## CS6140 Assignment1 Guanglei Wu

1.

1.1

(a)

\*\*\*Test for data set iris\*\*\* Accuracies

Eta: 0.05 Mean: 0.9533 SD: 0.0427

Eta: 0.10 Mean: 0.9467 SD: 0.0499

Eta: 0.15 Mean: 0.9467 SD: 0.0499

Eta: 0.20 Mean: 0.9467 SD: 0.0499

(b)

\*\*\*Test for data set spam\*\*\* Accuracies

Eta: 0.05 Mean: 0.9117 SD: 0.0171

Eta: 0.10 Mean: 0.8954 SD: 0.0147

Eta: 0.15 Mean: 0.8622 SD: 0.0183

Eta: 0.20 Mean: 0.8591 SD: 0.0115

Eta: 0.25 Mean: 0.8267 SD: 0.0253

1.2

(a)

Iris Confusion Matrix: Eta: 0.05

Actual

setosa versicolor virginica

setosa 50 0 0

Predict versicolor 0 46 3

virginica 0 4 47

The best prediction of iris class has about 95% accuracy. It does good job in categorizing classes of iris. For particular classes, setosa is perfectly distinguished from other 2 classes (50/50). It may because setosa has some very special characteristic comparing to the others. In contrast, 46/50 versicolors are classified correctly, 4 are misclassified as virginica. Whereas 47/50 virginicas are classified correctly, 3 are misclassified as versicolor. They are not as good as distinguishing setosa, but still good predictions. Misclassify may because the two classes can not be fully distinguished by the given attributes, we may add more attributes to improve the outcome. Or the data set is not big enough, makes some bias.

(b)

Spambase Confusion Matrix: Eta: 0.05

Actual

non-spam spam

Predict non-spam 2599 218

spam 189 1595

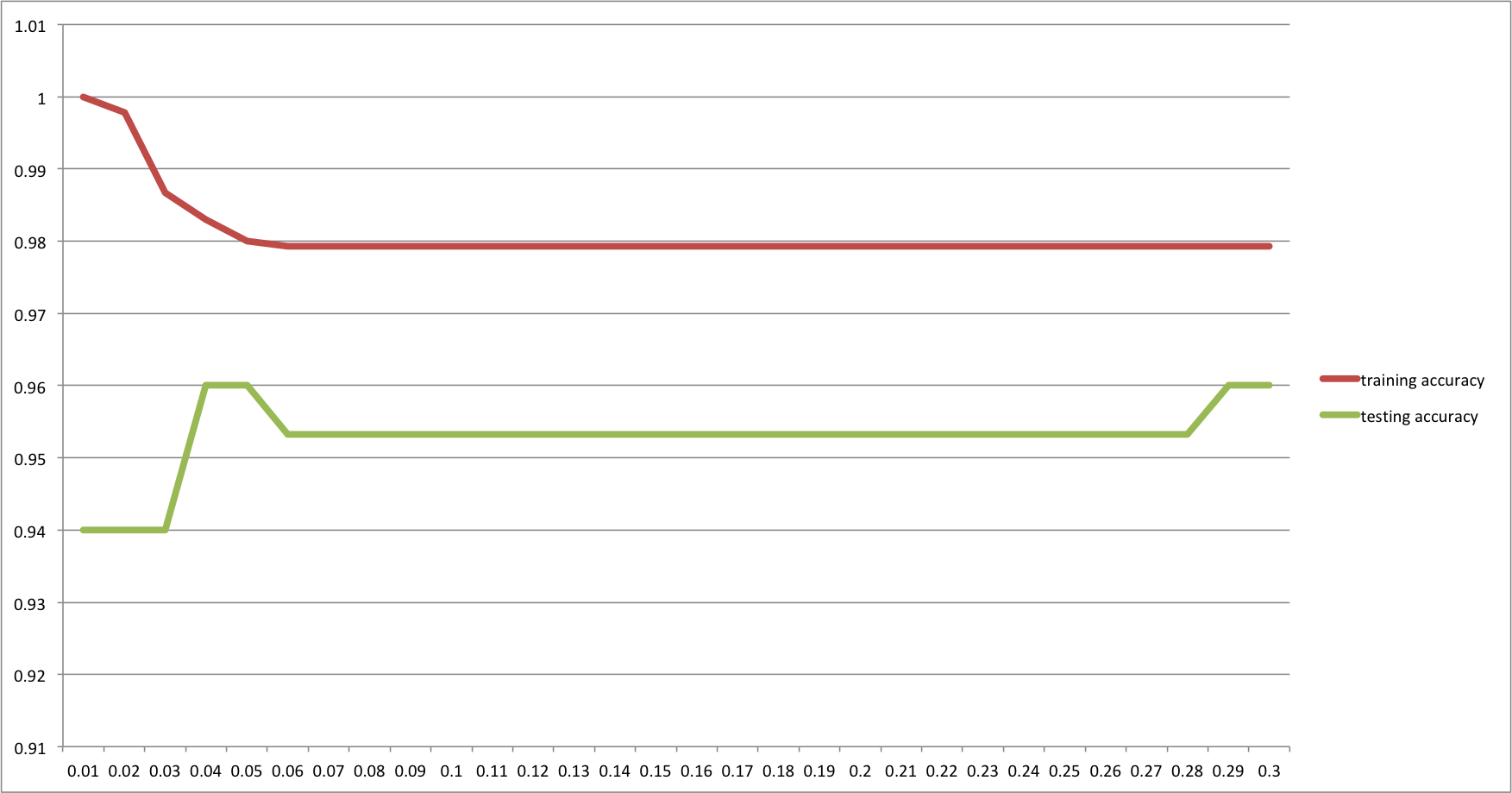
The best prediction of spam class has about 91% accuracy. About 7% non-spams are misclassified as spams. While about 12% spams are misclassified as non-spams. The system is good in distinguishing each class. However for the spam class, misclassifying spam as non-spam is less severe than misclassifying non-spam as spam. So in addition to improve general accuracy, we may want to improve the accuracy of classifying non-spam. We can achieve that by implementing cost matrix or weights for classes.

(c)

For Iris class, different values of EtaMin do not change much of the accuracies. As the results in (1.1a), for all the values of EtaMin, we almost always got 95% accuracies. This may because the data set is small, and the classes are easily distinguishable. In another word, the data is well split before the min size of the node comes to intervene.

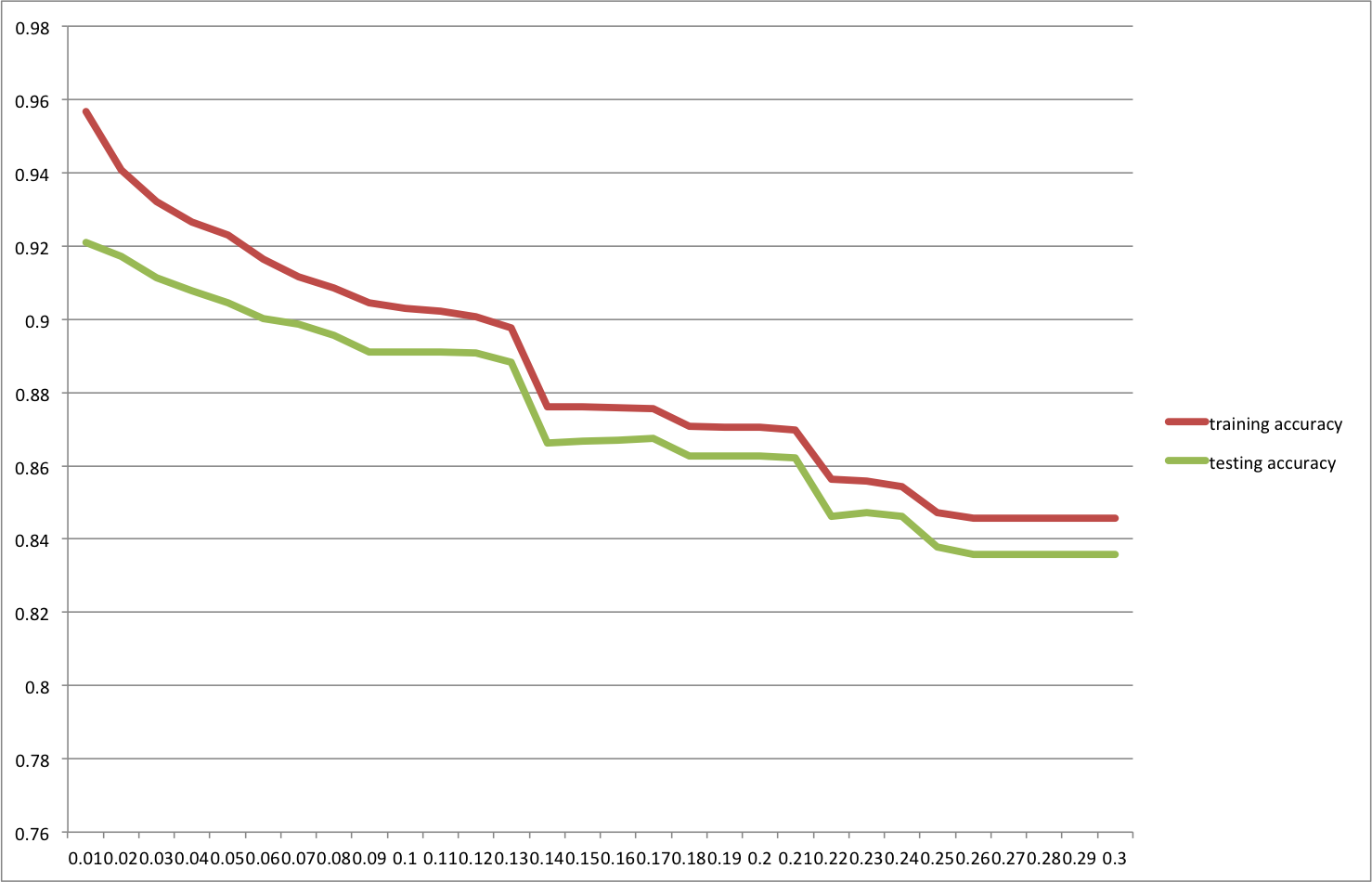
For Spambase class, different values of EtaMin do influence the accuracies. As the results in (1.1b), along with increase value of EtaMin, the means of accuracies decrease from 91% to 83%. This shows when the data set is large enough, stoping too soon makes each leaf node still contains too many values. This leads to underfitting bias. We can improve the prediction by use smaller EtaMin, or use post prune strategy.

(d) Iris



From the chart, x-axis is the EtaMin values, from 0.01 to 0.3. Red line is accuracy for training data. Green Line is accuracy for testing data. We can see that it shows overfitting when EtaMin is small, the model fits training data better, but fits testing data poorer. This because choosing too small a EtaMin makes the model grab noises of the data.

Spam



The spam data shows clear under fitting characteristic. As EtaMin increases, the accuracy of prediction drops. Because the tree stop growing too soon, leaf nodes still carries too much information that can be split to use.

2.

2.1

(a)

\*\*\*Test for multiway attributes\*\*\* Accuracies

Eta: 0.05 Mean: 0.9946 SD: 0.0021

Eta: 0.10 Mean: 0.9906 SD: 0.0031

Eta: 0.15 Mean: 0.9887 SD: 0.0032

(b)

\*\*\*Test for binary attributes\*\*\* Accuracies

Eta: 0.05 Mean: 0.9977 SD: 0.0026

Eta: 0.10 Mean: 0.9977 SD: 0.0026

Eta: 0.15 Mean: 0.9661 SD: 0.0061

2.2

(a)

Mushroom Multiway Confusion Matrix: Eta: 0.05

Actual

edible poison

Predict edible 4208 48

poison 0 3868

The model did very good job in classifying mushrooms. Overall 99.46% mushroom is classified correctly. 100% of the edible mushrooms are classified correctly. While 1.23% poison mushrooms are misclassified as edible. But misclassify poison mushrooms as edible is fatal. We may want to improve the accuracy of distinguishing poison mushroom. We can achieve that by implementing cost matrix or weights for classes.

Mushroom Binary Confusion Matrix: Eta: 0.05

Actual

edible poison

Predict edible 4192 7

poison 16 3909

The binary model seems did a similarly good job as the multiway model. Overall 99.77% mushrooms are classified correctly. 0.4% edible mushrooms are misclassified as poison. While 0.2% poison mushrooms are misclassified as edible. This model did a better job in classifying poison mushroom, which is more fatal.

(b)

It seems the optimal value of EtaMin for multiway and binary models are nearly the same (0.05 for each case, as in 2.1). In this case, it tells that multiway and binary models are very similar when selecting an appropriate EtaMin value. The difference between binary and multiway models are that some leaf nodes in binary model are "not something", whereas all the leaf nodes in multiway model are "something". It does not influence a lot if the tree size is well grown.

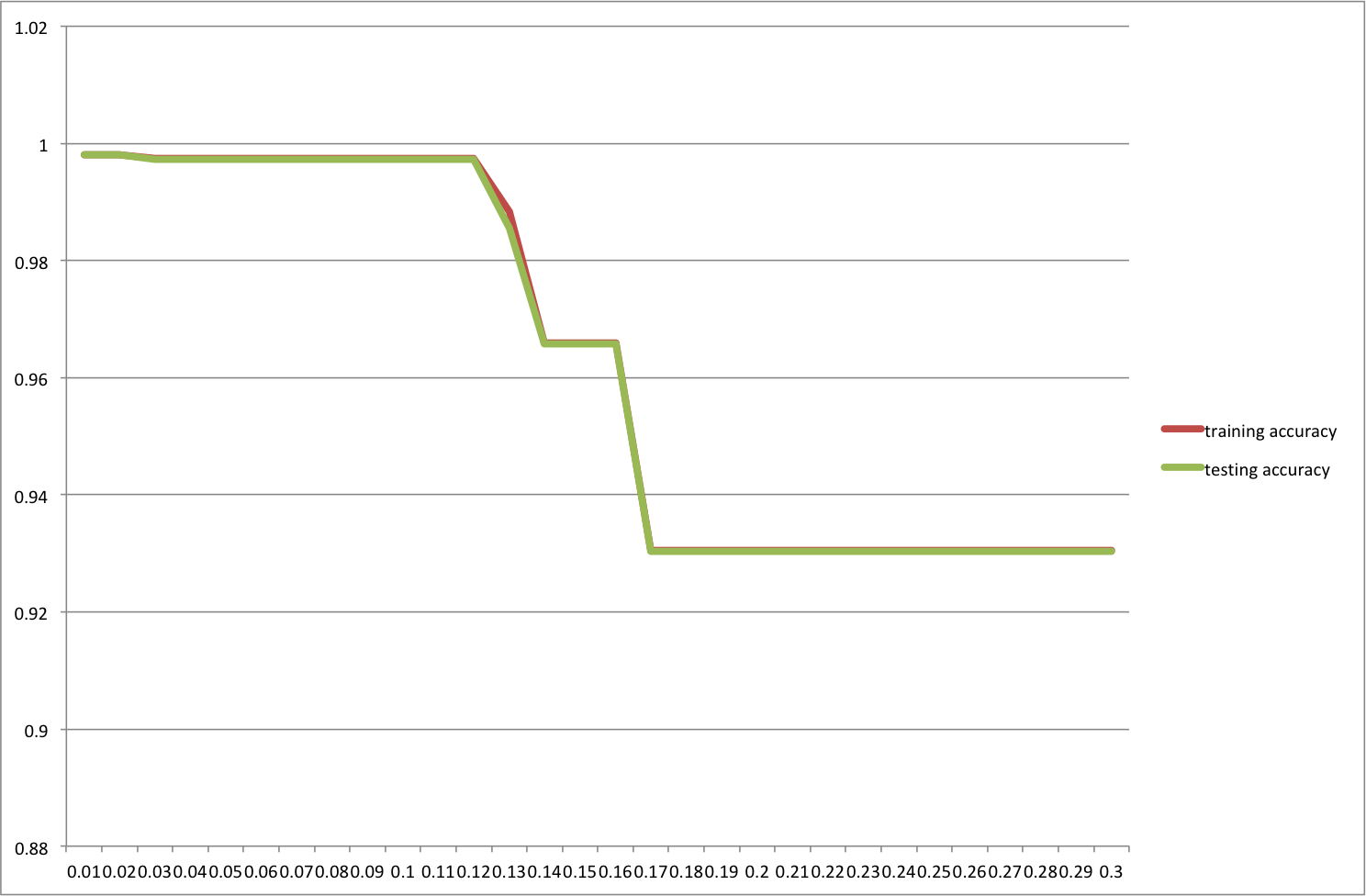
(c)

Mushroom multiway



The graph shows a great fits and prediction of the model. Increasing EtaMin only influence the accuracy a little bit. And not much differences between predictions of training and testing data. The data may be easy to classify before the tree grows too large.

Mushroom Binary



The binary model shows under fitting when EtaMin becomes large. Accuracies drop faster than multiway model. This is because the tree stops growing too soon, makes the model less comply with the training data, and makes less accurate predictions.

3.

(a)

The decrease in entropy is just the Mutual Information of q and b, where q represents the node, b represents the binary feature,

I(q ; b) = I(b ; q) = H(b) - H(b|q)

Since b is binary, H(b) <= - 2 \* 1/2 \* log2 (1/2) = 1, H(b|q) >= 0, H(b) - H(b|q) <= 1.

(b)

Suppose the feature has m values.

I(q; m) = H(m) - H(m|q)

H(m) <= - m \* 1/m \* log2 (1/m) = log2(m), also H(m|q) >= 0,

so I(q; m) <= log2(m)

4.

Gain(q, V) = iota(q) -

Since iota(q) is a determined value for a specific given q, choosing the value of Gain(q, V) is purely choosing the value of , which represents the weighted average of the descendent nodes iota. So maximizing Gain(q, V) is equivalent to minimizing . And since for a given split, |V|, Ni and Nq are determined values. So maximizing Gain(.,.) is equivalent to minimizing iota(.) over |V| children.

5.

Gini(q) =

For M > 2, (1-pqk) =

Gini(q) = =

=

6.

(a)

\*\*\*Test for regression housing\*\*\* SSEs

Eta: 0.05 Mean: 877.63 SD: 516.07

Eta: 0.10 Mean: 973.64 SD: 431.57

Eta: 0.15 Mean: 1236.96 SD: 424.37

Eta: 0.20 Mean: 1350.67 SD: 383.69

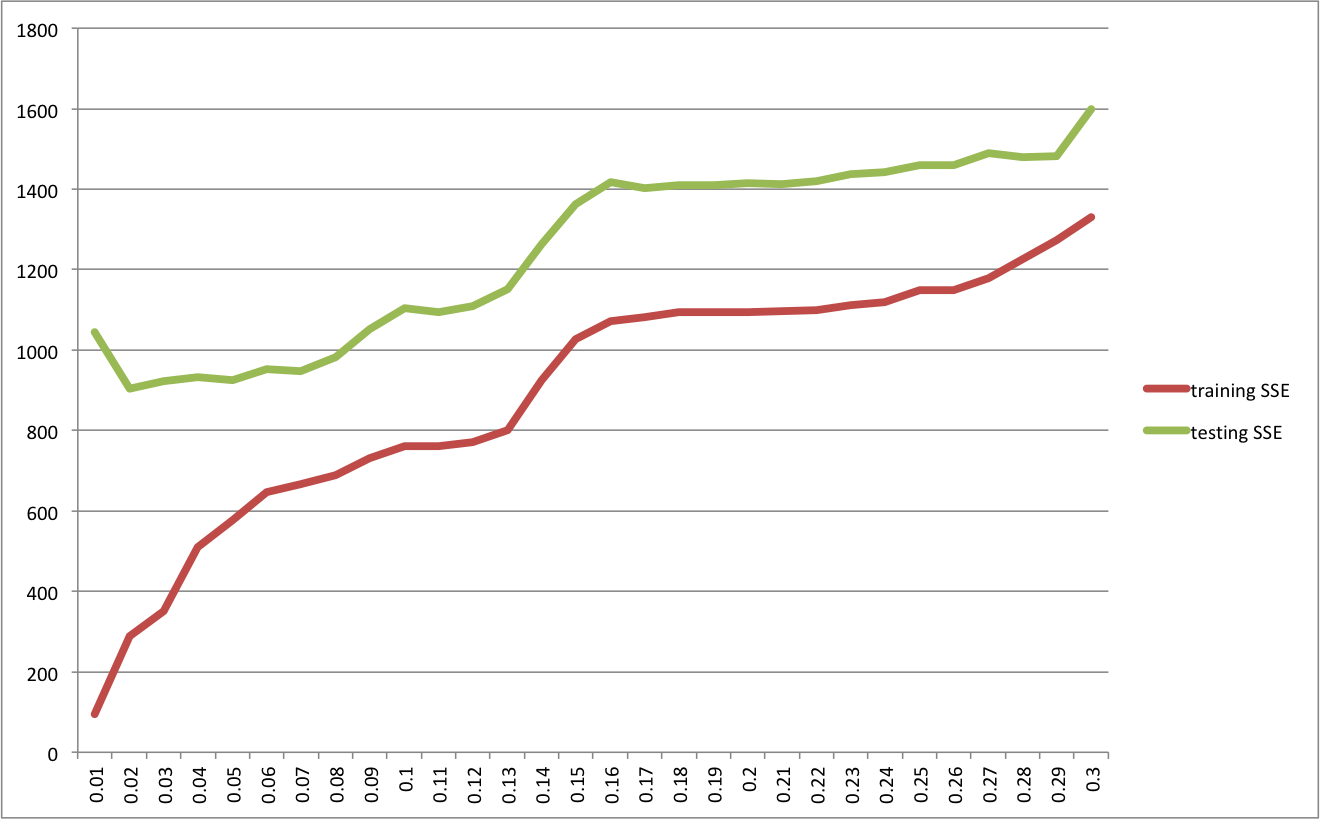
(b)

According to the output, different choices of Eta influences the results significantly. This may because the data set is diverged very much. That makes the each split of the tree node adds more information to the model. Early stop strategy may make the model less accurate.

(c) (same as b)

(d)

Housing



In order to make the SSE comparable for the sizes of training and testing data are not the same, the training SSE is divided by 9 as its size is 9 times of the testing data. The graph shows both under fitting and over fitting. For over fitting, it is caused by too small EtaMin, makes the model fits better for training data, but worse on testing data. For under fitting, it is caused by too large EtaMin, the model is not fully developed, fits worse on training and testing data.