CS412 HW4

Wangfei Wang

1 Trees/Ensemble Methods

a) Experiment with the value max leaf nodes which is the maximum number of nodes that the tree will build, greedy first. Plot a graph of cross validation errors where the x-axis is values of max leaf nodes in {5,10,15,20,30,40,50,75,100,200,500,1000,10000}. Use a logarithmic scale. Label this Figure 4.1.

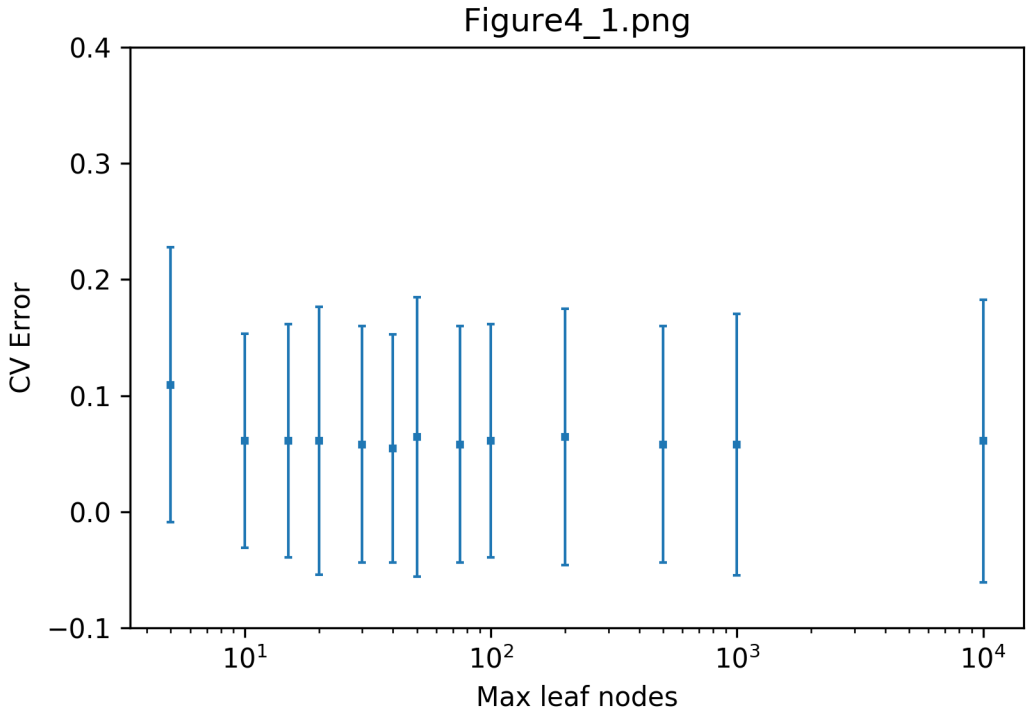

Figure 4.1. CV error for max leaf nodes.

b) Was there any evidence of over or underfitting? Which values of max leaf nodes exhibited this and explain why you think that is the case.

The errors seem to be similar for any max leaf nodes value. There is no increase in cross validation error after a certain point, so there is no over or underfitting.

c) Select the tree from part a) that has the lowest cross validation error. Plot the 2D decision region for this optimal model. Label this Figure 4.2.

The lowest CV error (upper boundary for 95 %) is achieved at max_leaf_nodes = 10. Therefore, max_leaf_nodes = 10 was used for decision boundary plot.

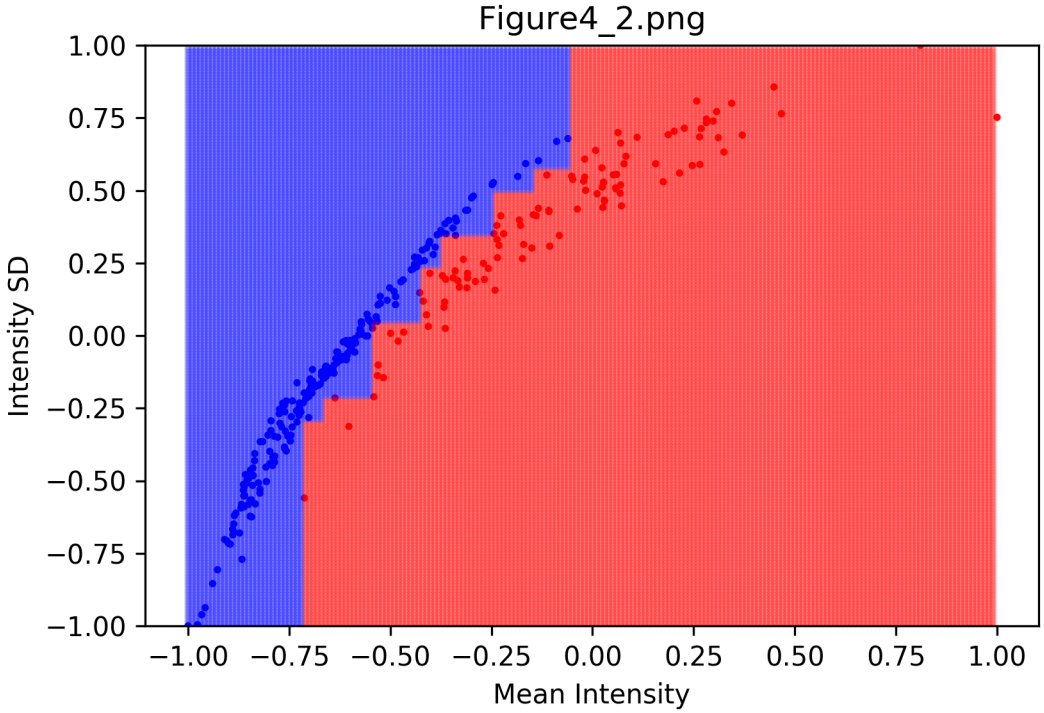

Figure 4.2. Decision boundary for max_leaf_nodes = 10.

d) Graduate Student Question: min impurity decrease and min impurity split are pre-pruning techniques. Let max leaf nodes go to default and find the optimal values of min impurity decrease and min impurity split in terms of cross-validation error. You may give your results in a graph or a table.

According to the documentation, min_impurity_split has been deprecated. Therefore, only min_impurity_decrease was investigated in this question. A set of different values of variable min_impurity_decrease in $\{0, e^{-4}, e^{-3}, e^{-2}, 0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ were considered. The CV errors are shown in Figure 4.d. When min_impurity_decrease was chosen to be bigger than and equal to 0.3, the CV errors jumped to be very high.

e) Now, experiment with the Random Forest Classifier. For this, you will produce 3 graphs each corresponding to a max leaf nodes value of 10,100,1000. For each of these three, plot the cv error $\sim$ n_estimators $\{5, 10, 15, 20, 30, 40, 50, 75, 100, 200, 500, 1000, 10000\}$.
Label these Figure 4.3, 4.4 and 4.5.

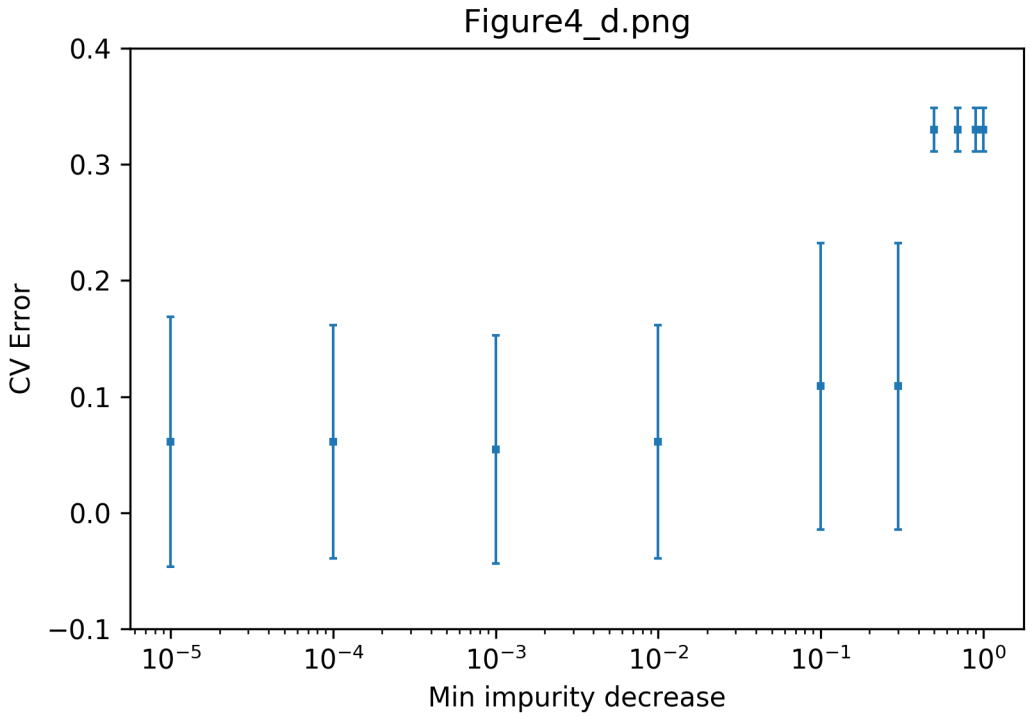

Figure 4.d. CV errors for min_impurity_decrease.

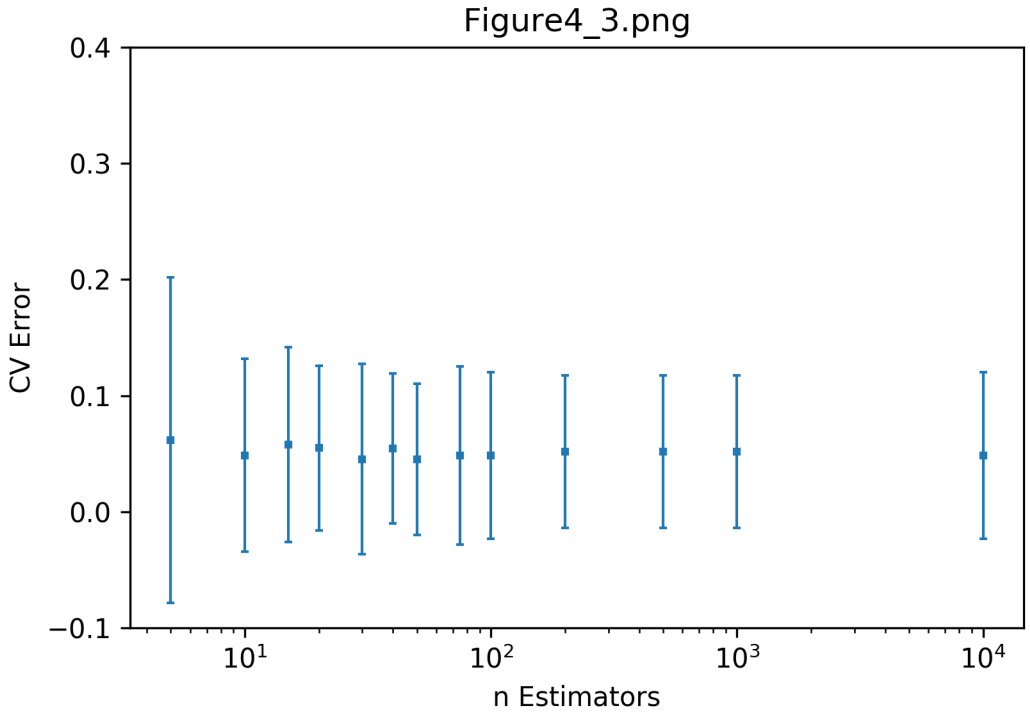

Figure 4.3. CV errors for different values of n_Estimators when max_leaf_nodes = 10.

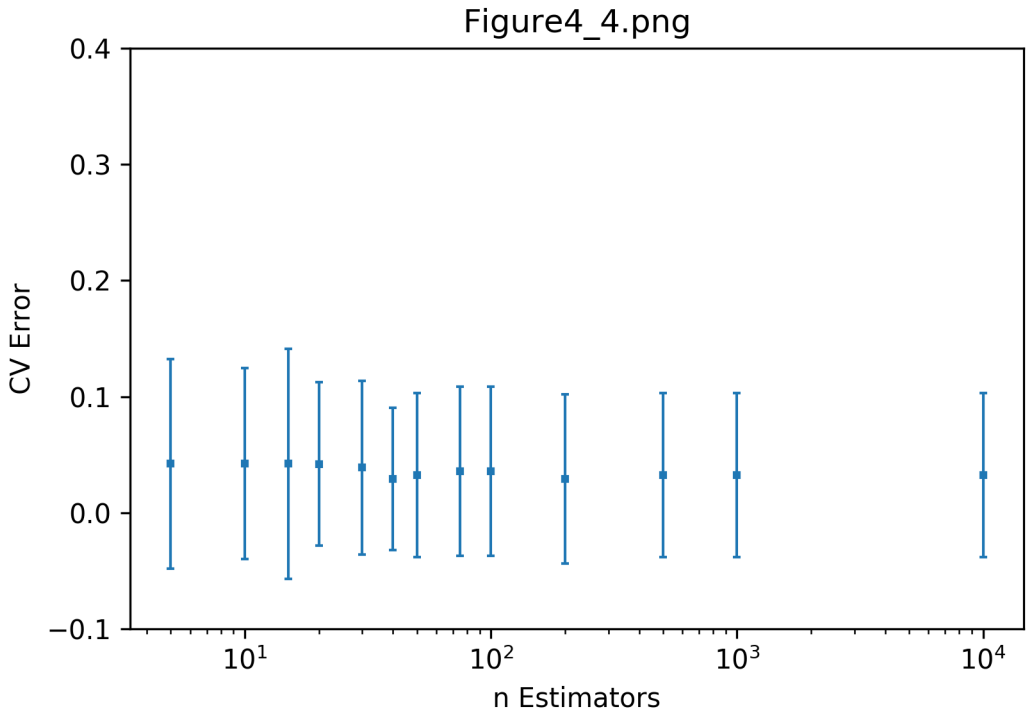

Figure 4.4. CV errors for different values of n_Estimators when max_leaf_nodes = 100.

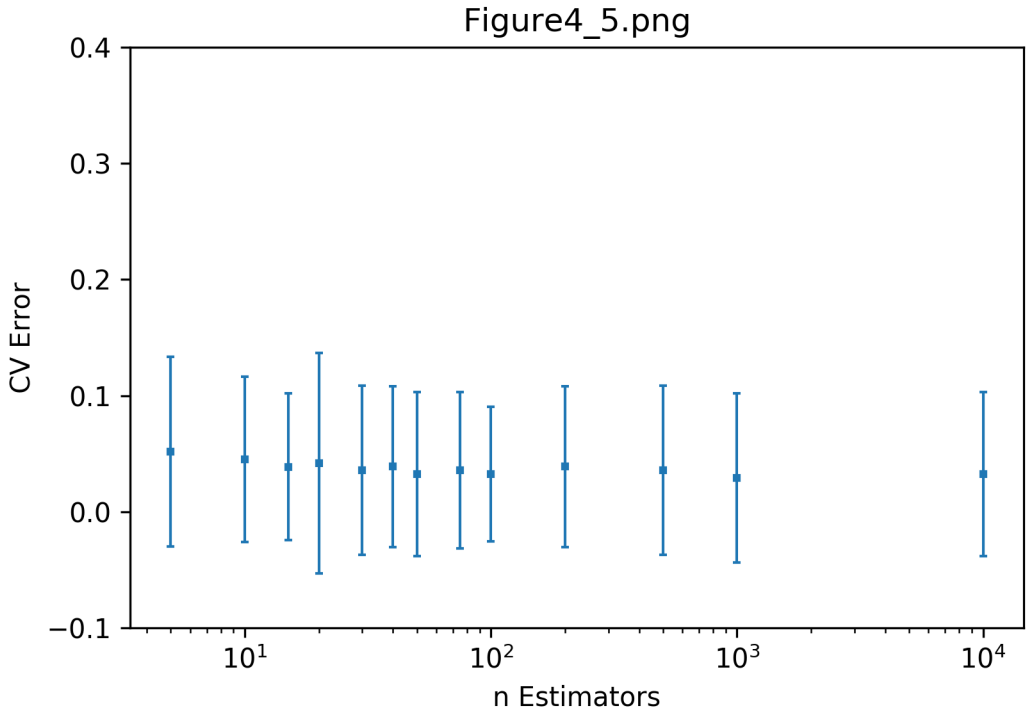

Figure 4.5. CV errors for different values of n_Estimators when max_leaf_nodes = 100.

f) Which values of max leaf nodes was most impacted by the bagging approach? Explain why you think this is the case. If there is no significant difference, explain that.

All three cases seem to be impacted by the bagging approach similarly, but max_leaf_nodes = 10 seems to be most impacted by the bagging approach. The variance of CV errors is the biggest among the three different max_leaf_nodes values. The variable defines the maximum number of possible leaf nodes. When the leaf numbers are small, the variances may be higher than the other cases and therefore got the most influenced by the bagging approach.

g) To what extent does n estimators impact the result? Was there a point at which more estimators did not make an impact? Identify this point.

This is the number of trees you want to build before taking the maximum voting or averages of predictions. n_estimators initially makes CV errors smaller as it go up, but as n_estimators exceed a certain threshold, the CV error gets stabilized and does not change very much.

For max_leaf_nodes = 10 case, after n_Estimators exceeds 75, the CV errors don't change much.

For max_leaf_nodes = 100 case, after n_Estimators exceeds 40, the CV errors don't change much.

For max_leaf_nodes = 1000 case, after n_Estimators exceeds 200, the CV errors don't change much.

Overall, all CV errors don't vary much with increasing n_Estimators.

h) Plot the decision region for the Random Forest classifier from above which has lowest cross-validation error. Label this Figure 4.6.

When max_leaf_nodes = 100 and n_Estimators = 40 minimizes the CV error. Decision boundary is shown in Figure 4.6.

i) Graduate Student Question: Experiment with different float values for max features. Create a table of cross-validation errors for max leaf nodes in {10,100,1000}, n estimators in {1,10,100,1000} and max features in {0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0}.

See Table 1.

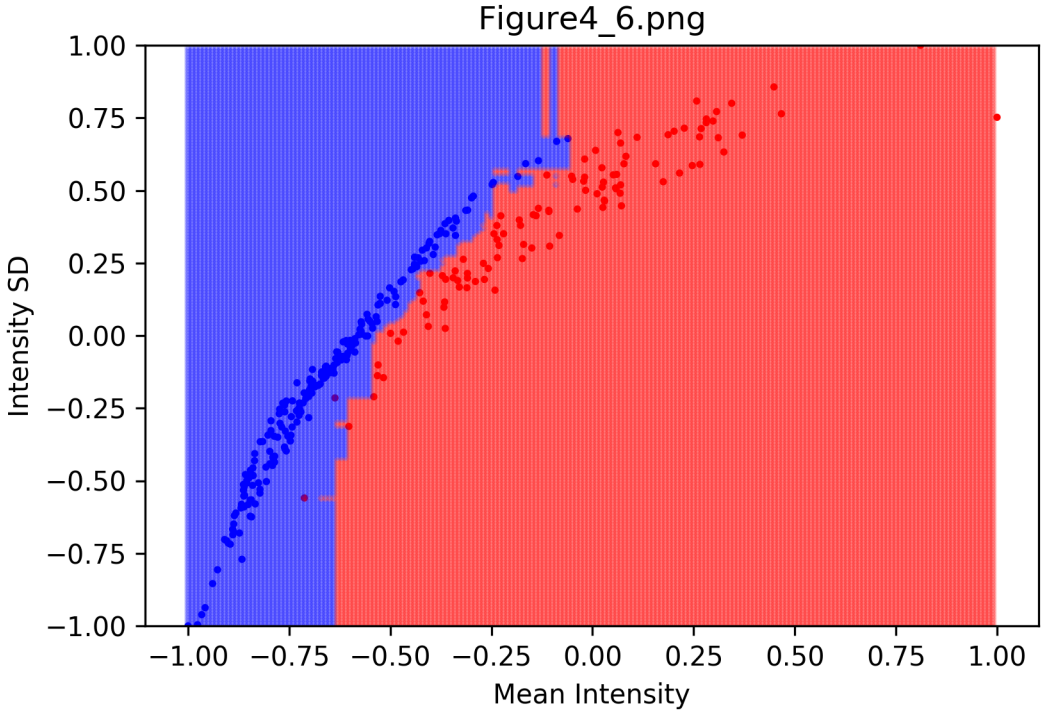

Figure 4.6. Decision boundary for optimal random forest model.

max_features	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
max leaf nodes = 10 n estimators = 1	0.0936	0.0964	0.0996	0.0836	0.0868	0.0737	0.0803	0.0967	0.0962	0.0771
max leaf nodes = 10 n estimators = 10	0.0486	0.0580	0.0675	0.0615	0.0708	0.0709	0.0643	0.0515	0.0644	0.0388
max leaf nodes = 10 n estimators = 100	0.0452	0.0453	0.0549	0.0516	0.0420	0.0516	0.0515	0.0549	0.0516	0.0325
max leaf nodes = 10 n estimators = 1000	0.0453	0.0516	0.0485	0.0516	0.0516	0.0516	0.0453	0.0516	0.0484	0.0325
max leaf nodes = 100 n estimators = 1	0.0708	0.0929	0.0711	0.0644	0.0775	0.0809	0.0614	0.0646	0.0739	0.0575
max leaf nodes = 100 n estimators = 10	0.0355	0.0548	0.0293	0.0357	0.0293	0.0452	0.0390	0.0421	0.0487	0.0325
max leaf nodes = 100 n estimators = 100	0.0388	0.0325	0.0357	0.0294	0.0261	0.0356	0.0293	0.0325	0.0325	0.0356
max leaf nodes = 100 n estimators = 1000	0.0325	0.0325	0.0325	0.0325	0.0325	0.0324	0.0325	0.0325	0.0325	0.0388
max leaf nodes = 1000 n estimators = 1	0.0645	0.0484	0.0804	0.0704	0.0680	0.0677	0.0707	0.0708	0.1097	0.0420
max leaf nodes = 1000 n estimators = 10	0.0484	0.0420	0.0357	0.0357	0.0326	0.0389	0.0261	0.0454	0.0421	0.0421
max leaf nodes = 1000 n estimators = 100	0.0326	0.0357	0.0325	0.0325	0.0356	0.0356	0.0325	0.0325	0.0357	0.0325
max leaf nodes = 1000 n estimators = 1000	0.0357	0.0325	0.0293	0.0325	0.0325	0.0325	0.0325	0.0325	0.0357	0.0324

Table 1: CV errors for different combinations of max_features, max leaf nodes and n estimators.

j) Now you will experiment with the AdaBoost Classifier. For base estimator = DecisionTreeClassifier(max depth=1), plot a graph of 10-fold cross-validation errors where the x-axis is n estimators in {1,5,10,100,1000,10000}. Label this figure 4.7.

See Figure 4.7.

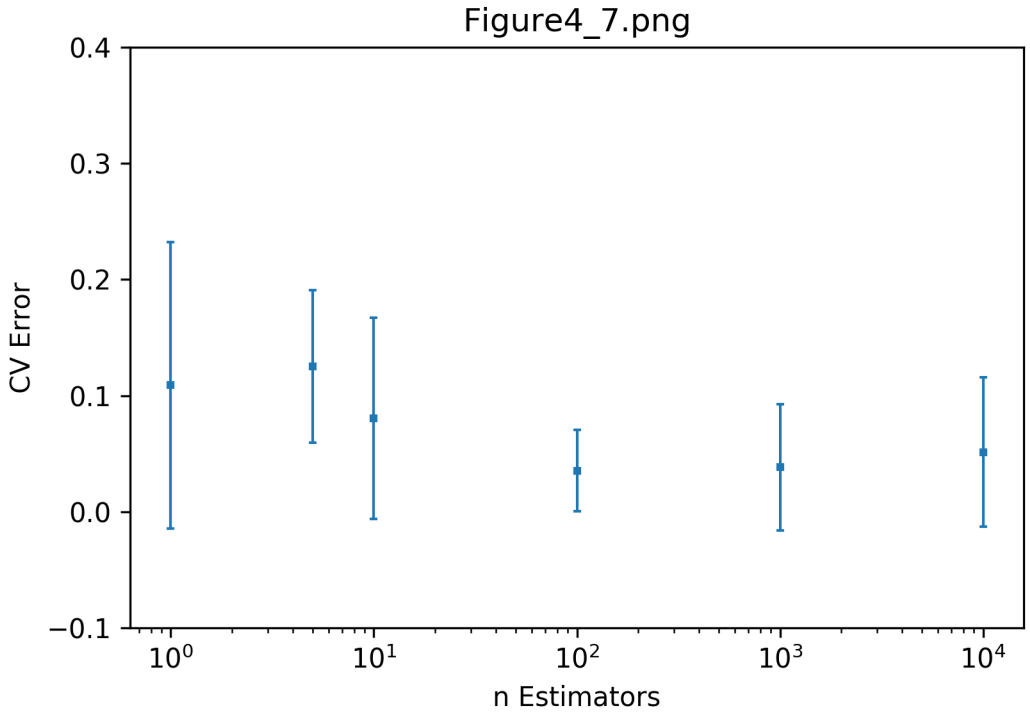

Figure 4.7. CV errors for max depth = 1 in relation to n estimators.

k) Repeat this for base estimator = DecisionTreeClassifier(max depth=10), Label this figure 4.8.

See Figure 4.8.

l) Repeat this for base estimator = DecisionTreeClassifier(max depth=1000), Label this figure 4.9.

See Figure 4.9.

m) Plot the decision boundary of the AdaBoost classifier which has the lowest cross-validation error. Label this figure 4.10.

The lowest CV error is reached when max depth = 1 and n estimators = 100. See Figure 4.10 for the decision boundary plot .

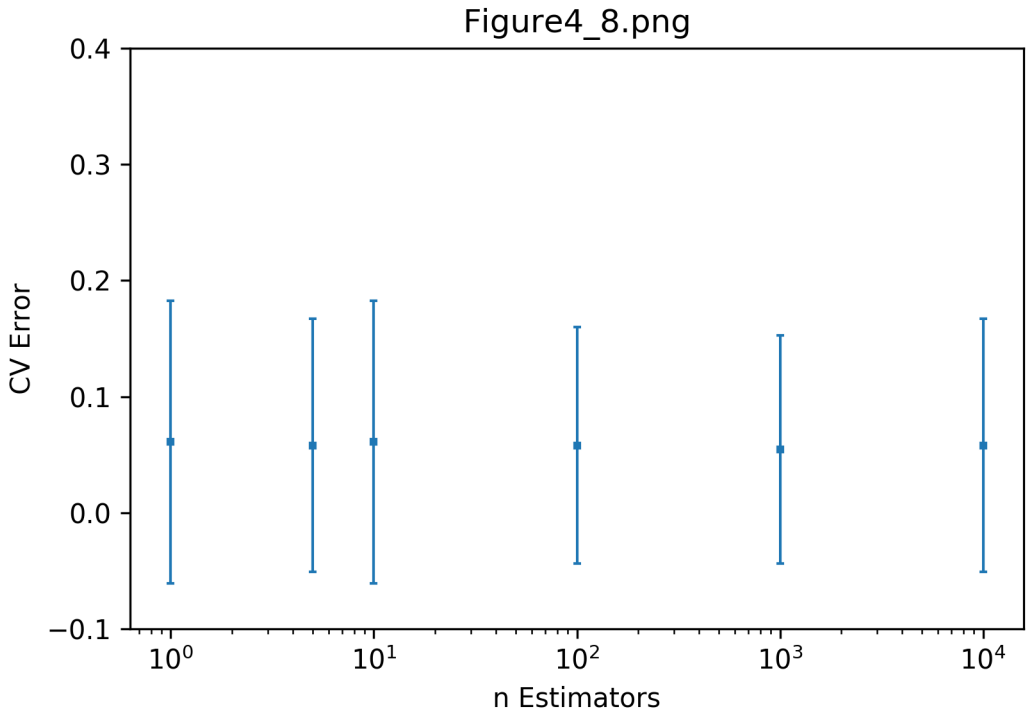

Figure 4.8. CV errors for max depth = 10 in relation to n estimators.

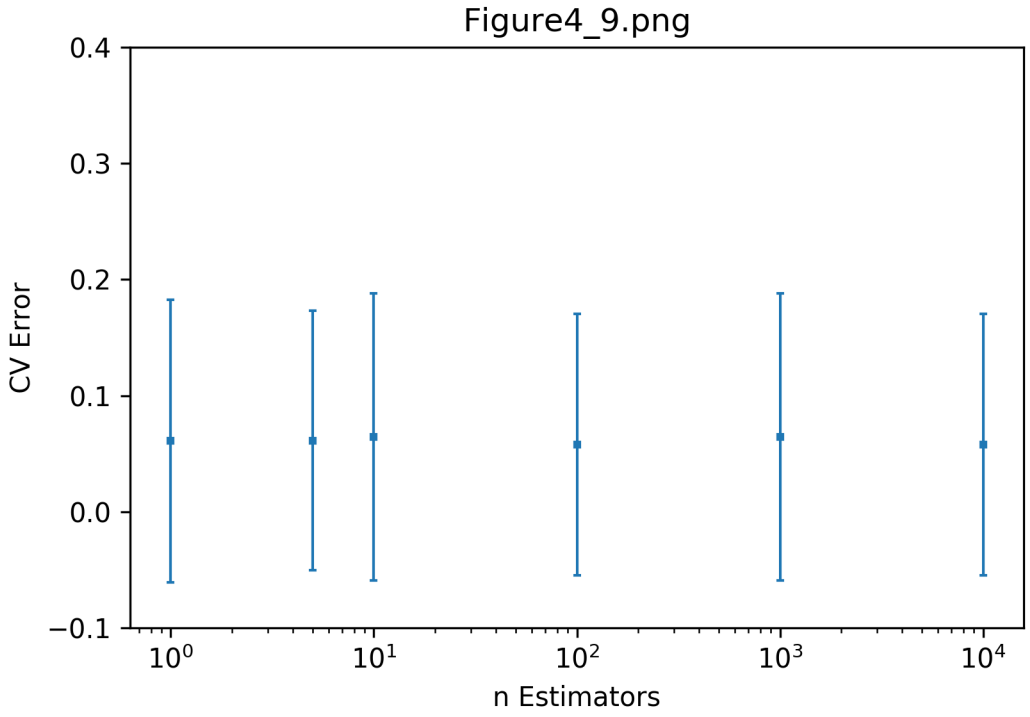

Figure 4.9. CV errors for max depth = 1000 in relation to n estimators.

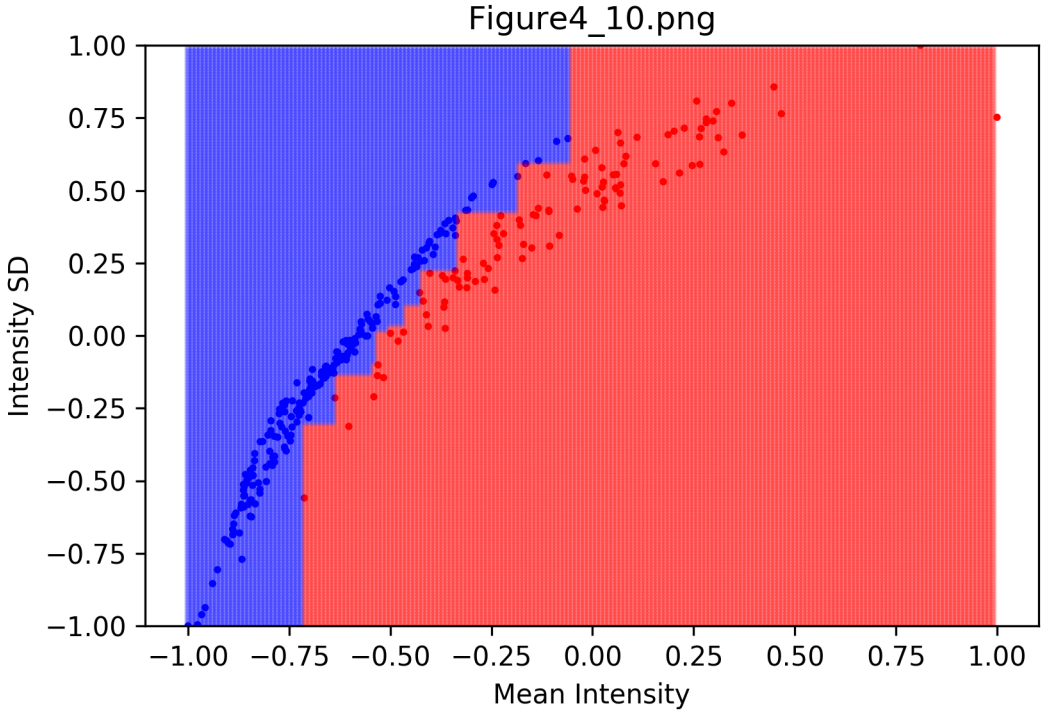

Figure 4.10. Decision boundary for optimal AdaBoost model.

2 Reporting Final Error

Parameters used in different models are shown in Table 2 below.

Model	Optimal parameter
Polynomial SVM	degree = 2, C = 38.8816
Neural Network	hidden_layer = 2, number_of_nodes_per_layer = 50
Random Forest	max_leaf_nodes=100, n_estimators=40
AdaBoost	max_depth=1, n_estimators=100

Table 2: Optimal models from previous HW

Train each of the model with optimal parameters shown in Table 2 with training set and then fit the test set with the model.

- Markov's inequality:

$$P(X \geq a) \leq \frac{E(X)}{a}$$

  Using confidence interval to calculate the a and then obtain the concentration bound upper bound.
  For example, if the confidence interval is 75%, then set $\frac{E(X)}{a}$ as 0.25 and solve a. Then the upper bound for E_out would be:

$$\frac{E(X)}{0.25}$$

  .

- Chebyshev inequality:

$$P\{|X - E(X)| \geq \epsilon\} \leq \frac{Var(X)}{\epsilon^2}$$

  Treat the error as the average of n independent Bernoulli trials. Var(X) = E_out $\times$ (1-E_out)/n. Using the confidence interval requirement to calculate the ϵ and then calculate the upper concentration bound.

- Hoeffding bound:

$$\epsilon \leq \sqrt{\frac{1}{2n} log(\frac{2}{\delta})}$$

  where n is the length of test set and δ is the 1-α upper bound of confidence interval. The upper concentration bound would be E_out + ϵ.

a) Which of the following models changed most dramatically with respect to confidence interval?

Optimal model	E_out	Upper concentration bounds								
		Markov			Chebyshev			Hoeffding		
		75%	95%	99%	75%	95%	99%	75%	95%	99%
Polynomial SVM	0.01361	0.05444	0.2722	1.3611	0.02017	0.02827	0.04639	0.04246	0.05204	0.05966
Neural Network	0.00961	0.0384	0.1922	0.9608	0.01513	0.02195	0.03721	0.03846	0.04804	0.05566
Random Forest	0.0232	0.09287	0.4644	2.3219	0.03174	0.04228	0.06583	0.05204	0.06162	0.06924
AdaBoost	0.03283	0.1313	0.6565	3.2826	0.0429	0.05537	0.08324	0.06168	0.0713	0.07888

Table 3: Error boundary for different models at different confidence intervals.

Markov. Because it is too conservative, and is useless when $a < E(X)$.

b) Given this data, which model would you select if presented with this problem by a client? Defend your answer.

I would choose neural network because it has smallest test error and smallest test error upper concentration bound. Also the training error of neural network is small too.

c) Are there any considerations other than final reported error that went into your decision? Why or why not?

We need to take stability of the models into account when deciding which model to choose. For instance, the initial value neural network picked randomly would influence the model's test error.

3 Feedback

a) What were some of the biggest challenges you faced in the first four assignments?
Writing our own NN code.

b) Which assignments/parts contributed most to your understanding of machine learning?
Writing our own NN code.

c) Which portions of the assignments took the most time?
Writing our own NN code.

d) Is there any advice or resources that you found that you think would have been helpful to get at the start?
Get a good understanding of linear algebra.

e) Leave any other comments here. There will also be another survey after spring break to share comments anonymously.
HW should not be dominated by running python packages. It is not very useful for us to either learn how to write python code or understand the ML topics. Equation derivation or other types of theoretical questions would be sometimes helpful for us to understand the topics.