QTM 347 Machine Learning

Lecture 15: Random forests and boosting

Ruoxuan Xiong

Suggested reading: ISL Chapter 8 and 10

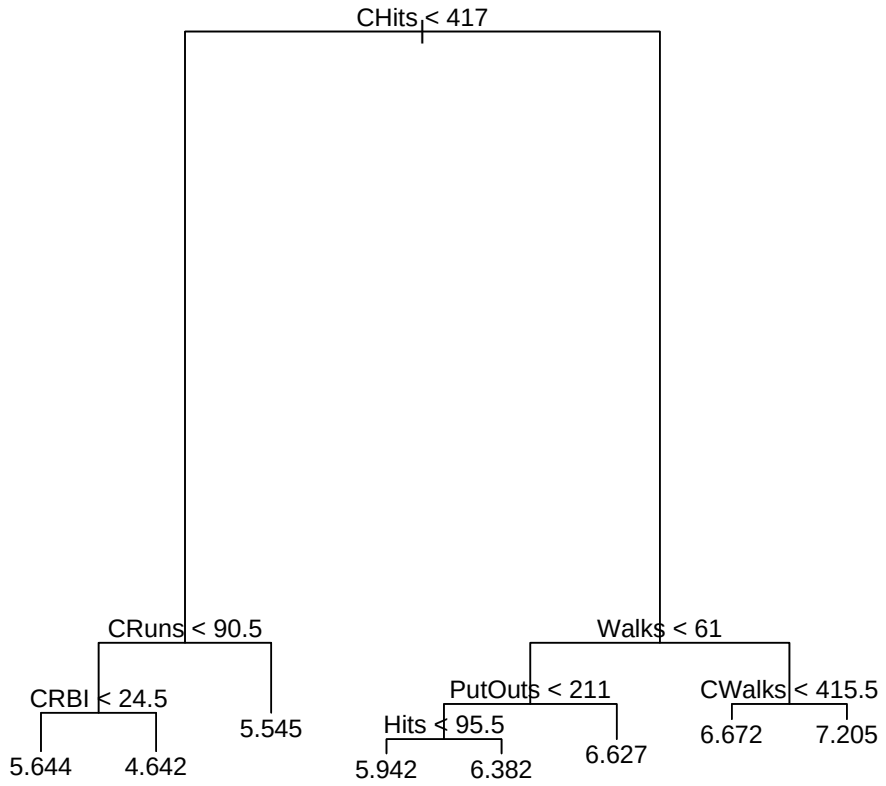
Lecture plan

- Random forests

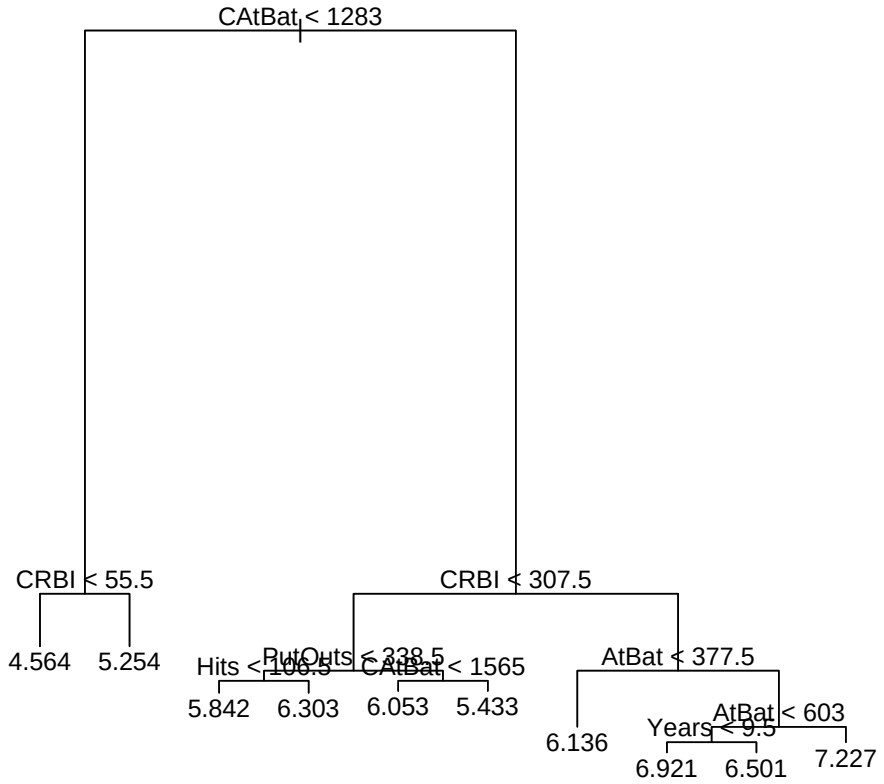
- Gradient boosting

Decision tree has a high variance

- **Example:** Predicting a baseball player's salary
  - Split the training data into two equal-sized parts at random creates disparity



Subsample 1



Subsample 2

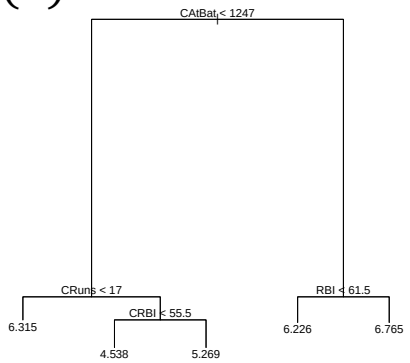
Bagging decision trees to reduce variance

- **Idea: Bootstrap aggregation** (mean / majority of predictions for regression / classification tasks)

Sample #1

X_1	Y_1
X_2	Y_2
X_1	Y_1
X_5	Y_5
X_4	Y_4

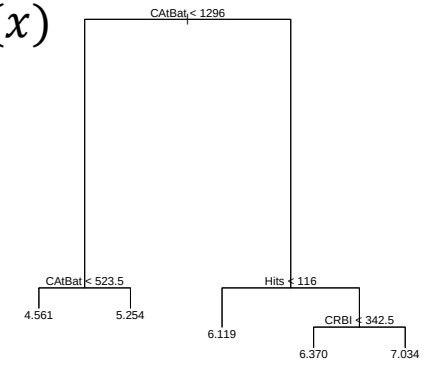
$\hat{f}^1(x)$



Sample #2

X_4	Y_4
X_1	Y_1
X_3	Y_3
X_2	Y_2
X_3	Y_3

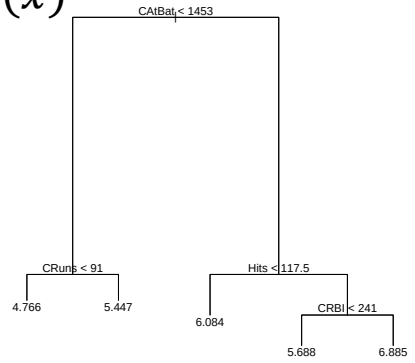
$\hat{f}^2(x)$



Sample #3

X_5	Y_5
X_2	Y_2
X_3	Y_3
X_2	Y_2
X_1	Y_1

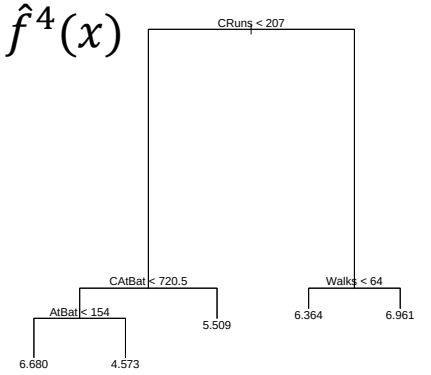
$\hat{f}^3(x)$



Sample #4

X_5	Y_5
X_3	Y_3
X_3	Y_3
X_1	Y_1
X_2	Y_2

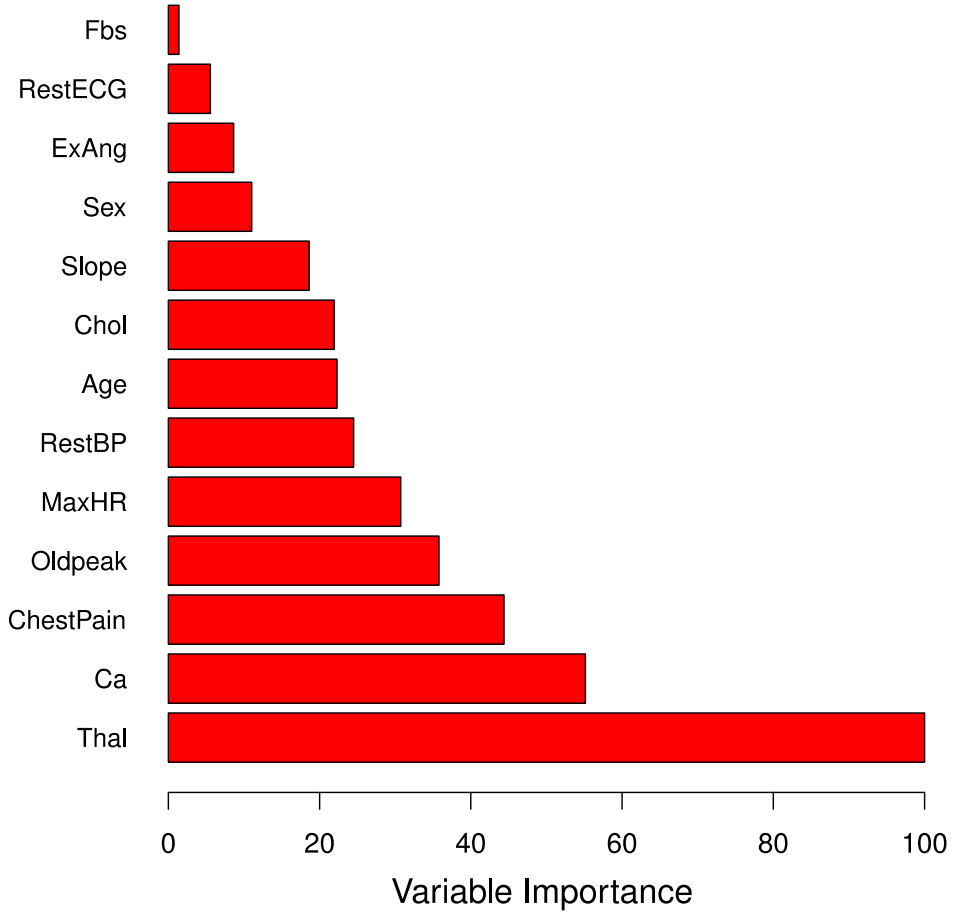
$\hat{f}^4(x)$



Bagging decision trees

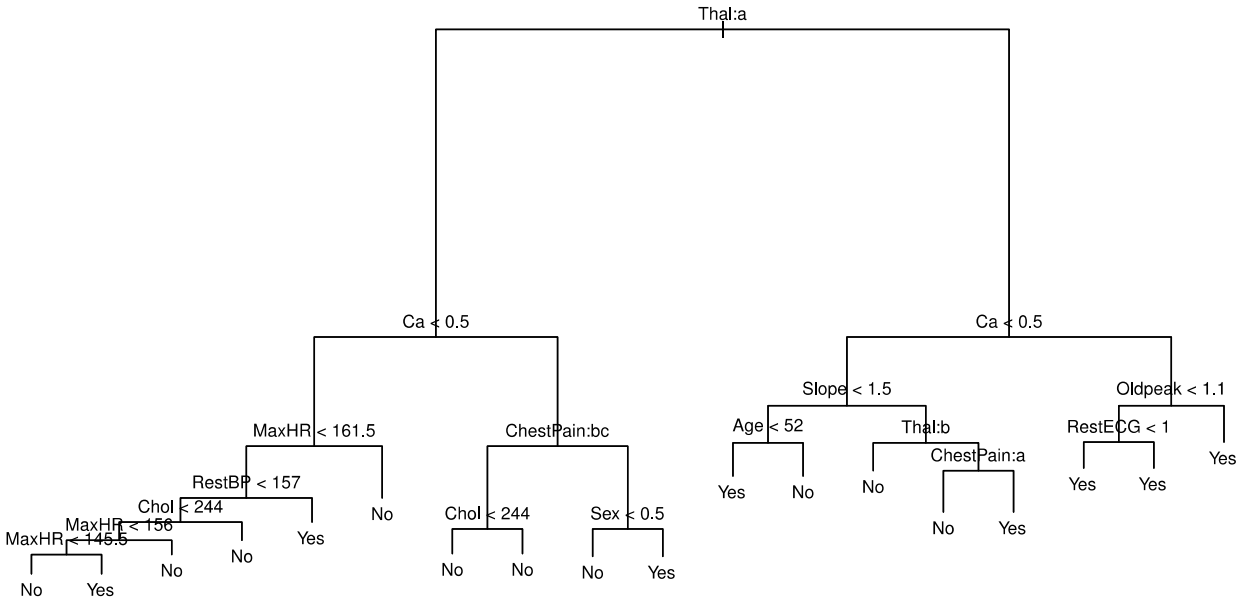
- **Disadvantage**: Loss of interpretability
  - Every time we fit a decision tree to a Bootstrap sample, we get a different tree
- **Solution: Variable importance**
  - For each predictor, add up the total amount by which the RSS (or Gini index) decreases every time we use the predictor in
  - Average the total over each Bootstrap estimate $T^1, \cdots, T^B$

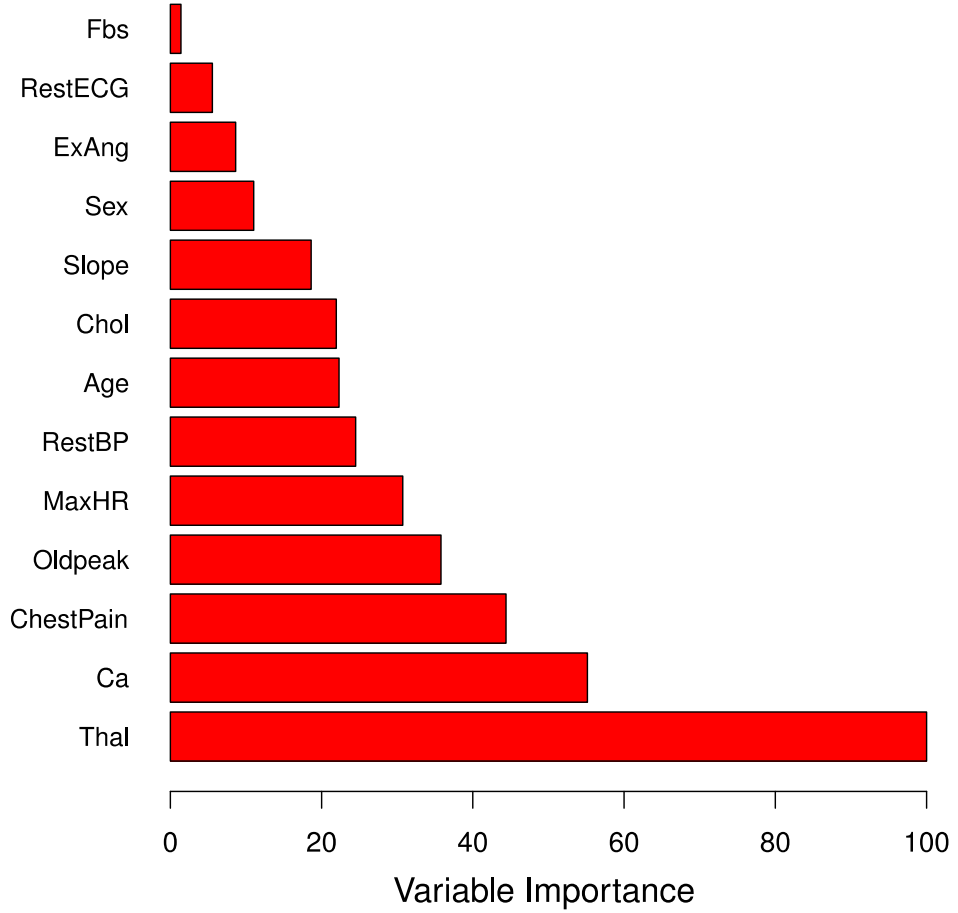
- **Example:** Predicting heart disease



Recall the classification tree to predict heart disease

- **Example**: Predict whether a patient with chest pain has heart disease based on Age, Sex, Chol (a cholesterol measure), and other heart and lung function measures

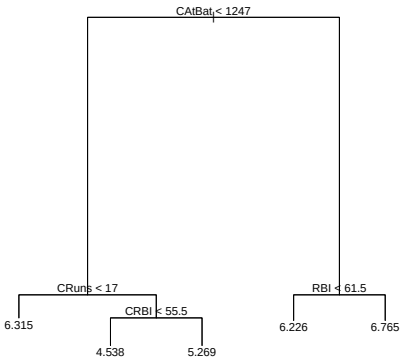




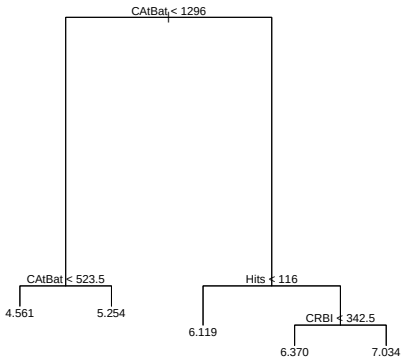
Bagging has a problem

- The trees produced by different Bootstrap samples can be very similar

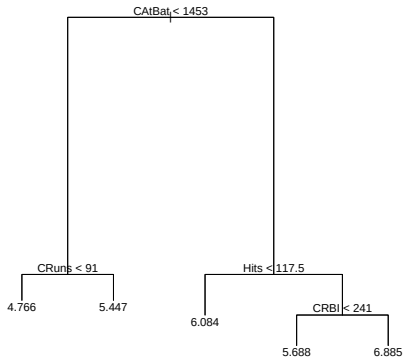
$\hat{f}^1(x)$



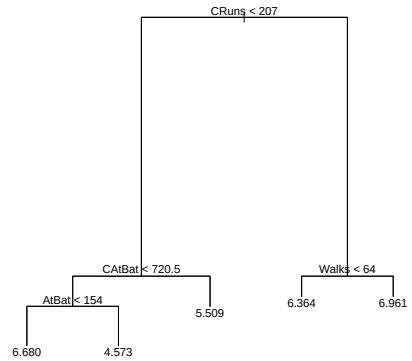
$\hat{f}^2(x)$



$\hat{f}^3(x)$



$\hat{f}^4(x)$



- Three decision trees first split by CAtBat

Random forests

- **Random forests: Bagging plus random sampling of predictors**
  - We fit a decision tree to each Bootstrap samples
  - When fitting the tree, we select a random subset of $m < p$ predictors to consider in each step
  - This will lead to very different trees from each sample
  - Finally, aggregate (mean or majority vote) the prediction of each tree

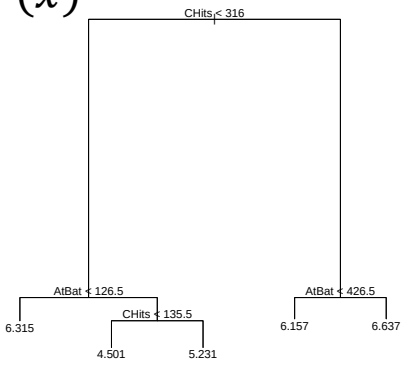
Random forests

- **Random forests** to predict a baseball player salary: $p = 19, m = 5$
  - $X_{i,j}$: jth predictor of observation i

Sample #1

$X_{1,4}$	$X_{1,17}$	$X_{1,9}$	$X_{1,6}$	$X_{1,1}$	Y_1
$X_{2,4}$	$X_{2,17}$	$X_{2,9}$	$X_{2,6}$	$X_{2,1}$	Y_2
$X_{1,4}$	$X_{1,17}$	$X_{1,9}$	$X_{1,6}$	$X_{1,1}$	Y_1
$X_{5,4}$	$X_{5,17}$	$X_{5,9}$	$X_{5,6}$	$X_{5,1}$	Y_5
$X_{4,4}$	$X_{4,17}$	$X_{4,9}$	$X_{4,6}$	$X_{4,1}$	Y_4

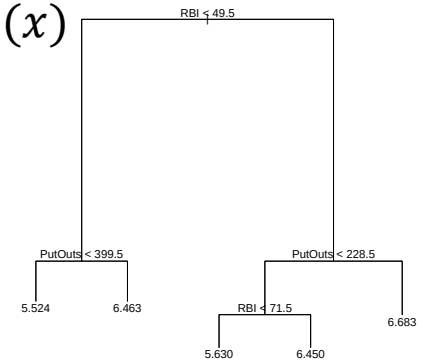
$\hat{f}^1(x)$



Sample #2

$X_{4,16}$	$X_{4,5}$	$X_{4,19}$	$X_{4,18}$	$X_{4,1}$	Y_4
$X_{1,16}$	$X_{1,5}$	$X_{1,19}$	$X_{1,18}$	$X_{1,1}$	Y_1
$X_{3,16}$	$X_{3,5}$	$X_{3,19}$	$X_{3,18}$	$X_{3,1}$	Y_3
$X_{2,16}$	$X_{2,5}$	$X_{2,19}$	$X_{2,18}$	$X_{2,1}$	Y_2
$X_{3,16}$	$X_{3,5}$	$X_{3,19}$	$X_{3,18}$	$X_{3,1}$	Y_3

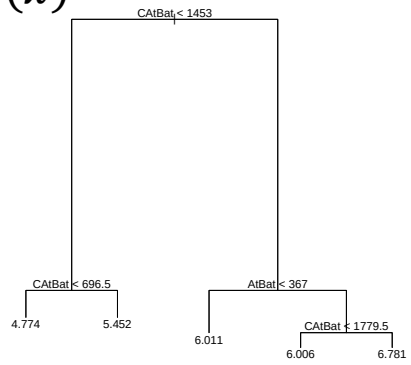
$\hat{f}^2(x)$



Sample #3

$X_{5,6}$	$X_{5,14}$	$X_{5,1}$	$X_{5,4}$	$X_{5,8}$	Y_5
$X_{2,6}$	$X_{2,14}$	$X_{2,1}$	$X_{2,4}$	$X_{2,8}$	Y_2
$X_{3,6}$	$X_{3,14}$	$X_{3,1}$	$X_{3,4}$	$X_{3,8}$	Y_3
$X_{2,6}$	$X_{2,14}$	$X_{2,1}$	$X_{2,4}$	$X_{2,8}$	Y_2
$X_{1,6}$	$X_{1,14}$	$X_{1,1}$	$X_{1,4}$	$X_{1,8}$	Y_1

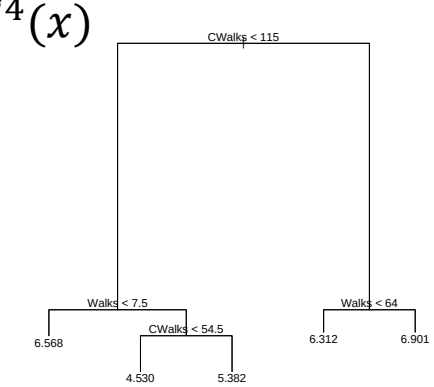
$\hat{f}^3(x)$



Sample #4

$X_{5,17}$	$X_{5,6}$	$X_{5,13}$	$X_{5,5}$	$X_{5,7}$	Y_5
$X_{3,17}$	$X_{3,6}$	$X_{3,13}$	$X_{3,5}$	$X_{3,7}$	Y_3
$X_{3,17}$	$X_{3,6}$	$X_{3,13}$	$X_{3,5}$	$X_{3,7}$	Y_3
$X_{1,17}$	$X_{1,6}$	$X_{1,13}$	$X_{1,5}$	$X_{1,7}$	Y_1
$X_{2,17}$	$X_{2,6}$	$X_{2,13}$	$X_{2,5}$	$X_{2,7}$	Y_2

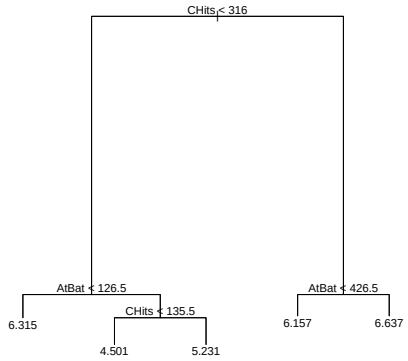
$\hat{f}^4(x)$



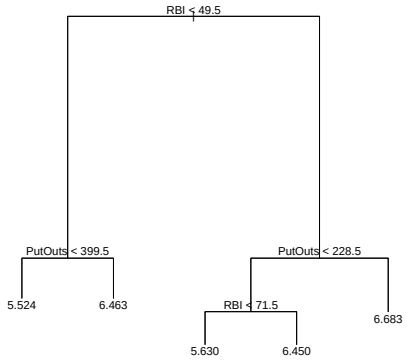
Random forests

	AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CAtBat	CHits	CHmRun	CRuns	CRBI	CWalks	League	Division	PutOuts	Assists	Errors	Salary	NewLeague
-Andy Allanson	293	66	1	30	29	14	1	293	66	1	30	29	14	A	E	446	33	20	NA	A

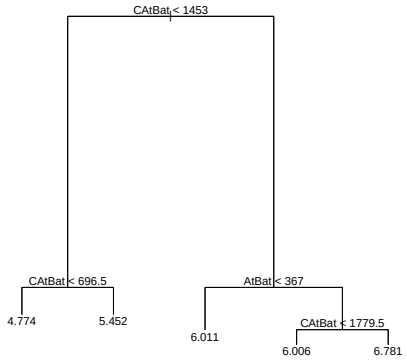
$\hat{f}^1(x)$



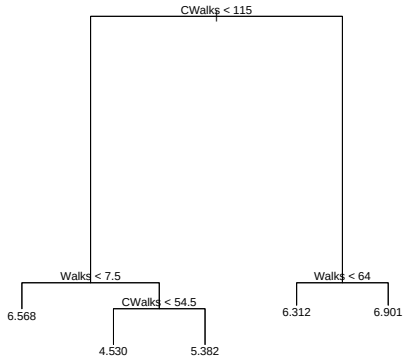
$\hat{f}^2(x)$



$\hat{f}^3(x)$



$\hat{f}^4(x)$



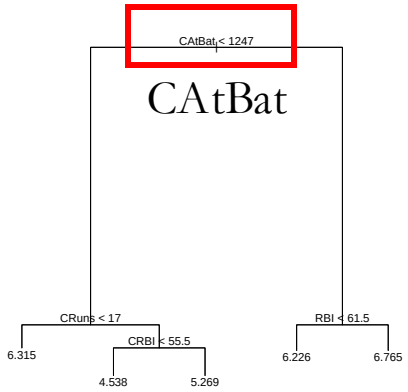
$$\hat{f}_{rf}(x) = \frac{1}{4}\{\hat{f}^1(x) + \hat{f}^2(x) + \hat{f}^3(x) + \hat{f}^4(x)\}$$

- More generally, if we have B bootstrapped training data sets, $\hat{f}_{rf}(x) = \frac{1}{B}\{\hat{f}^1(x) + \hat{f}^2(x) + \cdots + \hat{f}^B(x)\}$

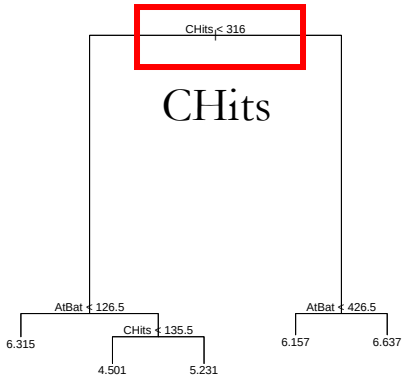
Bagging vs. random forests

$\hat{f}^1(x)$

Bagging

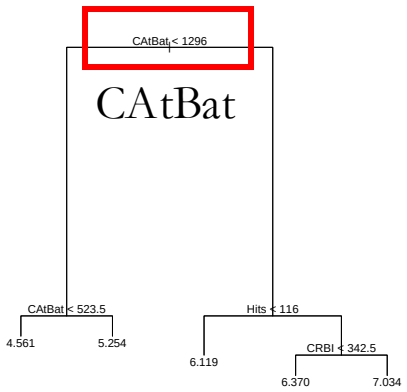


Random forests

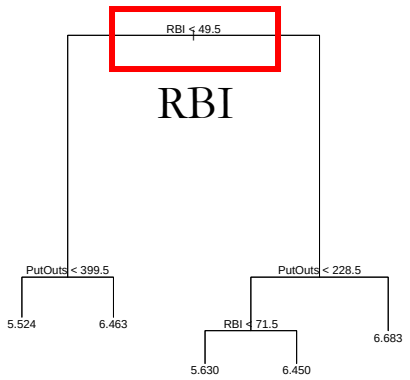


$\hat{f}^2(x)$

Bagging

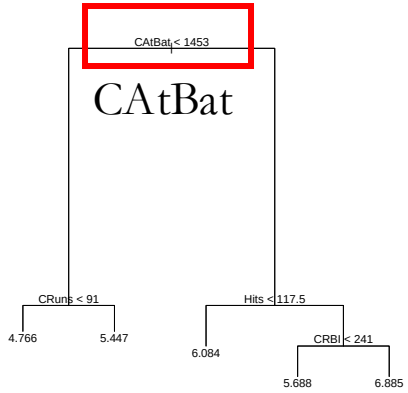


Random forests

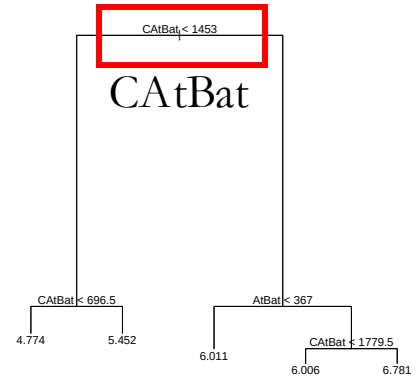


$\hat{f}^3(x)$

Bagging

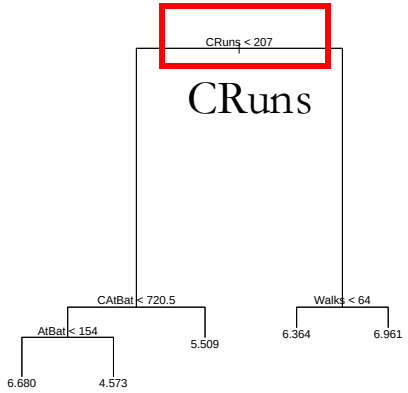


Random forests

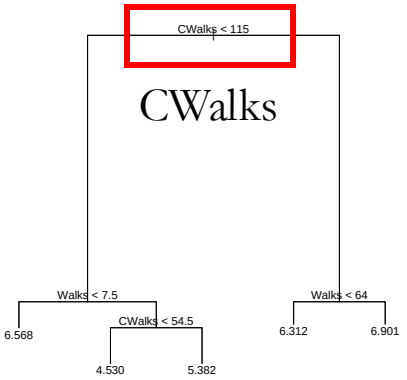


$\hat{f}^4(x)$

Bagging

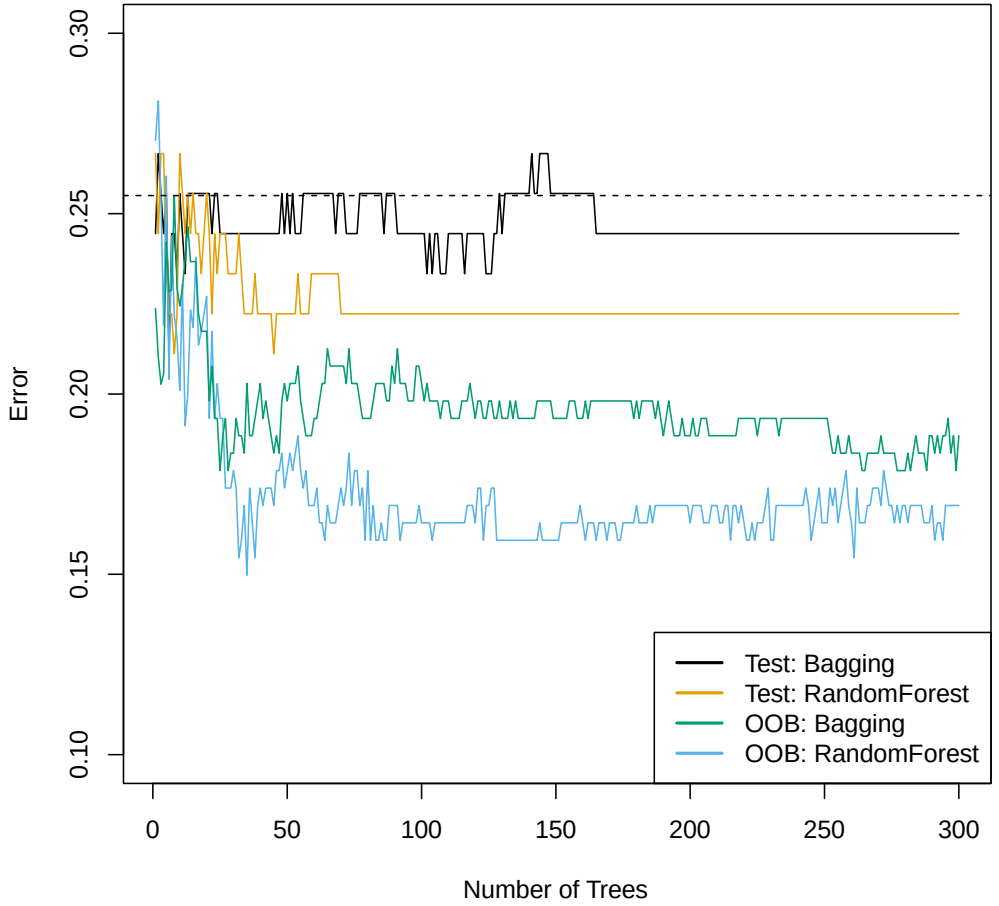


Random forests



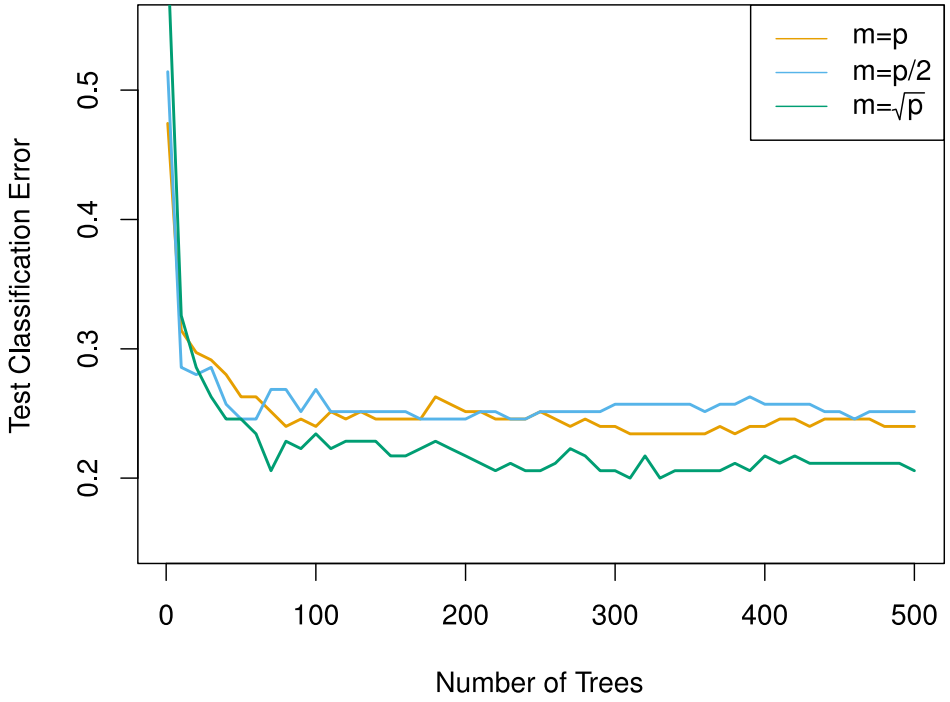
Bagging vs. random forests

- **Example:** Predict whether a patient with chest pain has heart disease
  - Random forests outperform bagging
  - $m = \sqrt{p}$



Random forests, choosing m

- **Example:** Predict cancer type (either normal or 1 of 14 different types of cancer) based on 500 genes
  - Error rate of a single tree: 45.6%
  - Using 400 trees is sufficient



- The optimal m is usually around $\sqrt{p}$, but this can be used as a tuning parameter

Boosted trees

- **Boosted trees**
  - Trees are grown sequentially using the information left from previously grown trees
  - Each tree is fit on a **modified version** of the original data
- Random forests involve a lot of randomness
- Boosting uses less randomness
- Boosting is often more scalable

Boosting

- **Step 1**: Set $\hat{f}(x) = 0$, and $r_i = y_i$ for $i = 1, \cdots, n$.
- **Step 2**: For $b = 1, \cdots, B$, iterate:
  - Fit a decision tree $\hat{f}^b$ with d splits ($d + 1$ terminal nodes) to the response $r_1, \cdots, r_n$
  - Update the prediction to

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

  - Update the residuals

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

- **Step 3**: Output the final model

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

Boosting

X_1	Y_1
X_2	Y_2
X_3	Y_3
X_4	Y_4
X_5	Y_5

$r_i^1 \leftarrow Y_i - \lambda \hat{f}^1(X_i)$

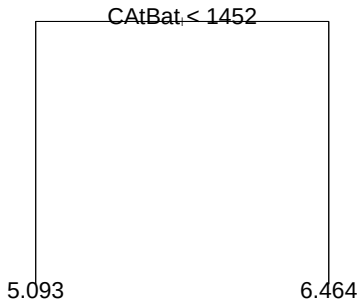
X_1	r_1^1
X_2	r_2^1
X_3	r_3^1
X_4	r_4^1
X_5	r_5^1

$r_i^2 \leftarrow r_i^1 - \lambda \hat{f}^2(X_i)$

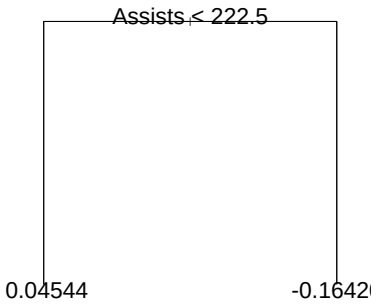
X_1	r_1^2
X_2	r_2^2
X_3	r_3^2
X_4	r_4^2
X_5	r_5^2

$\cdots$

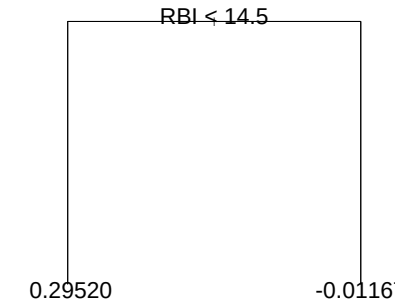
$\hat{f}^1(x)$



$\hat{f}^2(x)$



$\hat{f}^3(x)$



$$\hat{f}(x) = \lambda \hat{f}^1(x) + \lambda \hat{f}^2(x) + \lambda \hat{f}^3(x) + \cdots + \lambda \hat{f}^B(x)$$

Tuning parameters in boosting

- The number of trees B
  - Boosting can overfit if B is too large (a.k.a. early stopping)
  - Use cross-validation to select B
- The shrinkage parameter λ
  - Typical values are 0.01 or 0.001
  - Very small λ requires a large B to achieve good performance
- The number of splits/depth d in each tree
  - $d = 1$ works well
- Remark:
  - Also called **gradient boosting**
  - λ is learning rate

Boosting vs. random forests

- **Example:** Predict cancer type (either normal or 1 of 14 different types of cancer) based on 500 genes
  - $\lambda = 0.01$
  - Depth-1 trees outperform depth-2 trees
  - Both outperform random forests