Team squila: Finding the Higgs Boson Challenge

Irina Bejan, Nevena Drešević & Marija Katanić
School of Computer and Communication Sciences, EPFL Lausanne, Switzerland

***Abstract*—The goal of this paper is to present our approach in solving the Higgs boson challenge - to determine whether the event's signature is a result of the Higgs boson (signal) or is caused by other process/particle (background). The input data set consists of data collected at CERN and the paper presents our approach in understanding, preprocessing and classifying it. We explore various methods and propose a binary classifier based on logistic regression, which has achieved an accuracy of 0.8 on the AICrowd platform.**

I. Introduction

The Higgs boson is a particle that gives other particles their mass and its discovery is crucial, indicating the existence of new physics principles. Hence, the problem of classifying if a particle is a Higgs boson or not is undoubtedly significant and improvements on the current solutions are still needed and highly valuable.

The data set used for training the model is original data provided from CERN. Analyzing the semantics of the data, we observed multiple improvements can be derived from data wrangling and cleaning. We attempt to split the data according to jets and cover missing data, remove outliers, standardize and normalize. Further, we look into the distribution and correlation of features to do dimensionality reduction and simplify our optimization task, as well as feature expansion to achieve a smoother optimization. We use different methods, such as least squares and logistic regression, combined with different regularization techniques and fine-tune the models to get better particle predictions. This approach leads to an accuracy of 0.8 and an F1-score of 0.693 and it is described in detail in the following sections.

II. Data preprocessing

The data set received consists of 250000 samples and a number of 30 characteristics and we noticed that some features were defined using different scales, multiple data points were missing and most features were dependent on each other. Therefore, we focused on properly refining the data and used the following techniques:

- **Dependent features**
  It is a fast observation when looking at the data semantics[1] that most features are dependant on the feature called PRI_jet_num, being the only one with categorical values. Some features are undefined based on the jet number and this is difficult to capture in the model. We decided to split the data into four independent data sets for each value of the jet number, which can be 0, 1, 2, or 3 (some values can also be higher, in which case, they are regarded as 3) and have a different classifier for each data set. For each jet, we identified in features characteristics what columns are undefined for on each jet and dropped those columns. We also dropped the jet number column from each of the four datasets. To further overcome redundancy in our dataset, we tried using the PCA (Principal Component Analysis) technique. This is done by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible. Unfortunately, this method did not have much impact, this is likely because we already removed all unnecessary features for each of the jets. Figure 1 confirms this, as we can see there is almost no high correlation between features.

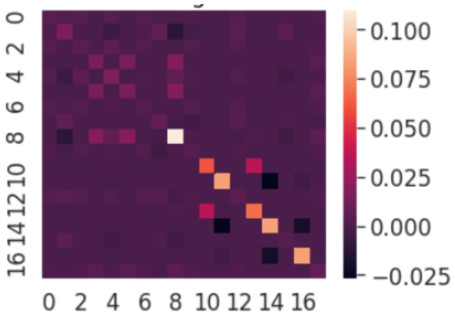

Fig. 1. Covariance matrix for data set for jet 0.

- **Missing values**
  To understand our data, we plotted the data points of each dataset and noticed that even after splitting, there were missing points - having the value of -999. To address those values, we replaced them with the mean of the feature values, so they would not affect the data distribution. However, we also tried replacing them with zeros and this made the training slightly more difficult.

- **Outliers**
  Plots also showed that there were outliers in multiple feature columns, as in Fig. 2. Our initial approach was removing them statistically considering how many times below or above the interquartile range the point is to the first and third quartile, respectively. Although iteratively reducing that factor, the model was overfitting very fast. Instead of removing, we analyzed each plot and deter-

[1] http://opendata.cern.ch/record/328

mined the threshold for capping the outliers.

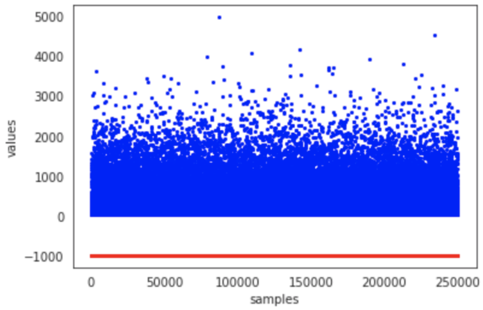

Fig. 2. Feature values for column $DER\_mass\_jet\_jet$. Red dots represent undefined values. The features are here capped to 2500.

- **Standardization & normalization**
  We standardize the training data by subtracting the mean and dividing by the standard deviation and reuse the same values when standardizing the test data. For logistic regression is very important to have scaling given the log likelihood computation.

- **Feature extension**
  To model our data better, we also experiment by doing polynomial feature transformation which improved our results significantly. Furthermore, we plotted the distribution of each of the initial features and we noticed some were very skewed (Fig. 3). To get a normal distribution, we do a log transformation and append these new feature columns before the polynomial expansion.

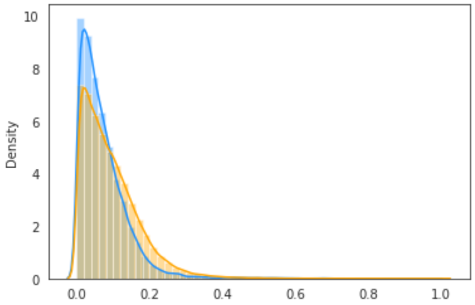

Fig. 3. Distribution of feature column DER_pt_tot after preprocessing. Blue line shows the signals and the yellow one the background samples.

III. Models hyperparameters

For training, we split each of the four individual dataset into training and validation sets, based on a ratio of 70% - 30%. The stopping condition for GD is validation loss increasing over previous losses to avoid overfitting and for SGD we used a fixed number of iterations depending on the hyperparameters. Because our final preprocessing methodology did not result in any skewness between training and testing data, our models maintained very close train/test loss and accuracy and did not overfit. During training, the weights given by best validation accuracy are saved and used for predictions. Because our function to optimize was not convex, we replaced the zeros weights initialization with He initialization which draws the weights from a normal distribution centered on 0 and variance based on the number of features.

Our first approach was the least squares regression technique and MSE loss on the initial dataset without splitting the inputs, getting a score of 0.76 using GD and around 40000 iterations until the improvements became very small. Using ridge regression helped in converging faster, but did not improve the final accuracy. Furthermore, we tried using logistic regression and we noticed training becomes much faster using SGD and big batches of 8192 samples, without compromising any degree of generality. As we tried to improve using both L1 and L2, we noticed the former helped achieve better results, while Lasso regression did not influence the learning much given we already removed useless features. Because the dataset was highly imbalanced, we computed the class weights for background and signal labels based on their percentage in the dataset and penalized more the mistakes in the underrepresented class (signal class), which also improved both the accuracy and the F1-score. Within the same parameters, we reran least squares and its accuracy improved by 0.116, using gradient descent which was converging very fast now compared to the previous attempt. For the learning rate and regularization, we tried different values on a log scale from 1 to 10^{-5} and observed which give a better trade-off of speed and loss fluctuation. For faster convergence, we decayed the gamma by a factor of 0.95 and a minimum gamma of 0.1.

IV. Results

The best results were obtained using logistic regression and SGD, starting with an identified learning rate and combined with L2 regularization with a factor of 10^{-6} and generally bigger batch sizes (8192, 1024, 512).

Accuracy	Dataset	Method used	Comments
0.760	All	Least squares GD	No splitting, capping, poly
0.842	Jet 0	Logistic reg SGD	gamma=0.3, batch=8192
0.782	Jet 1	Logistic reg SGD	gamma=0.2, batch=1024
0.803	Jet 2	Logistic reg SGD	gamma=0.2, batch=512
0.783	Jet 3	Logistic reg SGD	gamma=0.2, batch=128
0.807	All combined	Logistic reg SGD	
0.827	Jet 0	Least squares GD	gamma=0.05, reg=0.0001
0.757	Jet 1	Least squares GD	gamma=0.015, reg=0.0001
0.726	Jet 2	Least squares GD	gamma=0.015, reg=0.0001
0.777	Jet 3	Least squared GD	gamma=0.015, reg=0.0001
0.776	All combined	Least squares GD	

V. Conclusions

Our report shows how important data preprocessing is to improve results and shows that logistic regression can achieve the best results. However, the Higgs Boson challenge is still an open problem and future advancements can be done by exploring different techniques.