Predicting Higgs Boson Signals from Collision Events using Linear Models

Rahul Rajesh, Zhuang Xinjie, Chitrangna Bhatt
School of Computer and Communication Sciences, EPFL, Switzerland

***Abstract*—This paper analyses the effectiveness of using Linear Models (logistic/linear regression) to predict whether particle collision events originate from the Higgs Boson particle. After initial experimentation, feature processing and hyperparameter tuning was done to improve the accuracy. The final model (logistic regression) gave 81% accuracy on the test set.**

I. Introduction

The Higgs Boson is an elementary particle in the Standard Model of Physics which explains why other particles have mass. The aim of this paper is to help improve the analysis of the Higgs Boson. The goal is to predict whether a particle collision event was from the Higgs boson or from some other process/particle given a vector of features representing the decay signatures of the event [1]. This paper will show how we improved upon the given features and optimized hyperparameters for our models followed by a discussion on the final results obtained.

II. Models and Methods

This is a binary classification problem and thus, initially we considered a logistic regression model. A linear regression model (ridge regression) was used as a point of comparison. The results are shown in Table I below. The regularization constant (λ) was set to 0.05.

	Train Accuracy	Test Accuracy
Ridge Regression	0.740	0.740
Logistic Regression	0.639	0.639

Table I: Initial Experiment

However, the results showed that the logistic regression model had a worse accuracy as compared to the ridge regression model. The reasons for this are clear, the feature matrix needs to be augmented and transformed to fit the models better.

A. Data Exploration

The dataset provided consists of 250,000 events with 30 feature columns. The features consisted of primitive/raw quantities and derived data computed from those primitive quantities. All variables are floating point integers except for one variable *PRI_jet_num* which is an integer with value of 0,1,2,3. There are undefined values in the dataset indicated as -999.

B. Data Processing

1) *Label Encoding*: In order for the model to predict the output, the ”Higgs Boson” or ”Background” events have to be encoded as a number. For logistic regression, the encoding was chosen to be [0, 1] while for linear regression the encoding was [-1, 1].
2) *Imputation*: The -999 values among the variables would skew the predictions the model can produce. A good strategy is to replace the -999 with either the mean/median/mode of the column [2]. All three were tested and gave similar results. In the final model, mode or ’most frequent’ value was used for imputation.
3) *Log Transformation*: While analysing the features, we realised that a few variables were positively skewed (e.g. *DER_mass_MMC* or *PRI_jet_leading_pt*). As compared to a skewed variable, a variable that is symmetric or nearly so is better especially for the models we are using. To reduce the skew, an inverse log transformation was applied to all variables that had positive values. Experimentation showed better results after this transformation was performed.

$$x_{*,j} = log(1/1 + x_{*,j})$$

4) *Polynomial Basis*: A simple linear model can only draw a straight line between the points. This would not give good results when the data has a non-linear decision boundary. In order to work around this, feature expansion was done using a polynomial basis function. Powers of the feature matrix was taken and concatenated together to build a larger matrix.
5) *Standardization*: For models like linear regression, it is a good idea to standardize our feature set to reduce multicollinearity especially because we are using a polynomial basis to expand our feature matrix. We standardized our features by subtracting the mean and dividing by the standard deviation.

C. Choice of Model / Strategy

The linear models chosen to tackle this problem were Logistic Regression and Ridge Regression. Logistic Regression is ideal considering this is a classification problem and as pointed out earlier, Ridge Regression was used as a point of comparison.

L2 regularization with a constant (λ) was used to prevent overfitting and help with intermediate calculations. Gradient Descent was used for logistic regression and the closed form solution was employed for Ridge Regression. The models were run separately and the one that had the best accuracy was taken to be the final model.

Additional techniques like grouping the features based on *PRI_jet_num* and running separate models on them was considered initially. However, they did not yield better results due to the reduction in the feature space when training each model.

Tuning Hyperparameters: The models used had a number of hyperparameters, namely the constant for regularization (λ), the *degree (d)* used for feature expansion and the *step-size(γ)* for gradient descent. A grid search was carried out to find out the best value to be used for each of the hyperparameters. A 4-fold Cross Validation was used to verify the train/test accuracy and choose the optimal hyperparameter value.

Table II below summarizes the optimal values that were found:

	Ridge Regression	Logistic Regression
Lambda (λ)	1e-10	0.005
Degree (d)	10	2
Step-Size (γ)	-	0.08

Table II: Results of hyperparameter tuning

Furthermore, Stochastic Gradient Descent was not used despite its speed as it showed too much variance in the final accuracy.

III. Results and Discussion

The tables below showcase the train and test accuracy (again using a 4-fold cross-validation) after carrying out the steps above. The data below was trained using a sub-sample of the data to speed up execution.

	Train Accuracy	Test Accuracy
Ridge Regression	0.843	0.813
Logistic Regression	0.834	0.824

Table III: After Feature Processing

When we compared Tables I and III, it is clear that feature processing has a tremendous effect on the performance of the models. Both models were submitted to the online platform AICrowd to validate on the test set and the accuracies/F1-Score obtained are show below:

	Accuracy	F1-Score
Ridge Regression	0.792	0.671
Logistic Regression	0.815	0.718

Table IV: Leaderboard Submission

Between Ridge Regression and Logistic Regression, Logistic Regression performs better overall. This is expected behaviour considering that this is a classification problem. A Ridge Regression model may give an inaccurate assessment of the loss function especially since we are taking distance as our error (Mean-Squared Error). However, due to high degree/feature processing Ridge Regression still has a good accuracy score that is not too far off.

However, the results obtained are not perfect and there more improvements that could be done to improve the accuracy obtained.

- *Grid Search for hyperparameters*: Due to resource constraints, the grid search done for hyperparameter tuning did not cover all the possible values. There is a possiblity that there is a better combination of parameters that yield better results.
- *Correlation Matrix*: It is also good to plot a correlation matrix between features and drop features that are highly correlated to each other
- *Ensemble Techniques*: Even though this paper focuses on Linear Models, the accuracy for classification problems like this could be improved by using a combination of different models (e.g. Support Vector Machines / Decision Trees) and using voting/averaging the results

IV. Summary

The paper proves that simple Linear Models can be used to solve complex problems like the Higgs Boson Signal Prediction problem. A logistic regression model achieved a final accuracy of 81%. A comparison of the before and after results in this study show the importance of feature processing and hyperparameter tuning in the performance of Linear Models. Further strategies to improve the accuracy of our model such as ensemble techniques were also considered.

Acknowledgements

The authors thank our instructors Martin Jaggi and Rüdiger Urbanke for their guidance on the project.

References

[1] G. C. Claire Adam-Bourdariosa *et al.*, "Learning to discover: the higgs boson machine learning challenge." [Online]. Available: https://higgsml.lal.in2p3.fr/files/2014/04/documentation_v1.8.pdf

[2] W. Badr, "6 different ways to compensate for missing values in a dataset (data imputation with examples)." [Online]. Available: https://towardsdatascience.com