Bagging and Random Forests

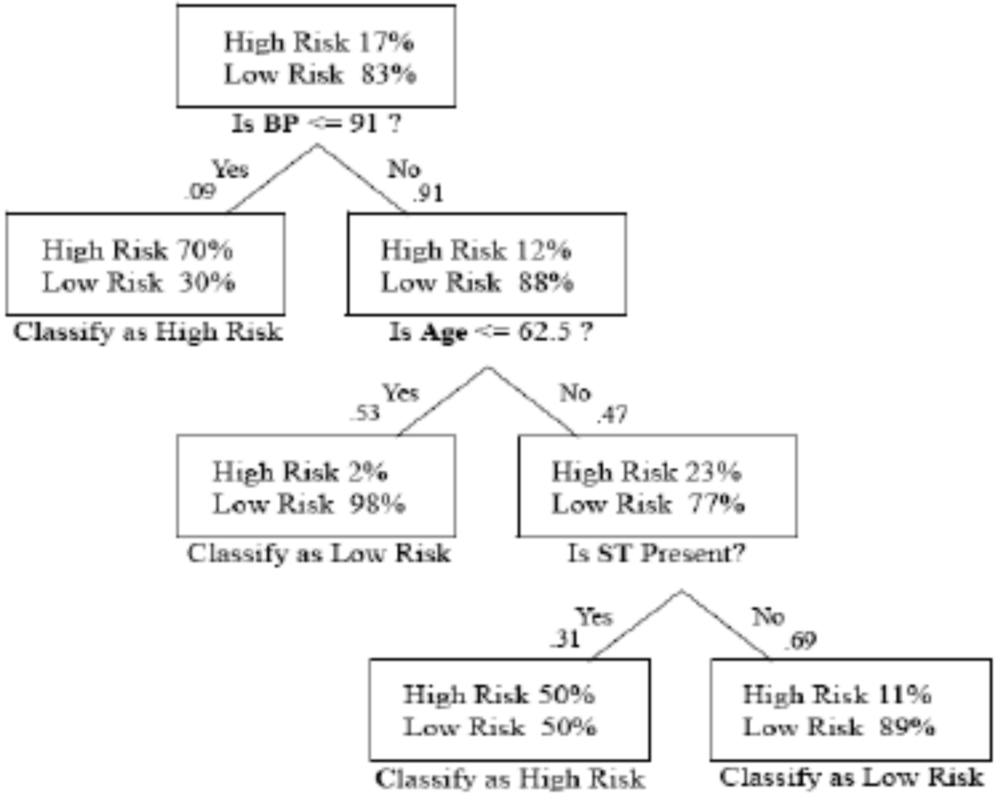
Sravani Vadlamani

Agenda

- Review of Regression Trees & Limitations
- Bagging
- Random Forests
- Empirical Example

Example of a Classification Tree

THE BINARY DECISION TREE (Figure 1)



Regression Trees

- Regression trees identify variables that are important for prediction in a different way
  - Stratifying the prediction space into several simple regions
  - Identifies variable and cut point on the variable to partition the data and does this repeatedly
  - Allows for non-linear associations and interaction effects
  - Simple methods useful for interpretation but not competitive in terms of accuracy
  - Can be applied to both regression and classification problems

Dividing the predictor space

- How is the division done? Theoretically, the region can be of any shape.
- Divide the predictor space into high-dimensional rectangles (2-dimensional space) or boxes (3 – dimensional space) for simplicity and ease of interpretation
- Goal is to find regions that minimize the residual sum of squares

$$\sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j}\right)^2$$

- $\hat{y}_{R_j}$ is the mean response for region j

How to divide the predictor space

- It is infeasible to consider every partition of the predictor space
- We use a top-down (greedy) approach
    - Recursive binary splitting
- Algorithm
    - Choose the best split (that minimizes RSS) of the predictor space
    - Divide the data into partitions based on the split
    - Choose the best split of the predictor space for one of the two partitions
        - Now you get three partitions
    - Repeat until a stopping criterion is reached (for e.g., no region contains more than 5 observations)

Issues with greedy approach

- Results in a tree optimized to the dataset
  - At each step, a cut off score on a variable is chosen that minimizes the residual sum of squares the most
- Can lead to overfitting
  - Fitting a model too closely to the training data set may make it difficult to replicate the model on test data (i.e., model is too complex)
  - This can be avoided by using a smaller tree
- Secondary issue with regression trees is the heavy dependence on the first split

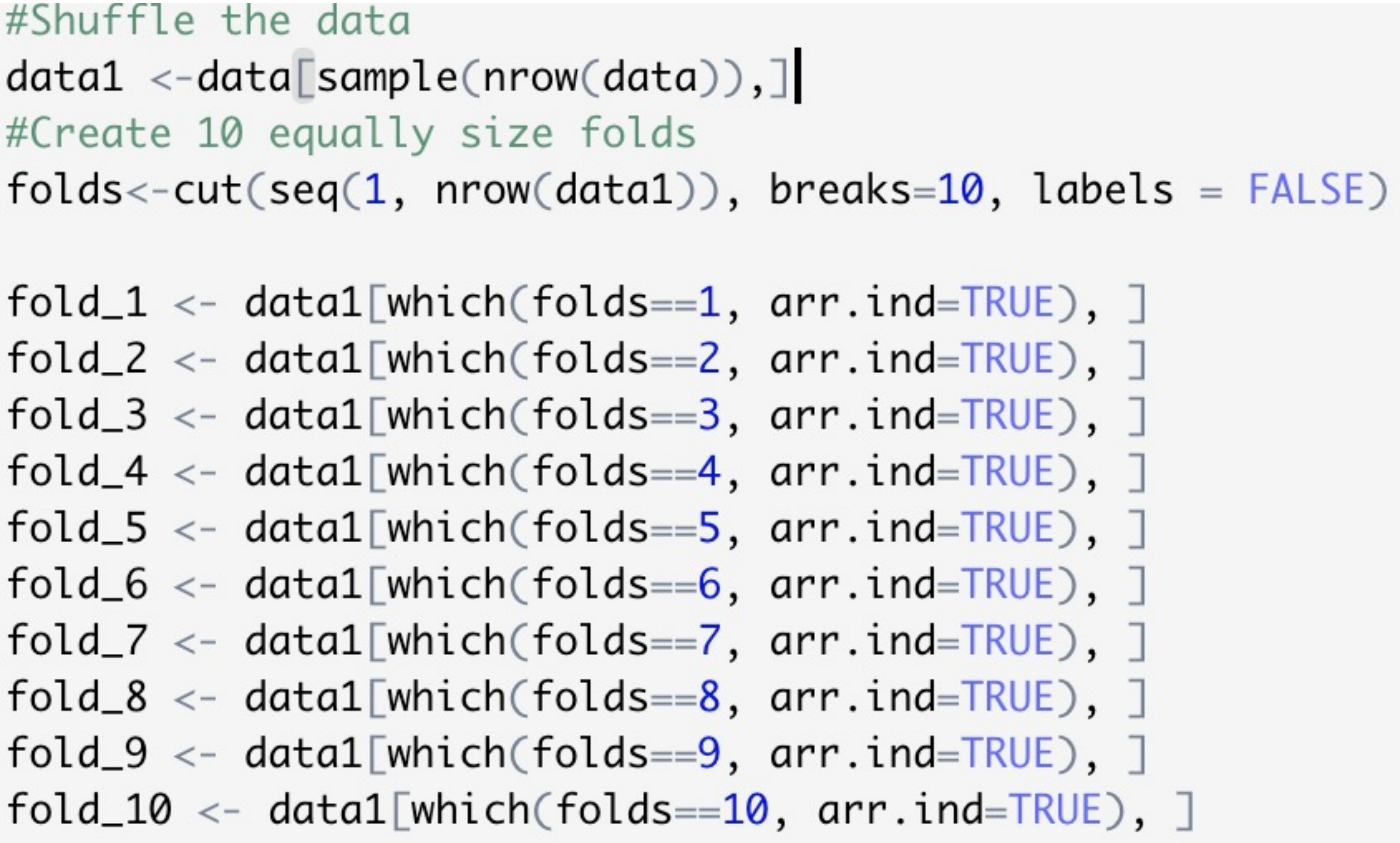
Advantages of Trees

- Easy to explain (than linear regression!)
- More closely mirrors human decision-making than the previously seen regression and classification methods
- Easy to display and interpret for a non-technical audience
- Handle qualitative predictors without creating dummy variables

Limitations of Trees

- Prone to instability even with minor data changes
  - Decision rules change even when the same variables are selected
- Relies heavily on the first split
- Variables that are correlated with the first split but not chosen may appear to be unimportant
- Trees tend to overfit the training data unless pruned
- Potentially have high bias

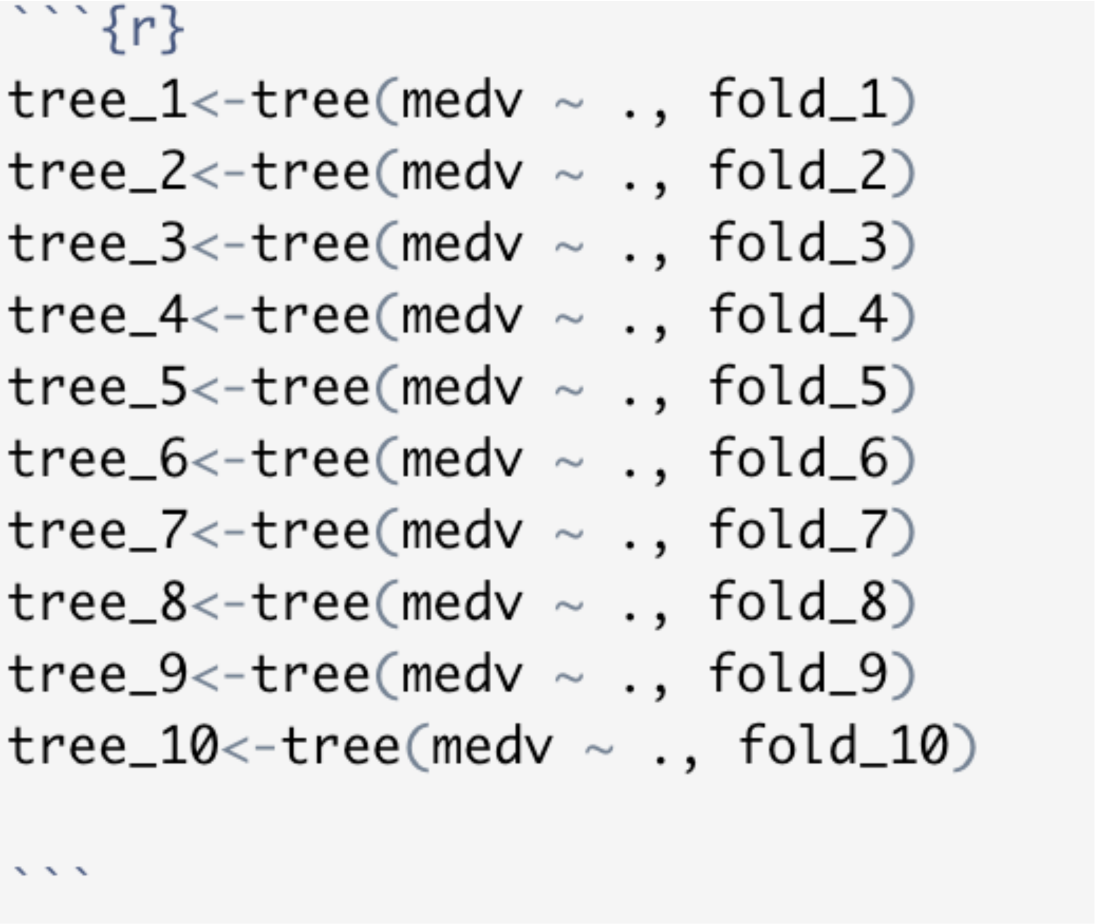
Boston Example – Create 10 datasets each with 10% of the data



```
#Shuffle the data
data1 <-data[sample(nrow(data)),]
#Create 10 equally size folds
folds<-cut(seq(1, nrow(data1)), breaks=10, labels = FALSE)

fold_1 <- data1[which(folds==1, arr.ind=TRUE), ]
fold_2 <- data1[which(folds==2, arr.ind=TRUE), ]
fold_3 <- data1[which(folds==3, arr.ind=TRUE), ]
fold_4 <- data1[which(folds==4, arr.ind=TRUE), ]
fold_5 <- data1[which(folds==5, arr.ind=TRUE), ]
fold_6 <- data1[which(folds==6, arr.ind=TRUE), ]
fold_7 <- data1[which(folds==7, arr.ind=TRUE), ]
fold_8 <- data1[which(folds==8, arr.ind=TRUE), ]
fold_9 <- data1[which(folds==9, arr.ind=TRUE), ]
fold_10 <- data1[which(folds==10, arr.ind=TRUE), ]
```

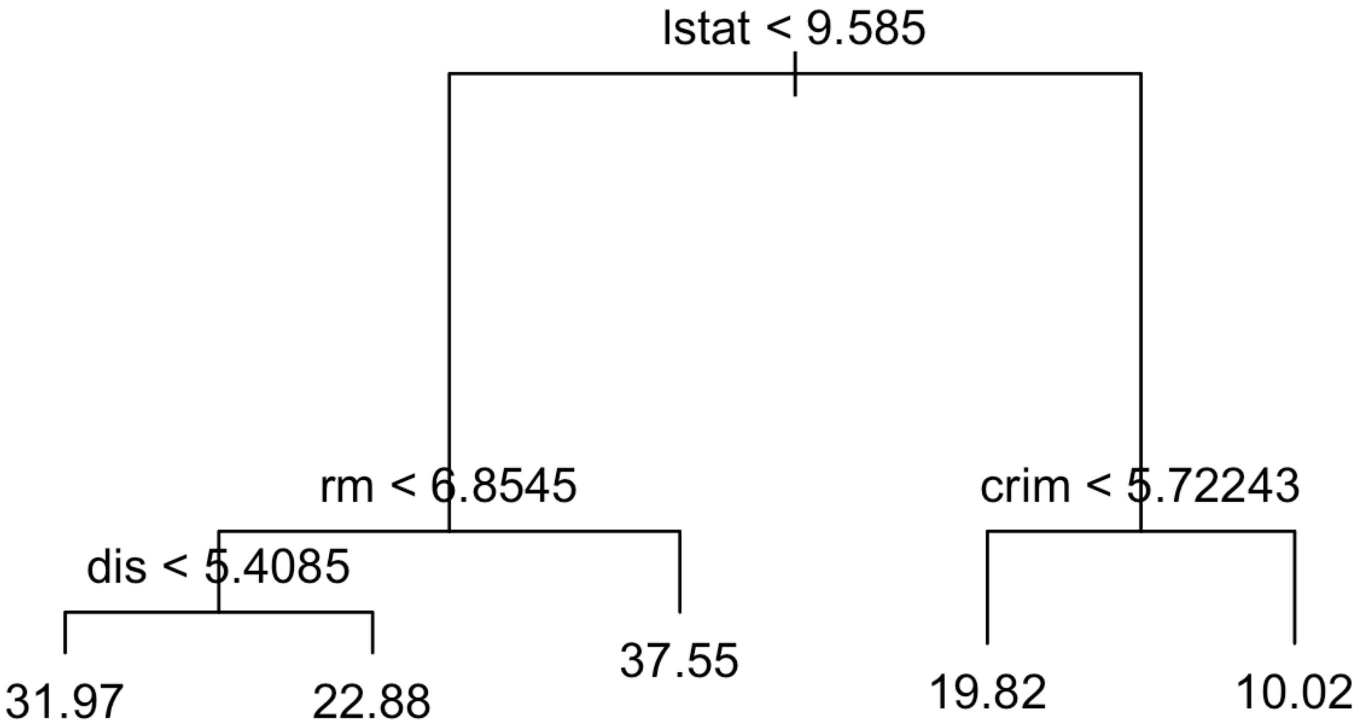
Boston Example – Fit a tree to each dataset



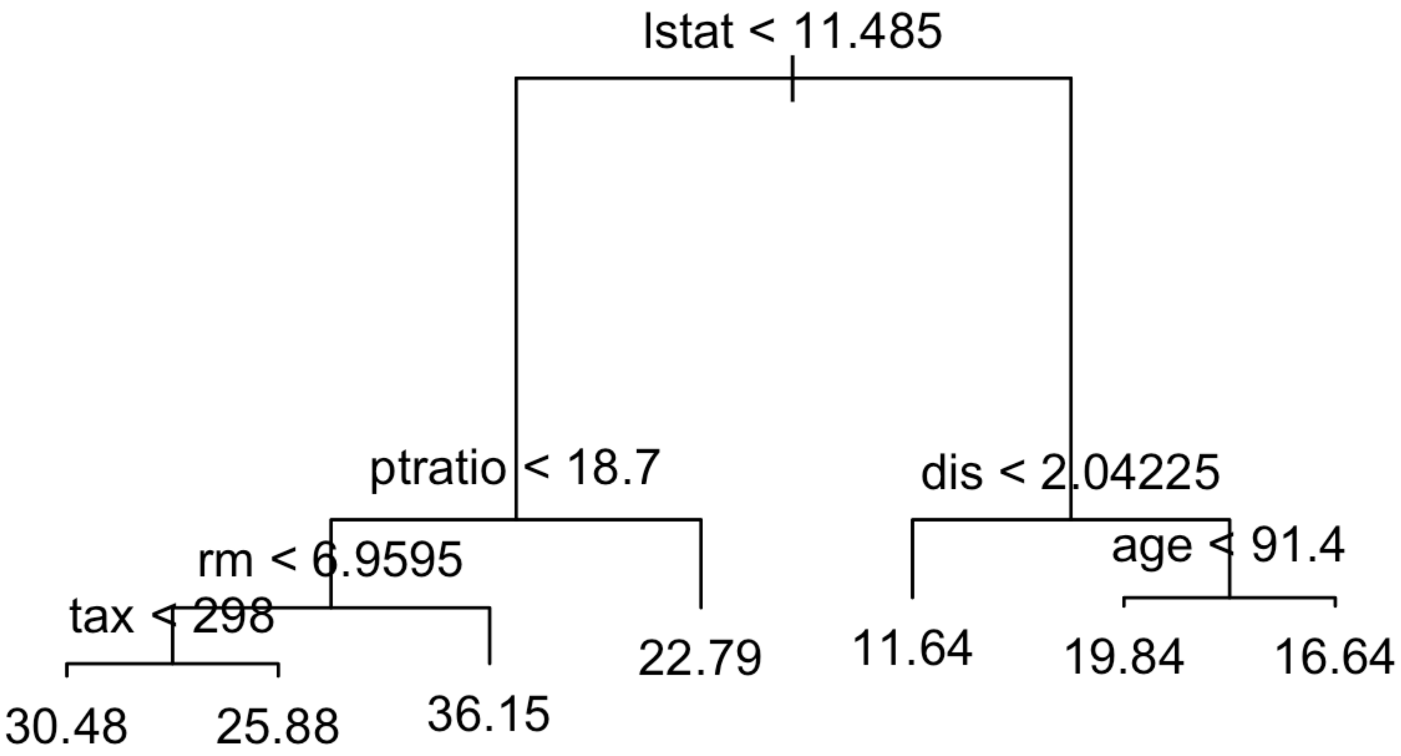
~~~
```{r}
tree_1<-tree(medv ~ ., fold_1)
tree_2<-tree(medv ~ ., fold_2)
tree_3<-tree(medv ~ ., fold_3)
tree_4<-tree(medv ~ ., fold_4)
tree_5<-tree(medv ~ ., fold_5)
tree_6<-tree(medv ~ ., fold_6)
tree_7<-tree(medv ~ ., fold_7)
tree_8<-tree(medv ~ ., fold_8)
tree_9<-tree(medv ~ ., fold_9)
tree_10<-tree(medv ~ ., fold_10)

```
~~~

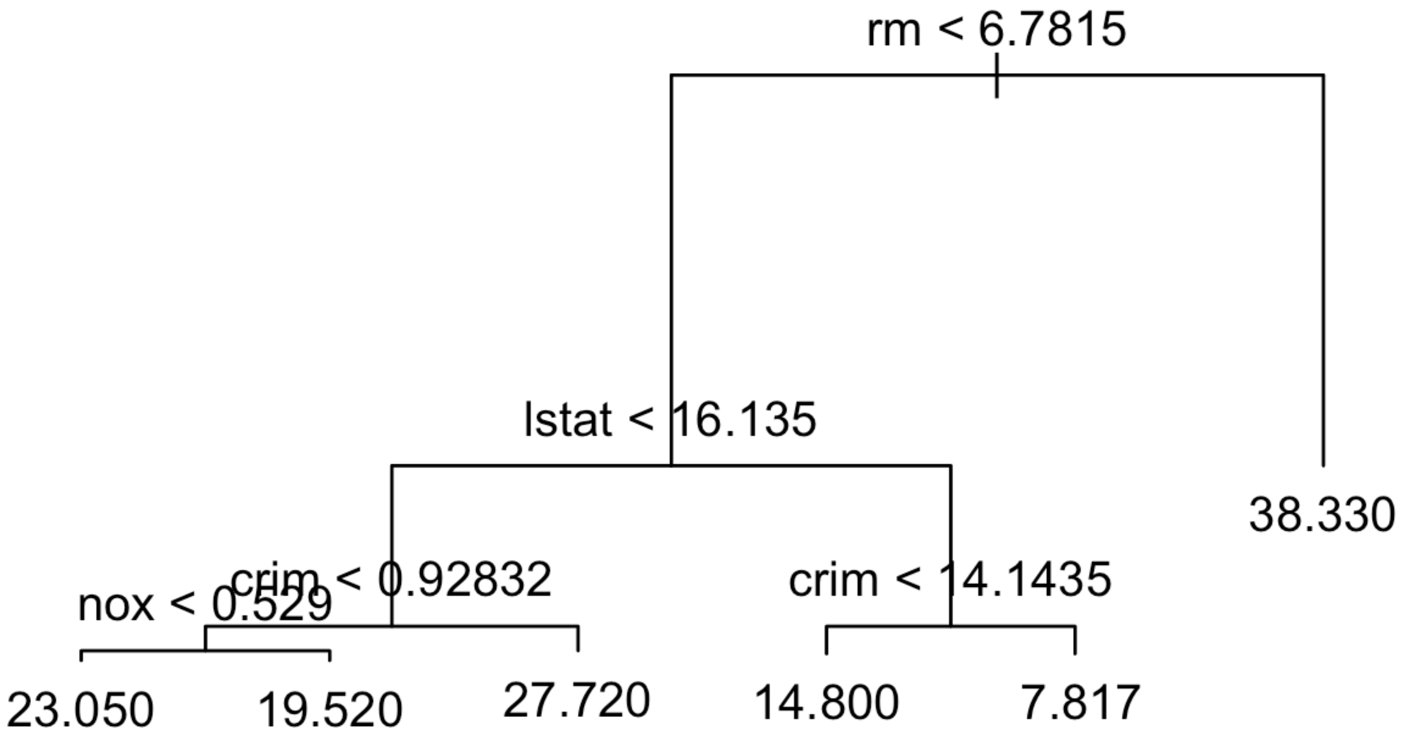
Boston Example – Tree 1



Boston Example – Tree 2



Boston Example – Tree 2



Trees

- Above figures show how trees can be sensitive to sampling from a single dataset
- Decision rules change even when the same variables are selected

Getting better Prediction

- Prediction is improved when we include more samples
- For example, if you want to estimate the population mean of a variable
  - Obtain a score from a single person – standard error of the mean is the population standard deviation
  - Obtain more scores – standard error of the mean shrinks $\frac{\sigma}{\sqrt{n}}$
- Can we use the same idea for trees and grow lots of trees?

Bagging

- Bagging = Bootstrap Aggregation
- Bootstrap
    - Resampling procedure where we sample a certain number of cases from our dataset with replacement
- Aggregation
    - The act of collecting together

Steps in Bagging

- Bootstrap our data
- Fit a tree to the bootstrapped data
- Obtain the predicted values
- Repeat B times
- Average the predicted values from the trees

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)$$

Bagging

- Big trees are grown and are not pruned
- How many trees should be grown?
- Cross-validation
    - In any given bootstrap sample, approximately 2/3 of the data is used to make a bagged tree and the remaining 1/3 observations are not used. These unused observations are called out-of-bag (OOB) observations and are used for testing.
    - OOB MSE is a valid estimate of the test error. When B is large, the OOB error estimate is approximately equal to the leave-one-out CV error.

Variable Importance

- Bagging leads to improved prediction. However, results are difficult to interpret as we do not have a single tree
- Improves prediction at the cost of interpretability
- Variable importance helps provide information on how each variable contributes to the overall prediction

Variable Importance Indices

- For each tree, the prediction error (MSE) on the out-of-bag portion of the data is recorded. The same is done after permuting each predictor variable. Difference in prediction errors are averaged over trees and normalized by the standard deviation of the differences
- Total decrease in node impurities from splitting on the variable averaged over all trees. In regression trees, it is based on the residual sum of squares

Random Forests

- Extension of bagging with a goal to decorrelate the trees
- Bagged trees are correlated i.e., they are independent but not identically distributed
- To decorrelate the trees, we choose a random sample of observations AND a random sample of predictors

Random Forests

- Build a number of trees using bootstrapped samples
- At each split in a tree, we choose a random sample of m predictors as split candidates from the full set of p predictors
- A new sample of m predictors is taken at each split

How to choose m

- When m is small, the trees are correlated to a smaller degree
- We typically choose m to be quite small (often $m = \sqrt{p}$ or $m = \log(p))$, so that the trees will be uncorrelated
- When predictors are highly correlated, it makes sense to have a smaller value of m