CS-433 Project 1: Higgs boson detection

Yann Vonlanthen, Tiago Kieliger, Benno Schneeberger
Ecole Polytechnique Federale de Lausanne, Switzerland

Abstract—**This report outlines our efforts to tackle the Higgs Boson Challenge on Kaggle. We compare different models and discuss their advantages and drawbacks and try different methods of data cleaning and feature expansion to achieve a robust and accurate prediction model.**

I. Introduction

The goal of this project is the prediction of the presence of a Higgs boson particle, given physical measurements of the ATLAS experiment. The data set was collected at CERN in 2014. [1] While some data points represent the measurements of a ”tau tau decay” indicating the presence of a Higgs boson particle, others are just background noise. In this report we will outline how we used Exploratory data analysis, Data cleaning and augmentation methods, as well as multiple Regression Models to tackle this prediction and classification task. Our contribution lies in the creation of three separate data sets with cleaned features, a comparison of the performance of logistic regression versus traditional linear methods as well as an analysis of the performance trade-offs that have to be made.

II. Models and Methods

A. Exploratory Data Analysis

In order to get an overview of the data set, we have computed the five-number summary of it, which consists in calculating the sample minimum, the lower quartile, the median, the upper quartile and the sample maximum. From this summary, we can observe that some features have more than half of their values that could not be measured and are thus missing (i.e. equal to -999). Furthermore, we noticed that there is only one categorical feature, called ”PRI jet num”. This feature is directly linked to the absence of some values. Concretely there are 3 cases; When jet num is 0, it can be observed that 10 features are always missing and one feature (PRI jet all pt) is equal to zero. When jet num is 1, 8 features are missing all the time. Finally, when jet num is equal to 2 or 3, there is no recurrent irregularity. Some meaningless values can also be found for the feature DER mass MMC, but it needs to be handled separately as it is not always missing depending on the number of jets. More details on the meaningless features can be found in the ”Exploratory Data Analysis” jupyter notebook.

B. Data Cleaning

We will leverage the above important observation by forming 3 different data sets depending on the value of the ”PRI jet num” column (i.e one for each jetnum in 0, 1 or $\{2,3\}$). As a second step we mitigated the effect of the meaningless values (-999) of the first column (”DER mass MMC”) by replacing them with the mean of the column, without taking the -999 values into account in the mean computation. With that approach, except for the first column, the -999 values will have no effect on the model anymore. Note that if we decide to standardize the data, this will effectively set the meaningless values of the first column to 0 and remove their contribution to the model altogether. As explained later on, we will indeed standardize the data when testing the logistic regression model.

C. Feature Expansion

We used a polynomial basis of degree n for the feature expansion: for each column x we add $x^2, x^3, ..., x^n$ to the existing features. This method of expanding the feature vector allows for the model to better represent non-linear data. We used automated tests to compare the accuracy and variance of different choices for the degree n, as it can be seen in Fig. 1.

D. Linear Regression

Note that the results described below are obtained using 5-fold cross-validation with accuracy as a performance metric. We decided to use 5 folds because it has been empirically shown [2] to yield a good bias-variance trade-off. Furthermore, the accuracy obtained through 5-fold cross-validation and our Kaggle submissions were very close which further consolidates this choice.

1) Baseline: As a baseline, we perform a basic linear regression with the closed form least squares method on the raw data set. This yields an accuracy of 74.39 ±0.54%.

2) 3 Models: We then applied the above described data cleaning steps by training 3 different linear models. To compute the prediction for a given sample, we select the corresponding model depending on the value of ”PRI jet num” of the sample. This improved on the baseline with an accuracy of 75.10 ±0.52%. From this result we conclude that the idea of dealing with meaningless values by training 3 models gives better results than leaving them in the data set.

3) Polynomial basis expansion: Finally we expanded the features using a polynomial basis and compared the results for degrees from 1 up to 10. Where degree 10 is an upper bound that takes a reasonable amount of time to compute. Fig. 1 shows the mean values and the maximum deviation of a polynomial feature expansion using 5-fold cross-validation. This graph illustrates the trade-off between accuracy and variance. We can see that degree 8 has the maximum mean accuracy but has a variance greater than that of degree, say 5, but which has a slightly lower mean accuracy.

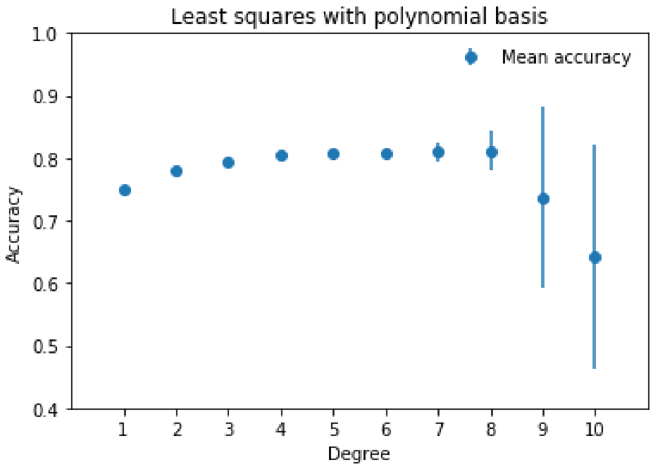

Fig. 1. Least squares with polynomial basis

E. *Logistic Regression*

While the Least-squares method has nice properties, the mean-squared error is not a good match to our classification task at hand. Note that our goal is purely to minimize the number of mis classified cases. While the mean-squared error penalizes positive and negative deviations from the label in the same way, for this task only a deviation towards the other labels value can potentially lead to a worse classification accuracy. [3] This implies that to achieve good results for a classification task using linear regression we need to train the model much more than what is actually necessary. Better models for classification usually rely on ideas using the *nearest neighbors*, as well as *linear or non-linear decision boundaries*. [3]

Logistic regression uses the maximum-likelihood criterion to obtain linear decision boundaries. By augmenting the features by a polynomial basis we can also use the same model to create non-linear decision boundaries. For these reasons we chose the logistic regression model in an attempt to outperform our results from linear regression.

Since the data set is big and using feature expansion increases the dimension space, we used stochastic gradient descent to try to converge to a minimum. While tuning the hyper parameters of our models, we realized that no regularization term is necessary, as the data is not linearly separable, and thus we did not observe weights tending towards infinity. (Note that this is only true for cleaned data, as for the raw data set obtaining convergence was much harder)

In order to have a fair comparison with our linear regression model, we ran our tuned model on the same data sets using 5-fold cross-validation.

III. Results

Below, a table containing the best accuracy achieved with linear regression and logistic regression before and after performing some operations on the data set. In the case of logistic regression, the hyper parameter gamma has been tunes in the three cases to yield the best accuracy.

	Accuracy of Linear Regression using normal equations
Raw Data	74.39 % ± 0.54 %
Data Cleaning	75.10 % ± 0.52 %
Polynomial expansion of degree 8	81.14 % ± 3.13 %

	γ	Accuracy of Logistic Regression using SGD
Raw Data	1×10^{-7}	62.99 % ± 1.55 %
Data Cleaning and standardization	2×10^{-3}	74.65 % ± 0.53 %
Polynomial expansion of degree 2	2×10^{-3}	77.56 % ± 2.56 %

IV. Discussion

We are using two different methods: linear regression and logistic regression. The benefits of using linear regression are that it gives us satisfying results while always converging to a solution and still requiring a low computational cost. However, linear regression with the least squares cost function is not the most appropriate method in the case of a binary classification task. Logistic regression is, in theory, a better fit, but in our case it performs worse than linear regression. This is probably due to the distribution of some features. Polynomial expansion improves the results significantly, but also increases the computational cost.

To slightly improve our results, we could have used additional methods [4] to identify outliers (other than the -999 values) and potentially remove them. The five-number summary can be used to identify the distribution of each feature and therefore use the most appropriate method. For example, if the distribution of a feature seems to follow Gaussian distribution, the values that are further than two or three standard deviations from the mean could be removed.

V. Summary

Through the tuning of different regression methods along with some data cleaning and feature processing, we are able to obtain an accuracy of 82.380 % on Kaggle. We have obtained the best result using linear regression using normal equations.We are splitting our data in 3 according to the number of jets and using polynomial expansion of degree 8.

REFERENCES

[1] Kaggle, "Higgs boson machine learning challenge," https://www.kaggle.com/c/higgs-boson, accessed: 2018-10-26.

[2] J. Brownlee, "A gentle introduction to k-fold cross-validation," https://machinelearningmastery.com/k-fold-cross-validation/, accessed: 2018-10-26.

[3] M. E. Khan, "Classification," *CS-433 class notes*, 2015.

[4] J. Brownlee, "How to use statistics to identify outliers in data," https://machinelearningmastery.com/how-to-use-statistics-to-identify-outliers-in-data/, accessed: 2018-10-26.