Machine Learning: Project 1
Higgs boson machine learning challenge

Quentin Deschamps, Emilien Seiler, Louis Le Guillouzic
School of Computer and Communication Sciences EPF Lausanne, Switzerland

***Abstract*—Machine learning techniques have been successfully applied to a bunch of problems from different disciplines. This paper introduces a machine learning method to predict if a certain event corresponds to the decayment of the Higgs boson given features extracted from the ATLAS experiment performed at the Large Hadron Collider at CERN [1]. The proposed algorithm is based on ridge regression with polynomial expansion, and obtained a categorization accuracy of 0.831 in the *AIcrowd* challenge.**

I. Introduction

The Higgs boson is an elementary particle in the Standard Model of physics. Its discovery at the Large Hadron Collider at CERN was announced in March 2013. In this project, apply machine learning techniques on actual CERN particle accelerator data are used to recreate the process of "discovering" the Higgs particle. It consists in building a binary classifier to predict whether an event corresponds to the decayment of a Higgs boson or not.

II. Methodology

A. Data handling

The first important part of this project was the preprocessing of the raw data. The features describing the events have a lot of meaning-less values represented by the value -999.0, which is outside the normal range of all variables.

1) Data splitting: In fact, a lot of features are dependent about the number of jets, corresponding to the feature called `PRI_jet_num`. This variable can take the values 0, 1, 2 or 3. Thus, we can consider this variable as categorical. Depending on this number, the other features can be undefined or not. The table I shows, for each number of jets, the number of features where all values are undefined.

Number of jets	0	1	2	3
Number of meaning-less features	11	7	0	0

TABLE I
Meaning-less features depending on the number of jets

So, to avoid this problem, the main idea is to **split the dataset in three subsets**:

- **Subset 1**: rows where `PRI_jet_num` $= 0$.
- **Subset 2**: rows where `PRI_jet_num` $= 1$.
- **Subset 3**: rows where `PRI_jet_num` ≥ 2.

In each subset, the meaning-less features are removed. Therefore, there are 18 features remaining for subset 1, 22 for subset 2 and 29 for subset 3.

2) Data transformation: After splitting the data, there are still some meaning-less values in the features (-999.0 values). The solution chosen is to **replace these values by the median of the feature**. The median is a good option due to its robustness.

Then, we apply a **log transformation** [2] on the subsets. This technique allows to reduce the impact of outliers. To perform the log transformation on negative features, we rescale the feature by adding the negative of its minimum plus 1. The equation 1 summarizes this log transformation on a feature x.

$$x_{\log} = \log(x - \min(x) + 1) \tag{1}$$

Afterwards, we **normalize** each feature such that the mean is zero and the standard deviation is one. It corresponds to the equation 2.

$$x_{\text{scaled}} = \frac{x - \text{mean}(x)}{\text{std}(x)} \tag{2}$$

These transformations are applied on each of the three subsets. When we have a training and a testing dataset, we use the minimum, the mean and the standard deviation of the training set on both training and testing set, so that both datasets are transformed in the same way.

B. Models

The second part of this project is to select relevant models and machine learning algorithms to perform this classification.

1) List of models: Different models are used to perform this binary classification by testing several loss functions and various approaches to optimize them.

The first loss function selected is the Mean Square Error (MSE). Gradient Descent (GD) as well as Stochastic Gradient Descent (SGD) are used to minimize it. MSE is also minimized by Normal equations with and without a regularization term.

The second loss function selected is the Logistic Loss. Gradient Descent with and without regularization is used to minimize it.

This brings us to six different models:

A. Linear regression using gradient descent
B. Linear regression using stochastic gradient descent
C. Least squares regression using normal equations
D. Ridge regression using normal equations
E. Logistic regression using gradient descent
F. Regularized logistic regression using gradient descent

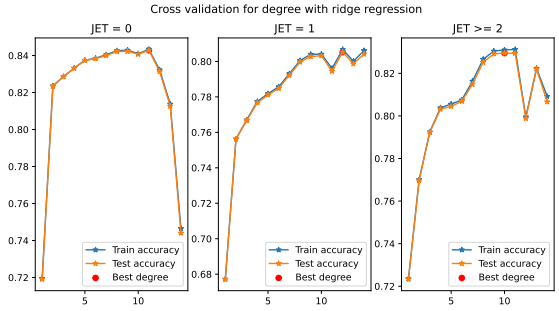

Fig. 1. Cross validation using ridge regression

2) Final model: Our final model which achieves the best accuracy is the **ridge regression** with **polynomial expansion**. An advantage of this model is that it uses normal equations to minimize the MSE function, which require much less computation and therefore time than using GD or SGD. The loss function is the following:

$$\mathcal{L}_d(w) = \frac{1}{2N_d} \sum_{n=1}^{N_d} \left[y_n - x_n^T w \right]^2 + \lambda \|w\|_2^2 \qquad (3)$$

Therefore, in this model, we have two parameters:

- λ the regularization factor, to avoid overfitting.
- d the degree of the polynomial expansion, to increase complexity.

Cross validation is used on these parameters (described in part II-C).

C. Cross validation

To get the best parameters for the ridge regression, we perform k-fold **cross validation**, with $k = 10$ folds. We looked for the best regularization parameter λ and the degree for polynomial expansion d. We tested λ in the set $\{10^{-4}, 10^{-3}, 10^{-2}\}$ and d in the range $[1, 15]$. The algorithm is used separatly on each of the three subsets described before, according to the number of jets. The figure 1 shows the accuracies obtained for each value of d, with $\lambda = 10^{-3}$.

The best parameters and the accuracies obtained for each subset with cross validation are illustrated in the table II.

Subset	Number of jets	Best λ	Best degree d	Accuracy
1	0	10^{-3}	13	0.844
2	1	10^{-3}	14	0.808
3	≥ 2	10^{-3}	13	0.836

TABLE II
BEST PARAMETERS OBTAINED WITH CROSS VALIDATION

III. RESULTS

Finally, the table III shows the performance of each model described in part II-B. The accuracy scores are obtained by splitting the training set delivered in a training and a testing set. The procedure used is the following:

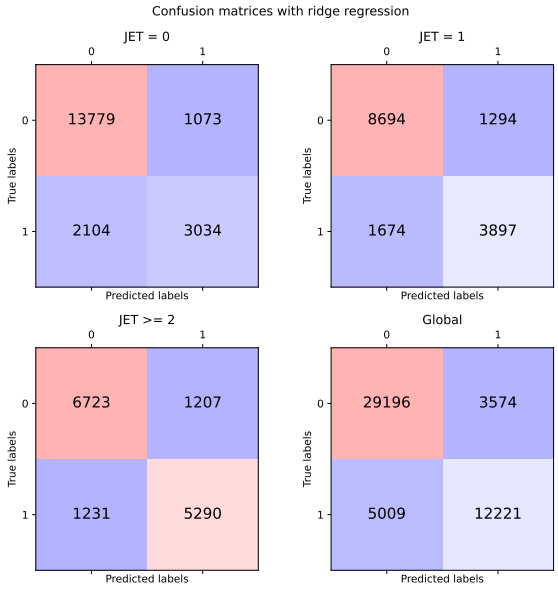

Fig. 2. Confusion matrix using ridge regression

1) The training and the testing dataset are splitted both in three subsets as described in the part II-A1. So, we obtain three training subsets and three testing subsets.
2) The subsets are cleaned following the procedure described in part II-A2. We note that we use the minima, means and standard deviations of the training subsets.
3) The optimization algorithm is runned on each of the three training subsets such that we obtain three vectors of weights.
4) The labels of the testing subsets are predicted.
5) The accuracy score is calculated for each testing subset, and the global accuracy is computed by merging the predictions.

Model	A	B	C	D	E	F
Accuracy	0.715	0.709	0.827	0.828	0.760	0.760

TABLE III
PERFORMANCE OF THE MODELS

We note that polynomial expansion is used for the models C and D. The confusion matrix of the figure 2 shows the predictions for each subset and the global predictions for the model D.

IV. CONCLUSION

With this project, we shown the importance of cleaning and preprocessing data before applying machine learning algorithms. In this case, it is much more interesting to split the dataset in subsets rather than using the whole dataset.

Acccording to our methodology, the best results are obtained with the ridge regression. Our best accuracy score obtained on *AIcrowd* is 0.831[1].

[1]Submission link

REFERENCES

[1] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau, "Learning to discover: the higgs boson machine learning challenge," CERN, Tech. Rep., 2014. [Online]. Available: https://higgsml.lal.in2p3.fr/files/2014/04/documentation_v1.8.pdf

[2] P. Huilgol, "Feature transformation and scaling techniques to boost your model performance," 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/07/types-of-feature-transformation-and-scaling/