CSCI4622 Machine Learning 2020 Spring

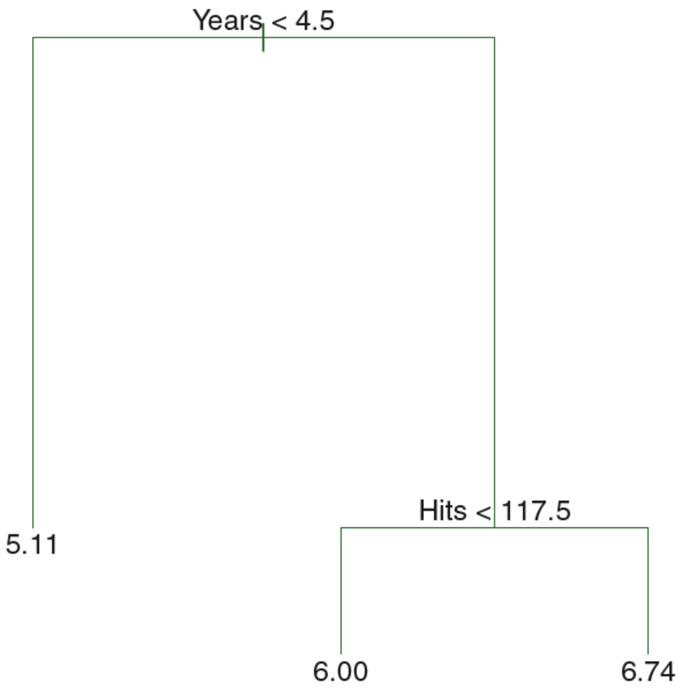
Ensemble method

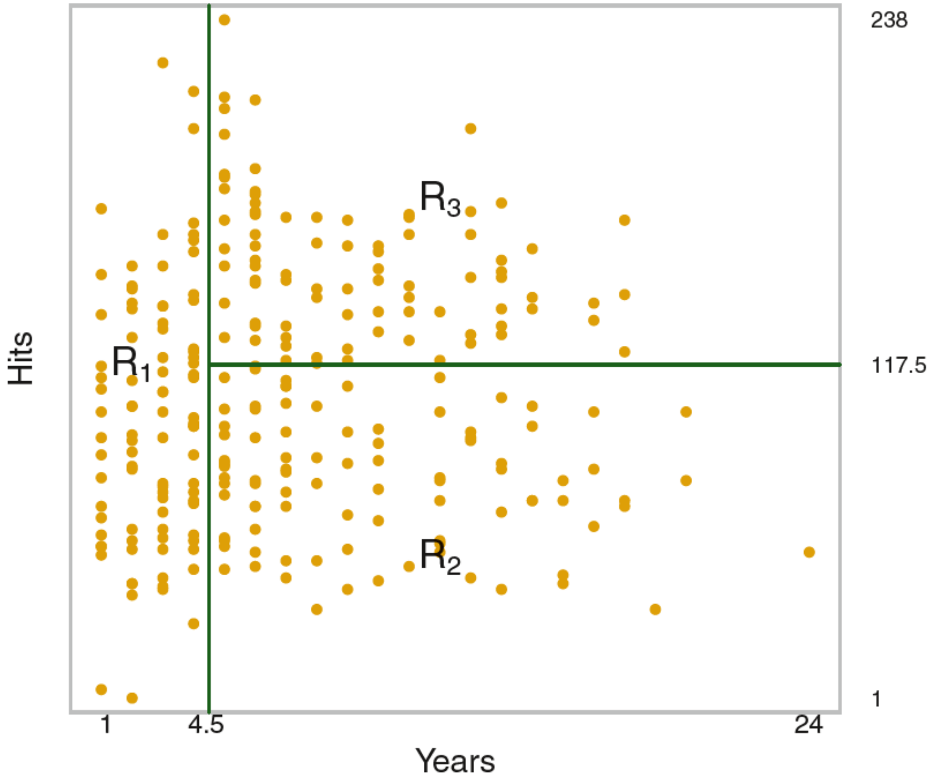
Geena Kim

*Slide contents adopted from ISLR material

Decision Tree

Split Rule: Minimize the metric (MSE, entropy, etc) of the boxes





Decision Tree Split Criteria

Regression Tree

MSE

$$H(X_m) = \frac{1}{N_m} \sum_{i \in N_m} (y_i - \bar{y}_m)^2$$

MAE

$$H(X_m) = \frac{1}{N_m} \sum_{i \in N_m} |y_i - \bar{y}_m|$$

Classification Tree

Gini

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk})$$

Entropy

$$H(X_m) = -\sum_k p_{mk} \log(p_{mk})$$

Information Gain

$$IG = E_{parent} - \frac{N_L}{N} E_L - \frac{N_R}{N} E_R$$

Improving Trees

Problems with a single Decision Tree

- Overfitting

- Trees are weak learner

Hyperparameter search

Grid Search Tip

- Give a range of values for each hyperparameter
- Measure a training time for one, then estimate how long for the loop
- Adjust number of values, range, or hyperparameters to include

max_depth

min_samples_split

min_samples_leaf

max_features

min_impurity_decrease

Decision Tree – Pruning

Minimal Cost-Complexity Pruning

$$R_\alpha(T) = R(T) + \alpha|T|$$

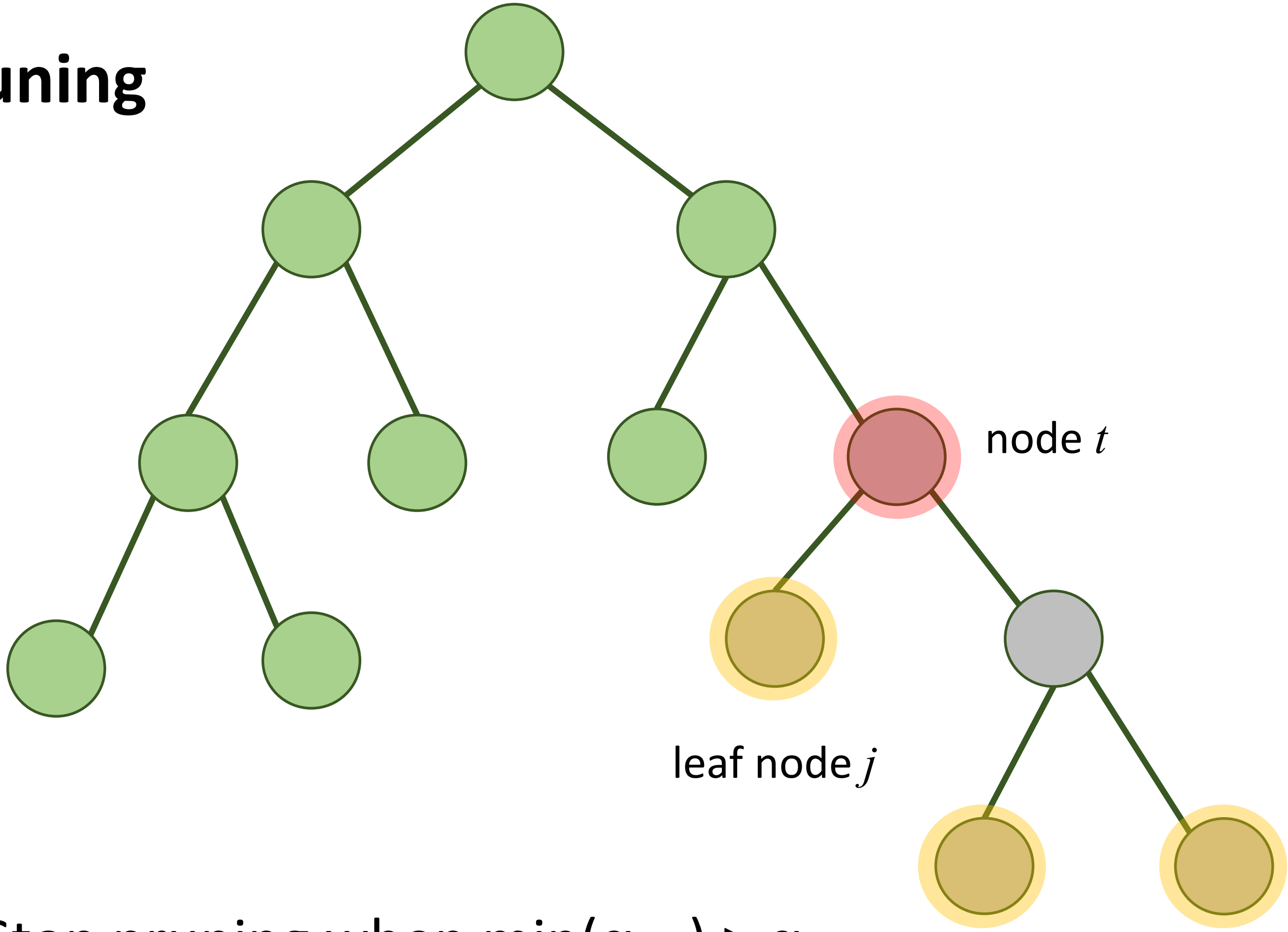
α : complexity parameter
|T|: number of leaf nodes of the subtree

Impurity at the node t

$$R(T_t) < R(t)$$

Sum of the impurities
at the leaf nodes of the subtree T_t

$$\alpha_{eff}(t) = \frac{R(t) - R(T_t)}{|T| - 1}$$



Stop pruning when min(α_{eff}) > α_{ccp}

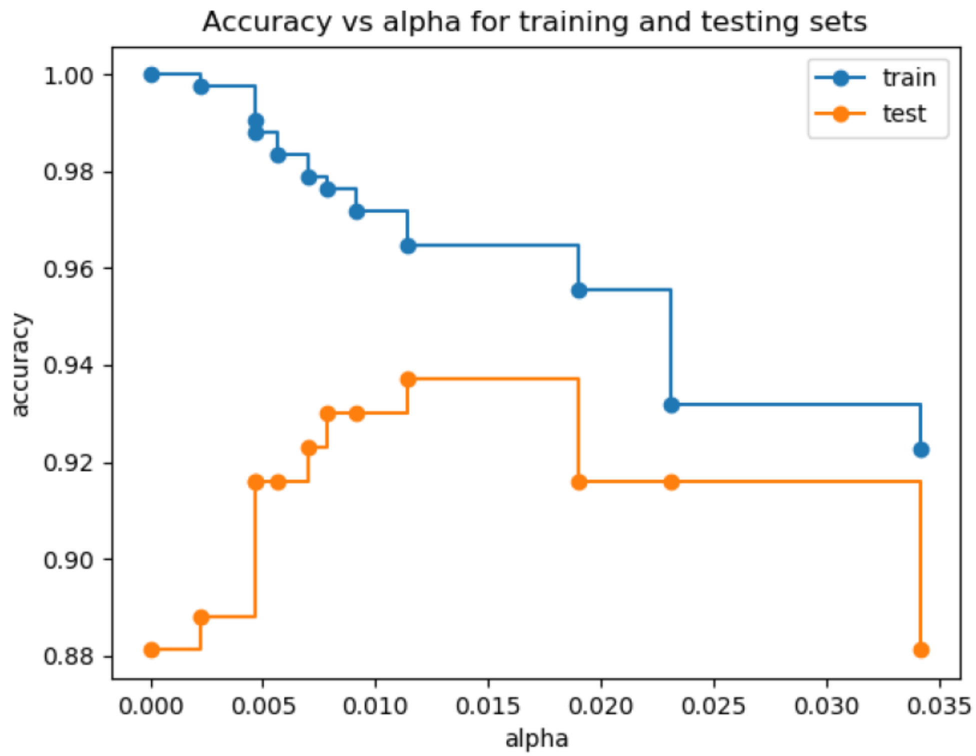
α_{ccp} : cost complexity parameter,
“`ccp_alpha`”

Decision Tree – Pruning

The cost complexity parameter(`ccp_alpha`) is a hyperparameter

How do we determine the right cost complexity parameter?

-> Use validation dataset (or cross-validation)



Ensemble method

What is an Ensemble?

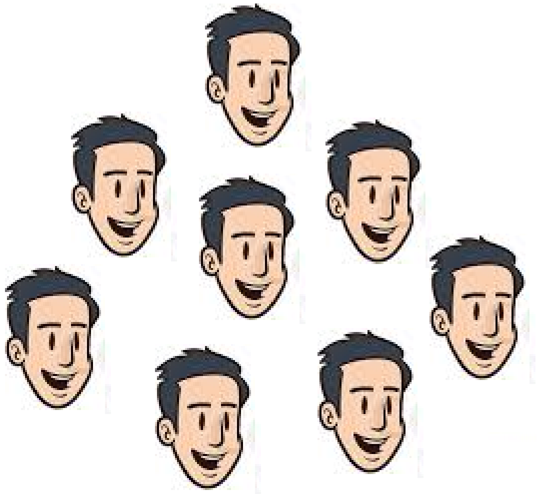
Ensemble method

What is an Ensemble?

- An individual model might be a weak-learner,
- Averaging models can predict better

Ensemble method

Diversity matters



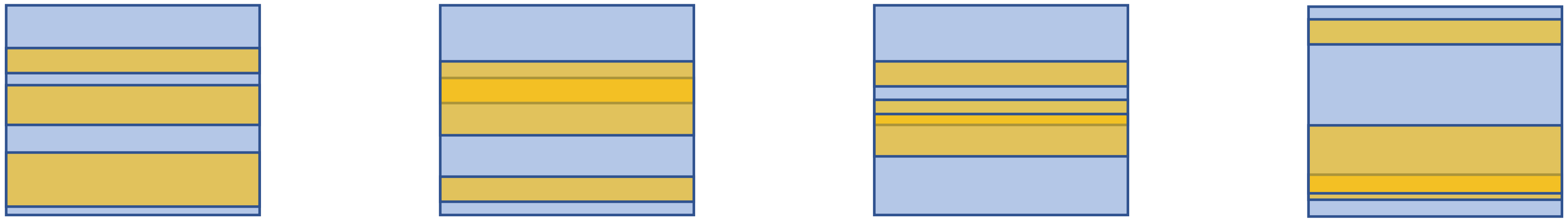
Ensemble method

How do we diversify our models?

- Idea1: Models trained on different data subset

Bagging

Bagging (Bootstrap-Aggregation)



STEP1: Randomly sample a subset of training data with replacement (Bootstrap)

STEP2: Grow a tree (without pruning) on the subset of data

STEP3: Ensemble the result (regression : average, classification : vote)

Out of Bag error (OOB) : test the grown tree on the rest of data, then average

Random Forest

Random Forest

Bagging : random sampling of data

+

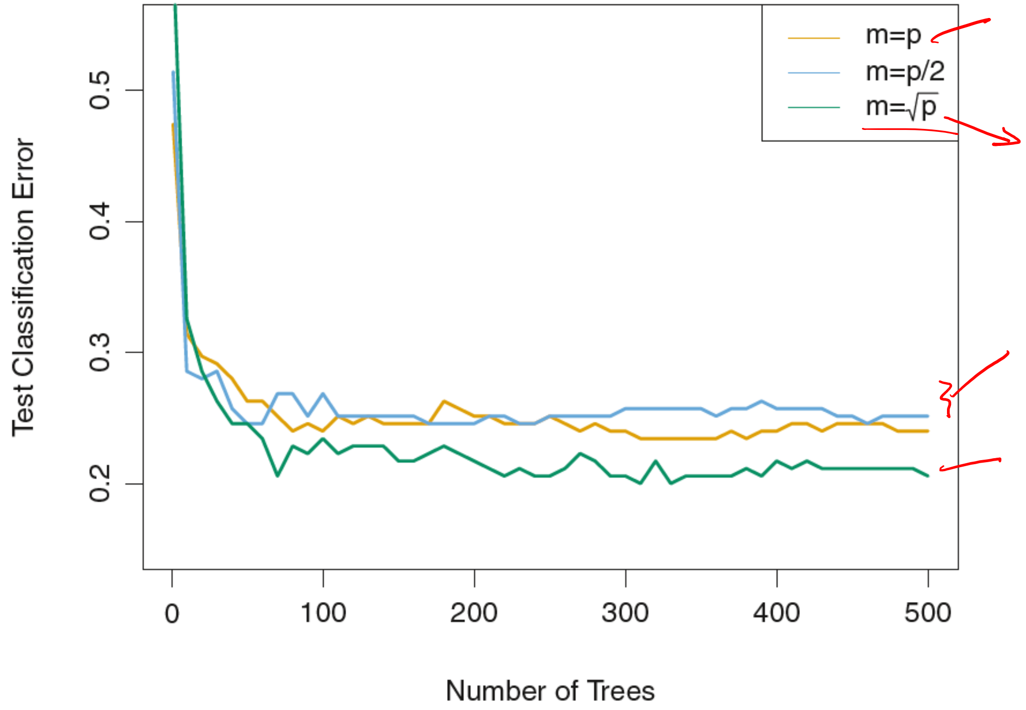
Decorrelation : random sampling of features

=

Random Forest

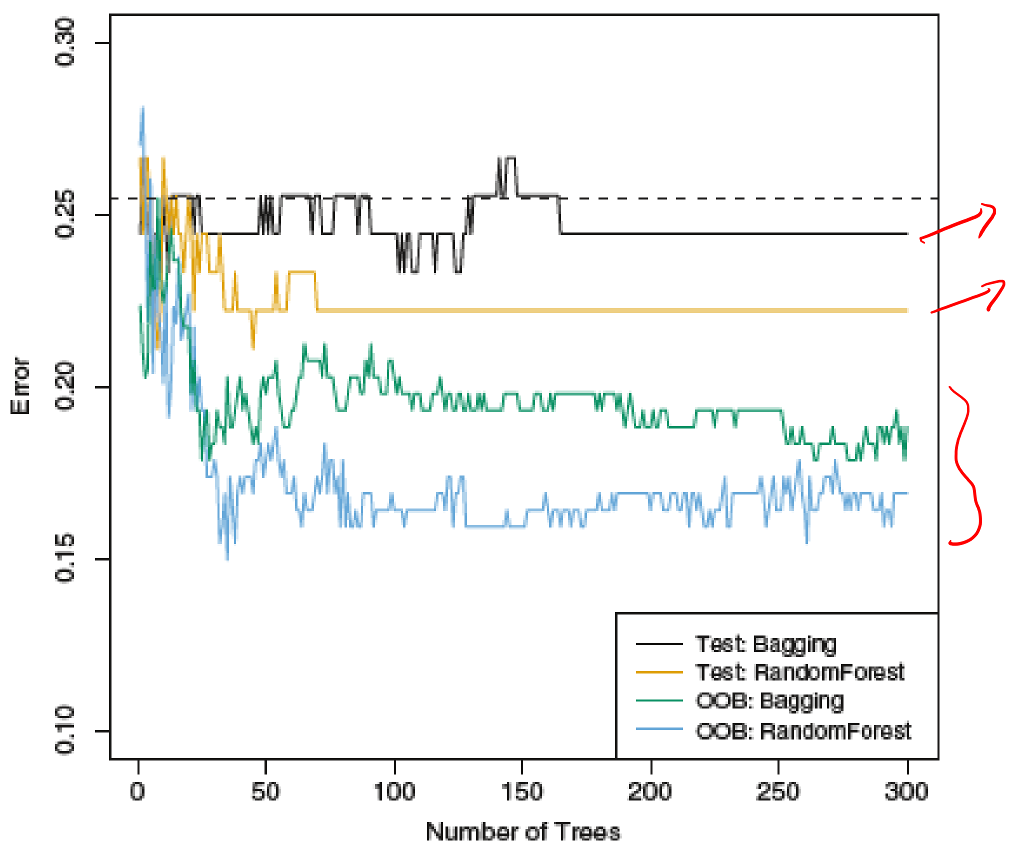
How do we sample features?

-> Rule of thumb : $\sqrt{n}$

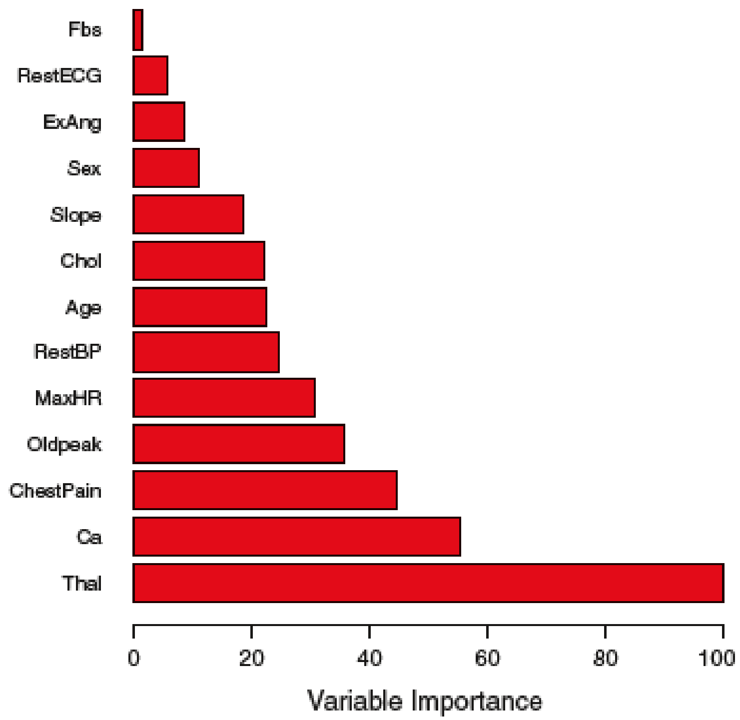


Power of an ensemble of trees

Increased performance



Built-in feature importance



Ensemble method

Is diversity always good?

Ensemble method

How do we gather models that solve the problem?

- Idea3: Include models that actually contribute
- Idea4: Train each sequentially to improve the error

Boosting

Boosting

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
2. For $b = 1, 2, \ldots, B$, repeat:
   (a) Fit a tree $\hat{f}^b$ with d splits ($d+1$ terminal nodes) to the training data (X, r).
   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:

   <1

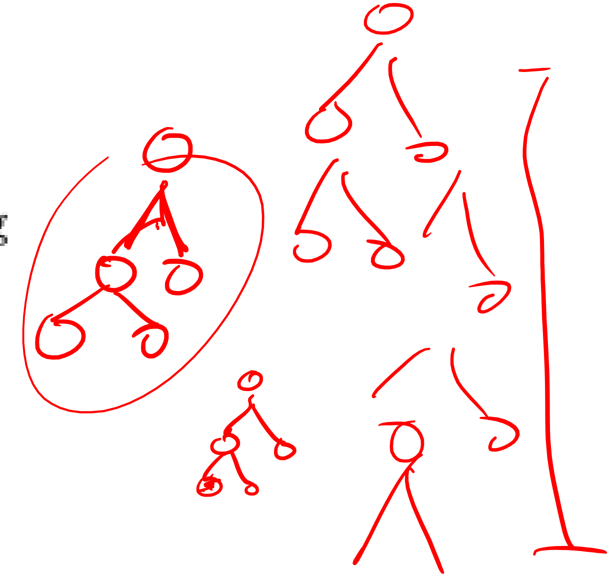
$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x).$$

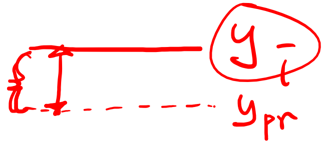
   (c) Update the residuals,

$$r_i \leftarrow r_i - \lambda \boxed{\hat{f}^b(x_i)}.$$

3. Output the boosted model,

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x).$$



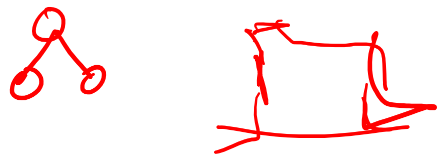


ISLR. *p.323*

Boosting- Tuning parameters

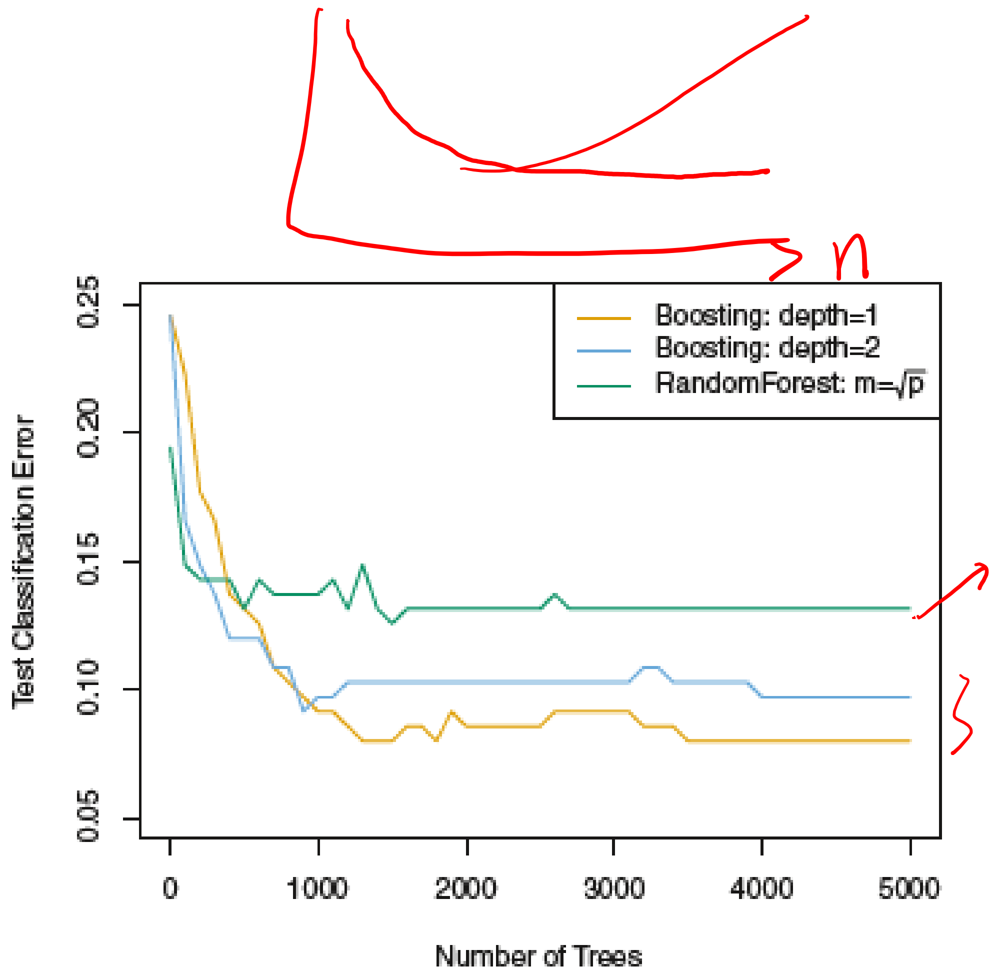
- Number of trees B
  If B too large, it can overfit

- Shrinkage parameter (boosting learning rate) λ
  typical values: 0.01~0.001
  trade-off with B



- Number of splits d in each tree
  as small as d=1~2 enough

RF vs. Boosting



Both RF and Boosting are tree ensembles

RF randomly subsamples features as well as on bootstrapping on data

RF grows large decorrelated trees to fit y

Boosting fits small trees to the residuals and additively add the small trees

RF cannot overfit while boosting can