# Project 1 on Machine Learning team Yoor

Sergei Volodin[1], Baran Nama[1], and Omar Mehio[1]

[1]EPFL
{*sergei.volodin,baran.nama,omar.mehio*}*@epfl.ch*

***Abstract*—A classification dataset from the Large Hadron Collider simulations is being studied. First, the data is thoroughly explored using visual aids. After that, several basic Machine Learning methods are applied on preprocessed data. Results are evaluated using cross-validation. Model overview is given for each considered algorithm and the best model is chosen.**

## I. Introduction

The Higgs boson is a famous elementary particle which was first predicted in 1960s and then discovered in 2012 [1]. Its famousness is due to two facts, first being the collosal amount of effort put into construction of the Large Hadron Collider and conducting the ATLAS experiment and the second being the fact that the Higgs boson is considered to be connected to the fact that particles have a mass.

The ATLAS experiment consists of protons colliding at near-relative speed. After collision, the resulting particles sometimes contain the Higgs boson. Itself, it is not detectable by LHC. However, it is possible to detect the particles that it is decaying to.

The data being studied comprises of 250 000 objects (train) each having 30 features: $\{(x_i, y_i)\}_{i=1}^{N}$, $x_i \in \mathbb{R}^D$, $N = 250000$, $D = 30$. Each object represents a collision of a stream of protons. The data was not obtained during the ATLAS experiment, but rather from the simulation [2]. Features represent properties of detected particles. It is required to determine if the particles represent the Higgs boson. The dataset has two classes: signal/noise.

The following paper claims that it is possible to use simple methods, such as Linear and Logistic Regressions to classify the data. In the following sections, the data is thoroughly studied and then the model is chosen based on reasoning and cross-validation.

## II. Models and Methods

Table I: Feature processing tricks and linear regression

| Feature trick added | Test accuracy |
|---|---|
| Constant feature | 0.745 |
| Standardization | 0.745 |
| Imputation | 0.747 |
| Binarization | 0.747 |
| Degree 3 poly | 0.785 |

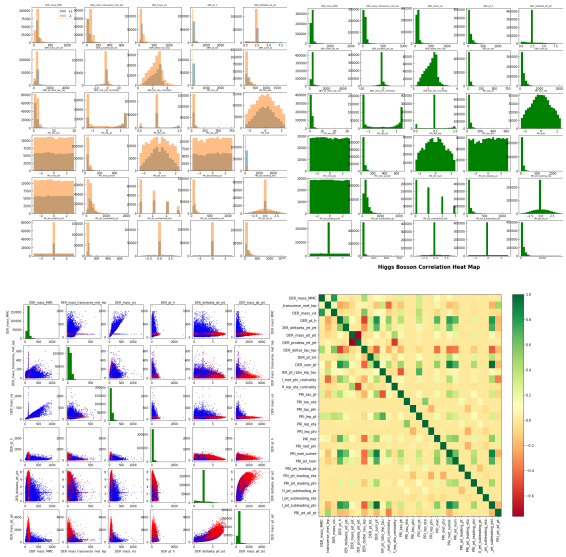


Figure 1: Exploratory data analysis using charts

First, the exploratory data analysis shows that most of the features (except one) are real-valued 1, 1-2. Moreover, it can also be seen that the test distribution does not differ from the training one 1, 1. Scatter plots 1, 3 show dependencies between features.To further explore the linear dependencies among our features, we computed the correlation matrix,1, 4. Some of the significant dependencies we observed include the relationship among following tuples $(DER_prodeta_jet_jet, DER_mass_jet_jet)$ and $(PRI_jet_subleading_phi, DER_sum_pt)$. Each of these tuples have a -1 and +1 correlation coefficient respectively. This implies that for each column in the tuple we can eliminate the other pair since we are able to deduce the same amount of knowledge from either one of them. In practice this didn't turn out to be efficient as our prediction with deducted features resulted in lower accuracy.

To begin with, a simple linear regression model is used on raw data (with a constant feature added), with 5-fold cross-validation to control overfitting. The results of each subsample are averaged to provide train/test accuracy. However, the model gives unsatisfying results on the hold-out dataset. As a means to deal with this issue, feature imputation and feature engineering is used: missing features are imputed with mean

values (and 'missing value' feature is added), categorical features are binarized. Moreover, it has been discovered that some of the features represent particle's mass, momentum or energy. Using the fact that squaring these quantities would make linear combinations of them meaningful in terms of physics ($E^2 = m^2 + p^2$), we introduce the polynomial basis. Table I shows the respective effects of these tricks.

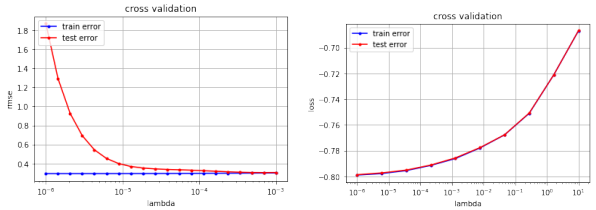


Figure 2: Degree 6 basis, MSE and Accuracy metrics

Applying ridge regression, it might be seen that it does not overfit in terms of accuracy (but does in terms of MSE 2) on the dataset given even for the 6th degree, meaning that there is no significant difference between the test and the train accuracies. Applying it for different degrees, one obtains results shown in the Table II. It can be seen that

## III. Models and Methods

Table II: Polynomial basis and linear regression, $\lambda = 0$

| **Degree** | **Test accuracy** | **MSE Test** | **MSE Train** | $\lambda$ best |
|---|---|---|---|---|
| 1 | 0.7470 | 0.3378 | 0.3377 | $10^{-6}$ |
| 2 | 0.7737 | 0.3483 | 0.3158 | $10^{-2}$ |
| 3 | 0.7850 | 0.3060 | 0.3040 | $10^{-4}$ |
| 4 | 0.7937 | 0.3030 | 0.3010 | 0.0004 |
| 5 | 0.7971 | 0.3050 | 0.3010 | 0.0006 |

As can be seen in the figures and results, while train MSE is continuously decreasing because of over-fitting, there is no significant change in test error after third degree polynomial basis. Therefore, feature extraction using polynomial basis does not seem to work for higher degrees. Furthermore, the accuracy of the model in Kaggle is not as good as what we expected (below 0.8), so that we decided to switch logistic regression since it uses a loss function tailored for the purpose of classification.

Applying logistic regression to the dataset with gradient descent, we were obliged to choose a very small learning rate , since while computing the gradient (logistic function) , the exponential of the vecotr product of Xw would result in an overflow in the computation, due to their large values. Hence we chose $\gamma = 10^-5$ as our learning rate. Offcourse this came as a tradeoff with respect to the loss threshold we were aiming for. The larger the learning rate the faster we can approach to our global minimum. To solve this problem ,we took two descions. First we implemented the Newton's method to compute the gradient in a much faster time. The second approach was to find if there exists a relation between magnitude $\gamma$ of the gradient step and the size of the dataset.Our findings indicated that magnitude $\gamma$ depends on the size of the dataset , hence we used batched version of stochastic gradient descent for logistic regression as the optimizer to stabilize the size of the dataset the optimizer uses for training.

## IV. Results

We have presented an application of linear, ridge and logistic regression to the dataset. The best model was logistic regression with best accuracy of 0.81.

## V. Discussion

Despite thorough analysis, our project lacks further consideration in the following directions. First, it can be seen that some of the features do not look alike the Gaussian distribution. It might prove beneficial to split such features into two using a certain threshold.

## VI. Summary

We have shown that it is possible to detect the Higgs boson using linear methods and feature augmentation.

## References

[1] en.wikipedia.org/wiki/Higgs_boson

[2] Experiment and feature description

[3] EPFL ML Project 1