CS-433: Machine Learning Project 1

Xingyu SU, Yuxing YAO, Haobo SONG

I. Introduction

The Higgs boson is an elementary particle in the Standard Model of physics which explains why other particles have mass. Thus, finding a method to discover Higgs boson is of great significance. In this project, we will apply machine learning techniques to actual CERN particle accelerator data to recreate the process of "discovering" the Higgs particle.

To recreate the process, we use the dataset provided by ATLAS experiment at CERN. This dataset includes N=250,000 samples, each of which contains 30 features of particle and also a label to state whether it is a Higgs boson. Thus, the data dimension is $(x_i, y_i)_{i=1}^{N}, x_i \in \mathbb{R}^{1\times 30}$. Each sample represents a collision of a stream of protons.

In this dataset, We apply linear regression, logistic regression and ridge regression with all different hyper-parameters and methods to discuss its effect.

II. Data Analysis

When provided, The dataset is divided into training dataset(TrainSet) and testing dataset(TestSet). The TrainSet's labels are visible while TestSet's labels are invisible. The TrainSet contains 85,667 positive labels and 16,433 negative labels. To start with, we would like to know if TrainSet and TestSet are in the same distribution.

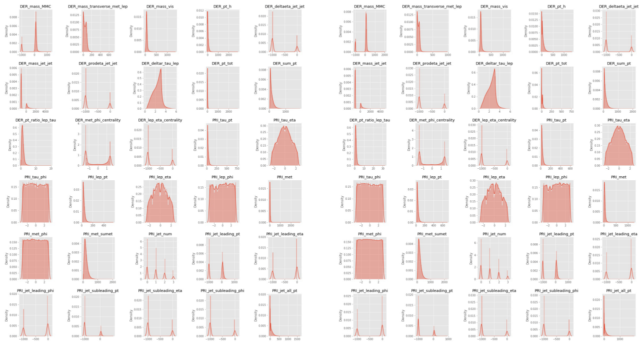

Fig. 1: Dataset Distribution. Left: TrainSet Distribution; Right: TestSet Distribution

Fig.1 (TrainSet in the left and TestSet in the right) show that their distributions are almost the same. Therefore, modifying dataset is not necessary.

The TrainSet contains 30 different features. Different features contribute differently to its classification, some may be strongly connected while others may be weakly connected. Feature correlations with label in Fig. 2 indicates DER_mass_transverse_met_lep, DER_met_phi_centrality, DER_mass_MMC are the most related features. Data augmentations are applied according to Fig. 2.

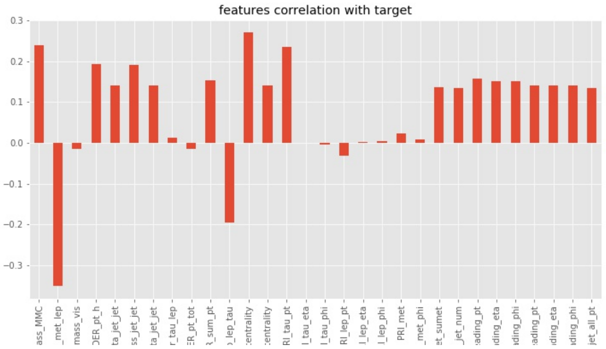

Fig. 2: Feature correlations with label

When analysing the complexity of data, we find that a large amount of outliers -999 exists in 11 feature columns, which shows a big difference with normal data. We assume -999 may represents missing data and needs data pre-processing.

III. Implementations

A. Pre-processing

Based on the observation that there exists outliers in our samples, to be specific, those features with value -999 in each feature column and value 0 in column 'PRI_jet_all_pt', we would like to pre-process these samples to eliminate the impact of outliers.

To pre-process the TrainSet, we first compute the arithmetic average and standard deviation of the inliers, named avg_{train} and std_{train}, and normalize them. Then we set all the outliers to 0 so that they would not contribute to the weighted sum. When pre-processing test set, we just directly apply avg_{train} and std_{train} for normalization and set outliers to 0 as what we did for TrainSet.

B. Data Augmentation

We implement the polynomial data augmentation to prevent our models from underfitting. That is, for each sample $X \in R^{30}$, we augment it to become $\{X, X^2, X^3, ..., X^{degree}\}$, based on the hyperparameter $degree$.

C. Cross Validation

To minimize the variance of our models, we adopt 5-fold cross validation while training.

D. Machine Learning Methods

In this project, we finish all the required implementations, including $least\_squares\_GD$, $least\_squares\_SGD$, $least\_squares$, $ridge\_regression$, $logistic\_regression$ and $reg\_logistic\_regression$. We additionally implement logistic regression with Newton's method, which turns out to be one of the best models we have on this dataset.

IV. Experiments

A. Ridge Regression

In this section, we test the performance of ridge regression with different combinations of hyperparameters $degree$ and λ. In Fig. 3 and Fig. 4 we show respectively how prediction accuracy will change according to $degree$ or λ when the other parameter is fixed.

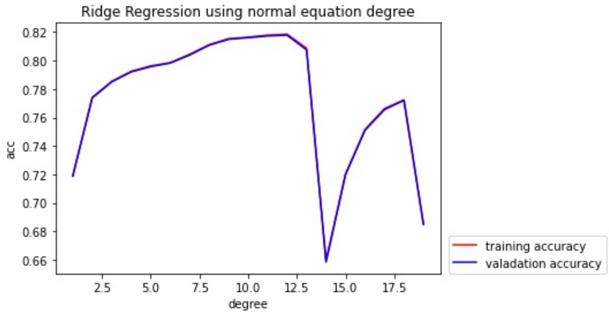

Fig. 3: Ridge regression accuracy trend with respect to degree

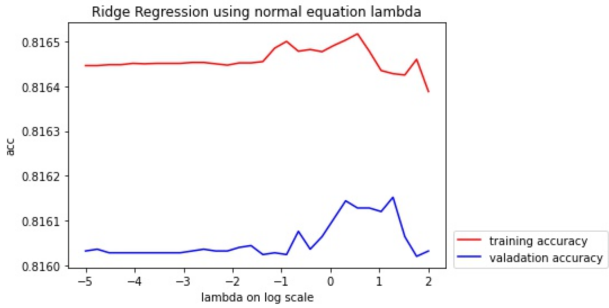

Fig. 4: Ridge regression accuracy trend with respect to λ

B. Logistic Regression

In this section, we test regularized logistic regression performance by Gradient Descend(GD) method and Newton method. Fig. 5 points out the data degrees' influence to accuracy in newton method. When degree equals to 4, Newton method gets its best accuracy. While in gradient descent, gradient descent reaches its best with degree=1. The best learning rate for GD method and newton method are 1 and 0.4.

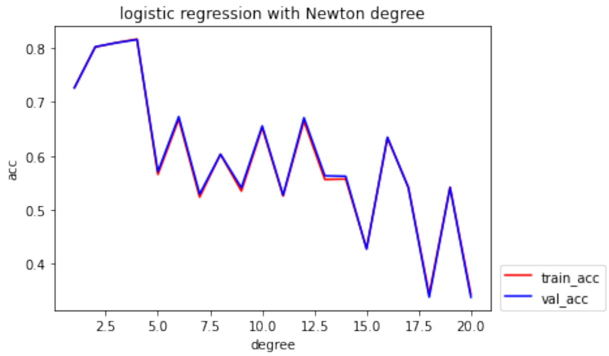

Fig. 5: Newton Method Performance with Different Degree

Fig. 6 illustrates GD method and Newton method loss in iterations with their best parameters. Newton method shows way better performance than Gradient Descent with more than 25% loss decreasing and more than 50% convergence time decreasing.

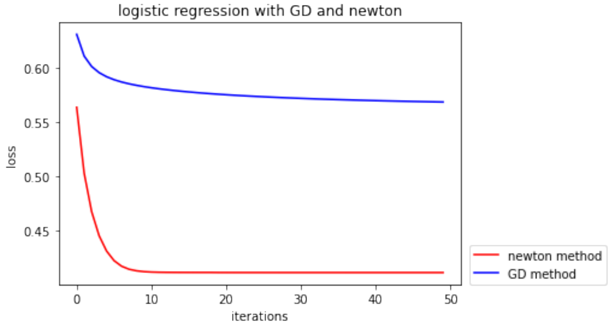

Fig. 6: Logistic Regression Loss with Different Method

With formulation $x^{(k+1)} = x^{(k)} - \lambda(H^{(k)})^{-1}\nabla f(x^k)$, which is a second-order approximation, Newton method can better fit $x^{(x+1)}$ and $x^{(x)}$ though it also results in higher computational complexity in every step. Thus, it can obtain quicker convergence and lower loss result.

C. Other Methods

We also implement linear regression with GD method and SGD method. These methods don't show better performance than methods mentioned above, so we don't explain it in detail.

D. Results

The best results of each method are implemented in the following table.

Method	Accuracy	F1 score
Least Squares(GD)	68.3%	0.590
Least Squares(SGD)	62.4%	0.519
Linear Regression	72.6%	0.444
Ridge Regression	**82.0%**	**0.722**
Logistic Regression(GD)	72.2%	0.662
Reg Logistic Regression(GD)	72.2%	0.662
Logistic Regression(Newton)	**81.8%**	**0.722**

The best result in all is achieved by Logistic Regression (Newton method) and Ridge Regression. All models we obtained have low variance and there is no overfitting.

V. Summary

Through this project, we have obtained a better understanding of different machine learning methods. We also learned that data pre-processing and augmentation can really make a difference for prediction tasks. Based on all these techniques, even if we just exploited some elementary machine learning models with limited computational resources, we can still achieve a relatively precise prediction on this complex Higgs Boson dataset.