Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

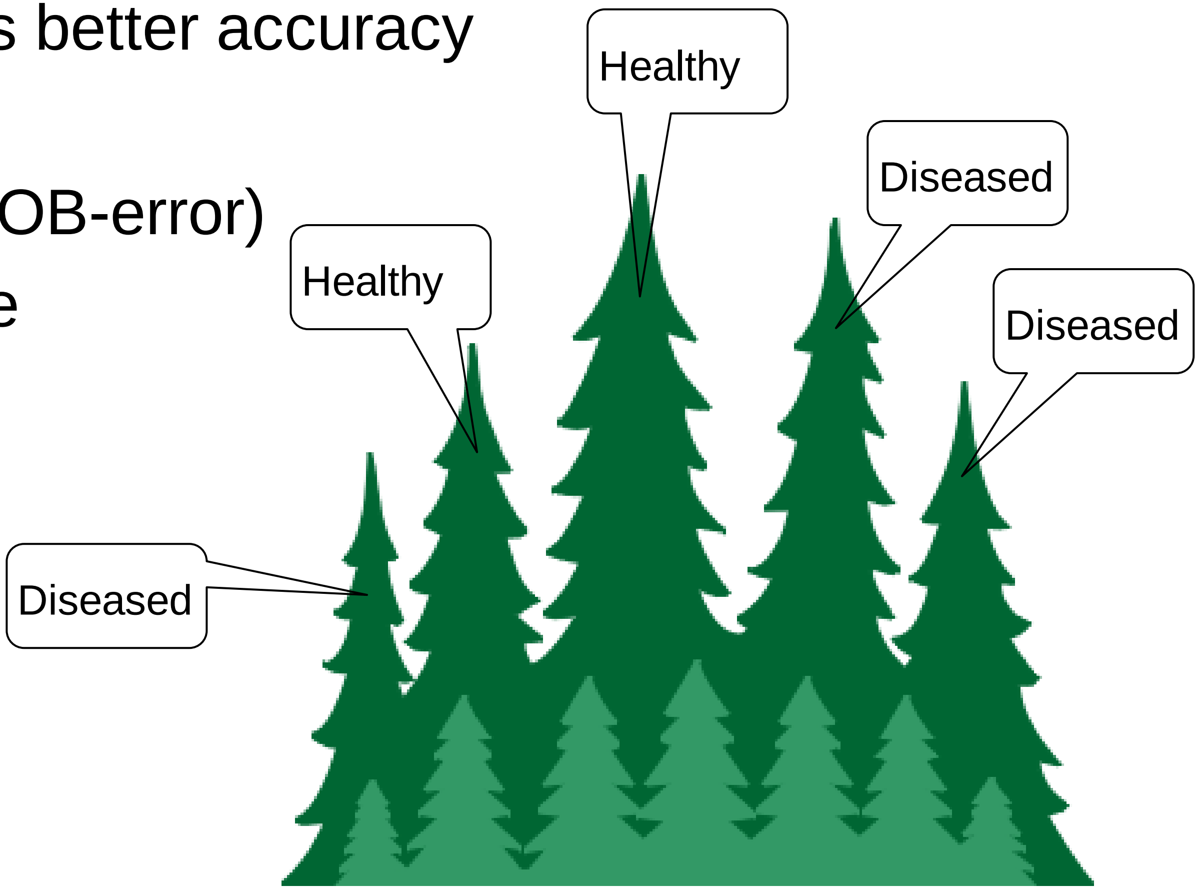
# Random Forest

Applied Multivariate Statistics – Spring 2013

# Overview

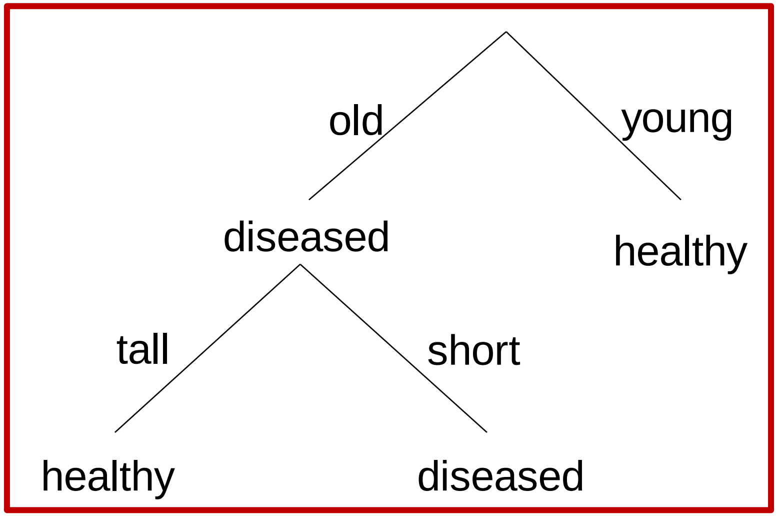
- Intuition of Random Forest
- The Random Forest Algorithm
- De-correlation gives better accuracy

- Out-of-bag error (OOB-error)
- Variable importance

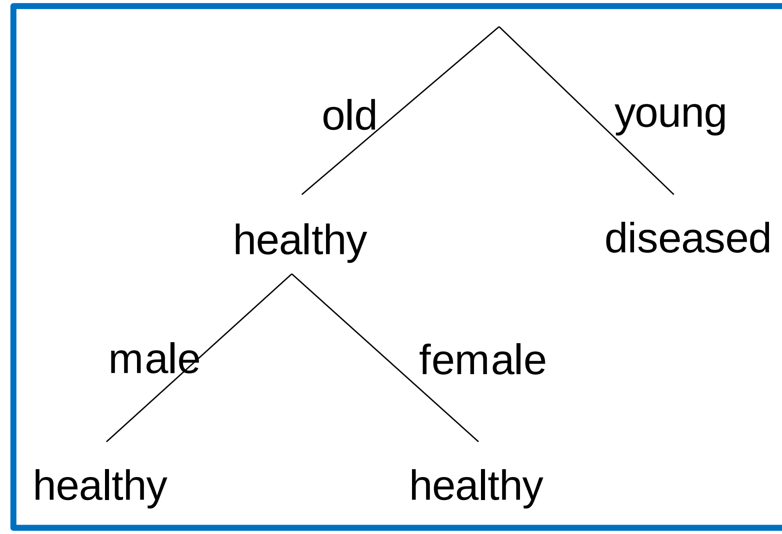


# Intuition of Random Forest

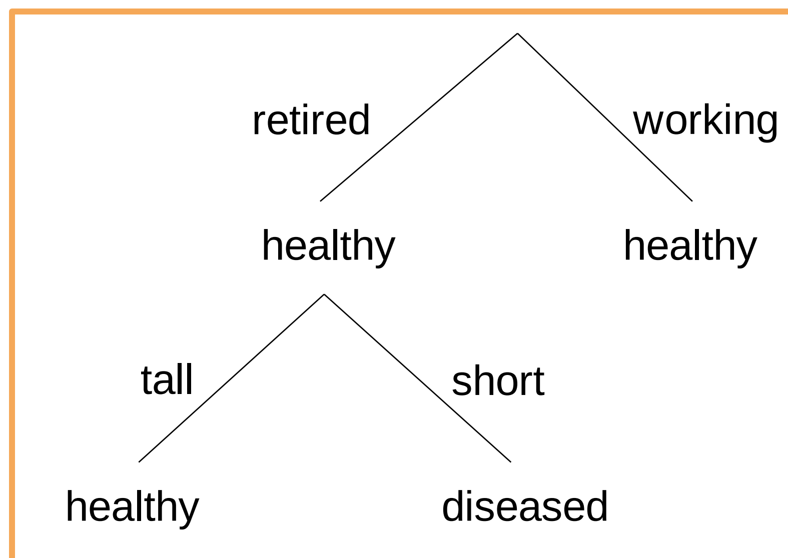
Tree 1




Tree 2




Tree 3




**New sample:**

old, retired, male, short

**Tree predictions:**

diseased, healthy, diseased

**Majority rule:**
**diseased**

# The Random Forest Algorithm

1. For $b = 1$ to $B$:

   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size $N$ from the training data.

   (b) Grow a random-forest tree $T_b$ to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $n_{min}$ is reached.

      i. Select $m$ variables at random from the $p$ variables.

      ii. Pick the best variable/split-point among the $m$.

      iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point $x$:

*Regression:* $\hat{f}_{\mathrm{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.

*Classification:* Let $\hat{C}_b(x)$ be the class prediction of the $b$th random-forest tree. Then $\hat{C}_{\mathrm{rf}}^B(x) = \textit{majority vote } \{\hat{C}_b(x)\}_1^B$.

# Differences to standard tree

- Train each tree on bootstrap resample of data
  (Bootstrap resample of data set with N samples:
  Make new data set by drawing **with replacement** N samples; i.e., some samples will probably occur multiple times in new data set)
- For each split, consider only m randomly selected variables
- Don’t prune
- Fit B trees in such a way and use average or majority voting to aggregate results

# Why Random Forest works 1/2

- Mean Squared Error = Variance + Bias$^2$
- If trees are sufficiently deep, they have very small bias

- How could we improve the variance over that of a single tree?

# Why Random Forest works 2/2

$$
\begin{aligned}
Var\left(\frac{1}{B}\sum_{i=1}^{B} T_i(x)\right) &= \frac{1}{B^2}\sum_{i=1}^{B}\sum_{j=1}^{B} Cov(T_i(x), T_j(x)) \\
&= \frac{1}{B^2}\sum_{i=1}^{B}\left(\sum_{j\neq i}^{B} Cov(T_i(x), T_j(x)) + Var(T_i(x))\right) \\
&= \frac{1}{B^2}\sum_{i=1}^{B}\left((B-1)\sigma^2\cdot\rho + \sigma^2\right) \\
&= \frac{B(B-1)\rho\sigma^2 + B\sigma^2}{B^2} \\
&= \frac{(B-1)\rho\sigma^2}{B} + \frac{\sigma^2}{B} \\
&= \rho\sigma^2 - \frac{\rho\sigma^2}{B} + \frac{\sigma^2}{B} \\
&= \boxed{\rho\sigma^2} + \boxed{\sigma^2\frac{1-\rho}{B}}
\end{aligned}
$$

i=j

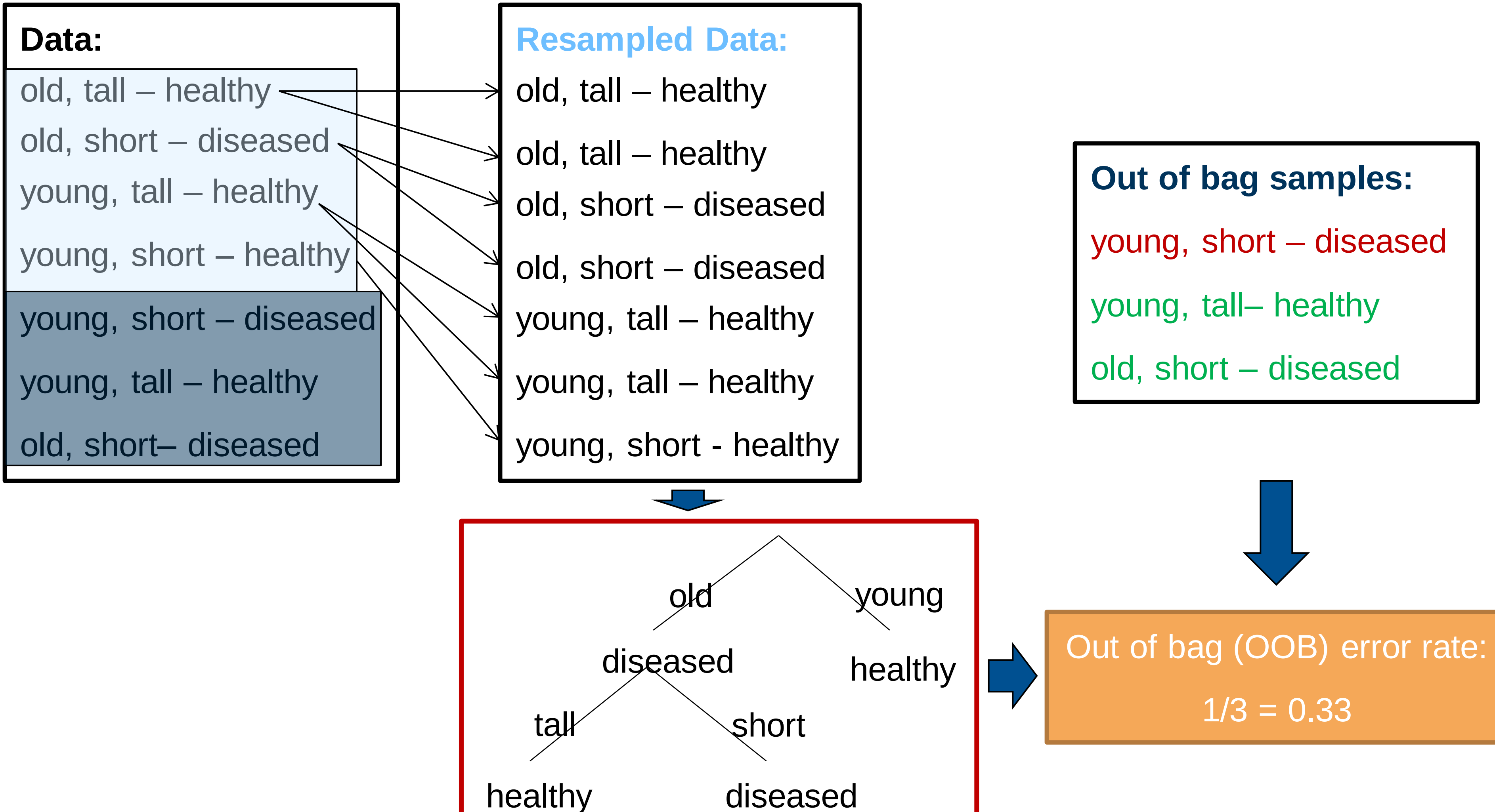
De-correlation gives better accuracy

Decreaes, if $\rho$ decreases, i.e., if m decreases

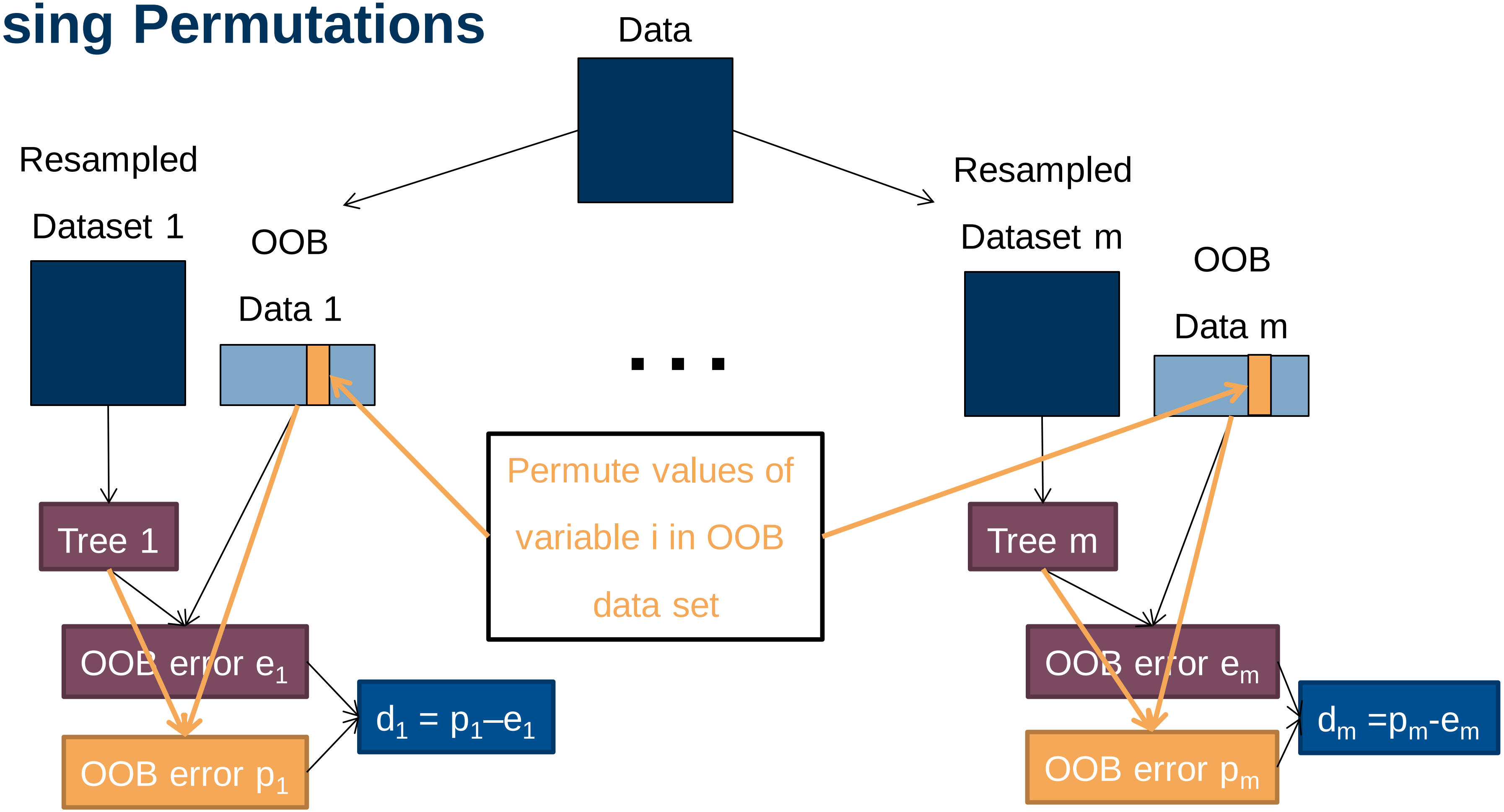
Decreases, if number of trees B increases (irrespective of $\rho$)

# Estimating generalization error: Out-of bag (OOB) error

- Similar to leave-one-out cross-validation, but almost without any additional computational burden



# Variable Importance for variable i using Permutations




$$\bar{d} = \frac{1}{m} \sum_{i=1}^{m} d_i$$

$$s_d^2 = \frac{1}{m-1} \sum_{i=1}^{m} (d_i - \bar{d})^2$$

$$v_i = \frac{\bar{d}}{s_d}$$

# Trees vs. Random Forest

+ Trees yield insight into decision rules

+ Rather fast

+ Easy to tune parameters

- Prediction of trees tend to have a high variance

+ RF has smaller prediction variance and therefore usually a better general performance

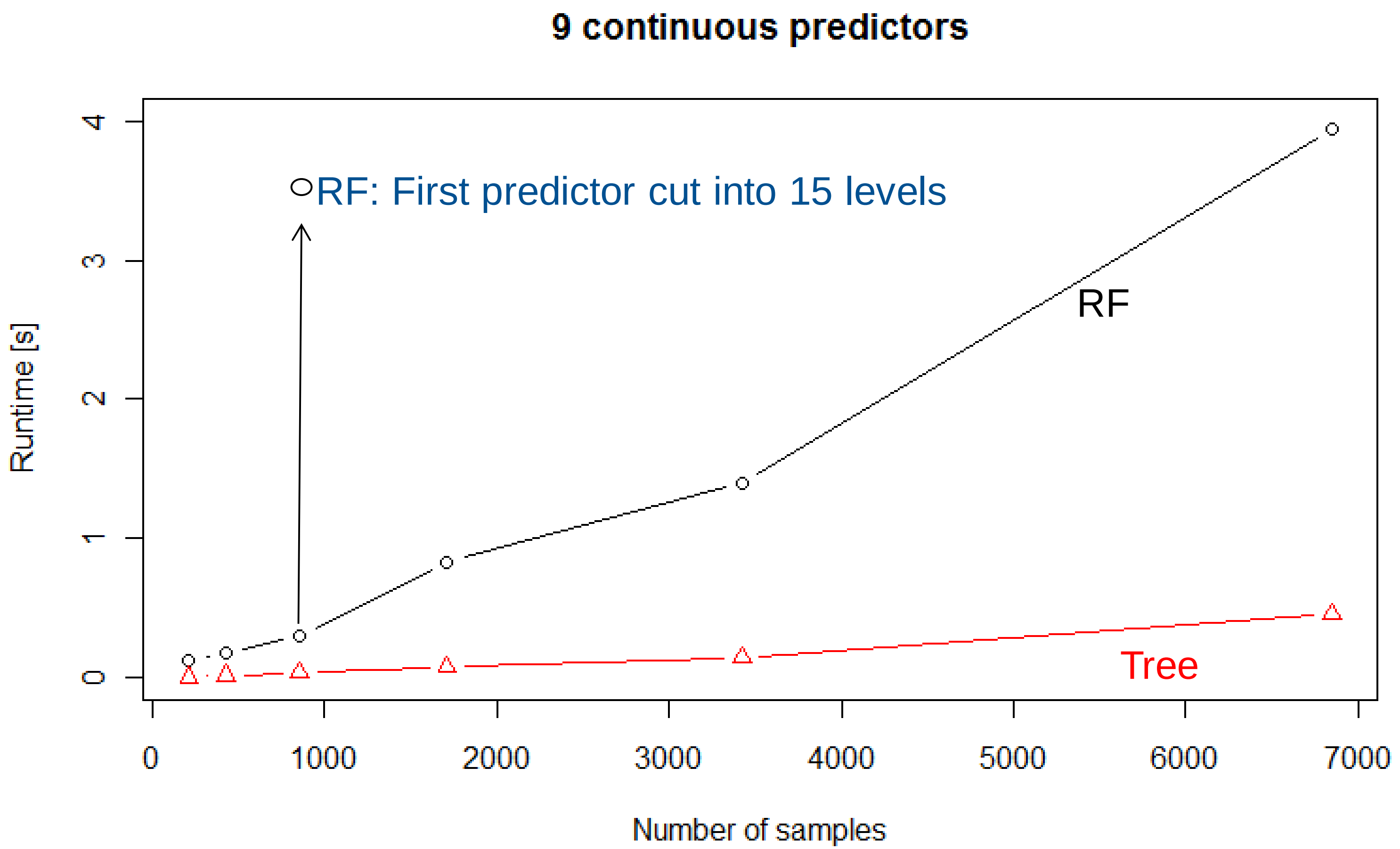
+ Easy to tune parameters

- Rather slow

- “Black Box”: Rather hard to get insights into decision rules

# Comparing runtime (just for illustration)

- Up to “thousands” of variables
- Problematic if there are categorical predictors with many levels (max: 32 levels)



# RF vs. LDA

+ Can model nonlinear class boundaries

+ OOB error “for free” (no CV needed)

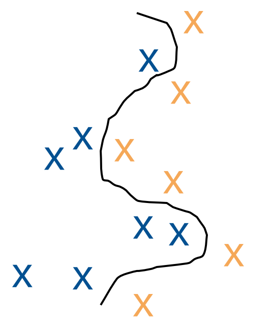
+ Works on continuous and categorical responses (regression / classification)

+ Gives variable importance

+ Very good performance

- “Black box”

- Slow



+ Very fast

+ Discriminants for visualizing group separation

+ Can read off decision rule

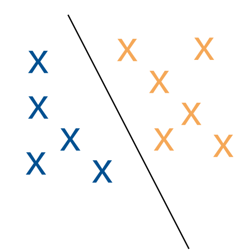
- Can model only linear class boundaries

- Mediocre performance

- No variable selection

- Only on categorical response

- Needs CV for estimating prediction error



# Concepts to know

- Idea of Random Forest and how it reduces the prediction variance of trees
- OOB error
- Variable Importance based on Permutation

# R functions to know

- Function “randomForest” and “varImpPlot” from package “randomForest”