EPFL Machine Learning Higgs Boson Challange Project

Ilker Gul, Can Kirimca, Ahmet Sencan

***Abstract*—The Higgs boson is a particle that its decay signature can identify. This project aims to utilize binary classification methods to determine if a Higgs boson generates a decay signature. This report includes our data processing techniques and compares the machine learning models on this task.**

I. Introduction

The Higgs boson is an elementary particle discovered at the Large Hadron Collider at CERN in 2013. It can be rarely generated by the collision of protons into one another. Due to its instability, the Higgs boson can only be observed indirectly by measuring its decay signature. This project aims to use binary classification techniques to determine whether a given decay signature belongs to a Higgs boson. To achieve this, we performed data cleaning and feature processing on the dataset and trained various models to measure their effectiveness for this particular task. We compared the models based on their accuracy on the test data and obtained the best accuracy using ridge regression.

II. Models and Methods

A. Implemented Methods

We implemented the following methods as asked in the project description:

- Linear regression using gradient descent
- Linear regression using stochastic gradient descent
- Least squares regression using normal equations
- Ridge regression using normal equations
- Logistic regression using gradient descent
- Regularized logistic regression using gradient descent

B. Data Pre-processing

EPFL Machine Learning Higgs data is separated into two sets: a training set of 250000 events and a test set of 568238 events. Each set contains an ID field and 30 feature fields. Additionally, the training set includes the labels for the events, which can be either "s" for a signal event or "b" for a background event. Each feature is either prefixed with "PRI" for raw data directly observed or "DER" for derived information that is produced using the raw data. To be able to use this data, we followed several data pre-processing techniques as follows:

1) Jet Categorization: According to challenge documentation, jets are characterized as pseudo particles, and they are originated from a high-energy quark or gluon. They appear in a detector as collimated energy deposits associated with charged tracks. The number of jets are reported as the "PRI_jet_num" and this discrete feature can take the values 0, 1, 2, or 3 (any higher number of jets is recorded as three). To fit a model specifically tailored, we separated the train and test data into four subsets, each according to this feature. Then, we made the predictions for the test set with the models obtained from the respective subset.

2) Handling Constant Features: After separating the data into four subsets, further analysis showed that there are some constant features for each. However, since our training is done separately for each of the four jet categories, keeping these constant features does not add any information and only increases the model's size. Therefore we removed these features.

3) Handling Missing Values: The challenge description mentions that certain functionalities might not be present for some events. This might be because this value simply does not make sense for the event, it might not be possible to compute it, or a problem with a sensor. Therefore, these features are represented with the value -999.

If the feature does not exist for the whole jet number, the values at the corresponding set will be -999 throughout the column. Hence it will be constant. As a result, it would have been removed at the previous section II-B2. If only some of the events of the jet category have these missing values, we replace them with the mean of the remaining points. This option is more suitable than just removing the events since it allows us to learn from every data point.

4) Handling Outliers: It can be observed that many of the features provided contain outliers. These values can cause the model to have a bias toward them. Hence to smooth the effects of them, we used a method based on statistics. First of all we obtained lower (L) and upper (U) bounds with the following equations:

$$U = Mean(x) + 2.2 * Std(x) \tag{1}$$

$$L = Mean(x) - 2.2 * Std(x) \tag{2}$$

Then any value less than L is assigned to be L, and any value greater than U is assigned to be U. Assuming that the data have a normal distribution, this process removes about 1.3% of the data from both the upper and bottom sides.

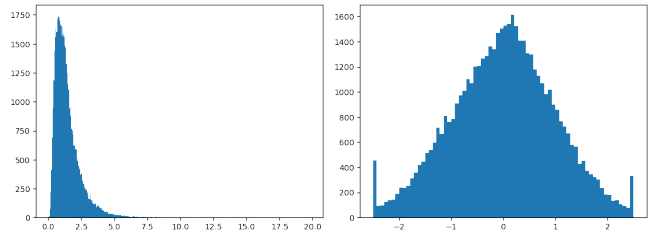
5) Removing Skewness: Skewness can be thought of as a measure of calculating if either side of a distribution has a longer tail. Even though some of the features in our data resemble a normal distribution, some features have quite a longer tail on one side of their peak than the other. In our analysis, we saw that all of these highly skewed features have a positive skew (the right tail is longer than the left

tail). Therefore, to remove this unevenness, we can take the logarithm of these features. They are shown in Table I. To save space, the indices of the columns are used as they are ordered in the given dataset.

Jet Number	Indices
0	0, 2, 3, 8, 9, 10, 11, 13, 16, 19, 21
1	0, 1, 2, 3, 8, 9, 10, 13, 16, 19, 21, 23, 29
2, 3	0, 1, 2, 3, 5, 8, 9, 10, 13, 16, 19, 21, 23, 26, 29

Table I: Logarithm Features for Jet Numbers

Figure 1 illustrates the effects of applying the data pre-processing methods explained so far as well as standardization on the column "DER_pt_ratio_lep_tau".



(a) Before processing (b) After processing

Figure 1: first figure shows the Unprocessed distribution of the "DER_pt_ratio_lep_tau" column. Second figure shows the result after taking its natural logarithm, standardization and outlier removal.

C. Feature Engineering

Linear models are inherently not capable of fitting distributions that are not linear. However, as explained in the lectures, one can apply polynomial expansion to the features of the input to extract more meaning from the data and make the tool more powerful. To leverage this, we extended every feature obtained after data pre-processing by adding a polynomial basis of degree, which we chose using a grid search and cross-validation.

III. Results and Discussion

1) Comparing the Machine Learning Models: We ran all the implemented models on the training set. We used 64 as the batch size to run the training that uses stochastic gradient descent. Nevertheless, even the iterative methods that use stochastic gradient descent were relatively slow to converge. Hence, we chose to use the ridge regression method to conduct more experiments and move faster. We also experimented with the least squares method, but its best results were below ridge regression results. These results are shown in the Table III.

2) Best Results and Hyper-parameter Tuning: In order to test our methods, we used cross validation with 4-fold. In our tests there are two hyper-parameters, over which we ran a grid search. These are the regularization parameter λ and the degree of the polynomial basis that we created. After a lot of experimenting our final grid search was over the lambda values: $[10^{-3}, 10^{-5}, 10^{-7}]$ and degree values: $[6, 7, 8]$. Best results are illustrated in the Table II

Jet Number	λ	Degree	Accuracy
0	1e-7	7	0.844 ± 0.002
1	1e-3	7	0.808 ± 0.003
2	1e-7	8	0.837 ± 0.002
3	1e-3	6	0.841 ± 0.003
Total	-	-	0.831 ± 0.002

Table II: Cross Validation Results. λ refers to the regularization parameter and Degree refers to the degree of the polynomial basis created during the data pre-processing.

3) Ablation Study: In this section, we conduct an extensive ablation study to see the effect of the data pre-processing methods we have used. We have ran a grid search over the hyper-parameters and record here the best results we obtained for each of the specifications. The results are illustrated in the Table III.

Method	Data Pre-processing Method					Accuracy
	J	O	M	S	ST	
Ridge Reg.	✓	✓	Mean	✓	✓	**0.831 ± 0.002**
Ridge Reg.	✓	✓	Median	✓	✓	0.829 ± 0.003
Ridge Reg.	✓	✓	Mean	✓		0.829 ± 0.002
Ridge Reg.	✓	✓	Mean		✓	0.828 ± 0.002
Ridge Reg.	✓		Mean	✓	✓	0.824 ± 0.001
Ridge Reg.		✓	Mean	✓	✓	0.821 ± 0.002
Least Squares	✓	✓	Mean	✓	✓	0.825 ± 0.001
Baseline (Random)						0.508 ± 0.001

Table III: Ablation study on the effect of data pre-processing methods. Column names stand for J: Jet Categorization, O: Handling Outliers, M: Handling Missing Values, S: Removing Skewness, ST: Applying Standardization.

The table clearly shows that the methods we use have an improving effect on the results. The most drastic decrease occurs when we remove the jet categorization or handling outliers.

IV. Conclusion

In this project, we tried to devise a solution to the given dataset by implementing various machine-learning approaches. During the implementation process, we saw that we needed to implement pre-processing in which we included outlier filtering, polynomial expansion, categorization concerning jet numbers, and logarithmic conversion to highly skewed features to produce a comprehensive solution to the problem. In conclusion, the best accuracy score was obtained using the ridge regression model, and we noticed that computational resource is essential for comprehensively testing logistic regression models.