CSS-433 Machine Learning Project 1

Daniel-Florin Dosaru
Sena Necla Çetin
Natasha Ørregaard Klingenbrunn
École Polytechnique de Lausanne, Switzerland

***Abstract*—The Higgs boson is an elementary particle that is sometimes created as the byproduct of two protons smashing, and can be identified by its decay signature. However, the problem remains that many decay signatures resemble each other, which can make it difficult to identify such a particle. Thus, in this paper we will outline our solution using Machine Learning to estimating the likelihood of a Higgs boson, given a vector of features representing the decay signature of a collision of protons.**

I. Introduction

To approach the problem, we first implemented six separate machine learning algorithms to provide a foundation of techniques with which to work with the input data. Next, as outlined in this report, we pre-processed our data and implemented our methods, taking into account how to refine our model at each step. Finally, we considered our results and further improvements that could be implemented.

II. Pre-Processing of Data

A. Raw Data

To begin, we decided to split the data with a ratio of 80% and train our model using least squares. For this initial input, we computed a training MSE of 0.339 and a validation MSE of 0.341 using least squares. This lead to an accuracy of 74.4% on the validation set for our original model.

B. Handling values of -999

Upon inspecting the feature matrix, we decided to clean our data by removing the 181886 rows with values of -999. However, this augmented both our training and validation MSEs using least squares. Consequently, we next tried to replace all occurrences of -999 in each column with the mean of all values in that column that were not equal to -999. This also slightly elevated the training and validation MSEs in comparison with the raw input. For this reason, we decided to simply leave the -999 values in our data-set (see Table I). As an additional step, we also ensured that there was no data for which the value was NaN.

C. Standardization and Bias Term

We then decided to standardize our test and training data to have a mean of 0 and a standard deviation of 1, using the following formula.

$$x_{standardized} = \frac{x - mean(x_{training})}{std(x_{training})}$$

We also wanted to introduce a bias term, that is, add a column of ones to the feature matrix. This ever so slightly decreased our training and validation MSEs (see Table I). This effect was so insignificant (0.1% increase in accuracy) that we decided ultimately not to clean, standardize, or introduce the bias term in our subsequent models.

Pre-processing	Training MSE	Validation MSE	Accuracy
Raw Data	0.3394	0.3408	74.4%
Removing -999	0.3694	0.3688	72.1%
Replacing -999	0.3436	0.3446	74.1%
Std. & Bias	0.3392	0.3406	74.5%

Table I
Results of Pre-processing techniques

D. Feature Correlation

Finally, we also considered removing some features altogether, particularly features that where highly correlated. We produced a heatmap for raw data to demonstrate the correlation between the features, where darker hues indicate a larger magnitude of correlation. We can observe that the majority of the features have a fair amount of positive correlation between them. We also graphed the correlation after removing the -999 values and noticed that this effectively decreased the correlation between the majority of the features. We can deduce that this is due to the high number of -999 values in our data.

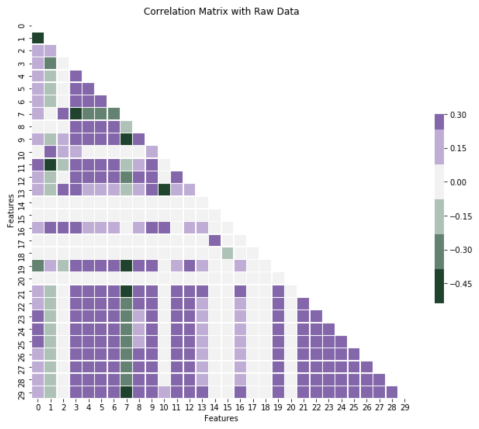

Figure 1. Correlation between Features for raw data

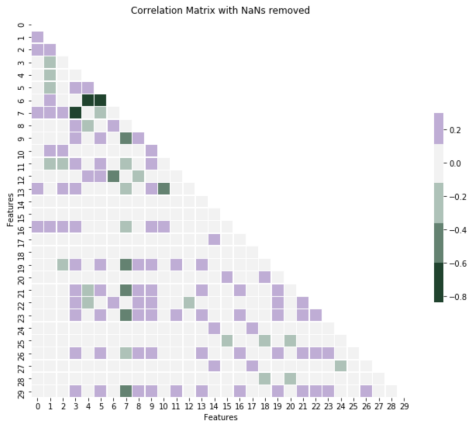

Figure 2. Correlation between Features for data with all rows with -999 removed

III. Feature Augmentation

At this point we noticed that for all the prior pre-processing, our training and validation MSEs were relatively high yet very close in value. From this, we deduced that in regards to the bias-variance trade-off, we err on the side on high variance. Consequently, we noted that it may be beneficial to augment the feature matrix. In order to accomplish this, computed the weights, training MSE, and validation MSE using the least squares method at different degrees of feature augmentation (see Table II). We saw the most success when augmenting our feature matrix to the second degree, giving us an accuracy of 76.5%. Thus we conclude that our raw data itself is underfitting the data, while elevating the degree over a degree of 2 consequently overfits the data.

Degree	Training MSE	Validation MSE	Accuracy
1	0.339427	0.340768	74.4%
2	0.322411	0.323429	76.5%
3	0.370321	0.372009	71.6%
4	0.414190	0.416315	69.5%

Table II
MSE and Accuracy for different degrees of feature augmentation

IV. Feature Augmentation with Regularization

However we also considered that we could augment our features to a higher degree while fighting a high model complexity by introducing a regularization parameter. Thus, we want to compute the ideal combination such that there is a sufficiently high degree augmentation that captures the data set while a low enough complexity as to not overfit the data. To accomplish this, for each degree and lambda, we computed the weights using ridge regression. We tracked the best combination that yielded the lowest validation MSE. More specifically, we found that at a degree of 6 and lambda 1e-15 we got a validation MSE of 0.3032, and hereby an accuracy of 78.9%, which is a clear improvement from using feature augmentation alone.

V. Logistic Regression

To further improve on our last model, we wanted to use logistic regression as it is better suited to classification problems than linear regression. However, as shown in Table III, we ended up with a lower accuracy using the Logistic regression, even after running 5000 iterations of the algorithm. For this reason, we eventually returned to our ridge regression model.

Method	Gamma	Lambda	Loss	Accuracy
Log Reg.	10^-8	/	42	69.9%
Reg Log Reg.	10^-8	1	42	72.3%

Table III
Logistic regression performance

VI. Discussion

Initially we were surprised to achieve 74.4% accuracy from the start, using least squares on the raw data. To improve our model further, we discussed handling the -999 values better. Keeping them in our data intuitively seems like a poor decision, though our results reflect that it had no significant negative impact in comparison to removing the rows entirely or replacing each individual instance. With more time we would have liked to further work with the data to find a cleaner solution to handle the instances of -999.

In addition to this, we also considered many models before deciding on ridge regression, and spent a long time working to refine our logistic regression function under the assumption that this would lead to the best result. Namely, ridge regression is not the obvious choice of model for a classification problem, where most typically logistic regression would produce better results. However, this we later realized may be due to the fact of classifying our data with the outputs -1 and 1 rather than 0 and 1 which is the interval for which the sigmoid function is effective in preventing unnecessary penalization. Eventually, we decided to return to the ridge regression function as our best model.

VII. Summary

In conclusion, we pre-processed our data by cleaning, standardizing and adding a bias term, only to discover that the raw data gave us the most consistent results. We also considered the effect of high data correlation but additionally decided against removing any features in entirely. We manipulated our input set using feature augmentation, then processed our data using 6 machine learning methods to finally arrive at our best model: ridge regression with degree augmentation of 6 and lambda of 1e-15, which yielded us 78.9% accuracy overall.

Acknowledgements

The authors thank the professors and assistants of EPFL CSS-433 for their resources and guidance.