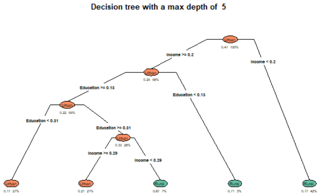
Machine Learning for Official Statistics & SDGs

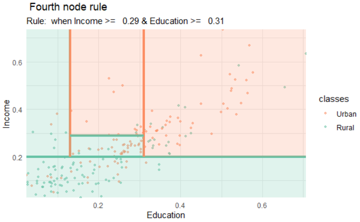
Random Forest

UNITED NATIONS
SIAP
Statistical Institute for Asia and the Pacific

[From Trees to Forest]

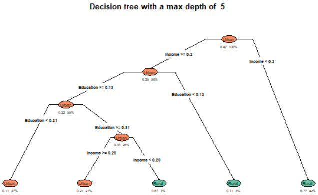
Trees are methods for classification or regression analysis.

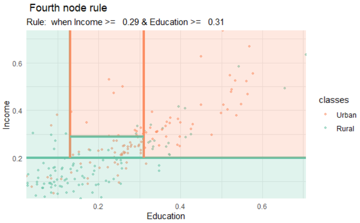




[From Trees to Forest]

Trees are methods for classification or regression analysis.

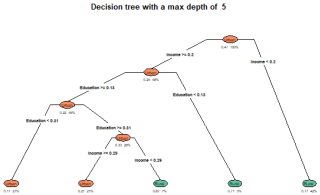


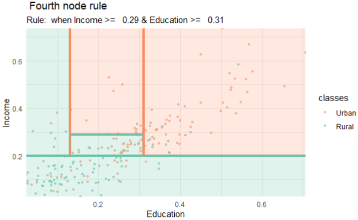


- Trees are based on recursive binary splits

[From Trees to Forest]

Trees are methods for classification or regression analysis.

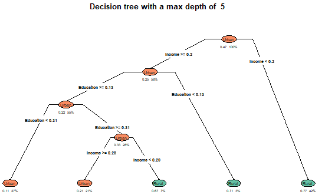


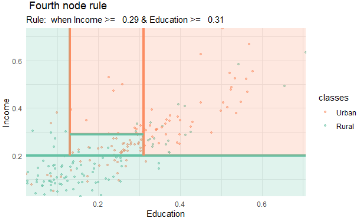


- Trees are based on recursive binary splits
- The structure is simple and corresponds to regions in the variable's space

[From Trees to Forest]

Trees are methods for classification or regression analysis.

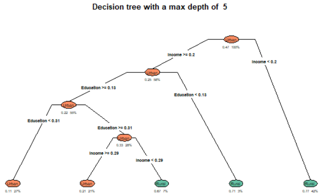


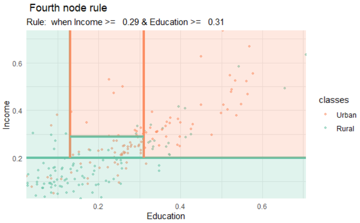


- Trees are based on recursive binary splits
- The structure is simple and corresponds to regions in the variable's space
- Each node is based on a the value of one variable and a threshold

[From Trees to Forest]

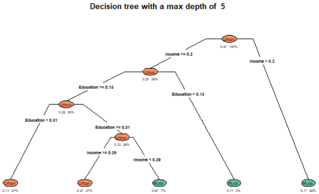
Trees are methods for classification or regression analysis.

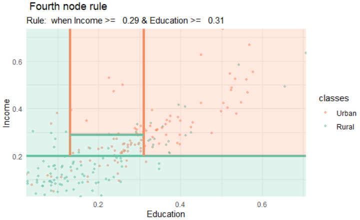




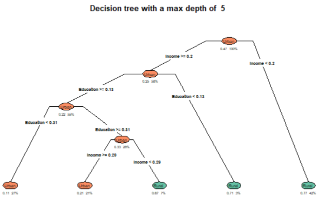
- Trees are based on recursive binary splits
- The structure is simple and corresponds to regions in the variable's space
- Each node is based on a the value of one variable and a threshold
- Trees can be very detailed and prone to **over fitting**

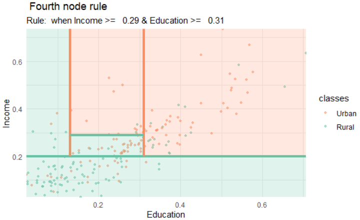
[Problems with trees]





[Problems with trees]

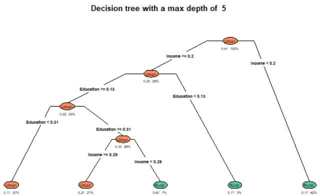


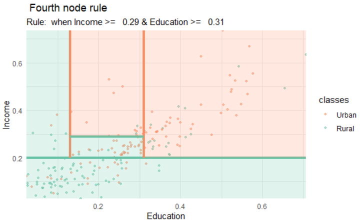


By structure:

- Trees are splitting the variable's space into "rectangles"

[Problems with trees]

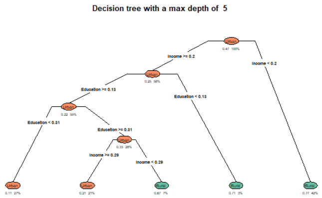


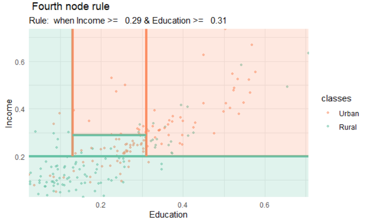


By structure:

- Trees are splitting the variable's space into "rectangles"
- Predictions using a tree may not be very accurate

[Problems with trees]

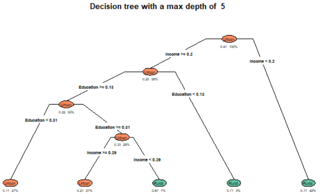


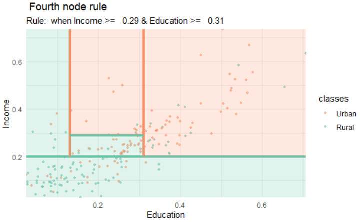


By structure:

- Trees are splitting the variable's space into "rectangles"
- Predictions using a tree may not be very accurate
- Trees are not robust to changes in the data

[Problems with trees]





By structure:

- Trees are splitting the variable's space into "rectangles"
- Predictions using a tree may not be very accurate
- Trees are not robust to changes in the data
- Trees are prone to overfiting

[Problems with trees]

The decision trees suffer from *high variance*:

[Problems with trees]

The decision trees suffer from *high variance*:

- Take a sample and build a tree

[Problems with trees]

The decision trees suffer from *high variance*:

- Take a sample and build a tree
- Divide the sample in two parts and build a tree on each part

[Problems with trees]

The decision trees suffer from *high variance*:

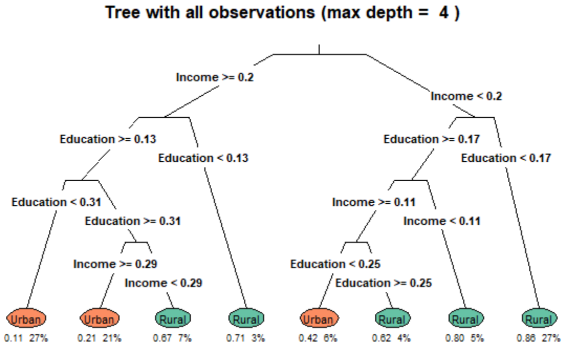
- Take a sample and build a tree
- Divide the sample in two parts and build a tree on each part

$\hookrightarrow$ The two trees will likely be very different

[Example on a simple tree]

[EXAMPLE ON A SIMPLE TREE]

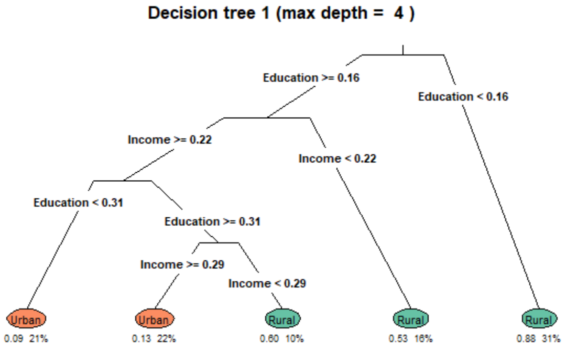
Which tree do you trust?



Tree using all observations

[Example on a simple tree]

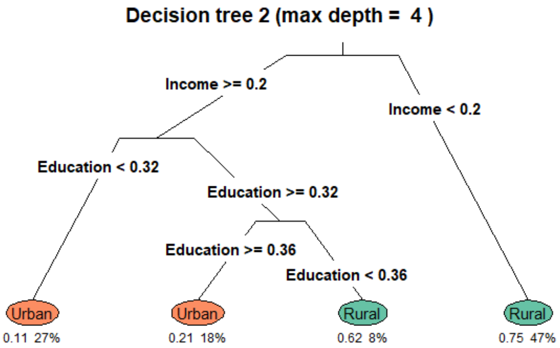
Which tree do you trust?



First tree with 50% of observations

[Example on a simple tree]

Which tree do you trust?

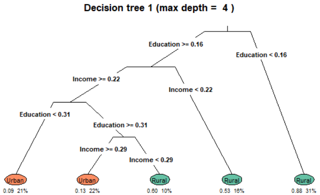


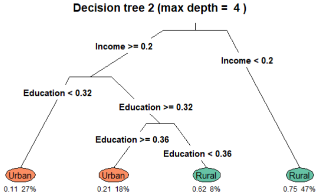
Second tree with 50% of observations

[Example on a simple tree]

Which tree do you trust?

Outcomes are different from one tree to another





$\hookrightarrow$ Instability of the outcome (or high variance)

[How to use trees?]

Alternative to simple tree models:

[How to use trees?]

Alternative to simple tree models:

- Aggregating the results of trees can help

[How to use trees?]

Alternative to simple tree models:

- Aggregating the results of trees can help

$\hookrightarrow$ "*Bagging*"

[How to use trees?]

Alternative to simple tree models:

- Aggregating the results of trees can help

$\hookrightarrow$ "*Bagging*"

- Some trees may be too similar (correlated) and have to be "decorrelated" using random draws of the variable used

[How to use trees?]

Alternative to simple tree models:

- Aggregating the results of trees can help

$\hookrightarrow$ "*Bagging*"

- Some trees may be too similar (correlated) and have to be "decorrelated" using random draws of the variable used

$\hookrightarrow$ "*Random forest*"

[How to use trees?]

Alternative to simple tree models:

- Aggregating the results of trees can help

↪ "*Bagging*"

- Some trees may be too similar (correlated) and have to be "decorrelated" using random draws of the variable used

↪ "*Random forest*"

- Trees can be constructed *sequentially*

[How to use trees?]

Alternative to simple tree models:

- Aggregating the results of trees can help

↪ "*Bagging*"

- Some trees may be too similar (correlated) and have to be "decorrelated" using random draws of the variable used

↪ "*Random forest*"

- Trees can be constructed *sequentially*

↪ "*Boosting*"

[Some simple mathematical tricks]

Aggregating the results of trees can help decrease the variance

[Some simple mathematical tricks]

Aggregating the results of trees can help decrease the variance

- Intuitively, if $U_i, \ldots U_B$ are iid with variance σ^2, then the average of these variables has a lower variance

[Some simple mathematical tricks]

Aggregating the results of trees can help decrease the variance

- Intuitively, if $U_i, \ldots U_B$ are iid with variance σ^2, then the average of these variables has a lower variance

$$
\begin{aligned}
Var\left(\frac{1}{B}\sum_{i=1}^{B} U_i\right) &= \frac{\sigma^2}{B} \\
&< \sigma^2
\end{aligned}
$$

[Some simple mathematical tricks]

Aggregating the results of trees can help decrease the variance

- Intuitively, if $U_i, \ldots U_B$ are iid with variance σ^2, then the average of these variables has a lower variance

$$
\begin{aligned}
Var\left(\frac{1}{B}\sum_{i=1}^{B} U_i\right) &= \frac{\sigma^2}{B} \\
&< \sigma^2
\end{aligned}
$$

- This is a simple principle that is used here

[Bagging]

Bagging stands for *Bootstrap Aggregating* uses a simple logic:

[Bagging]

Bagging stands for *Bootstrap Aggregating* uses a simple logic:

1. Draw B bootstrapped samples from the original sample

[Bagging]

Bagging stands for *Bootstrap Aggregating* uses a simple logic:

1. Draw B bootstrapped samples from the original sample
2. Construct B trees, one on each sample

[Bagging]

Bagging stands for *Bootstrap Aggregating* uses a simple logic:

1. Draw B bootstrapped samples from the original sample
2. Construct B trees, one on each sample
3. *Average* the result to get a prediction with a lower variance

[Bagging]

Bagging stands for *Bootstrap Aggregating* uses a simple logic:

1. Draw B bootstrapped samples from the original sample
2. Construct B trees, one on each sample
3. *Average* the result to get a prediction with a lower variance

[Bagging]

Bagging stands for *Bootstrap Aggregating* uses a simple logic:

1. Draw B bootstrapped samples from the original sample
2. Construct B trees, one on each sample
3. *Average* the result to get a prediction with a lower variance

In classification, we do not average but take the *majority vote* rule

[*Bagging*]

Bagging scheme:

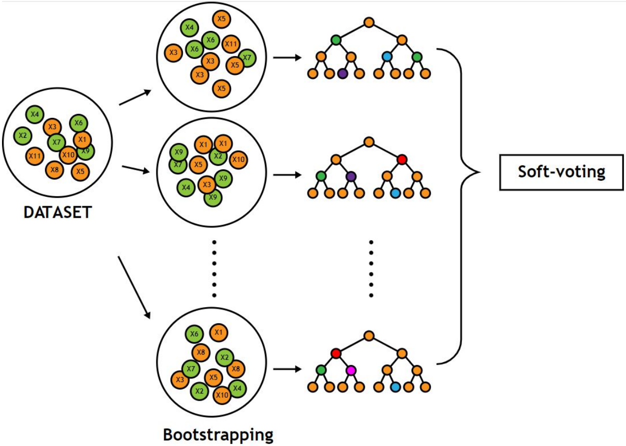


Image from igem.org

[The problem with *Bagging*]

Bootstrapped (*Bagging*) trees use the same set of variables

[The problem with *Bagging*]

Bootstrapped (*Bagging*) trees use the same set of variables

- Aggregating *correlated* trees will **not** decrease variance

[The problem with *Bagging*]

Bootstrapped (*Bagging*) trees use the same set of variables

- Aggregating *correlated* trees will **not** decrease variance
- Intuitively, if $U_i, \ldots U_B$ are with variance σ^2, **and cross-correlation** ρ, then the average has a variance:

$$Var\left(\frac{1}{B}\sum_{i=1}^{B} U_i\right) \quad = \quad \rho \cdot \sigma^2 + \frac{\sigma^2}{B}$$

[THE PROBLEM WITH *Bagging*]

Bootstrapped (*Bagging*) trees use the same set of variables

- Aggregating *correlated* trees will **not** decrease variance
- Intuitively, if $U_i, \ldots U_B$ are with variance σ^2, **and cross-correlation** ρ, then the average has a variance:

$$Var\left(\frac{1}{B}\sum_{i=1}^{B} U_i\right) = \rho \cdot \sigma^2 + \frac{\sigma^2}{B}$$

- This could be "high" if ρ is "high"

[The problem with *Bagging*]

Bootstrapped (*Bagging*) trees use the same set of variables

- Aggregating *correlated* trees will **not** decrease variance
- Intuitively, if $U_i, \dots U_B$ are with variance σ^2, **and cross-correlation** ρ, then the average has a variance:

$$Var\left(\frac{1}{B}\sum_{i=1}^{B} U_i\right) = \rho \cdot \sigma^2 + \frac{\sigma^2}{B}$$

- This could be "high" if ρ is "high"

$\hookrightarrow$ *Random forest* use a *decorrelation* algorithm by adding ***randomness*** in the set of variables used

[Random Forest]

Random forest use two mechanisms:

[Random Forest]

Random forest use two mechanisms:

- Bootstrap aggregating (*bagging*)

[Random Forest]

Random forest use two mechanisms:

- Bootstrap aggregating (*bagging*)

$\hookrightarrow$ Construct B trees, on B bootstrapped samples

[Random Forest]

Random forest use two mechanisms:

- Bootstrap aggregating (*bagging*)

$\hookrightarrow$ Construct B trees, on B bootstrapped samples

- **Random** selection of variables (*Feature sampling*)

[Random Forest]

Random forest use two mechanisms:

- ► Bootstrap aggregating (*bagging*)
- $\hookrightarrow$ Construct B trees, on B bootstrapped samples
- ► **Random** selection of variables (*Feature sampling*)
- $\hookrightarrow$ At each node, randomly select only **m** variables

[Random Forest]

Random forest use two mechanisms:

- Bootstrap aggregating (*bagging*)

$\hookrightarrow$ Construct B trees, on B bootstrapped samples

- **Random** selection of variables (*Feature sampling*)

$\hookrightarrow$ At each node, randomly select only **m** variables

- The resulting predictor will have a lower variance

$$Var_{Random\ forest} = \rho \cdot \sigma^2 + \frac{\sigma^2}{B}$$

[Random Forest]

Random forest use two mechanisms:

- Bootstrap aggregating (*bagging*)

$\hookrightarrow$ Construct B trees, on B bootstrapped samples

- **Random** selection of variables (*Feature sampling*)

$\hookrightarrow$ At each node, randomly select only **m** variables

- The resulting predictor will have a lower variance

$$Var_{Random\ forest} = \rho \cdot \sigma^2 + \frac{\sigma^2}{B}$$

Common practice (p = *nb of variables*):

[Random Forest]

Random forest use two mechanisms:

- Bootstrap aggregating (*bagging*)

$\hookrightarrow$ Construct B trees, on B bootstrapped samples

- **Random** selection of variables (*Feature sampling*)

$\hookrightarrow$ At each node, randomly select only **m** variables

- The resulting predictor will have a lower variance

$$Var_{Random\ forest} = \rho \cdot \sigma^2 + \frac{\sigma^2}{B}$$

Common practice (p = *nb of variables*):

- $m \approx \sqrt{p}$ in classification

[Random Forest]

Random forest use two mechanisms:

- Bootstrap aggregating (*bagging*)

↪ Construct B trees, on B bootstrapped samples

- **Random** selection of variables (*Feature sampling*)

↪ At each node, randomly select only **m** variables

- The resulting predictor will have a lower variance

$$Var_{Random\ forest} = \rho \cdot \sigma^2 + \frac{\sigma^2}{B}$$

Common practice (p = *nb of variables*):

- $m \approx \sqrt{p}$ in classification
- $m \approx p/3$ in regression

[Random Forest]

Random forest scheme:

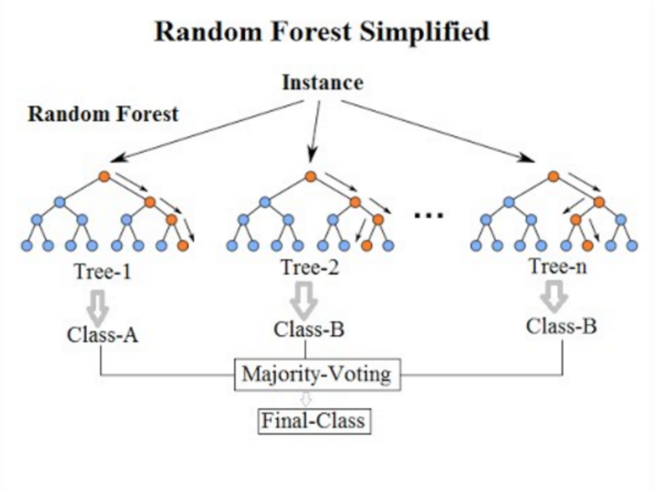


Image from *Venkata Jagannath* (wikimedia.org)

[Optimizing Random Forest]

Random forest are sensitive to several *hyperparameters*

[Optimizing Random Forest]

Random forest are sensitive to several *hyperparameters*

- All the parameters used to build a *tree*

[Optimizing Random Forest]

Random forest are sensitive to several *hyperparameters*

- All the parameters used to build a *tree*
  - *Purity* criterion (Gini *vs* Entropy)

[Optimizing Random Forest]

Random forest are sensitive to several *hyperparameters*

- All the parameters used to build a *tree*
  - *Purity* criterion (Gini *vs* Entropy)
  - Tree *Depth*

[Optimizing Random Forest]

Random forest are sensitive to several *hyperparameters*

- All the parameters used to build a *tree*
  - *Purity* criterion (Gini *vs* Entropy)
  - Tree *Depth*
  - *Complexity* parameter *Cp*

[Optimizing Random Forest]

Random forest are sensitive to several *hyperparameters*

- All the parameters used to build a *tree*
  - *Purity* criterion (Gini *vs* Entropy)
  - Tree *Depth*
  - *Complexity* parameter *Cp*
  - . . .

[Optimizing Random Forest]

Random forest are sensitive to several *hyperparameters*

- All the parameters used to build a *tree*
  - *Purity* criterion (Gini *vs* Entropy)
  - Tree *Depth*
  - *Complexity* parameter *Cp*
  - . . .
- Parameters used to construct the *Forest*

[Optimizing Random Forest]

Random forest are sensitive to several *hyperparameters*

- All the parameters used to build a *tree*
  - *Purity* criterion (Gini *vs* Entropy)
  - Tree *Depth*
  - *Complexity* parameter *Cp*
  - ...
- Parameters used to construct the *Forest*
  - Number of variables m used in each node

[Optimizing Random Forest]

Random forest are sensitive to several *hyperparameters*

- All the parameters used to build a *tree*
  - *Purity* criterion (Gini *vs* Entropy)
  - Tree *Depth*
  - *Complexity* parameter *Cp*
  - ...
- Parameters used to construct the *Forest*
  - Number of variables m used in each node
  - Number of trees

[Optimizing Random Forest]

Random forest are sensitive to several *hyperparameters*

- All the parameters used to build a *tree*
  - *Purity* criterion (Gini *vs* Entropy)
  - Tree *Depth*
  - *Complexity* parameter *Cp*
  - ...
- Parameters used to construct the *Forest*
  - Number of variables m used in each node
  - Number of trees
  - Minimum number of observations per leaves

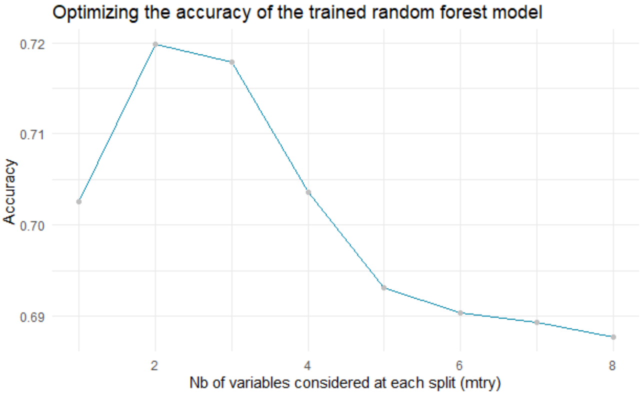
[Optimizing Random Forest]

Random forest are sensitive to several *hyperparameters*

- All the parameters used to build a *tree*
  - *Purity* criterion (Gini *vs* Entropy)
  - Tree *Depth*
  - *Complexity* parameter *Cp*
  - ...
- Parameters used to construct the *Forest*
  - Number of variables m used in each node
  - Number of trees
  - Minimum number of observations per leaves
  - ...

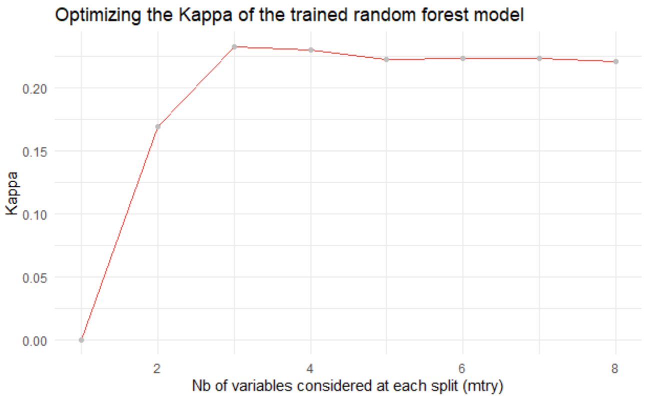
[Random Forest on an Example]

[Random Forest on an Example]



The impact of the number of variables (*m*) on accuracy

[Random Forest on an Example]



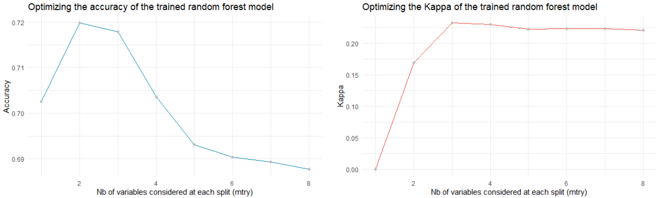
The impact of the number of variables (***m***) on *kappa*

[Random Forest on an Example]

The impact of the number of variables (m):

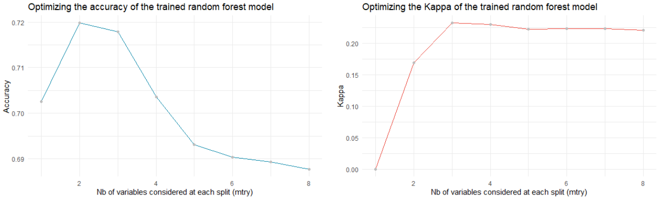
[Random Forest on an Example]

The impact of the number of variables (m):



[Random Forest on an Example]

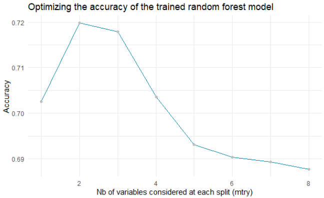
The impact of the number of variables (m):

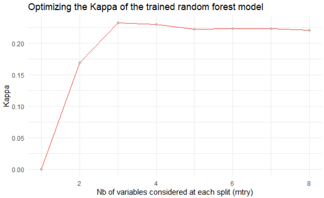


Why are the curves decreasing after a threshold?

[RANDOM FOREST ON AN EXAMPLE]

The impact of the number of variables (m):



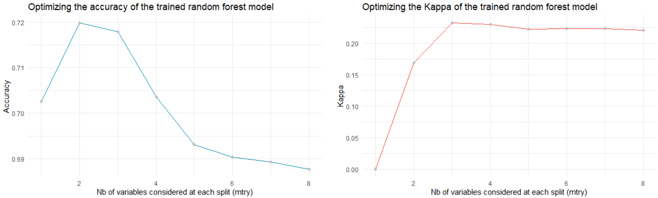


Why are the curves decreasing after a threshold?

- The variance depends on $\rho \cdot \sigma^2$

[Random Forest on an Example]

The impact of the number of variables (***m***):



Why are the curves decreasing after a threshold?

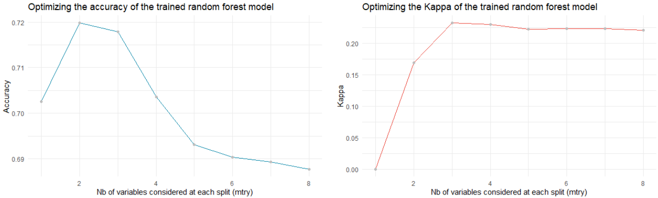
- The variance depends on $\rho \cdot \sigma^2$

$\hookrightarrow$ *Trees using similar variables are very similar (same predictions)*

$\nearrow m \implies \rho \nearrow$

[Random Forest on an Example]

The impact of the number of variables (m):



Why are the curves decreasing after a threshold?

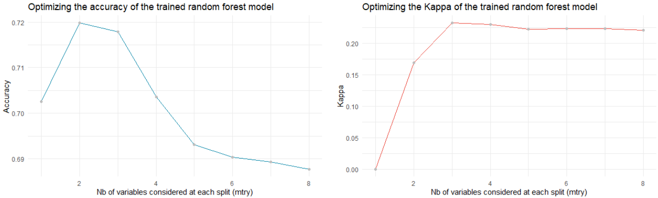
- The variance depends on $\rho \cdot \sigma^2$

$\hookrightarrow$ *Trees using similar variables are very similar (same predictions)*
$\nearrow m \implies \rho \nearrow$

- Optimum for **3** (out of 7) regressors only at each node

[Random Forest on an Example]

The impact of the number of variables ($\boldsymbol{m}$):



Why are the curves decreasing after a threshold?

- The variance depends on $\rho \cdot \sigma^2$

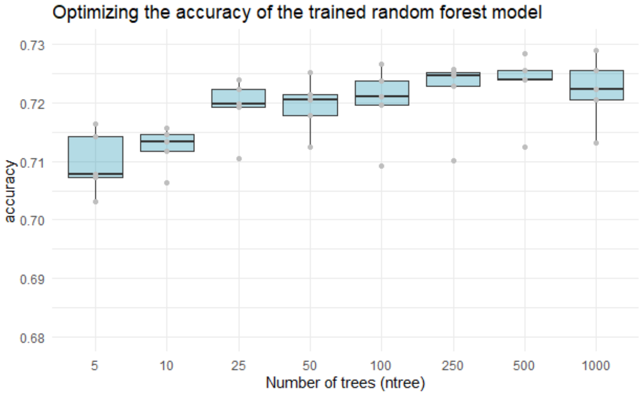
$\hookrightarrow$ *Trees using similar variables are very similar (same predictions)*
$\nearrow m \Longrightarrow \rho \nearrow$

- Optimum for **3** (out of 7) regressors only at each node

$\hookrightarrow$ Rule of thumb: $m \approx \sqrt{p}$

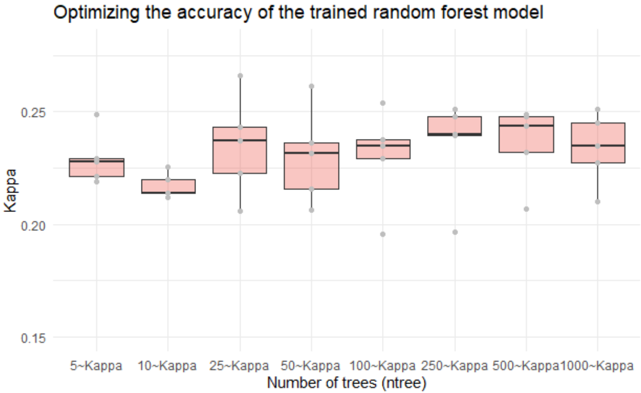
[Random Forest on an Example]

[Random Forest on an Example]



The impact of the number of trees on accuracy

[Random Forest on an Example]



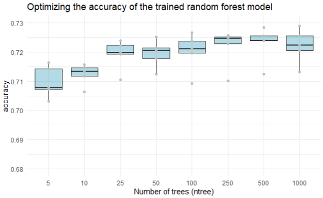
The impact of the number of trees on *kappa*

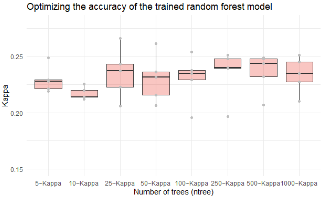
[Random Forest on an Example]

The impact of the number of trees:

[Random Forest on an Example]

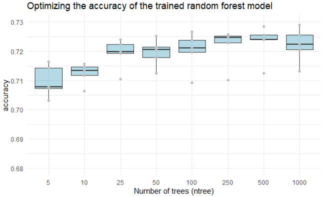
The impact of the number of trees:

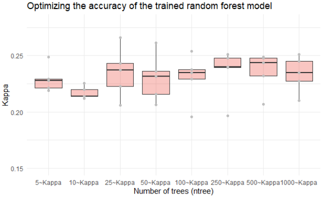




[Random Forest on an Example]

The impact of the number of trees:

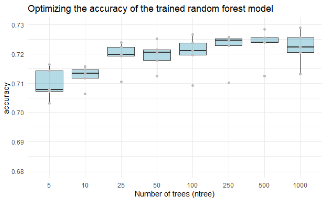


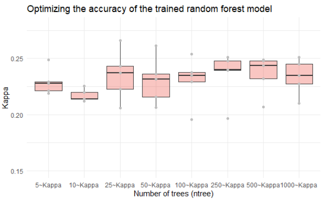


- No clear optimum $\hookrightarrow$ Moderate number of trees is OK

[Random Forest on an Example]

The impact of the number of trees:



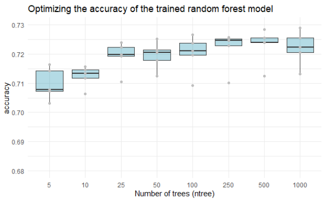


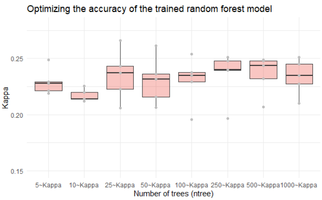
- No clear optimum $\hookrightarrow$ Moderate number of trees is OK
- The variance depends on $\frac{\sigma^2}{B}$

  $\nearrow B \Longrightarrow \frac{\sigma^2}{B} \searrow$

[Random Forest on an Example]

The impact of the number of trees:





- No clear optimum $\hookrightarrow$ Moderate number of trees is OK
- The variance depends on $\frac{\sigma^2}{B}$

  $\nearrow B \Longrightarrow \frac{\sigma^2}{B} \searrow$

$\hookrightarrow$ More tree reduces variance (up-to a certain point)

[Improvements]

Trees can also grow differently:

[Improvements]

Trees can also grow differently:

- Random Forest grow *independently*

[Improvements]

Trees can also grow differently:

- Random Forest grow *independently*
- "Vote" or average of the outcome from leaves

[Improvements]

Trees can also grow differently:

- Random Forest grow *independently*
- "Vote" or average of the outcome from leaves
- Trees can be constructed *sequentially*

[Improvements]

Trees can also grow differently:

- Random Forest grow *independently*
- "Vote" or average of the outcome from leaves
- Trees can be constructed *sequentially*

$\hookrightarrow$ "*Boosting*"

[Boosting]

Boosting helps construct trees sequentially

[BOOSTING]

Boosting helps construct trees sequentially

- Based on weak learners

[Boosting]

Boosting helps construct trees sequentially

- Based on weak learners

$\hookrightarrow$ Very simple trees grow on each other's *"mistakes"*

[BOOSTING]

Boosting helps construct trees sequentially

- Based on weak learners

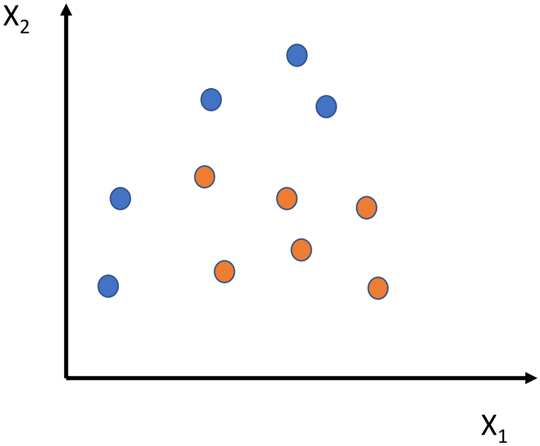
$\hookrightarrow$ Very simple trees grow on each other's *"mistakes"*

- *"Mistakes"* at tree nb t are overweighed for tree nb $t + 1$

[Boosting]

[Boosting]

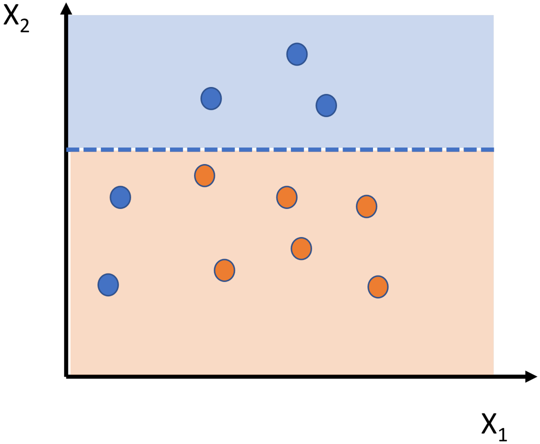
Boosting a tree (iterative process)



Two classes of observations: Orange and blue

[Boosting]

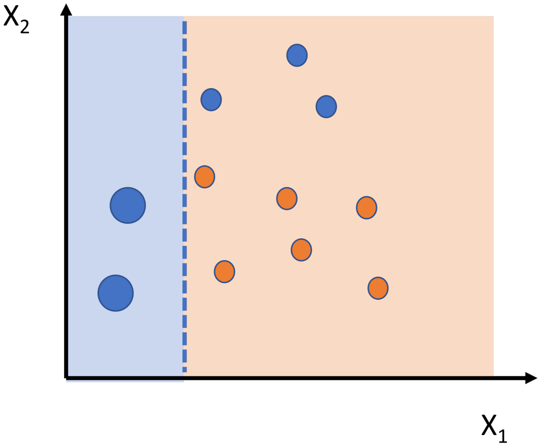
Boosting a tree (iterative process)



First weak learner

[Boosting]

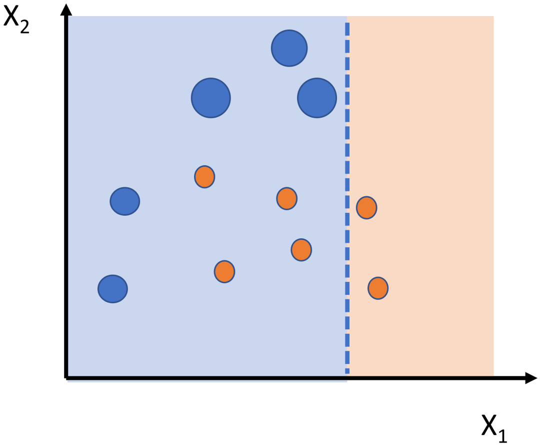
Boosting a tree (iterative process)



Second weak learner. Misclassification overweighed

[Boosting]

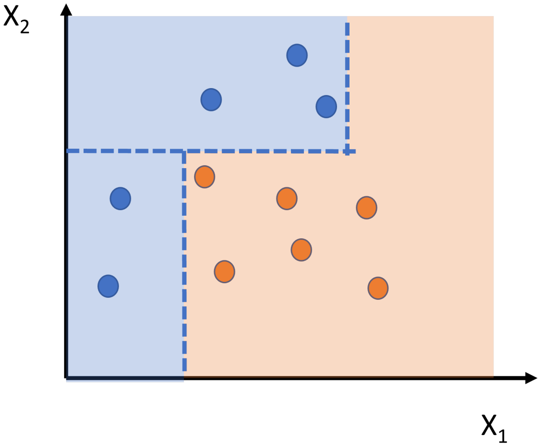
Boosting a tree (iterative process)



Third weak learner. New misclassification overweighed

[Boosting]

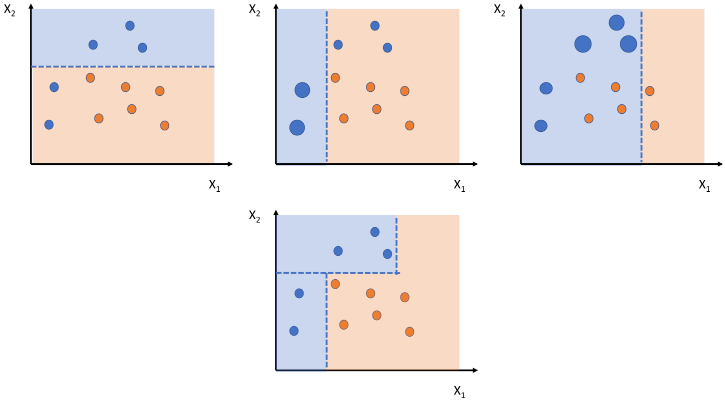
Boosting a tree (iterative process)



Combining weak learners.

[Boosting]

Boosting a tree (iterative process)



Combining weak learners. Classification tree completed

[Boosting]

Boosting has several features

[Boosting]

Boosting has several features

- Extremely powerful

[Boosting]

Boosting has several features

- Extremely powerful
- Based on simple trees (weak learners)

[Boosting]

Boosting has several features

- Extremely powerful
- Based on simple trees (weak learners)
- The "weight" placed on "mistakes" is important

[Boosting]

Boosting has several features

- Extremely powerful
- Based on simple trees (weak learners)
- The "weight" placed on "mistakes" is important

$\hookrightarrow$ Several choices *gradient boosting*

[Boosting]

Boosting has several features

- Extremely powerful
- Based on simple trees (weak learners)
- The "weight" placed on "mistakes" is important

$\hookrightarrow$ Several choices *gradient boosting*

- Sequential procedure (no parallel computation)

[Boosting]

Boosting has several features

- Extremely powerful
- Based on simple trees (weak learners)
- The "weight" placed on "mistakes" is important

↪ Several choices *gradient boosting*

- Sequential procedure (no parallel computation)

↪ *Xgboosting*

[Takeaways]

- Random forest are simple and easy to interpret

[Takeaways]

- Random forest are simple and easy to interpret
- *Random forest* use two mechanisms:
  bagging + (random) feature selection

[Takeaways]

- Random forest are simple and easy to interpret
- *Random forest* use two mechanisms:
  bagging + (random) feature selection
  - Bootstrap aggregating (*bagging*)

[Takeaways]

- Random forest are simple and easy to interpret
- *Random forest* use two mechanisms:
  bagging + (random) feature selection
  - Bootstrap aggregating (*bagging*)

  $\hookrightarrow$ Construct B trees, on B bootstrapped samples

[TAKEAWAYS]

- Random forest are simple and easy to interpret
- *Random forest* use two mechanisms:
  bagging + (random) feature selection
  - Bootstrap aggregating (*bagging*)

  ↪ Construct B trees, on B bootstrapped samples
  - **Random** selection of variables (*Feature sampling*)

[TAKEAWAYS]

- Random forest are simple and easy to interpret
- *Random forest* use two mechanisms:
  bagging + (random) feature selection
  - Bootstrap aggregating (*bagging*)
  - ↪ Construct B trees, on B bootstrapped samples
  - **Random** selection of variables (*Feature sampling*)
  - ↪ At each node, randomly select only **m** variables

[Takeaways]

- Random forest are simple and easy to interpret
- *Random forest* use two mechanisms: *bagging* + (random) feature selection
  - Bootstrap aggregating (*bagging*)
  - ↪ Construct B trees, on B bootstrapped samples
  - **Random** selection of variables (*Feature sampling*)
  - ↪ At each node, randomly select only **m** variables
  - The resulting predictor will have a lower variance

$$Var_{Random\ forest} = \rho \cdot \sigma^2 + \frac{\sigma^2}{B}$$

[TAKEAWAYS]

- Random forest are simple and easy to interpret
- *Random forest* use two mechanisms:
  bagging + (random) feature selection
  - Bootstrap aggregating (*bagging*)

  ↪ Construct B trees, on B bootstrapped samples
  - **Random** selection of variables (*Feature sampling*)

  ↪ At each node, randomly select only **m** variables
  - The resulting predictor will have a lower variance

$$Var_{Random\ forest} = \rho \cdot \sigma^2 + \frac{\sigma^2}{B}$$

- Many variations in r*andom forest* exist (*boosting*, *gradient boosting*, *Xgboosting*)

[TAKEAWAYS]

- Random forest are simple and easy to interpret
- *Random forest* use two mechanisms:
  bagging + (random) feature selection
  - Bootstrap aggregating (*bagging*)

    ↪ Construct B trees, on B bootstrapped samples
  - **Random** selection of variables (*Feature sampling*)

    ↪ At each node, randomly select only **m** variables
  - The resulting predictor will have a lower variance

$$Var_{Random\ forest} = \rho \cdot \sigma^2 + \frac{\sigma^2}{B}$$

- Many variations in r*andom forest* exist (*boosting*, *gradient boosting*, *Xgboosting*)
- Several parameters to adjust: Nb of trees, nb of variables in each node, minimum number of obs. in leaves/nodes, tree complexity, stopping rules, etc.

[Takeaways]

- Random forest are simple and easy to interpret
- *Random forest* use two mechanisms: *bagging* + (random) feature selection
  - Bootstrap aggregating (*bagging*)

  ↪ Construct B trees, on B bootstrapped samples
  - **Random** selection of variables (*Feature sampling*)

  ↪ At each node, randomly select only **m** variables
  - The resulting predictor will have a lower variance

$$Var_{Random\ forest} = \rho \cdot \sigma^2 + \frac{\sigma^2}{B}$$

- Many variations in r*andom forest* exist (*boosting*, *gradient boosting*, *Xgboosting*)
- Several parameters to adjust: Nb of trees, nb of variables in each node, minimum number of obs. in leaves/nodes, tree complexity, stopping rules, etc.
- Implemented in many software!