Finding the Higgs Boson
CS-433 Machine Learning: Project 1

Devrim Celik, Xavier Oliva i Jürgens, Nina Mainusch
Department of Computer Science, École polytechnique fédérale de Lausanne (EPFL), Switzerland

***Abstract*—The Higgs boson challenge aims to recreate the process of discovering the Higgs particle in order to improve the discovery significance of the experiment [1]. Being embeded in the Machine Learning course CS-433 of the EPFL, our approach to tackle this challenge by exploring the potential of machine learning methods will be described in this report. Implementing the Ridge Regression method, we managed to achieve an accuracy of 82.47% on the platform AIcrowd [2].**

I. Introduction

The Higgs boson exists solely for a short period of time, before it decays into other particles. Using simulated data with features characterizing events detected by ATLAS, the challenge is to estimate the likelihood that a given signature of an event was the result of a Higgs boson (signal) or some other process (background). ATLAS is a particle physics experiment taking place at the Large Hadron Collider at CERN that searches for new particles and processes using head-on collisions of protons of extraordinarily high energy [3]. By preprocessing the data set, tuning hyperparameters and comparing the performance of various machine learning models, we aim to achieve the highest possible accuracy on the evaluation data of the platform AIcrowd.

II. Models and Methods

For our project we were given a matrix of features representing the decay signature of a collision event, and we were asked to predict whether this event was signal (encoded as 1) or background (encoded as -1). Missing values were encoded as -999. The training and test data sets are characterized by 30 features, consisting of 250000 and 568238 data points respectively. 65.7% of the training data are labeled as background and 34.3% as signal, resulting in a slightly unbalanced dataset.

A. Exploratory Data Analysis

We started by exploring the data at hand. For each of the 30 features we investigated the proportion of missing values they incorporate. Interestingly enough, there are only eleven features (columns) that contain missing values with a proportion of either 70.98%, 39.96% or 15.25%. Furthermore, if the same two features exhibit 70.98% of missing values, the missing values are at the exact same position. This led us to the consideration that the missing values do not occur at random, but are rather deterministic artifacts resulting from different experimental settings. We started splitting the data set into six mutually exclusive groups, where in each group there are whole columns of missing data, but all other features are complete. See Table I for the exact features missing in each group. Supporting our hypothesis, we noticed that certain categorical features take exactly one value throughout a respective group.

Group	1	2	3	4	5	6
Missing Features	None	0	1,5,6,7,13, 24,25,26, 27,28,29	1,5,6,7 13,27 28,29	5,6,7,13, 24,25,26, 27,28,29	5,6,7, 13,27, 28,29

Table I
Missing features in each of the six groups.

B. Data Preprocessing

1) Missing values: Regarding the missing values, the first intuition would be to discard all data rows containing a missing value or to impute it by an estimated value based on the remaining feature values. However, this would result in either a tremendous data loss or in highly artificial data points, given that in certain features 70.98% and 39.96% of the data are missing. Instead we decided to split the data set into the six previously mentioned mutually exclusive groups, and to drop the features that exhibit missing values in each group.

Supporting our hypothesis that the missing values do not occur at random, we found out that if there are missing values in the test data, they appear in the exact same pattern as in the training data. This means that each new data point can be unambiguously allocated to one of the six groups. In Figure 1, the percentage of training data belonging to each group is displayed.

2) Model Architecture: Due to the apparently deterministic pattern in missing values, we decided against training one model for all data points, but rather train six models separately for each group. For the test data, we first assigned them to a group according to their combination of missing values and then got the predicted target value by the model corresponding to the respective group.

3) Feature transformation: For all groups we added a bias term for the regression analysis and we explored the effect of standardizing features, outlier detection, polynomial feature expansion and the removal of correlated features,

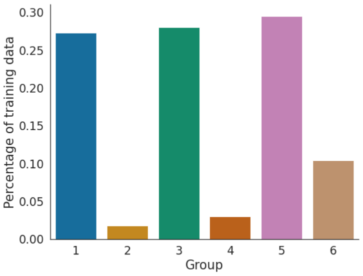

Figure 1. The six groups and the respective percentage of training data for each. The percentage of test data for each group is the same (except for the positions after the decimal point).

where a feature that displayed an absolute Pearson's correlation score of more than $\rho = 0.95$ would be removed. Regarding the slightly unbalanced data set, we tried to equalize the amount of samples in every class by duplicating data points labeled as signal. Unfortunately, this had a negative effect on the performance of the models.

C. Machine learning models

Initially we implemented six machine learning methods according to the project description: Least Squares (Stochastic) Gradient Descent, Least Squares and Ridge Regression (RR) using the closed-form expression, (Regularized) Logistic Regression using (Stochastic) Gradient Descent. Additionally, we implemented a Support Vector Machine using Gradient Descent, and Least Squares Regression using mini-batch Gradient Descent.

D. Testing and Tuning

To train the models, we applied 4-fold-cross-validation to locally estimate the accuracy of each model and avoid overfitting. Our approach was to iterate for each group over all possible preprocessing options, models and hyperparameter settings, in order to allow the models to individually adapt to the designated group. This was done in a grid-search setup.

III. Results

Interestingly enough, Ridge Regression turned out to be the best model for all groups. We observed a considerable improvement in accuracy over all groups if we applied outlier detection and correlation analysis, while we saw a deterioration if class equalization was performed. A quick glance at Figure 2 yields insights about the best parameter specifications of the polynomial feature expansion and the values of lambda: we can see that unevenly labeled groups have no clear preference for any specific polynomial degree (if it is greater than two) or value of lambda. The evenly labeled groups on the other hand seem to prefer smaller polynomial degrees (but not degree one) and again no specific value for lambda. In accordance with the heatmap, the best polynomial degrees for the feature expansion for each group were $7, 10, 9, 25, 9,$ and 28 respectively, and the best values of lambda ranged between 0.000075 and 0.001. Having obtained these optimal values, we trained the models one final time on the whole training data set, in order to utilize all available training samples. Our final model achieved an overall accuracy of 82.47% on AIcrowd and 82.64% on our local cross-validation test set.

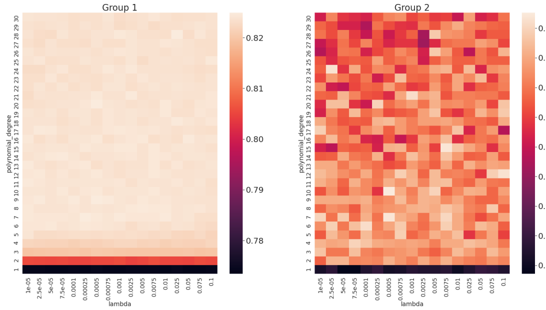

Figure 2. Heatmap displaying the accuracy of Ridge Regression models with given parameter specifications for lambda and the polynoial degree for even and uneven groups (represented by group one and group two).

IV. Discussion

Unexpectedly a single model, i.e. Ridge Regression, outperformed all other models. One possible reason might be that most other models are of an iterative type and our grid-search approach did not allow to explore their performance beyond 1500 iteration due to time constraints. Maybe the most interesting conclusion we can draw from our grid-search results is the fundamentally different behaviour between the evenly and unevenly labeled groups. They exhibit dissimilar characteristics when it comes to the number of associated samples and the best accuracy we can achieve on them. The greatest share of samples is in the uneven groups, thus explaining why the Ridge Regression model is not sensible to varying degrees of polynomials or values of lambda. The even groups on the other hand are sensible to changes in the parameter specifications, since the minority of data samples is assigned to them.

V. Summary

In this project we applied different machine learning algorithms to predict the likelihood that a given signature of an event was the result of a Higgs boson signal or some other process. While exploring the data we detected a certain pattern in the occurrence of missing values, divided the data in six groups respectively and trained a separate model for each group. The test data is perfectly allocable to the groups according to its missing values. We tested and tuned different models and hyperparameters. The best performance was achieved with Ridge Regression, obtaining 82.47% accuracy on AIcrowd and 82.64% on a local cross-validation test set.

References

[1] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kegl, and D. Rousseau, “Learning to discover: the higgs boson machine learning challenge,” *http://higgsml.lal.in2p3.fr/documentation*, vol. 9, 2014.

[2] “Epfl machine learning: Higgs,” https://www.aicrowd.com/challenges/epfl-machine-learning-higgs/leaderboards, accessed: 2020-10-19.

[3] “Higgs boson machine learning challenge,” https://www.kaggle.com/c/higgs-boson, accessed: 2020-10-18.