Higgs Boson Detection: Machine learning approach

Zhecho Mitev Kylian Do Nascimento Anthony Yazdani
zhecho.mitev@epfl.ch kylian.donascimento@epfl.ch Anthony.Yazdani@etu.unige.ch

***Abstract*—The recent breakthroughs in machine learning offers a whole range of new methods that are highly appreciated throughout the various industries. In this project we develop a method which predicts whether a signal is a Higgs boson or a background noise. Our approach is based on one of the most famous models belonging to generalized linear models, the logistic regression.**

I. Introduction

The Higgs boson is an elementary particle, which aims to explain why particles have mass. The particle was initially discovered in 2012 at CERN [1]. A major problem is that the particle decays extremely fast and it is not directly observable; therefore the decay features are considered by researchers to detect the Higgs boson. Machine learning (ML) has been continuously used to solve various aspects of this problem [2], [3], however they struggle to produce a satisfactory Higgs boson classification model. This project aims to create a supervised model which can accurately predict whether a particle is Higgs boson or not, based on its decay signature.

II. Methodology

A. Exploratory data analysis

We observe that the data contains 30 features in total and that all are continuous except for one (PRI_jet_num). We further notice that there is a significant amount of missing values (-999). In Section II-C we introduce the different methods which address this problem. Additionally, we observed that the features have skewnesses different from zero, however they are relatively too small in absolute values to have a harmful impact on the modeling.

B. Data splitting according to PRI_jet_num

It has been found that the observations concerning the different discrete states of PRI_jet_num shows different proportions of missing values. In doing so, we investigate the properties of the dataset in each of these states by proposing the following approach:
We hypothesize that for a state $j = 0, 1, 2, 3$, a feature $X^{(i,j)}$ for $i = 1, ..., d$, contributes to linearly separate labels if its average when we observe a Boson ($\Theta^{(i,j)}$) is statistically different from its average when we do not observe a Boson ($\Phi^{(i,j)}$).
Assuming $\Theta^{(i,j)} \perp \Phi^{(i,j)}$, an indicator of whether this feature is useful would be to calculate the p-value of the following statistical test: $H_0 : \Theta^{(i,j)} = \Phi^{(i,j)}$ and $H_a : \Theta^{(i,j)} \neq \Phi^{(i,j)}$.
By the law of large numbers [4], the corresponding p-value is:

$$p_{value}^{(i,j)} = 2 \times \mathbf{P}\left(Z > \frac{|\Theta^{(i,j)} - \Phi^{(i,j)}|}{\hat{\sigma_\Theta}^{(i,j)} + \hat{\sigma_\Phi}^{(i,j)}}\right)$$

Where $Z \sim N(0, 1)$, $[\hat{\sigma_\Theta}^{(i,j)}, \hat{\sigma_\Phi}^{(i,j)}]$ are the standard deviations and $\mathbf{P}(.)$ is computed using Riemann's sum approximation of integrals. [5]
We observe that the dataset does not share the same significant features for the different states of PRI_jet_num. In doing so, we decided to split the dataset into four subsets that corresponds to the different states of PRI_jet_num.

C. Data processing

In this section, we explain the steps of our data processing. The procedure is described by the following pseudocode.

Separate the observations w.r.t **PRI_jet_num**;
for *each dataset* **do**
  Remove features with no variance;
  Impute missing values;
  Perform data augmentation;
  Scale the data;
  Perform change of basis;
  Add a bias term;
end

Algorithm 1: Data Processing

More formally, the procedure is the following:

After separating the data into four subsets, we remove the features with no variances.

Secondly, three techniques of imputation were considered. Respectively, mean imputation, median imputation and stochastic k-nearest neighbor imputation.[6]

When it comes to data augmentation we choose to use polynomial basis and pairwise interactions. More formally, the augmentation of our four different datasets has the following form:

$$X_{augmented} = [X, X^2, ..., X^K, (X^{(i)} \times X^{(j)})_{1 \leq i < j \leq d}]$$

Where K is the higher monomial of the polynomial augmentation and the product ($\times$) is the element-wise product.

For optimization purposes, we decided to scale the data by subtracting the mean and dividing by the standard deviation.

Additionally, we represent our features in an orthogonal base [7]. In doing so, we proceed as follows:

$$\tilde{X} = X_{augmented} \cdot (\mathcal{A}^{-1})^T$$

Where $\mathcal{A}$ has the eigenvectors of the augmented feature covariance matrix as columns and $\tilde{X}$ is the orthogonal dataset.

Finally, we add a bias term after the change of basis as follows:

$$\dot{X} = [\mathbf{1}, \tilde{X}]$$

III. Logistic regression and Gradient-based optimization

All four models are learned by minimizing the following loss function:

$$L(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \left[ln[1 + exp(x_n^T \cdot \mathbf{w})] - y_n x_n^T \cdot \mathbf{w} \right] + \lambda \left\| \mathbf{w} \right\|_2^2$$

Where λ is the penalty parameter.

We will be interested in three variations of Gradient-based optimization [6], namely, Newton's method (NM), Gradient descent (GD) and Stochastic Gradient descent (SGD). Mathematically, they are of the following form:

$$\mathbf{w}_{n+1} = \begin{cases} \mathbf{w}_n - \gamma \cdot [\nabla^2 L(\mathbf{w}_n)]^{-1} \cdot \nabla L(\mathbf{w}_n) & \text{for NM} \\ \mathbf{w}_n - \gamma \cdot \mathbf{I}_d \cdot \nabla L(\mathbf{w}_n) & \text{for GD} \\ \mathbf{w}_n - \gamma \cdot \mathbf{I}_d \cdot \nabla \tilde{L}(\mathbf{w}_n) & \text{for SGD} \end{cases}$$

Where $\mathbf{w}_n$ is the parameter vector at step n, γ is the learning rate, $\mathbf{I}_d$ is the identity matrix of size d^2, $L(\mathbf{w}_n)$ is the loss function and $\tilde{L}(\mathbf{w}_n)$ is the loss function computed on a random subset of the data points.

Although the Newton's method have a greater complexity than GD or SGD due to the calculation of the hessian, we implement the method due to its powerful convergence rate. Our results show that the method reaches a good solution ($\left\| \nabla L(\mathbf{w}_n) \right\|_\infty \leq 10^{-4}$) much faster than the other optimization methods.

IV. Hyper-parameter tuning

We perform 5 Fold cross validation for all four models as follows: We choose an imputation method, and for every degree ($k = 1, ..., 20$) of the polynomial basis augmentation, we find the optimal penalty parameter λ which maximize the accuracy. To do so, we used a golden search algorithm by assuming concavity of the accuracy w.r.t λ.
We observe that the choice of imputation has no valuable impact on the accuracy. For the sake of simplicity and because of its robust properties, we keep the median imputation in our final models. Secondly, we notice that the optimal choice of λ for all four models is approximately zero.

Moreover, to determine the probability threshold for labelling a datapoint as a Higgs boson, we perform a grid search on the training sets and find the optimal threshold (t^*) for each of them.

V. Results

In this section, we discuss our results and show the error plots of the 5 Fold cross validations using median imputation and setting the penalty parameters to zero.

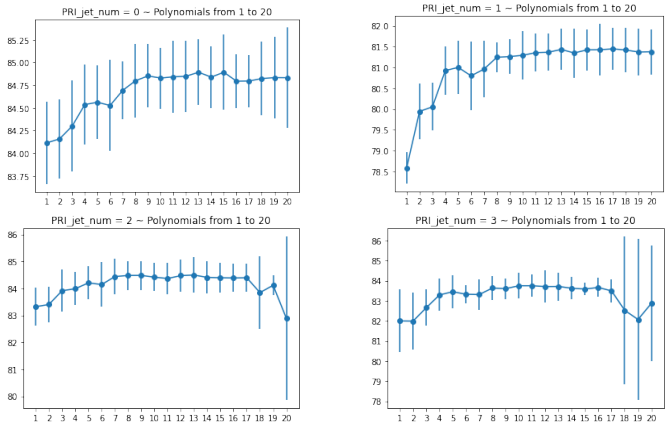

Fig. 1. 5 Fold cross validation error plots with 95% confidence intervals.

We observe from Fig. 1, that increasing the degree is beneficial up to a certain point. More precisely, optimum parameters and results are shown in the table below.

Model	k*	Mean 5F-CV accuracy	Std
PRI jet num = 0	13	84.9%	0.0019
PRI jet num = 1	17	81.45%	0.0026
PRI jet num = 2	13	84.5%	0.0033
PRI jet num = 3	10	83.75%	0.0032

By combining our four models, we were able to achieve a total accuracy of 83.8% and a F1_{score} of 75% on the test set.

VI. Conclusion

In conclusion, we model our dataset using four different logistic regressions. By doing so, we were able to achieve an accuracy of 83.8% on the test set. Despite a reasonable accuracy, a lot of different classification methods are suitable to increase the latter. For example, we can quote support vector machine [8]. Additionally, other improvements can be done on the data processing part. First of all, outlier imputation or deleting could be done. Secondly, the knn imputation was probably under-performing because of a large amount of features. This imputation technique might be improved by reducing the features space into a much smaller one using singular value decomposition.

REFERENCES

[1] G. Aad, T. Abajyan, B. Abbott, J. Abdallah, S. Abdel Khalek, A. Abdelalim, O. Abdinov, R. Aben, B. Abi, M. Abolins, and et al., "Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc," *Physics Letters B*, vol. 716, no. 1, p. 1–29, Sep 2012. [Online]. Available: http://dx.doi.org/10.1016/j.physletb.2012.08.020

[2] K. Lasocha, E. Richter-Was, D. Tracz, Z. Was, and P. Winkowska, "Machine learning classification: Case of higgs boson cp state in h→ decay at the lhc," *Physical Review D*, vol. 100, no. 11, Dec 2019. [Online]. Available: http://dx.doi.org/10.1103/PhysRevD.100.113001

[3] A. Buzatu, "Machine learning for higgs boson physics at the lhc," *Higgs Couplings 2018*, p. 1–29, Dec 2018.

[4] P.-L. Hsu and H. Robbins, "Complete convergence and the law of large numbers," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 33, no. 2, p. 25, 1947.

[5] A. Knowles, "Analysis l - university of geneva," 2016.

[6] "Machine learning - epfl," 2020.

[7] R. Bro and A. K. Smilde, "Principal component analysis," *Analytical Methods*, vol. 6, no. 9, pp. 2812–2831, 2014.

[8] I. Steinwart and A. Christmann, *Support vector machines*. Springer Science & Business Media, 2008.