Higgs Boson Classification Using Regression Techniques

Efe Acer, Murat Topak, Daniil Dmitriev
CS433 Machine Learning, EPFL, Project 1

***Abstract*—Machine learning algorithms have become significant in various scientific fields due to the fact that they can make sense of complex, high dimensional data sets. Our work focuses on the application of such algorithms to overcome the problem of binary classification on CERN's Higgs Boson data set. This report proposes a procedure that uses specific regression techniques to work towards a solution to binary classification.**

I. Introduction

The Higgs Boson in simple terms is a highly unstable collection of very strong and weak charges, which are obtained immediately prior to high kinetic energy collisions of composite particles. These collisions generate a huge amount of data that is gathered in large data sets for further research. Our data set is obtained from the measurements in the CERN ATLAS experiments. Even though, the measurements reflect laws of quantum physics, they involve background noise. Our aim is to process the raw measurement data to eliminate the background noise to some extend, and then apply particular regression techniques to construct an accurate classifier. The most suitable regression technique and the methods we used to process the raw data will establish our classification procedure.

II. Models and Methods

A. Data Pre-processing

With pre-processing, prominent errors in the measurements can be cleared from the data set. In our exploratory data analysis, we had the following observations in order:

1) `PRI_jet_num` *is a discrete feature restricted with values 0, 1, 2 and 3*

   Jets are pseudo particles that may appear in the detector when other particles collide. `PRI_jet_num` is apparently a categorical feature denoting the number of jets appeared in an experiment. Other features include various measurements related to the angles between the jets and masses of the jets. Since such measurements have direct correlation with the number of jets, we split the raw data set into four, which are labeled with their corresponding jet number.

2) *There are many zero variance features*

   After splitting the data sets, we observed that some features have zero variance, i.e. they have the same value regardless of the data point. These features are simply removed from the data set due to the fact that they are non informative (see Table I).

Feature name \ Jet number	0	1	2	3
`DER_deltaeta_jet_jet`	✓	✓	-	-
`DER_mass_jet_jet`	✓	✓	-	-
`DER_prodeta_jet_jet`	✓	✓	-	-
`DER_lep_eta_centrality`	✓	✓	-	-
`PRI_jet_leading_pt`	✓	-	-	-
`PRI_jet_leading_eta`	✓	-	-	-
`PRI_jet_leading_phi`	✓	-	-	-
`PRI_jet_subleading_pt`	✓	-	-	-
`PRI_jet_subleading_eta`	✓	✓	-	-
`PRI_jet_subleading_phi`	✓	✓	-	-
`PRI_jet_all_pt`	✓	✓	-	-

Table I: Zero variance features for each jet number data set. (✓denotes the features with zero variance, they are removed.)

3) *There are many* `NULL` *values in the features*

   We spotted many features with the `NULL` value (-999) and replaced these missing values with the median of the non `NULL` values for the particular feature, with the hope of capturing the correct distribution.

4) *There are many outliers in the features*

   Outliers are values that diverge from the overall pattern of the samples. We defined:
   $lb = \overline{x} - 2 \times \sigma(x)$ and $ub = \overline{x} - 2 \times \sigma(x)$, where x is the feature vector, $\overline{x}$ is the mean of x and $\sigma(x)$ is the standard deviation of x.
   We considered every value outside of the interval $[lb, ub]$ as an outlier and rounded to the closest endpoint. Figure 1 and Figure 2 are given to illustrate how effective this is to capture the underlying normal distribution of a particular feature.

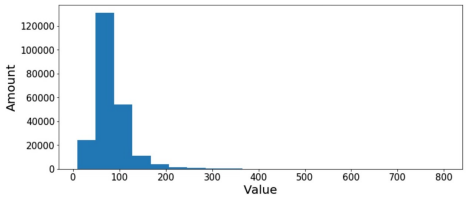

Fig. 1: distribution of `DER_mass_vis` before processing outliers (for jet number 0)

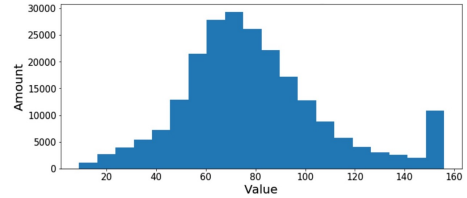

Fig. 2: distribution of `DER_mass_vis` after processing outliers (for jet number 0)

After these pre-processing procedures, we obtained four data sets labeled with the corresponding jet number, with no zero variance features, no NULL values and no outliers, both for training and testing.

B. Implementing Regression Techniques

Six fundamental regression techniques were implemented to see which one is the best fit for the task. In order to have a rough understanding about the techniques, we ran each one of the six on the raw data set by splitting 0.8 of it to a training set and 0.2 of it to a test set. We had to standardize the data for the iterative algorithms, since non standardized data led to computational overflows. We have also non-exhaustively optimized the hyperparameters and number of iterations for each algorithm. The performance of each technique together with the relevant parameters are given in Table II.

	γ	λ	max-iter	%
Gradient Descent ✓	0.05	-	1000	71.672
Stochastic Gradient Descent ✓	0.005	-	3000	70.594
Least Squares	-	-	-	74.402
Ridge Regression	-	10^{-6}	-	74.396
Logistic Regression ✓	10^{-6}	-	3000	75.137
Regularized Logistic Regression ✓	10^{-6}	0.1	3000	75.134

Table II: Performance of regression techniques on the raw data set. (✓means that the data is standardized before running the algorithm, γ denotes step size, λ denotes the regularization parameter, max-iter is the number of iterations and % is the percentage of correct predictions)

The performance outcomes went along with our theoretical expectations. Hence, at the beginning we decided to proceed with the Regularized Logistic Regression. This technique is theoretically better than others, since it predicts the *probabilities* of the class labels and penalizes over-fitting. However, we encountered the problem of having large iteration numbers for the logistic loss to converge near its minimum. This problem drastically increased the time required for hyperparameter tuning. Thus, we proceeded with the second best option, Ridge Regression.

C. Feature Engineering

It is rarely the case in modern physics that features such as mass, angles and kinetic energy have a linear relationship. Considering that these features usually have a non-linear complex relation, we expanded each feature using a certain polynomial basis. We have also added cross-terms by pairs as new features to enforce the significance of the product of different features. Then, we decided to add the logarithm and square root of the initial features as new ones for the reason that these functions frequently arise in modern physics. After all, we noticed that there are many features denoting the angles between the jets for the data sets with jet number greater than 1, so we added the sine and cosine values of the initial features to the feature matrices of these data sets.

Let x_i be a feature of the initial feature vector $X_{D\times 1}$, where D is the number of dimensions.
$\forall x_i \forall x_j \in X_{D\times 1} \mid 0 \leq i \neq j \leq D$:

(i) Polynomial expansion (degree) : $\{1, x_i, x_i^2, ..., x_i^{degree}\}$
(ii) Logarithm: $\{\ln(x_i)\}$
(iii) Square root: $\{\sqrt{x_i}\}$
(iv) Sine, cosine: $\{\sin(x_i), \cos(x_i)\}$
(v) Cross terms: $\{x_i x_j\}$

are the new features we added and tested.
The effect of using different combinations of these new features are given in Table III. We chose the best combination for each jet number.

Features added \ Jet number	0	1	2	3
(i)	84.46%	80.90%	83.74%	84.19%
(i) + (ii) + (iii) + (v)	84.85%	81.35%	-	-
(i) + (ii) + (iii) + (iv) + (v)	-	-	84.91%	85.13%

Table III: Correct prediction rates for the train set when particular features are added. (The model is constructed using ridge regression with optimal hyperparameters.)

D. Tuning Hyperparameters

After we agreed on Ridge Regression, an exhaustive grid search became necessary to find the best hyperparameters. Hence, we set up a grid search for the degree of the polynomial basis and the regularization parameter lambda. For each pair of hyperparameters explored by the grid search, we ran a 10 fold cross validation to avoid overfitting while we were maximizing the percentage of correct predictions. We obtained the following hyperparameters:

	degree	λ
jet number 0	6	6×10^{-5}
jet number 1	6	2.3×10^{-3}
jet number 2	6	4.6×10^{-9}
jet number 3	6	5.7×10^{-5}

Table IV: Optimal hyperparameters for ridge regression for each jet number

III. Results and Discussion

The results we have obtained throughout our work were in accordance with our theoretical expectations. For practical reasons mentioned earlier, such as time constraints, we could not incorporate Regularized Logistic Regression to our procedure. Given more computational power, we believe that we could have achieved better results using Regularized Logistic Regression. However, we are gladsome since our procedure achieved a correct prediction rate of 83.639% on the test ran in the Kaggle platform's public leaderboard.

IV. Conclusion

Our work showed us that one must approach a machine learning problem in various ways. We saw that no regression technique works without a comprehensive data pre-processing and a good feature engineering. We learned that domain specific knowledge is one of the key aspects of this field, since it helps to identify useless features together with the important ones. We also noticed that computational power plays a huge role by making hyperparameter tuning much faster. In conclusion, we believe that our procedure is not perfect but well-reasoned and educatory.