Random Forest for Classification Problems

Raphael, Arkadiusz and Burak

Uni Bonn

January 15, 2020

Overview

Decision Tree: Example

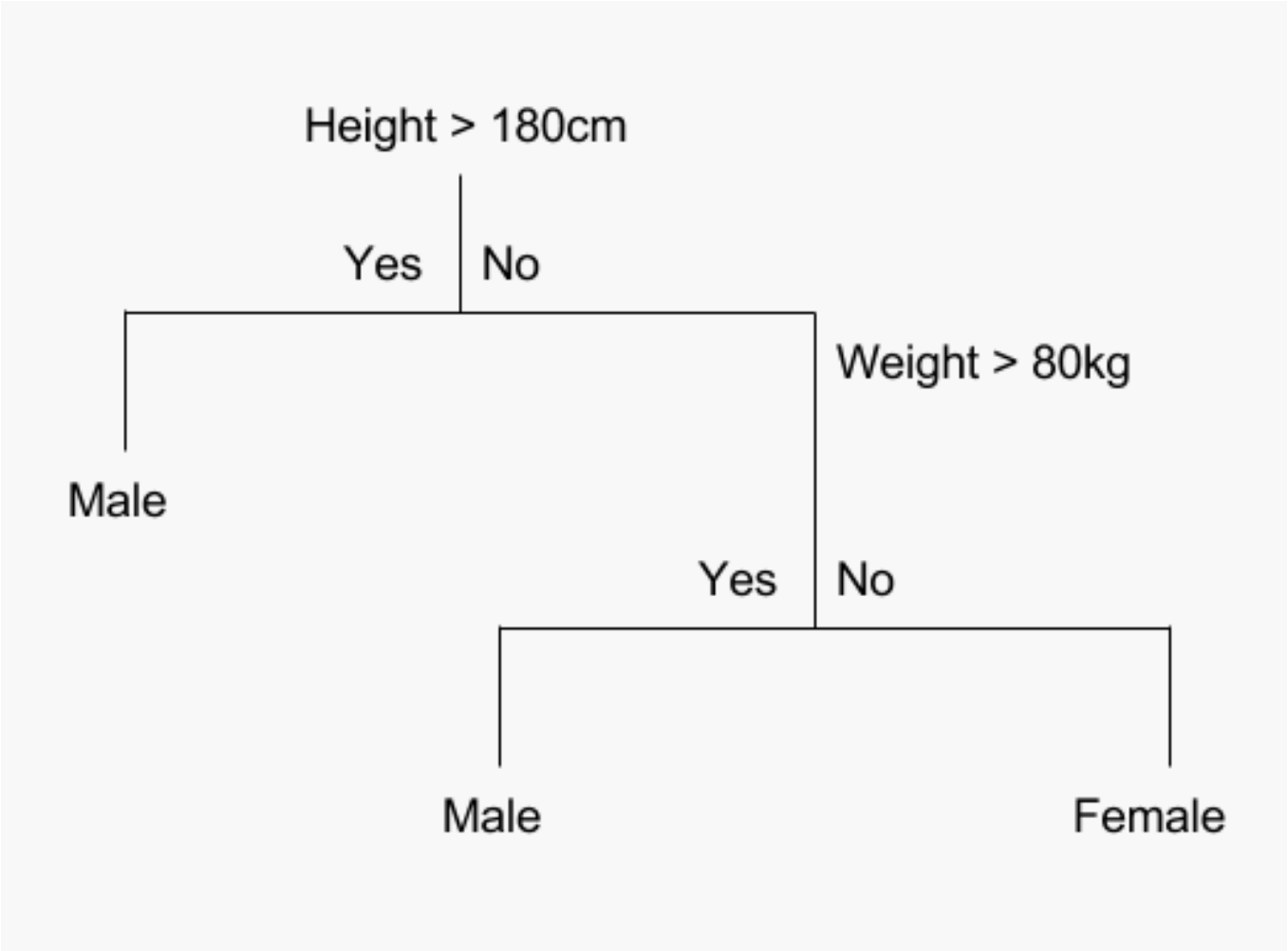


Figure: Source:[3]

Decision Tree: Tree Building Process

A tree is grown starting from the root node by repeatedly using the following steps on each node (also called binary splitting) [1]:

(i) **Find best split s for each feature X_m:** For each feature X_m, there exist $K-1$-many potiential splits whereas K is the number of different values for the respective feature. Evaluate each value $X_{m,i}$ at the current node t as a candidate split point (for $x \in X_m$, if $x \leq X_{m,i} = s$, then x goes to left child node t_L else to right child node t_R). The best split point is the one that maximize the splitting criterion $\Delta i(s,t)$ the most when the node is split according to it. The different splitting criteria will be covered in the next chapter.

(ii) **Find the nodes best split:** Among the best splits for each feature from Step (i) find the one s^*, which maximizes the splitting criterion $\Delta i(s,t)$.

(iii) **Satisfy stopping criterion:** Split the node t using best node split s^* from Step (ii) and repeat from Step (i) until stopping criterion is satisfied.

Decision Tree: Purity Measures

Gini Measure

$$i(t) = \sum_{c \in C} p(c|t)(1 - p(c|t)) = 1 - \sum_{c \in C} p_c^2 \tag{1}$$

Information Entropy

$$i(t) = \sum_{c \in C} p(c|t) log(p(c|t)) \tag{2}$$

Expected Generalization Error

Let $D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ and $y = f(x) + \epsilon$.
The decomposition of a model's expected generalization error is

$$\boldsymbol{Err}(f(x)) = Noise(x) + [Bias(\hat{f}(x))]^2 + Var(\hat{f}(x)) \tag{3}$$

Noise is irreducible and independent of the model.
There is a trade-off between $bias^2$ and variance.
Adjusting parameters to decrease variance increases $bias^2$ and vice versa.

Bias-Variance Trade-off

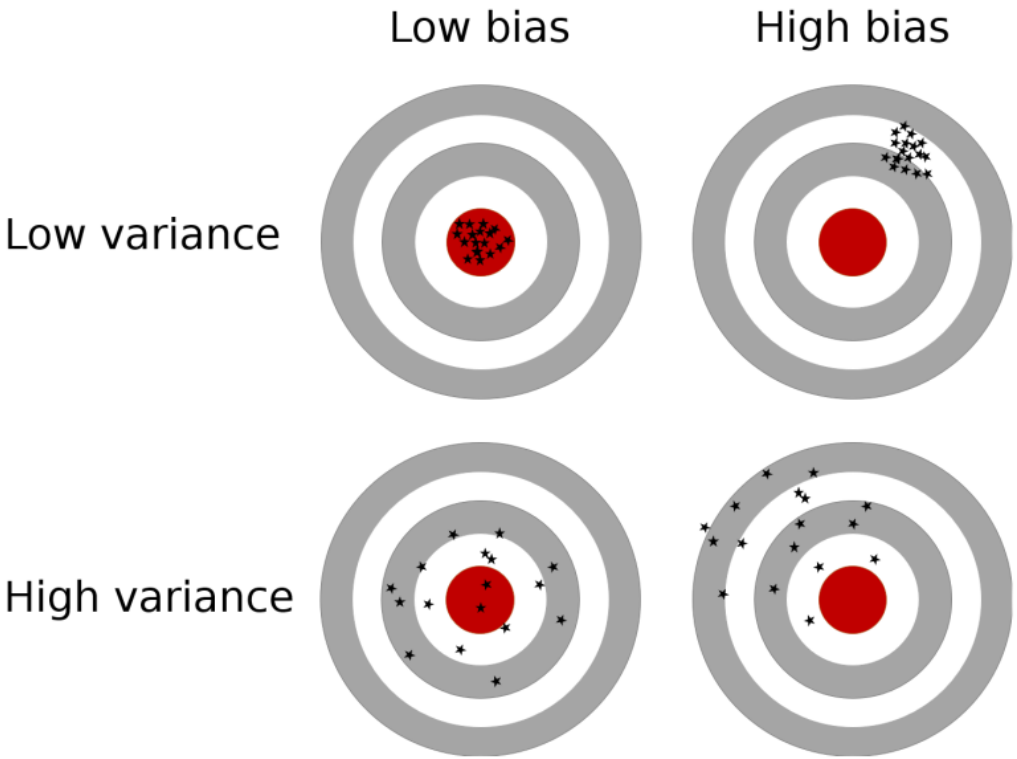
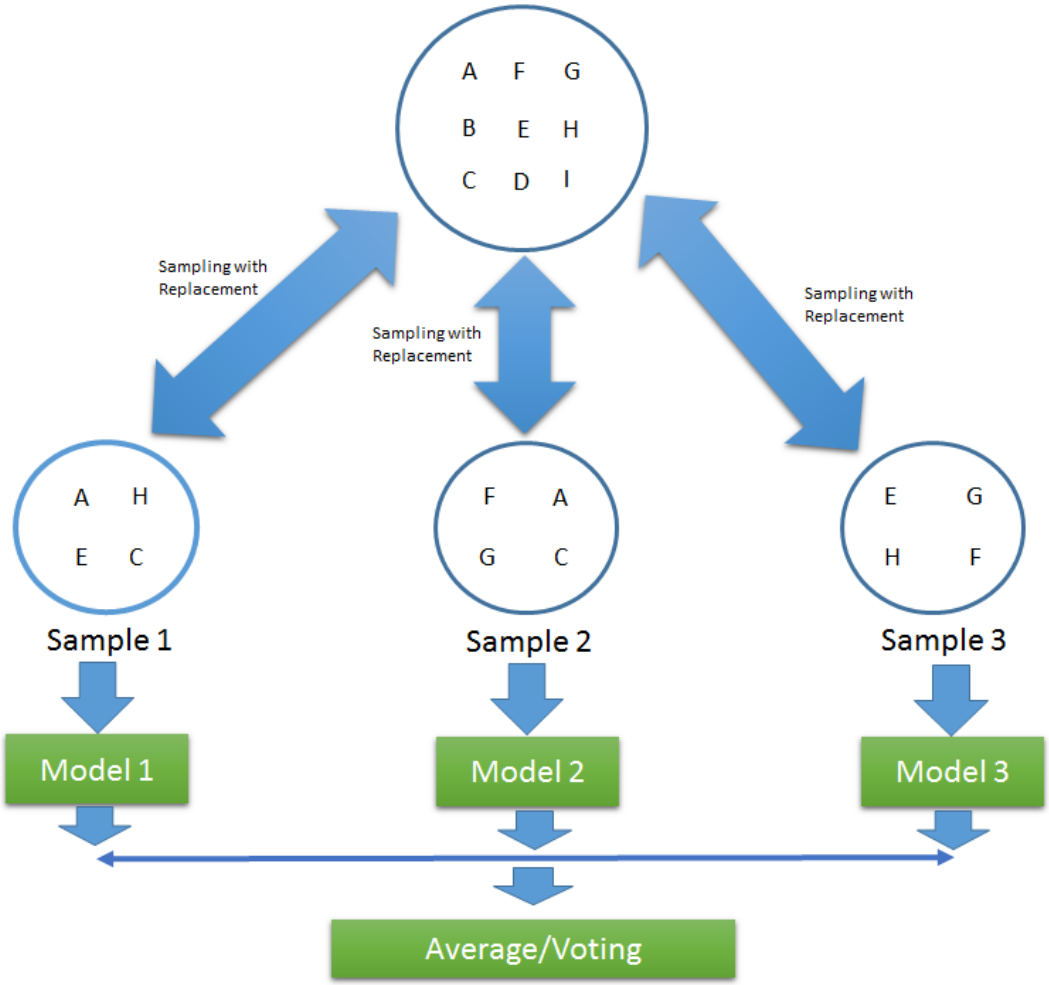


Figure: Illustration of bias-variance trade-off [5]

Decision trees generally have low bias and high variance [2].

Bagging

1. created for methods with high variance
2. reduces variance and gives better predictions
3. improvement of bagging: Random Forest



Random Forest

An ensemble of randomly trained decision trees, so in other words random forest was defined by L. Breiman:

Theorem

A random forest is a classifier consisting of a collection of tree-structured classifiers $\hat{T}_{\theta_b}(\mathbf{x}), b = 1, ..., B$ *where the* θ_b *are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input* $\mathbf{x}$.

Random Forest is an extension and improvement over bagging:

1. Like in bagging, multiple decision trees are built
2. Improvement: an injection of randomness is made

Random Forest: randomness in the model

Two key concepts that makes decision forest "random" are:

1. Random sampling of training data points when building trees
2. Random subsets of features considered when splitting nodes. Recommended number of variables:
   a. For classification: $\lfloor \sqrt{n} \rfloor$
   b. For regression: $\lfloor \frac{n}{3} \rfloor$

Random Forest: algorithm

Algorithm 1: Random Forest for Regression or Classification

1. For $b = 1$ to B:
   a. Draw a bootstrap sample θ_b of size N from the training data.
   b. Grow the Random Forest tree T_{θ_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 n 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 $\{T_{\theta_b}\}_1^B$

Mathematical Explanation

Let $D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ and $T_{D,\theta}$ is fully grown decision tree trained on set D with using parameters θ. Random Forest estimate of an observation x^* is

Majority Voting

$$\boldsymbol{RF}_{D,\theta_1,\theta_2,...,\theta_B}(x^*) = \underset{c \in Y}{argmax} \sum_{b=1}^{B} 1(\hat{T}_b(x^*) = c) \tag{4}$$

Soft Voting

$$\boldsymbol{RF}_{D,\theta_1,\theta_2,...,\theta_B}(x^*) = \underset{c \in Y}{argmax} \frac{1}{B} \sum_{b=1}^{B} \hat{p}_{D,\theta_b}(Y = c | X = x^*) \tag{5}$$

where $\hat{p}_{D,\theta_b}(Y = c | X = x^*)$ is the probability estimates of a tree.

Majority Voting

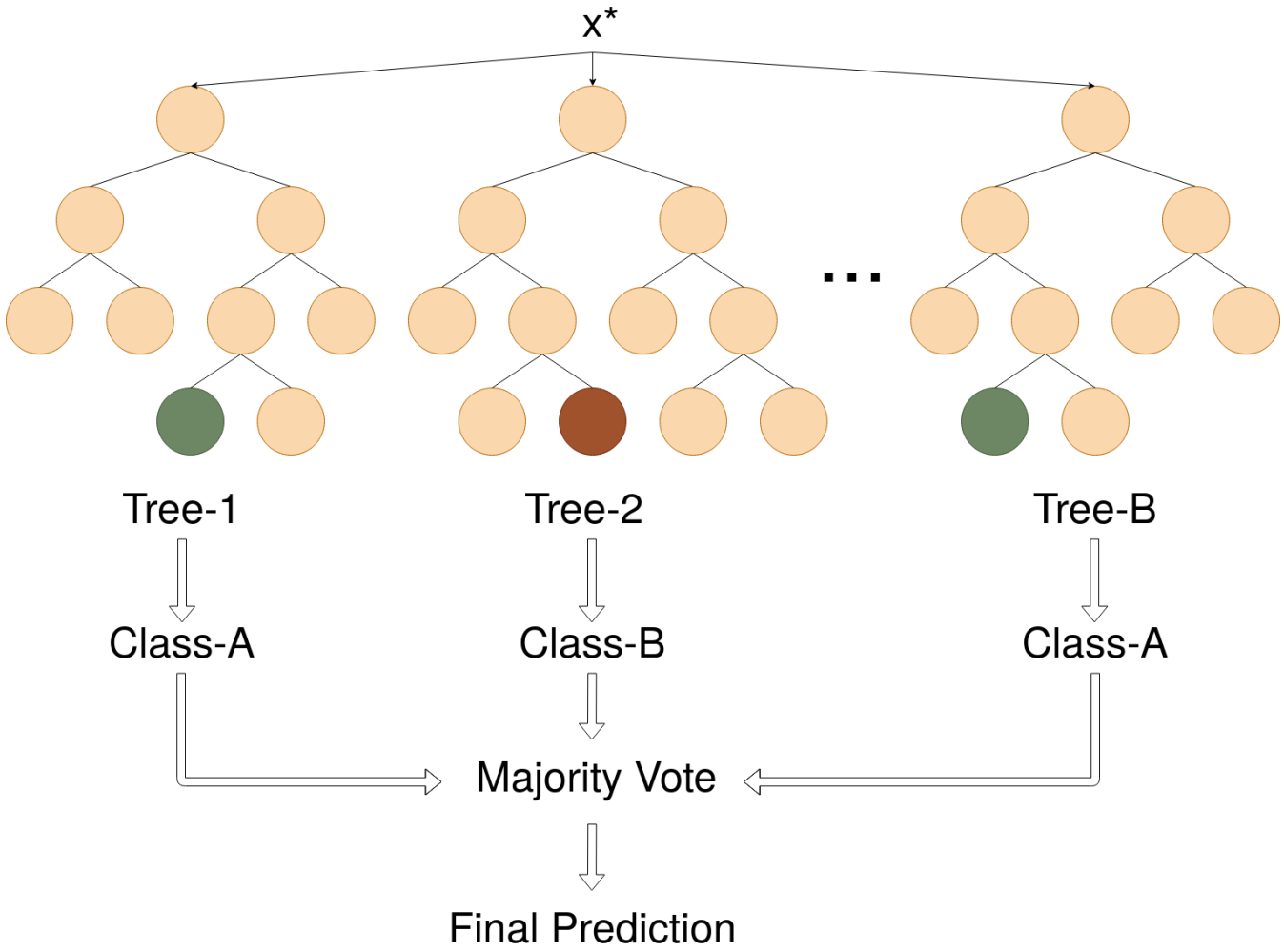


Figure: Majority Voting Illustration

Soft Voting

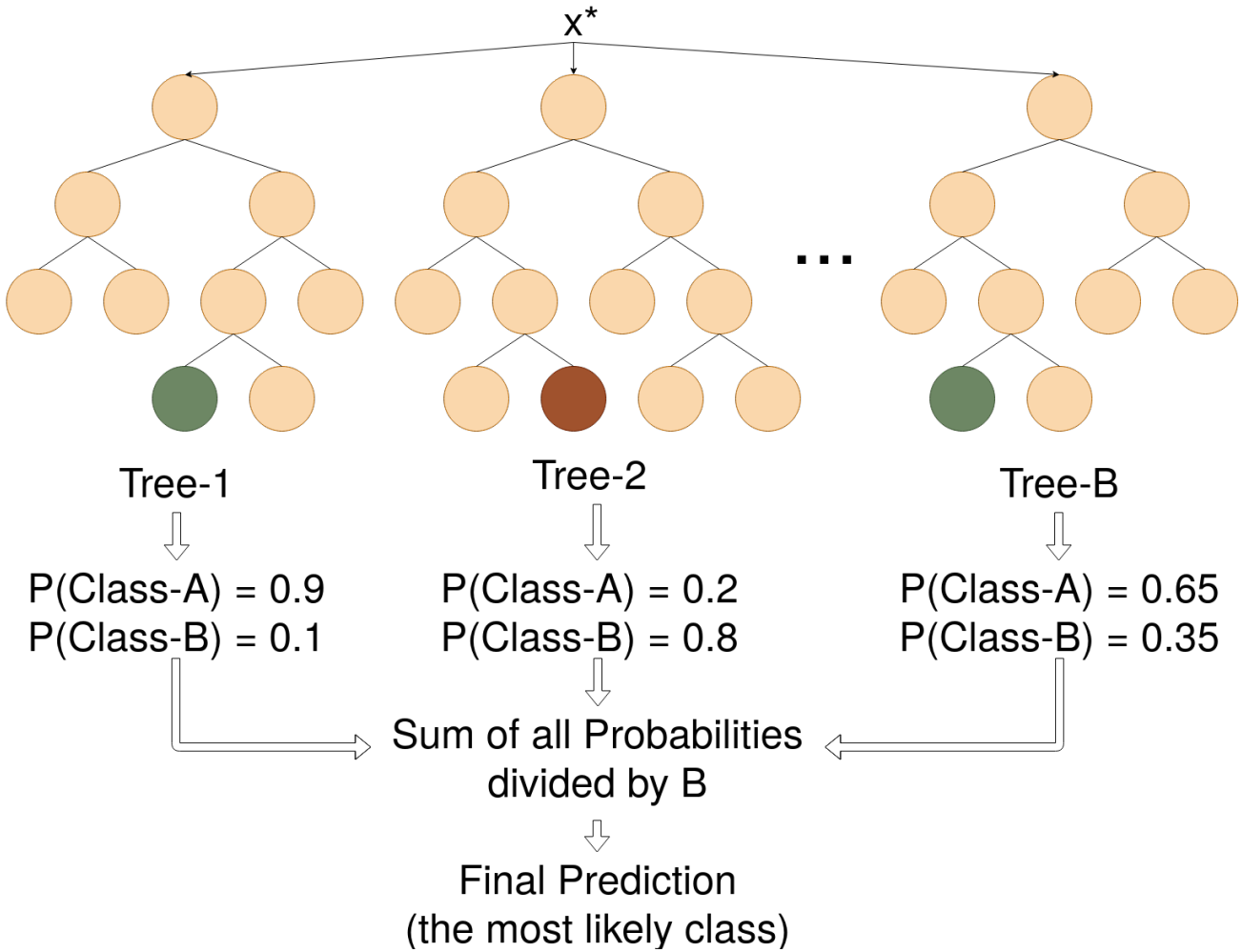


Figure: Soft Voting Illustration

Bayes Model

Theoretically, there exists a model that minimizes the generalization error and can be derived analitically independent of the model [4]. Conditioning the expected generalization error on X gives:

$$\mathbb{E}_{X,Y}\{L(Y,\phi_\beta(X))\} = \mathbb{E}_X\{\mathbb{E}_{Y|X}\{L(Y,\phi_\beta(X))\}\} \tag{6}$$

Point-wise minimization of inner term yields:

$$\phi_\beta = \underset{c\in Y}{\textit{argmin}}\, \mathbb{E}_{Y|X=x}\{L(Y,c)\} \tag{7}$$

ϕ_β is Bayes Model and as mentioned $\boldsymbol{Err}(\phi_\beta)$ is the irreducible error.

The Expected Generalization Error

The expected generalization error of $T_{D,\theta}$ is

$$\boldsymbol{Err}(T_{D,\theta}) = \mathbb{E}_{X,Y}\{L(Y, T_{D,\theta}(X))\} \tag{8}$$

with the decomposition as

$$\boldsymbol{Err}(T_{D,\theta}) = noise(x) + bias^2(x) + var(x) \tag{9}$$

where

$$noise(x) = \boldsymbol{Err}(\phi_\beta)$$
$$bias^2(x) = (\phi_\beta(x) - \mathbb{E}_D\{T_{D,\theta}(x)\})^2$$
$$var(x) = \mathbb{E}_D\{(\mathbb{E}_D(T_{D,\theta}(x)) - T_{D,\theta}(x))^2\}$$

Random Forest

Random Forest Estimator for regression shares the same idea with soft-voting classification;

$$\boldsymbol{RF}_{D,\theta_1,\theta_2,\ldots,\theta_B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_{D,\theta_b}(x) \tag{10}$$

Taking expectation gives;

$$\begin{aligned}\mathbb{E}_{D,\theta_1,\theta_2,\ldots,\theta_B}\{\boldsymbol{RF}_{D,\theta_1,\theta_2,\ldots,\theta_B}(x)\} &= \mathbb{E}_{D,\theta_1,\theta_2,\ldots,\theta_B}\{\frac{1}{B}\sum_{b=1}^{B} T_{D,\theta_b}(x)\} \\ &= \frac{1}{B}\sum_{b=1}^{B}\mathbb{E}_{D,\theta_b}\{T_{D,\theta_b}(x)\} \\ &= \mu_{D,\theta}(x) \end{aligned} \tag{11}$$

where $\mu_{D,\theta}(x)$ is the average prediction of all ensembled trees.

$Bias^2$ of Random Forest

$Bias^2$ of a tree in regression setting

$$[bias(T_{D,\theta})(x)]^2 = (\phi_\beta(x) - \mathbb{E}_D\{T_{D,\theta}(x)\})^2 \tag{12}$$

When we extend our findings to Random Forest;

$$[bias(\boldsymbol{RF}_{D,\theta})(x)]^2 = (\phi_\beta(x) - \mu_{D,\theta}(x))^2 \tag{13}$$

Correlation Coefficient

For any two trees $T_{D,\theta'}$ and $T_{D,\theta''}$ trained with the same data D and different growing parameters θ' and θ'', we can define the correlation coefficient as follows

$$\rho(x) = \frac{\mathbb{E}_{D,\theta',\theta''}\{T_{D,\theta'}(x)\, T_{D,\theta''}(x)\} - \mu^2_{D,\theta}(x)}{\sigma^2_{D,\theta}(x)} \tag{14}$$

$\rho(x)$ is close to 1 $\implies$ highly correlated trees and randomization has no significant effect.
$\rho(x)$ is close to 0 $\implies$ non-correlated and perfectly random prediction of two trees

Variance of Random Forest

$$\mathbb{V}_{D,\theta_1,\theta_2,\ldots,\theta_B}\{\boldsymbol{RF}_{D,\theta_1,\theta_2,\ldots,\theta_B}(x)\} = \rho(x)\sigma^2_{D,\theta}(x) + \frac{1-\rho(x)}{B}\sigma^2_{D,\theta}(x) \quad (15)$$

As $B \to \infty$, $\mathbb{V}\{\boldsymbol{RF}_{D,\theta_1,\theta_2,\ldots,\theta_B}(x)\}$ converges to $\rho(x)\sigma^2_{D,\theta}(x)$.

Due to randomization $\rho(x) < 1 \implies \mathbb{V}\{\boldsymbol{RF}_{D,\theta_1,\theta_2,\ldots,\theta_B}(x)\} < \sigma^2_{D,\theta}(x)$.

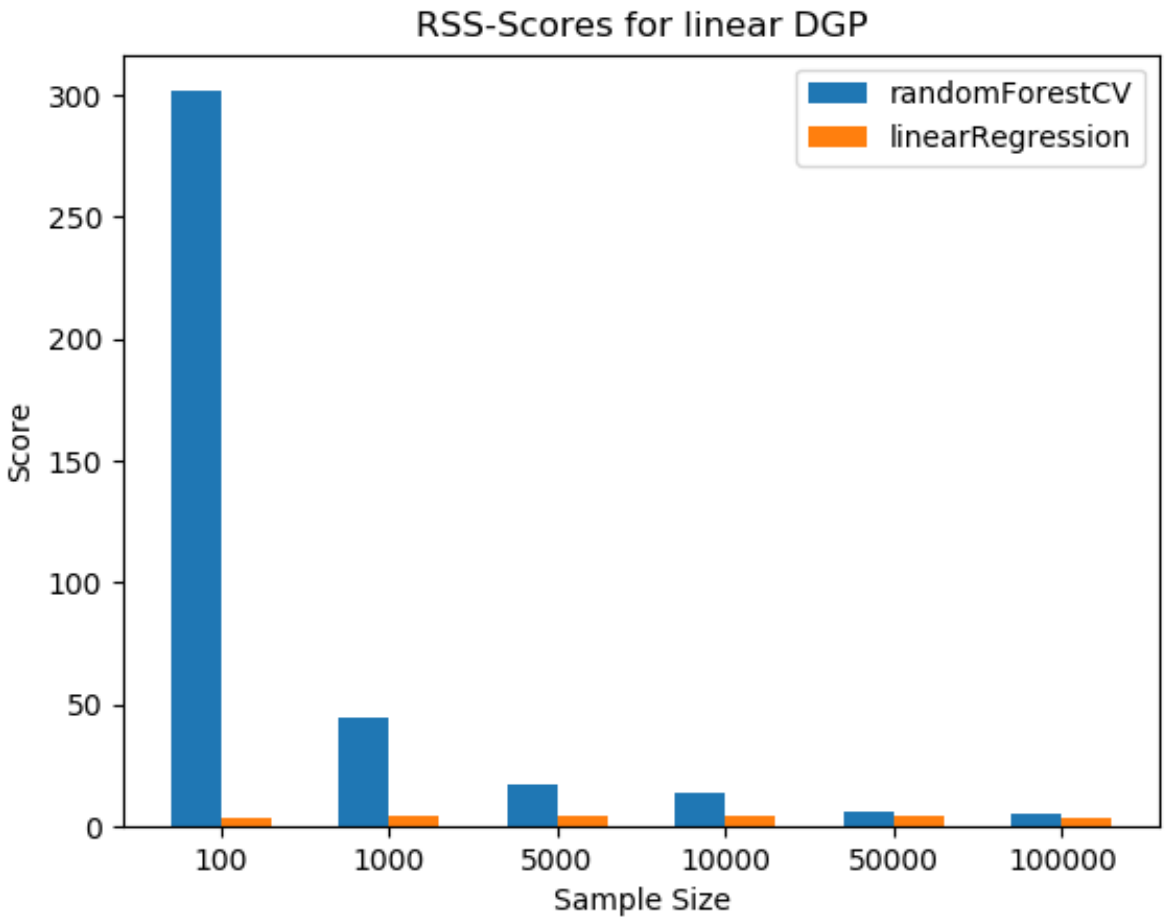
Simulation Study: Linear DGP

The linear DGP generates the data tuples (y, x_1, x_2, x_3) as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon, \tag{16}$$

whereas $(\beta_0, \beta_1, \beta_2, \beta_3) = (0.3, 5, 10, 15)$, $x_1, x_2, x_3 \sim \mathcal{N}(0, 3)$, and $\epsilon \sim \mathcal{N}(0, 1)$.

Decision Tree: Linear DGP Results



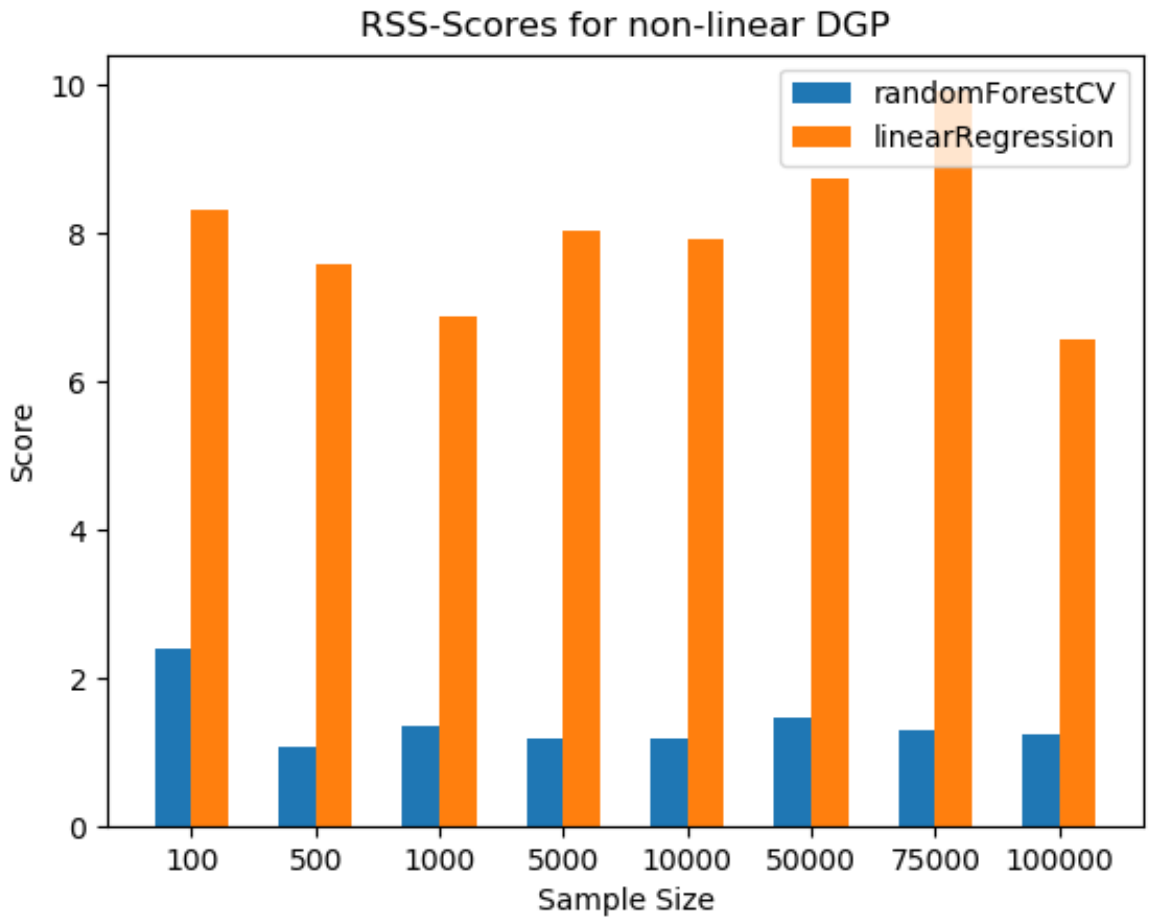
Simulation Study: Non-Linear DGP

The non-linear DGP generates the data tuples (y, x_1, x_2) as follows:

$$y = \beta_0 + \beta_1 \mathbb{1}(x_1 \geq 0, x_2 \geq 0) + \beta_2 \mathbb{1}(x_1 \geq 0, x_2 < 0) + \beta_3 \mathbb{1}(x_1 < 0) + \epsilon, \quad (17)$$

whereas $(\beta_0, \beta_1, \beta_2, \beta_3)$, x_1, x_2 and ϵ are the same as in the previous DGP.

Simulation Study: Non-Linear DGP Results



Real Data

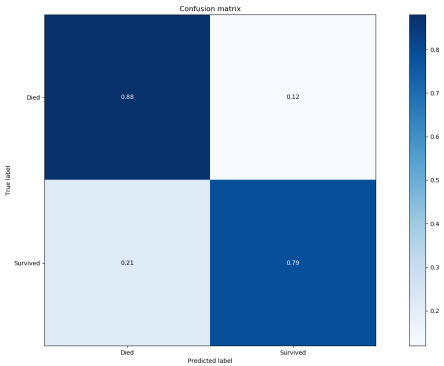
Data: Titanic data
Method used:

1. Random Forest
2. AdaBoost
3. Gradient Boosting Classifier

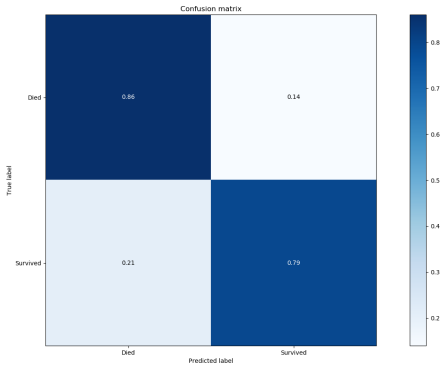
Goal: Given features of passengers predict which passengers survived the Titanic shipwreck

Real Data: results

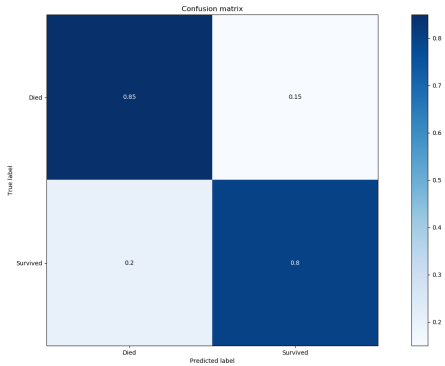
Random Forest
Accuracy: 84,32%



AdaBoost
Accuracy: 82.8%



Gradient Boosting
Accuracy: 82,8%



Conclusion

The End

References

- L. Breiman et al. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. ISBN: 9780412048418. URL: https://books.google.de/books?id=JwQx-WOmSyQC.
- Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.
- Gareth James et al. "An Introduction to Statistical Learning: with Applications in R". In: (2013). URL: https://faculty.marshall.usc.edu/gareth-james/ISL/.
- Gilles Louppe. "Understanding random forests". In: *University of Liège* (2014).
- Daan van der Valk and Stjepan Picek. *Bias-variance decomposition in machine learning-based side-channel analysis*. Tech. rep. Cryptology ePrint Archive, Report 2019/570, 2019.