionally considering interaction between features, i.e. the mixed products of the features, to test whether certain variable could be correlated with each other. This was done by the parameter Interaction = True when building the polynomials. From the result, we notice that the interaction between features had an impact. The gained accuracy was of 0.822 with penalizing term $\lambda = 1\text{e}{-10}$, second degree of polynomial expansion for Ridge and a slightly better accuracy of 0.810 with learning rate $\gamma = 2.15\text{e}{-07}$ and second degree of polynomial expansion for logistic.

Result	Ridge	Logistic
A	0.744	0.463
B	0.792	0.781
C	0.822	0.810
D	0.824	0.826

Table I
Modeling approaches with respective accuracy on ridge and logistic

D. Splitting the dataset

For the last approach, we applied the splitting of the dataset as mentioned before. We obtained 4 subsets, each one containing a different number for the variable PRI_jet_num. For each subset, we deleted the features in which only null values were present, since they are not significant. Then, four models were created for each subset using cross-validation whose purpose was to determine the following parameters: polynomial degree, if the mixed terms are present or not in the model (represented by the boolean value interactions), lambda for the Ridge Regression and the gamma step size for the Logistic Regression. We tested the created models by applying the corresponding model to each data point in the test dataset (for instance, the model obtained for PRI_jet_num=0 was applied only to the data points with PRI_jet_num=0). We noticed an increase of the overall accuracy on the train data. The results of each sub dataset (namely tX_1, tX_2, tX_3, tX_4) are:

Ridge:

- tX_0: accuracy of 0.839, degree=2, $\lambda = 3.59\text{e}{-05}$
- tX_1: accuracy of 0.805, degree=2, $\lambda = 3.59\text{e}{-04}$
- tX_2: accuracy of 0.839, degree=2, $\lambda = 4\text{e}{-04}$
- tX_3: accuracy of 0.838, degree=2, $\lambda = 0.0059$

Respectively, for the logistic:

- tX_0: accuracy of 0.840, degree=2, $\gamma = 2.78\text{e}{-06}$
- tX_1: accuracy of 0.805, degree=2, $\gamma = 2.78\text{e}{-06}$
- tX_2: accuracy of 0.830, degree=2, $\gamma = 2.78\text{e}{-05}$
- tX_3: accuracy of 0.834, degree=1, $\gamma = 3.59\text{e}{-05}$

In all cases we obtained interaction=True.

IV. Conclusion

From the results, the split was observed to be the method producing the most accurate models and predictions. We eventually reached an accuracy of 0.824 for Ridge Regression and 0.826 for Logistic Regression on the test data. So, eventually, we can state that the best model is obtained by:

- Using Logistic Regression
- Removing outliers, performing standardization and feature expansion
- Adding mixed products
- Splitting the dataset as previously described.