Machine Learning Project 1

Maxime Schoemans
Pedro de Tavora Santos
Yannick Paul Klose
Department of Computer Science, EPFL , Switzerland

***Abstract*—In the process of discovering the Higgs Boson it was crucial to separate between experiments that resulted in a Boson and experiments that were just related to background noise. This project aimed to solve this problem by using different Machine Learning algorithms applied on the original CERN dataset.**

We were supposed to distinguish between the two given sets of data: one for training our models, having the results of the predictions and one for testing our model. To this effect we performed exploratory data analysis on both sets to know which information was relevant and which not, we cleaned the data, separated it in several smaller datasets, performed feature expansion in the multiple datasets and finally proceeded to use ridge regression on the multiple datasets previously obtained. We analysed our models and tested them using cross-validation resulting in a match with the test set of approximately 78%.

I. Introduction

The aim of this project, as previously mentioned, is to train a model that can classify the samples of the new test dataset into Higgs Bosons and background noice. During the process of finding the best fitting model, we applied different methods of our toolbox, such as logistic regression, stochastic gradient decent and ridge regression. We describe our process in section II and our results in section III.

II. Models and Methods

A. Data analysis

The first thing we did was analyze the training dataset and try to find some intrinsic relationships between the parameters. Our main focus was to assign missing values (-999) to a certain pattern.

We found out that there was a relationship between the parameters *PRI_jet_num* (the only discrete parameter in the dataset) and the parameters of the leading and subleading as well as four other parameters (leading: *PRI_jet_leading_pt*, *PRI_jet_leading_eta*, *PRI_jet_leading_phi*; subleading: *PRI_jet_subleading_pt*, *PRI_jet_subleading_eta*, *PRI_jet_subleading_phi*; other: *PRI_jet_all_pt*, *DER_deltaeta_jet_jet*, *DER_mass_jet_jet*, *DER_prodeta_jet_jet*). A jet number of 0 resulted in the most missing values, which included all of the previous mentioned parameters. By increasing the jet number fewer and fewer missing values occurred. With a jet number of 1 the leading jet and *PRI_jet_all_pt* parameters could now be assigned non-missing-values. However, if the jet number was larger than 2, there were no missing values at all.

After this we decided to see which percentage of values were missing from the feature columns and we obtained the following groups: one feature with 15% of missing values, 3 features with 40% of missing values and 7 features with 70% of missing values.

The only feature that had missing values that were not explained by the number of jets used in the experience was the parameter *DER_mass_MMC* with 15% of missing values. The cleaning process is explained in the next section.

B. Data cleaning

To clean the data we decided to split the data into 4 different groups depending on the number of jets the experiment had. This allowed us to make not one but 4 different models that depended on the number of jets the experiment had. After splitting the dataset we determined again the number of missing values in the features for each dataset and the only one that still had missing values was *DER_mass_MMC* since, as seen before, was not dependant on the number of jets used. We decided to replace all the missing values with the mean of the feature for each dataset, since the percentages of missing values for this feature were low for all datasets.

To finish the cleaning process we normalized the features to make it easier to derive a model from the presented data.

C. Derived model

For our model we opted for a ridge regression algorithm that we adapted to perform classification: after the ridge regression algorithm we had continuous values of y. We then used the sigmoid function to compress all the possible values of y into the range 0-1 and depending of the value of a selected threshold we attributed the values of -1 if the value was lower than the threshold and 1 if the value was higher. This method was chosen instead of the regular logistic regression because it gave us far better results in terms of time expended and cross validation.

We also built a model of expanded features to make our models more precise. To this effect we built a polynomial of degree d and we made additional features in the matrix that were the square, the cube,..., until to the power of d of each one of the features, amplifying the number of features by a factor of d times the number of initial features.

III. Experiments

Our first experiments were made with the logistic regression algorithm. While comparing it to our training dataset results we found error rates of high 20/low 30 percentages for several values of the lambda and gamma parameters. Since this was not good enough for us, we tried the adapted ridge regression explained in the 'Derived Model' section of this report and the error rate decreased to around 25%, so the we opted for this approach.

After deciding the main regression model we adjusted for our successive experiments the parameters of the sigmoid decision threshold (prob), the degree of the polynomial (d) and the penalizing factor (λ). The gamma parameter used in the logistic regression is not used here. For the adjustment of these parameters we used grid search over values ranging from 0.5 to 0.7 for the threshold, 2 and 5 for the degree and values ranging from 1 to 10^{-5}. To decide on the best model depending on these parameters, we used 4-fold cross-validation and chose the model resulting in the lowest test loss.

After these processes the we still had an error rate of around 20%. Since this result indicated that improvements could be made, the we decided to try feature expansion: firstly by just building polynomials of degree d with the initial features and their powers $(x_n, x_n^2, .., x_n^d)$ and after by using all of the possible cross terms and adding them to the already existing features $(x_n, x_n * x_{n+1}, x_n * x_{n-1}, etc)$. The first method produced an improvement of the results, leading to better models and a reduction of the error rate to around 19% but the second model was too slow, since with the addition of the cross terms the number of features was immense. After a successful iteration of the algorithm we found out that the improvements were not significant enough to justify the time expended in this process so we opted for the first method of feature expansion.

Finally we decided to use this method of feature expansion in association with the logistic regression, in a last ditch effort to use this regression algorithm but no improvements were made in comparison to the adapted ridge regression algorithm.

IV. Results

By comparing our results to the real training sets, we obtain percentages of error of 17% for dataset_0 (the elements of the general dataset with *PRI_jet_num* = 0), 22% for dataset_1 (the elements of the general dataset with *PRI_jet_num* = 1), 22% for dataset_2 (the elements of the general dataset with *PRI_jet_num* = 2) and finally 19% for dataset_3 (the elements of the general dataset with *PRI_jet_num* = 3). Our final result on kaggle was computed by comparing our predictions to the real values of y for the test data set and our best score was 0.78912, with a reproducible score of 0.77525.

V. Conclusion

As a conclusion the ridge regression achieved the best results for our project. However, there are still two ways to improve our result. One method would be to remove outliers and perform a more advanced feature expansion and selection. Another approach to get better predictions would be to use more complex models not yet seen in class, such as neural networks.

References

[1] M. Jaggi and M. E. Khan, "Optimization," 2016, ecole Polytechnique Federale de Lausanne.

[2] M. E. Khan, "Least squares," 2015, ecole Polytechnique Federale de Lausanne.

[3] ——, "Regularization: Ridge regression and lasso," 2015, ecole Polytechnique Federale de Lausanne.

[4] M. E. Khan and R. Urbanke, "Model selection," 2016, ecole Polytechnique Federale de Lausanne.

[5] ——, "Bias-variance decomposition," 2016, ecole Polytechnique Federale de Lausanne.

[6] M. E. Khan, "Logistic regression," 2015, ecole Polytechnique Federale de Lausanne.

[7] J. Brownlee, "Discover feature engineering, how to engineer features and how to get good at it," 20014, https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/.