Machine Learning: CS-433 Class Project 1

Serif Soner Serbest, Erdem Bocugoz, Asli Yorusun
School of Computer and Communication Sciences, EPFL, Switzerland

***Abstract*—This report is the first class project of the Machine Learning course taught at EPFL. The task is to develop an algorithm using simulated data with features characterizing events detected by ATLAS so as to classify events into "tau tau decay of a Higgs boson" versus "background" [1].**

I. Introduction

In this project, our objective is to predict CERN's simulated particle collision events as either a Higgs Boson signal or background noise as of binary classification on a real life data-set by applying several regression and classification machine learning algorithms that we have seen so far as well as the exploratory data analysis, feature processing, and the comparison of different models using cross-validation. [1] The implementation and the discussion of the algorithms are explained in detail respectively in Section II and in Section III.

II. Models and Methods

A. Initial Implementation of the Given ML Methods

We implemented six Machine Learning methods as it is requested and we carefully tuned hyperparameters. We tested our implementations with raw data provided by CERN. Initial performances of the methods are tested on data-set presented in the table below.

	Parameters used				
Methods	Lambda	Gamma	Degree	Max_Iter	Pred (%)
Gradient Descent	/	0.1	1	500	74.43
Stochastic GD	/	0.1	1	500	74.43
Least Squares	/	/	1	/	74.50
Ridge Regression	0.0001	/	7	/	80.80
Logistic Regression	/	0.5	1	1000	72.79
Regularized Logistic Regression	0.0001	0.5	1	1000	72.75

TABLE I
Initial Implementation of the ML Algorithms

- As shown in the Table I, Ridge Regression outperformed other regression and classification algorithms with 80.8% of accuracy in predicting particle collision events as either a Higgs Boson signal or background noise.
- As it is expected there is no difference between the performances of Gradient Descent and Stochastic Gradient Descent methods although Stochastic Gradient Descent is faster. Least Squares method performs slightly better and significantly faster than the two methods mentioned above.
- Although we are working on classification problem instead of regression problem, we could not obtain good results in logistic regression and regularized logistic regression methods. However, we obtain better results with classification methods after data processing which will be discussed in Section III.
- After obtaining initial results of the methods, we applied exploratory analysis and decided that using all given features as they are is not a good approach to handle the binary classification problem. We then decided to use certain feature engineering methods which are explained in detail in Section II-B in order to find more reliable and accurate results.

B. Data Analysis and Processing

1) Description: The data-set includes 250.000 events with 30 features columns. Features consist of primitive variables which are raw quantities measured by the detector, and derived variables which are computed by the physicists using primitive variables [2]. All variables are floating point except number of jets labeled as *PRI_jet_num* in the data-set. As a result of observing mean, standard deviation and histogram of each feature, we detected certain features which have significantly large dispersion. Finally, -999.0 is used in data-set to indicate whether the value is meaningless or cannot be computed.

2) Categorization: PRI_jet_num label represent the number of pseudo particles called jets that appear in detector which can take values 0, 1, 2 and 3. We decided to categorize the data according to number of jets. We created 3 subsets of data-set where *PRI_jet_num* is 0, 1, 2 and 3. We merged 2 and 3 to obtain more equally divided subsets. As a result of categorization, we observed that the columns with *NaN* values, except *DER_mass_MMC*, are divided into sub-columns with either completely *NaN* values or without any *NaN* values. It is explained by the fact that some physical calculations of the features require one or more jets.

	Ratio of Nan		
Features	Subset 0	Subset 1	Subset 2
DER_mass_MMC	0.26	0.09	0.06
DER_deltaeta_jet_jet	1.0	1.0	0
DER_mass_jet_jet	1.0	1.0	0
DER_lep_eta_centrality	1.0	1.0	0
PRI_jet_leading_pt	1.0	0	0
PRI_jet_leading_eta	1.0	0	0
PRI_jet_leading_phi	1.0	0	0
PRI_jet_subleading_pt	1.0	1.0	0
PRI_jet_subleading_eta	1.0	1.0	0
PRI_jet_subleading_phi	1.0	1.0	0

TABLE II
Ratio of *NaN* values in features according to subsets

3) Feature Transformation and Standardization: In each subset, we eliminate columns with same values. For *DER_mass_MMC* column, we applied different imputation methods such as substituting the values with mean, median, mod of the column. Although there is not significant difference in

[1]For more information about the Higgs Boson Machine Learning Challenge, you can see the official challenge at https://www.kaggle.com/c/higgs-boson.

results, we selected median method. During the analysis of the features, we observed that the distributions of 11 columns show right skewness. We applied logarithm transformation to reshape these distributions to increase performance of our learning method. Finally, we column-wise standardized our data-set.

$$x_{*,j} \rightarrow \log\left(1 + x_{*,j}\right) \tag{1}$$

4) Polynomial Expansion: We expand our data-set in a polynomial way by taking powers of feature matrix, namely X, and concatenating them together. On top of that, we create new features by multiplying every column of X with each other. Our intuition about this approach is such that each feature represents a concept in physics such as mass, energy, momentum and angle. Hence, dependencies of these concept are presumably non-linear. That intuition motivates us to multiply columns to expand our feature dimension.

$$X = (1 \mid X^k) \text{ for } k \in \big(0, d\big) \tag{2}$$

$$X = (X \mid (x_{*,i} * x_{*,j})) \text{ for each } i \text{ , } j \text{ where } i \neq j \tag{3}$$

5) Model and Hyperparameter Selection: After building our learning model with implemented methods on processed data, we obtained best results from ridge regression and regularized logistic regression algorithms. In order to get best predictions, we needed to select optimal lambda value to penalize the cost function and optimal degree for polynomial expansion in the above-mentioned methods. In the decision process of hyperparameters, we used 10-fold cross-validation method. We aimed to select an optimal lambda value to avoid under-fitting or over-fitting scenarios. For ridge regression algorithm, test error is significantly greater than train error when penalty value is less than 10^-7, which is an indication of over-fitting. Moreover, for lambda values greater than 10^-1 both test and train error start to increase, which indicates that the model is under-fitting. Since we are aiming to avoid both of those cases, our cross-validation function returns value of lambda, at a point where test error is the minimum.

III. Results and Discussion

We obtained our results with Ridge and Regularized Logistic Regression with tuned hyperparameters as it is seen in Table III.

	Lambda	**Gamma**	**Degree**	**Max_Iter**	**Accuracy(Training)**
Method	**Subset 0**				
Ridge Regression	0.0535	-	4	-	0.8369
Regularized Logistic Regression	0.0001	0.2	2	10000	**0.8384**
	Subset 1				
Ridge Regression	0.0001	-	12	-	**0.8119**
Regularized Logistic Regression	0.0001	0.2	2	10000	0.7955
	Subset 2				
Ridge Regression	7.91E-7	-	12	-	**0.8473**
Regularized Logistic Regression	0.0001	0.2	2	10000	0.8328
					Overall Accuracy
Ridge Regression	-	-	-	-	**0.8322**
Regularized Logistic Regression	-	-	-	-	0.8175

TABLE III
PARAMETERS AND TRAINING ACCURACY FOR FINAL RESULTS

In Regularized Logistic Regression, we faced with certain technical difficulties with polynomial degree higher than 2. In our platform, we encountered with an overflow warning in sigmoid function due to the exponential part of the function. We take certain precautions to overflow issue by rounding off the values to the closest value our platform can represent. However, we could not obtain the desired results because of our manipulation of the cost function. Nonetheless, we obtain good results in subset 0 with polynomial degree of 2.

From the results, we conclude that Ridge Regression performs better in the data-set comparing to Regularized Logistic Regression. However in Subset 0, Regularized Logistic Regression method provides higher accuracy than Ridge Regression. By using only Ridge Regression method, we obtained 83.176% accuracy on Kaggle leader board. To increase our accuracy, we decided to use a hybrid approach by training Subset 0 with Regularized Logistic Regression and the other sets with Ridge Regression. With this approach, we obtained 83.216% accuracy on Kaggle leader board.

Since feature engineering is the crucial part of the data-set, as a future work, we would like to refer certain feature engineering methods to increase the performance. Checking correlations of features with each other and the predictions would be beneficial to identify and increase the weight of the important features. We obtain better results by adding new features to our dataset as it is discussed in II-B4. Furthermore, new features can be added to the data by applying other feature generation methods.

IV. Summary

In this project, we applied machine learning methods that we learned in class to the Higgs Boson data-set to predict it is signal or background noise. Throughout the project, we noticed that feature engineering is the most important part of the predictions. We first preprocessed the data then divided it into subsets according to their pseudo particle categories. Consequently, we performed polynomial expansion on our feature matrix to expand our feature dimension. After these steps, our accuracy scores improved around 10%. This improvement can clearly be seen by comparing logistic regression scores in Table I and Table III. Lastly, we used an hybrid approach by to increase our accuracy. As a future work, we pointed out the overflow problem in classification methods and certain feature engineering methods to improve the performance.

References

[1] EPFL, "EPFL Machine Learning Lecture - Project 1," *URL hhttps://www.kaggle.com/c/epfml18-higgs/l*, 2018.

[2] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kegl, and D. Rousseau, "Learning to discover: the Higgs Boson machine learning challenge," *URL http://higgsml.lal.in2p3.fr/documentation*, 2014.