# Machine Learning Project 1

Kuan Tung
*Institute of*
*Electrical Engineering, EPFL*
*kuan.tung@epfl.ch*

Chun-Hung Yeh
*Institute of*
*Electrical Engineering, EPFL*
*chun-hung.yeh@epfl.ch*

De-Ling Liu
*Institute of*
*Financial Engineering, EPFL*
*de-ling.liu@epfl.ch*

***Abstract*—Machine learning nowadays can be applied to various scientific areas to achieve accurate predictions from a large amount of data. In this task, we exert some classic machine learning techniques to do classification on the Higgs Boson dataset.**

## I. Introduction

In this report, we show the implementation of the six basic ML methods we have seen in classes and labs to solve the problem. In addition, we describe how we improved the basic ML methods by analyzing the dataset, better feature processing and working on more robust models.

## II. Data Preprocessing

Data preprocessing is used to transform raw data into an explainable data set. In this project, we mainly deal with missing value and select proper features to enhance the performance.

After digging into the data set, we found that within all the 30 features, 11 of them contain missing value. Without any preprocessing, we first ran a linear regression and obtained an accuracy 0.74 on training data. On the contrary, when we completely exclude variables with missing value from our model, the accuracy is 0.73 on training set. Hence, we could infer that features with missing value can still contribute to the prediction as they preserve some information in both missing and observed scenario, since whether a value is missing could be view as a binary feature. Thus, in the basic ML models, we train on the full data set without dropping any missing value and feature.

However, to obtain a better accuracy we need a better solution to handle the missing value problem. We checked out the full description of the features and found that most of the features with missing value are undefined when the `PRI_jet_num` equals to 0 or 1, since these features require one or more jets to be measurable such as the distance and the angle between the two jets. Therefore, grouping the data set by the feature `PRI_jet_num` can eliminate the missing value resulted from the physical constraints of the number of jet in the event. Also, the information contains in these missing value is simply the number of jet in the event and it can be attributed to the bias term in each group.

For the last step of data preprcessing, we noticed that the missing value in the feature `DER_mass_MMC` will not be deleted by the previous step. Given that the feature is the estimated mass of the Higgs boson candidate and the empirical density of the feature in the training set is bell-shaped, we decided to replace it with the median of the same column in order not to affect the standardization. Lastly, the clean data set after the preprocessing will be adopted in the improved ML method.

## III. Models and Methods

### A. Basic ML Methods

The following are the six ML methods we implemented:

*1) Linear Regression Using Gradient Descent (GD):*

First, we normalized the features with their respecting mean and standard deviation. We did not normalize the `PRI_jet_num` feature since it was a discrete variable. An all one vector was added to the feature matrix to present the bias term. The loss and gradient were calculated using the equations mentioned in lecture 2a. The initial weights were set to be all zeros. We then updated the weights. Finally, we repeated the above process until the maximum iteration (we set it to be 1000) was reached or the difference of the two recent losses was smaller than a threshold.

*2) Linear Regression Using Stochastic GD:*

Basically the same as in III-A1, the only difference was when calculating gradient, instead of using the entire data, we randomly selected a mini-batch of data. In this project, the mini-batch size was set to 1.

*3) Least Squares Regression Using Normal Equations:*

Using normal equations instead of gradient descent to find the weights. We used *np.linalg.solve* to solve the weights.

*4) Ridge Regression Using Normal Equations:*

Basically the same as in III-A3, the only difference was an added L2 regularization term in the left side term in the normal equations.

*5) Logistic Regression Using GD:*

The procedure was the same in III-A1. However, we changed the equation to calculate loss and gradient mentioned in lecture 5b.

*6) Regularized Logistic Regression Using GD:*

Basically the same as in III-A5, the only difference was an added L2 regularization term in the loss function and the gradient.

### B. Improved ML Methods

Apart from the basic ML methods, we designed two different methods to improve our model for increasing accuracy.

*1) Ridge Regression with Preprocessing:*

Based on our physics knowledge, we assume that there might exist some complex relationship between these physical quantities. As a result, we expanded each feature by a certain polynomial basis and added cross terms as new features to capture the difference in higher degree and the interaction between features. In addition, we noticed that there are some features in the data set representing the angles between the jets, so we applied trigonometric functions on the original features to create more non-linear features.

*2) Regularized Logistic Regression with Preprocessing:*

We also worked on a penalized logistic regression model. Since we found that over 90% of the event with a missing `DER_mass_MMC` is in group b, in this model, we decided to assign the events which their `DER_mass_MMC` is missing into a new group. Meanwhile, we adopted four-fold cross validation to determine the hyperparameter in each group.

*3) Neural Network:*

Lastly, we tried to implement a neural network with a 64 nodes hidden layer. Even though the nature of neural network is that it will perform feature transformation within the network, we still conducted two experiments, one with feature preprocessing and one without, to verify this neural network property. We implemented RMSprop, Adam optimizer, and used stochastic gradient descent to improve the model. We reported the best setting in the following.

For neural network with preprocessing, the best optimizer and mini-batch size is Adam and 64; for neural network without preprocessing, the best optimizer and mini-batch size is RMSprop and 250.

## IV. Results

### A. Basic ML Methods

The original data was split to training data and validation in the ratio 80:20. We used grid search to find the best parameters for each method.

The results of testing accuracy are shown in Figure 1. We can see that both logistic regression based methods achieved the highest accuracy. This is reasonable because the problem is a classification problem, and the loss function of logistic regression is designed for classification. The linear regression using stochastic GD got the lowest accuracy. We assumed it is because there are too many training samples (200000). With mini-batch-size set to 1 and max iteration set to 1000, there was not enough iteration to find a better weight. If we are allowed to increase the mini-batch-size, this method will definitely perform better. We also noticed that the regularized methods performed the same as the normal version. The reason behind it could be because all the methods did not overfit the data, the regularization was not needed.

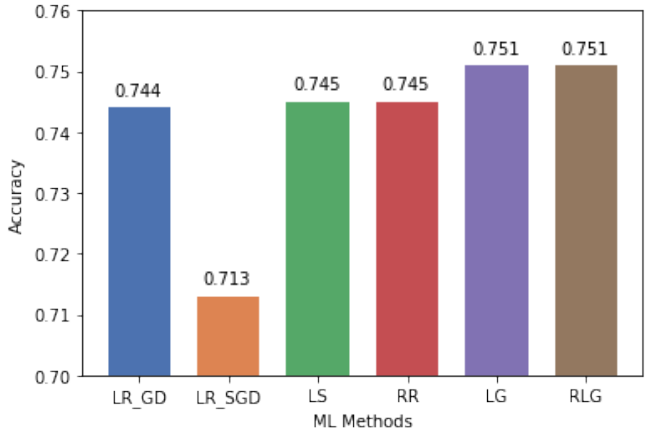


Figure 1. Testing accuracy of the basic ML Methods.

### B. Improved ML Methods

From Table 1, we found out that the performance of our neural network model is the best among the improved ML methods, and that neural network without preprocessing performed even slighty better than neural network with preprocessing. Regularized logistic regression performed much better than the basic one, but still lower the neural network model. On the other hand, ridge regression with fine feature engineering only improved slightly with respect to the basic method.

After parameter sweeping, we can see that neural network is indeed the powerful ML algorithm in classification. We also demonstrate the capability of neural network that it can preprocess the data inside its architecture. The performance of the regression models, however, is highly dependent on the quality of data preprocessing.

Table I
Comparison of improved ML methods

| Model | Accuracy | F1-score |
|---|---|---|
| RR with preprocessing | 0.753 | 0.651 |
| RLG with preprocessing | 0.821 | 0.73 |
| Neural Network with preprocessing | 0.838 | 0.75 |
| Neural Network w/o preprocessing | 0.839 | 0.754 |

## V. Summary

This report presents a exploring process of building and improving the machine learning system. Among all the basic model, logistic regression achieves the best performance. Neural network outperforms all the other models, showing its capacity to learn the correlation inside the data.

In the future work, we can implement different optimizer to approximate the optimal value of loss function, such as Adagrad, Newton method. Also, we can increase the depth and try some modern deep neural network techniques to create a more capable model. For instance, implementing batch normalization if we add more hidden layers, or using dropout to avoid the overfitting, which we have already encountered in the end in our neural network model.