Project 1 on Machine Learning team Yoor

Sergei Volodin[1], Baran Nama[1], and Name here 3[1]

[1]EPFL
{*sergei.volodin,baran.nama,email3*}@*epfl.ch*

***Abstract*—A classification dataset from LHC is being studied. First, the data is thoroughly explored using visual aids. Several basic Machine Learning methods are applied. Results are evaluated using cross-validation. Model overview is given and the best model is chosen.**

I. Introduction

Claim: it is possible to use simple methods for this dataset

1) What is the data (Simulation from LHC, details from physics)
2) What are we trying to do? Get the best classification score
3) Overview of data, diagrams of features, feature selection, feature augmentation
4) Methods and their choice (Linear regression, logistic regression, ridge regression) because of simplicity

II. Models and Methods

1) Least squares. Problem: missing data, overfit
2) Mean imputation. Problem: overfit, meaninglessness for some features
3) Feature binarization, add new feature 'feature missing', add squares for features for mass
4) Ridge regression using k-fold. Problem: low accuracy (?)
5) Logistic regression
6) Nearest neighbors?

A. K-Fold Cross validation for hyper parameter

In the previous chapter, we tried to train a basic linear model to estimate test label from input data. When we try to validate our model, we simply use train data to measure how well our model estimates. However, the problem is that our model is overfitting or under fitting the train data so that train error might not seem to be a good indication of how well the learner will generalize to an independent/ unseen data set. At this point, we decided to use cross validation technique to estimate our model performance for new data.

The second problem in previous chapter is that model parameters are simply unbounded, which can be resulted in overfitting to train data. Therefore, we need to introduce ridge regression to regulate our model parameters. However, we need to hyper parameter optimization for ridge regression in order to predict best model parameters for linear regression. Cross validation technique is suitable to select best hyper parameter for ridge regression. The pseudocode algorithm for this chapter is given below:

- Setup a lambda set for testing ridge regression hyper parameters
- Estimate model parameters using each lambda values in the lambda set for ridge regression
- Apply k-fold cross validation for each model, calculate and store train and test error
- After calculating train and test error for all possible lambda values, choose the model parameters which has lowest test error
- Visualize train and test error for each lambda values to check model correctness visually

In the first setup, we did not introduce any higher degree polynomial basis other than input parameters. In this setup, the best lambda value: 10^{-6}, test error: 0.3378 and train error (for best test error): 0.3377 (k-fold: 5 for all test benches). The result plot is given below (Figure -1):

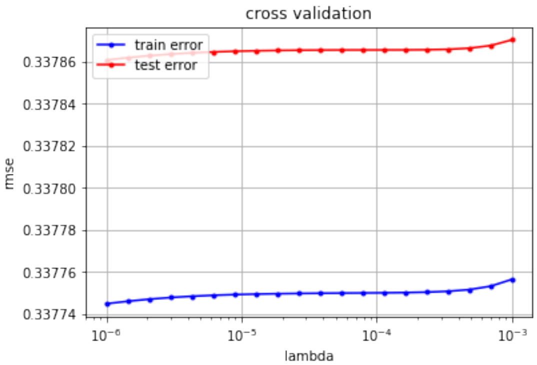

Figure 1: MSE - Lambda plot for first degree linear model

After testing the first degree linear model, we decided to add several degree polynomial basis to decrease bias and increase accuracy. The result plot is given below (Figure -2,3,4 and 5):

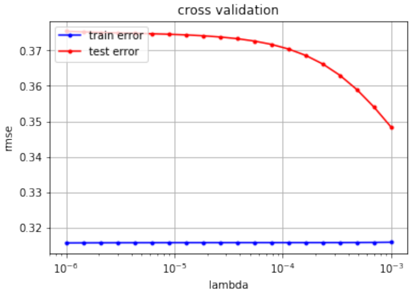

Figure 2: MSE - Lambda plot for second degree linear model **The best lambda value: 0.01, test error: 0.3483 and train error (for best test error): 0.3158**

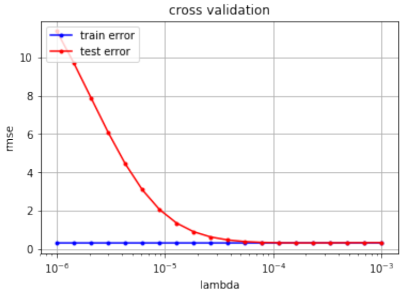

Figure 3: MSE - Lambda plot for third degree linear model **The best lambda value: 0.0001, test error: 0.306 and train error (for best test error): 0.304**

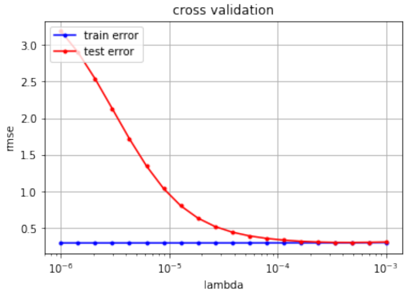

Figure 4: MSE - Lambda plot for fourth degree linear model **The best lambda value: 0.0004, test error: 0.303 and train error (for best test error): 0.301**

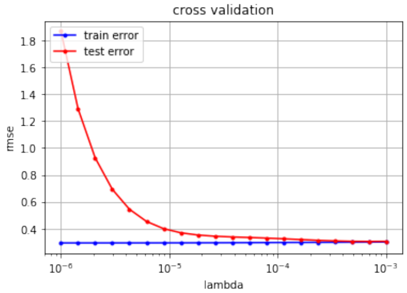

Figure 5: MSE - Lambda plot for fifth degree linear model **The best lambda value: 0.0006, test error: 0.305 and train error (for best test error): 0.301**

As can be seen in the figures and results, while train error is continuously decreasing because of over-fitting, there is no significant change in test error after third degree polynomial basis. Therefore, feature extraction using polynomial basis does not seem working for higher degrees. Furthermore, the accuracy of the model in Kaggle is not as good as what we expected (0.79 - 0.8), so that we decided to switch logistic regression, which will be discussed in the next chapter, to enhance accuracy.

III. Results

Shows that accuracy is good enough meaning that model selection was good

IV. Discussion

State that we can improve the accuracy by using non-linear classifiers?

V. Summary

We have shown that it is possible to detect the Higgs boson using linear methods and feature augmentation.