# Evolutionary Genetics

LV 25600-01 | Lecture with exercises | 4KP

## Random Forest

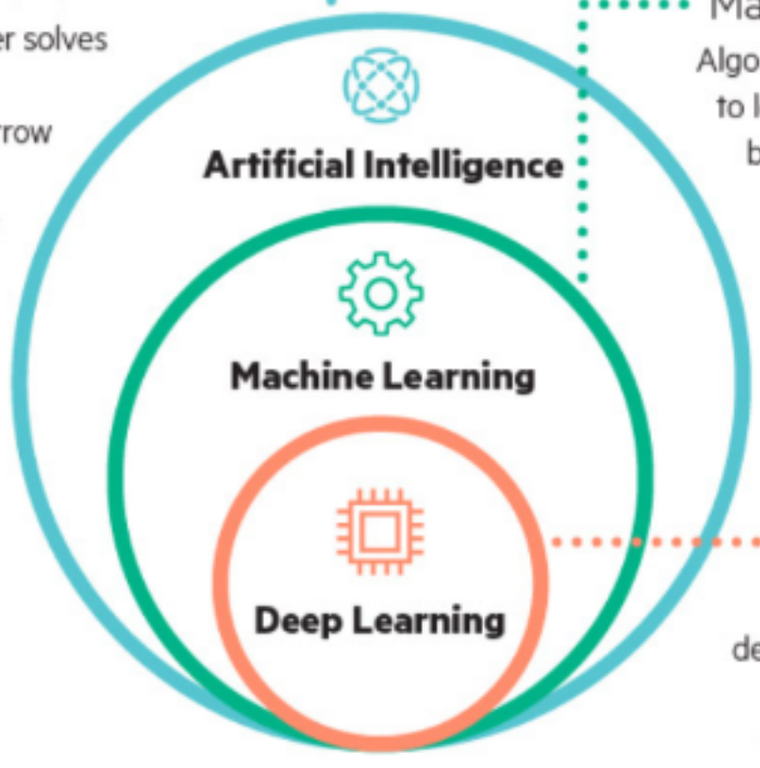
Random forest is a **supervised learning algorithm**. The "forest" is an ensemble of **decision trees** trained with the bagging method. Random Forest generates multiple decision trees and merges them together to build predictions.

## What Makes a Machine Intelligent?

While AI is the headliner, there are actually subsets of the technology which can be applied to solving human problems in different ways.

### Artificial Intelligence (AI)

A process where a computer solves a task in a way that mimics human behavior. Today, narrow AI—when a machine is trained to do one particular task—is becoming more widely used, from virtual assistants to self-driving cars to automatic tagging your friends in your photos on Facebook.

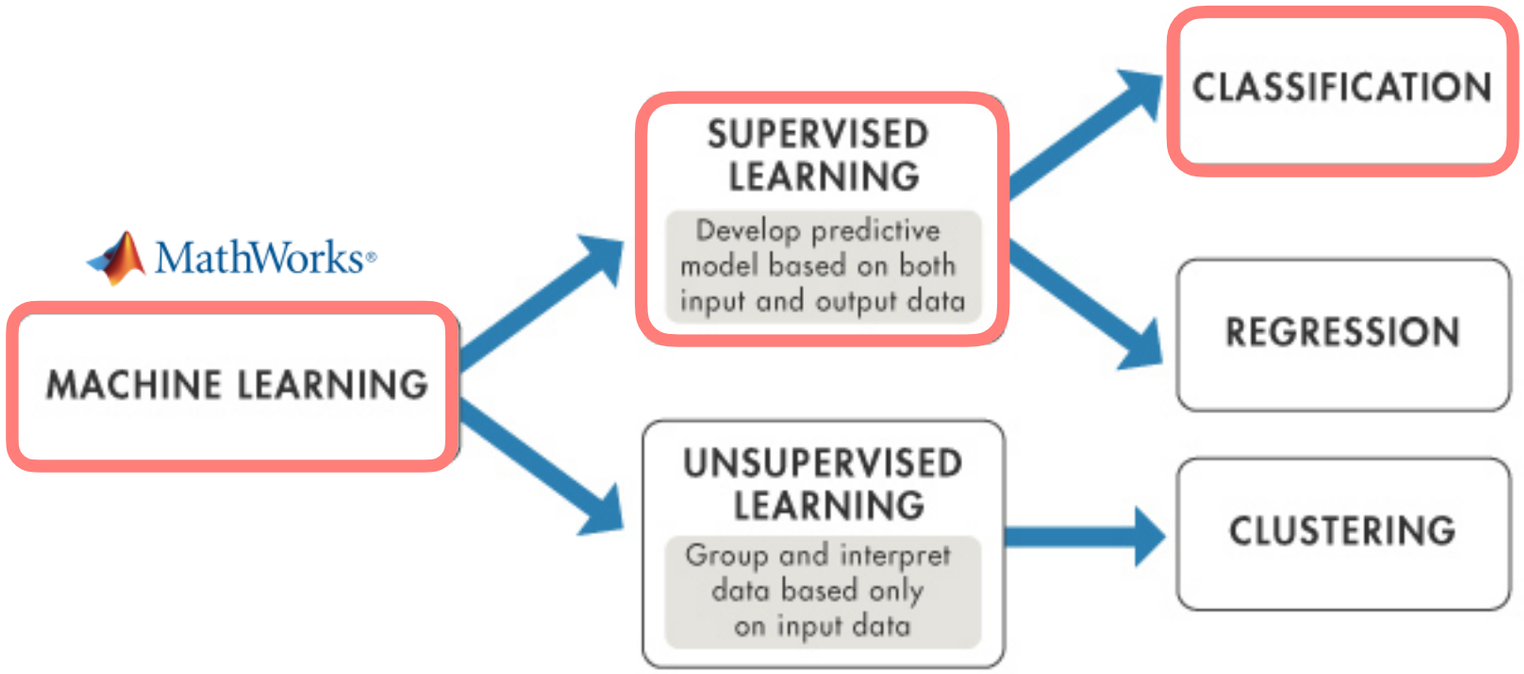



### Machine Learning (ML)

Algorithms that allow computers to learn from examples without being explicitly programmed.

### Deep Learning (DL)

A subset of ML which uses deep artificial neural networks as models and does not require feature engineering.

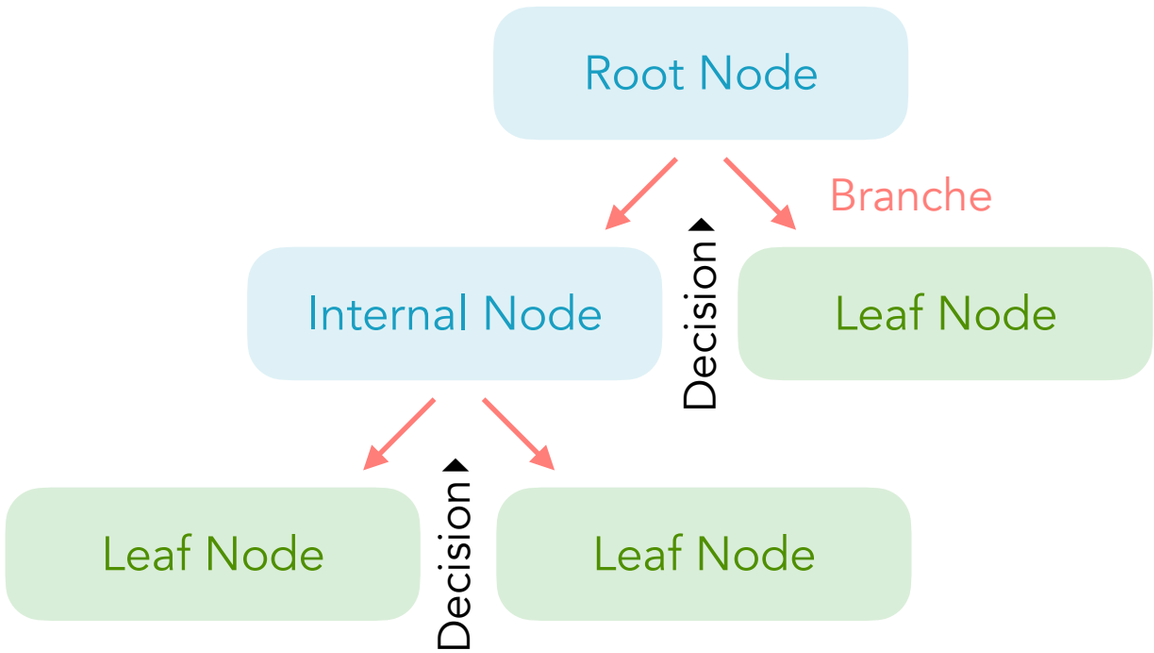



- **Supervised Learning** (direct feedback, predict outcome)
- **Unsupervised Learning** (no feedback, find hidden structure)
- **Reinforcement Learning** (reward system, decision process)

The supervision refers to the fact that the target values (y) provide a supervisory role, which indicates to the learner the task it needs to learn. Unsupervised learning, in contrast to supervised learning, includes a set of statistical tools to better understand and describe your data, but performs the analysis without a target variable. In essence, unsupervised learning is concerned with identifying groups in a data set.

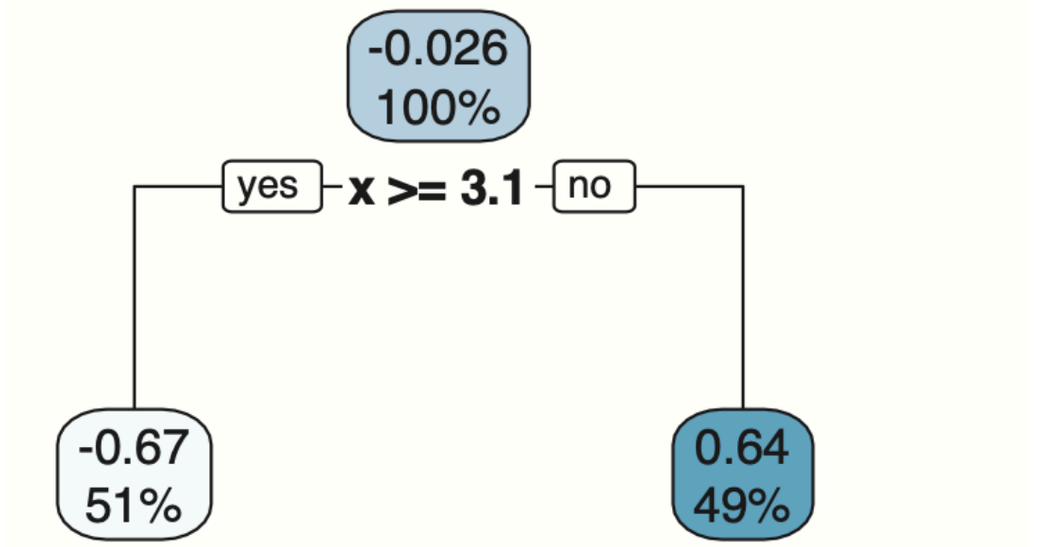
GDC Genetic Diversity Centre Zurich

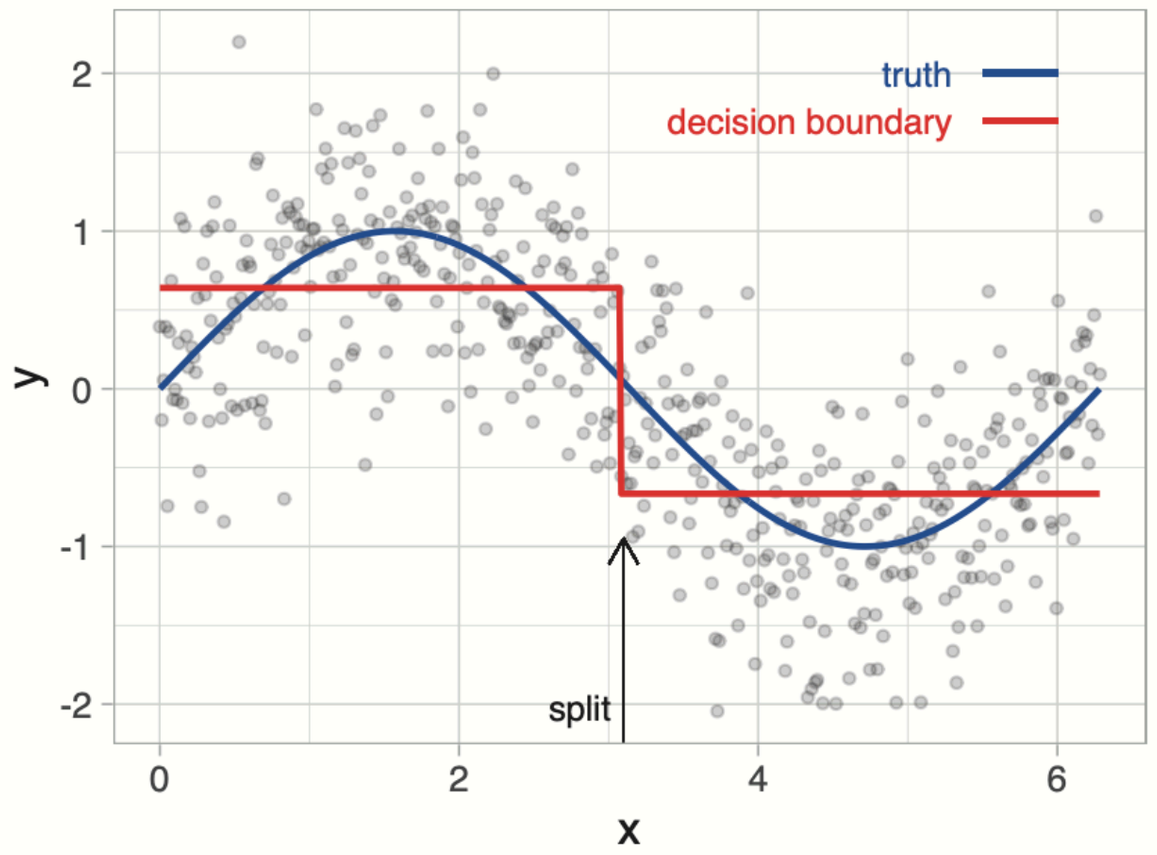
# Decision Trees




Decision Types
Binary: TRUE / FALSE
Threshold: e.g. bigger x
Factor: blue, yellow, green

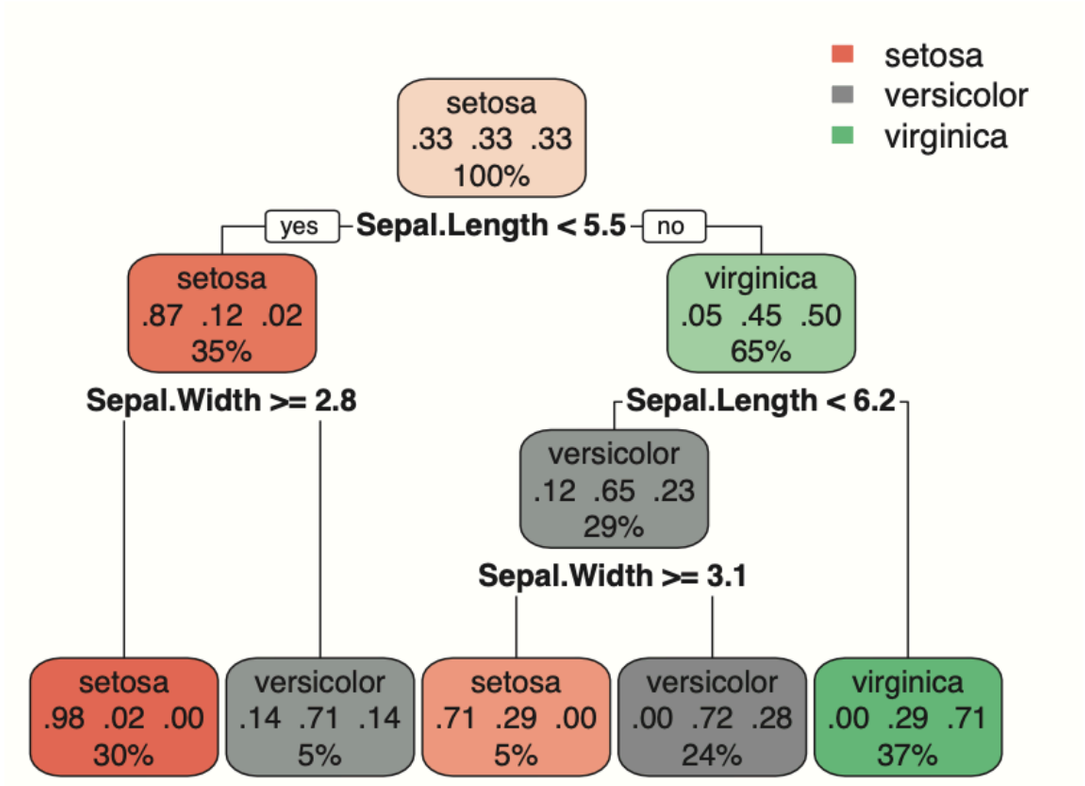
Tree-based models are a class of nonparametric algorithms that work by partitioning the feature space into a number of smaller (non-overlapping) regions with similar response values using a set of splitting rules. Such divide-and-conquer methods can produce simple rules that are easy to interpret and visualise with tree diagrams.

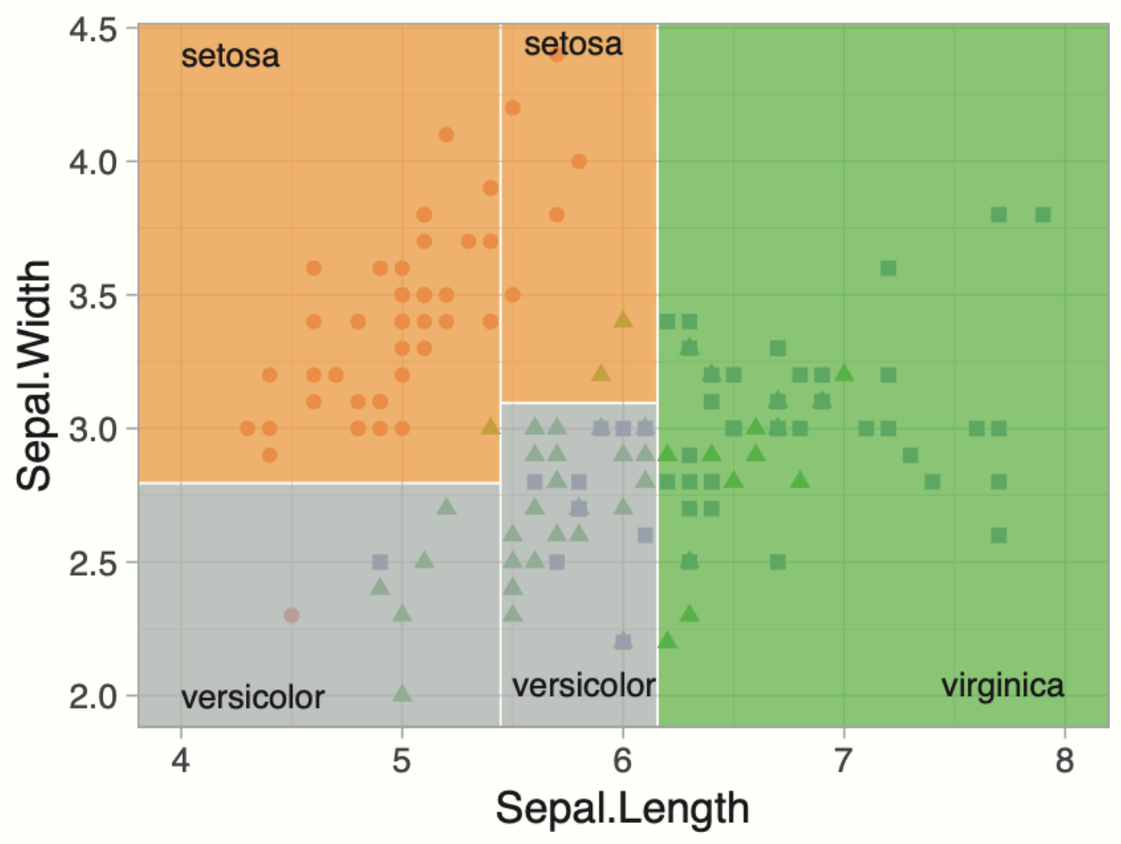



Decision tree illustrating the single split on feature x. The resulting decision boundary illustrates the predicted value when x < 3.1 (0.64) and when x > 3.1 (-0.67).

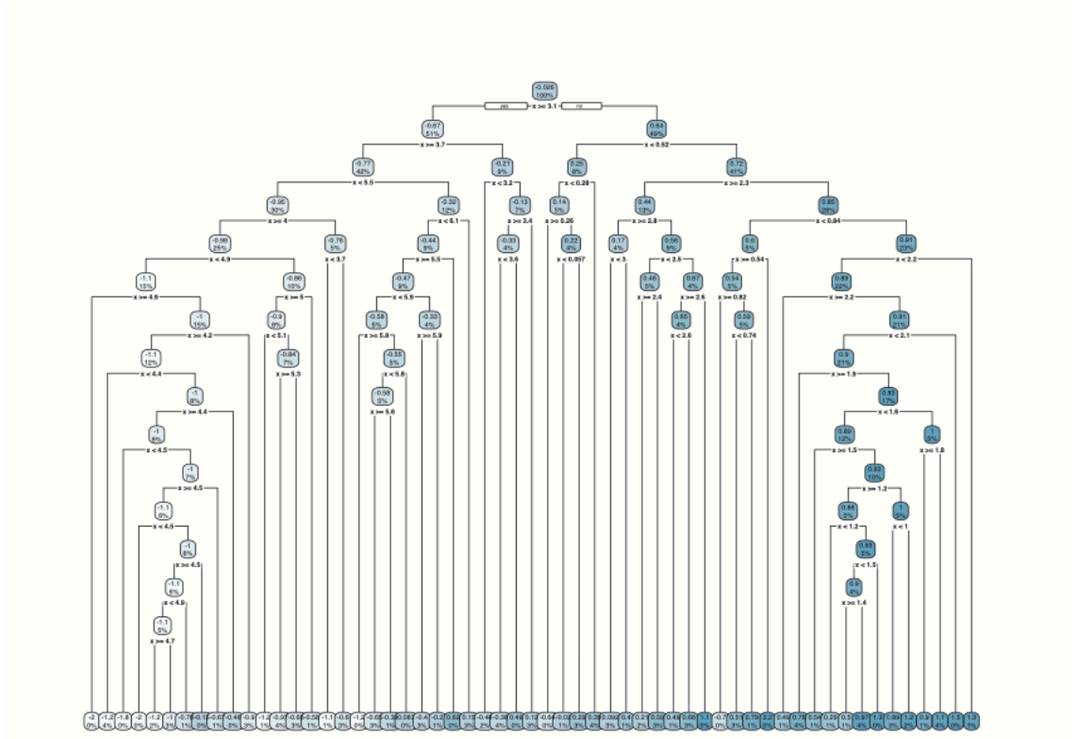
Source: Boehmke & Grenwell (2020)

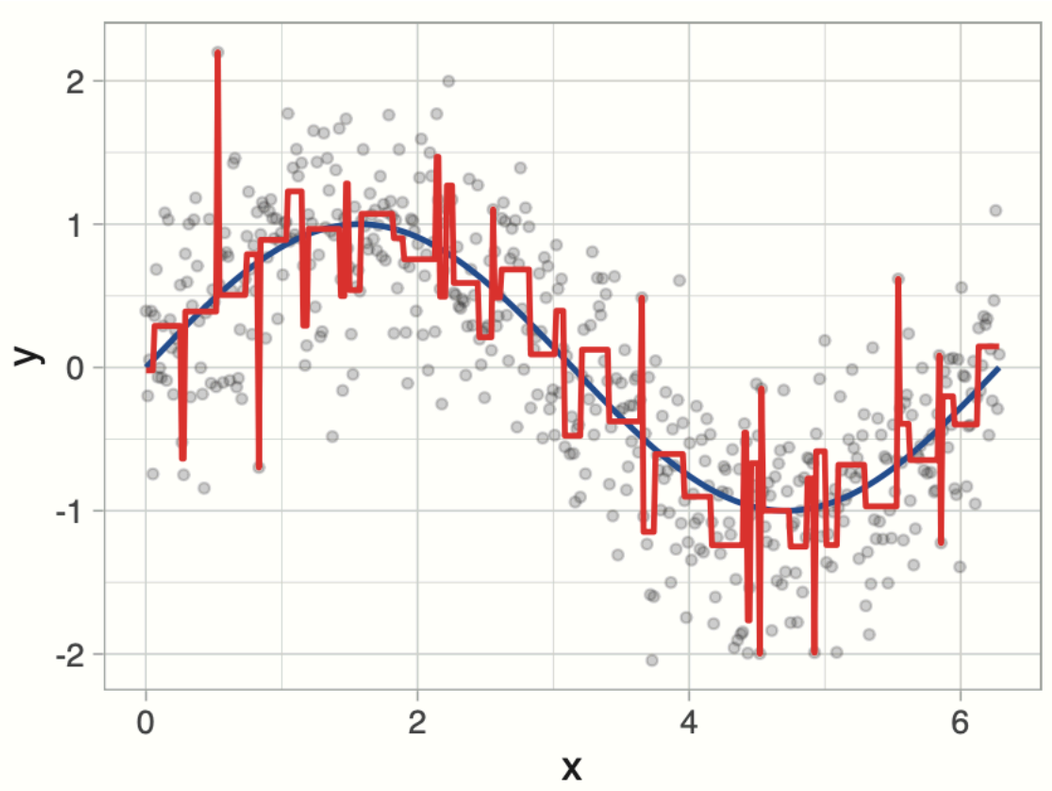



Decision tree for the iris classification problem (left). The decision boundary results in rectangular regions enclosing the observations. The class with the highest proportion in each region is the predicted value (right).

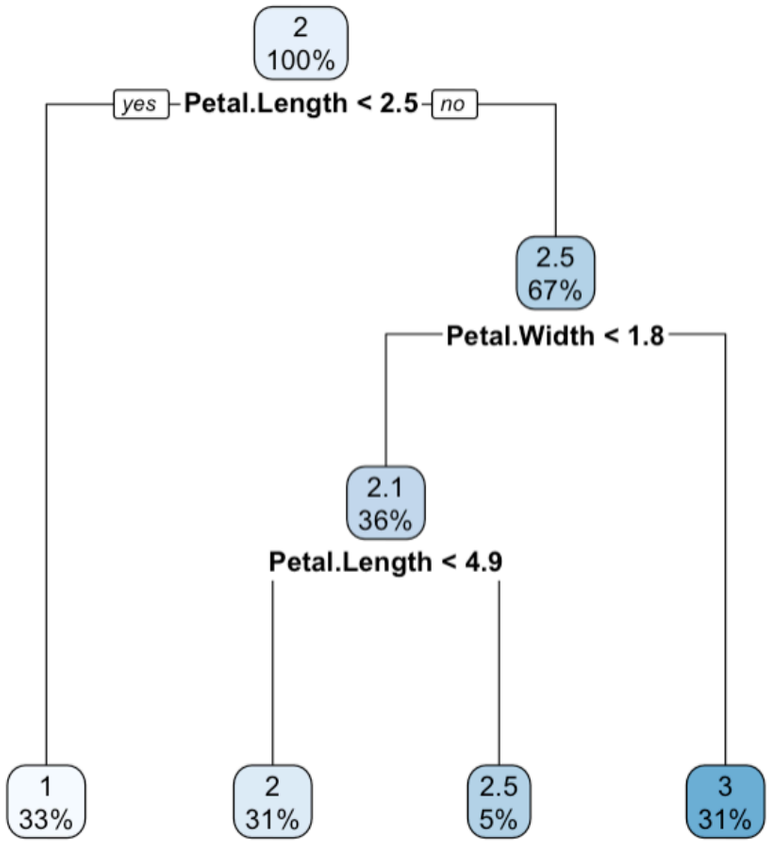
Source: Boehmke & Grenwell (2020)






Overfit decision tree - How deep (i.e. complex) should we make the tree? If we grow an overly complex tree, we will tend to overfit to our training data, resulting in poor generalisation performance.

Source: Boehmke & Grenwell (2020)




```
rpart::rpart(
  formula = Species ~ .,
  data = train_iris,
  method = "class"
)

# n= 150
#
# node), split, n, deviance, yval
#       * denotes terminal node
#
# 1) root 150 100.0000000 2.000000
#   2) Petal.Length< 2.45 50   0.0000000 1.000000 *
#   3) Petal.Length>=2.45 100  25.0000000 2.500000
#     6) Petal.Width< 1.75 54   4.5370370 2.092593
#      12) Petal.Length< 4.85 46   0.9782609 2.021739 *
#      13) Petal.Length>=4.85 8   2.0000000 2.500000 *
#     7) Petal.Width>=1.75 46   0.9782609 2.978261 *
```

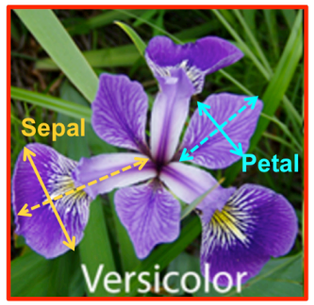
- Find root node by comparing Gini impurity scores*.
- Calculate Gini impurity scores.
- If the node itself has the lowest score it becomes a leaf node.
- If the node has a higher score it becomes an internal node and the separations with the lowest score will be the next node.

* Classification: Gini Impurity
Regression: Sum of Squared Errors (SSE)

* Gini score: small value indicates that a node contains predominantly observations from a single class.

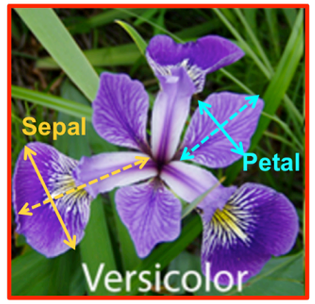
Predictors — Response

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 7.0 | 3.2 | 4.7 | 1.4 | versicolor |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| … | … | … | … | … |
| 6.3 | 3.3 | 6.0 | 2.5 | virginica |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |
| 6.5 | 3.0 | 5.8 | 2.2 | virginica |
| … | … | … | … | … |



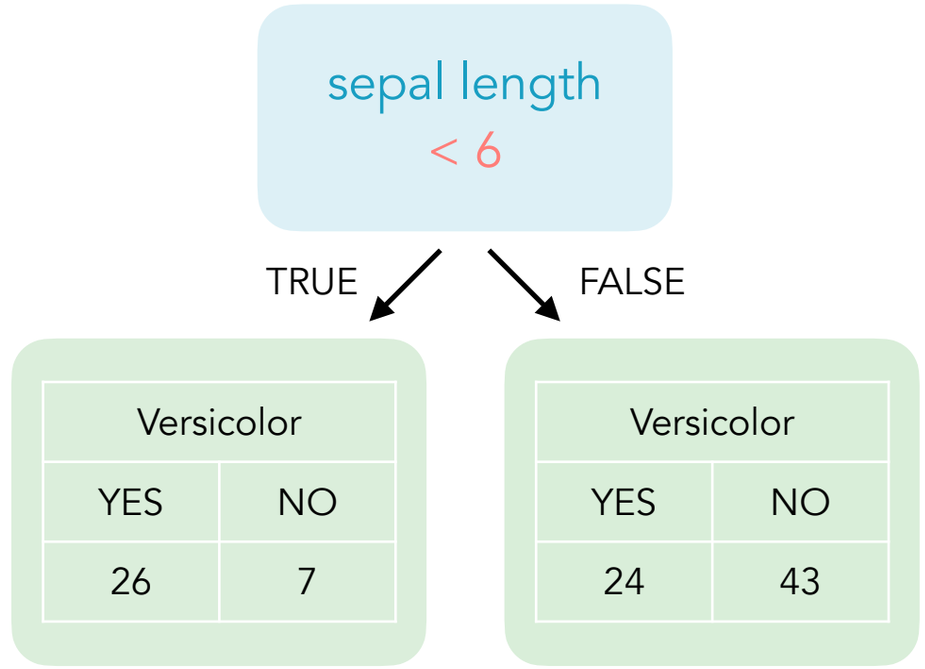


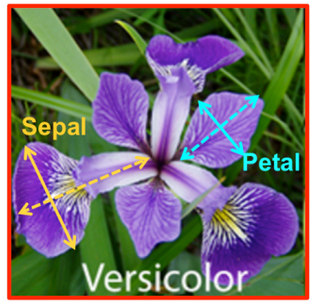

We use a slightly modified data set based on the famous (Fisher's or Anderson's) **iris data** set. It contains measurements in centimetres of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are *Iris setosa*, *versicolor*, and *virginica*.




subset of two species

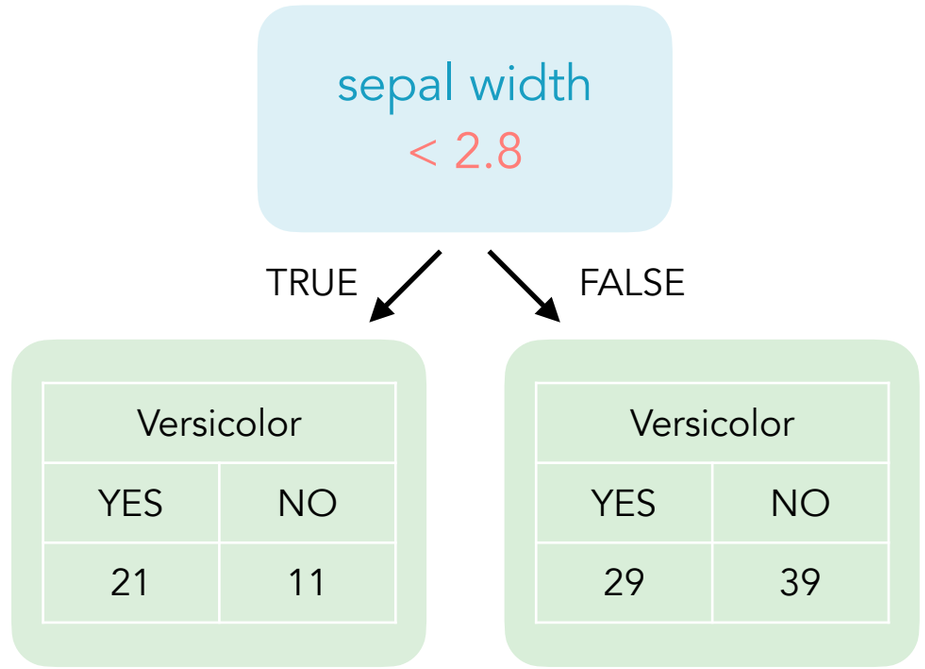
| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 7.0 | 3.2 | 4.7 | 1.4 | versicolor |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| … | … | … | … | … |
| 6.3 | 3.3 | 6.0 | 2.5 | virginica |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |
| 6.5 | 3.0 | 5.8 | 2.2 | virginica |
| … | … | … | … | … |

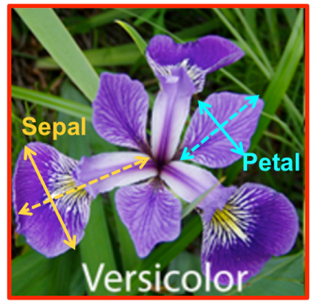





subset of two species

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 7.0 | 3.2 | 4.7 | 1.4 | versicolor |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| … | … | … | … | … |
| 6.3 | 3.3 | 6.0 | 2.5 | virginica |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |
| 6.5 | 3.0 | 5.8 | 2.2 | virginica |
| … | … | … | … | … |






subset of two species

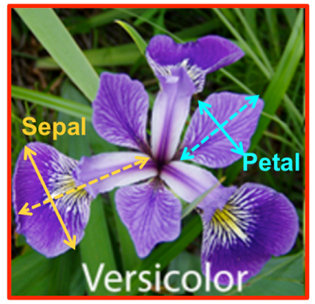
| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 7.0 | 3.2 | 4.7 | 1.4 | versicolor |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| … | … | … | … | … |
| 6.3 | 3.3 | 6.0 | 2.5 | virginica |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |
| 6.5 | 3.0 | 5.8 | 2.2 | virginica |
| … | … | … | … | … |

Petal Length
< 5

TRUE ↙ ↘ FALSE

| Versicolor | |
|---|---|
| YES | NO |
| 48 | 6 |

| Versicolor | |
|---|---|
| YES | NO |
| 2 | 44 |




subset of two species

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 7.0 | 3.2 | 4.7 | 1.4 | versicolor |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| … | … | … | … | … |
| 6.3 | 3.3 | 6.0 | 2.5 | virginica |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |
| 6.5 | 3.0 | 5.8 | 2.2 | virginica |
| … | … | … | … | … |

Petal Width
<1.5

TRUE ↙ ↘ FALSE

| Versicolor | |
|---|---|
| YES | NO |
| 35 | 1 |

| Versicolor | |
|---|---|
| YES | NO |
| 15 | 49 |

GDC Genetic Diversity Centre Zurich

sepal length
<6

TRUE ↙ ↘ FALSE

| Versicolor | |
|---|---|
| YES | NO |
| 26 | 7 |

| Versicolor | |
|---|---|
| YES | NO |
| 24 | 43 |

petal length
<5

TRUE ↙ ↘ FALSE

| Versicolor | |
|---|---|
| YES | NO |
| 48 | 6 |

| Versicolor | |
|---|---|
| YES | NO |
| 2 | 44 |

sepal width
<2.8

TRUE ↙ ↘ FALSE

| Versicolor | |
|---|---|
| YES | NO |
| 21 | 11 |

| Versicolor | |
|---|---|
| YES | NO |
| 29 | 39 |

petal width
<1.5

TRUE ↙ ↘ FALSE

| Versicolor | |
|---|---|
| YES | NO |
| 35 | 1 |

| Versicolor | |
|---|---|
| YES | NO |
| 15 | 49 |

## Gini Impurity Measure

$$gini = 1 - \left( \frac{\text{propability of TRUE}}{\text{number of cases}} \right)^2 - \left( \frac{\text{propability of FALSE}}{\text{number of cases}} \right)^2$$

## Gini Impurity Measure

$$gini = 1 - \left(\frac{\text{propability of TRUE}}{\text{number of cases}}\right)^2 - \left(\frac{\text{propability of FALSE}}{\text{number of cases}}\right)^2$$

The meaning of gini explained based on extremes:

→ Perfect Case

| YES | NO |
|---|---|
| 100 | 0 |

$$gini = 1 - \left(\frac{100}{100}\right)^2 - \left(\frac{0}{100}\right)^2 = 0$$

→ Worst Case

| YES | NO |
|---|---|
| 50 | 50 |

$$gini = 1 - \left(\frac{50}{100}\right)^2 - \left(\frac{50}{100}\right)^2 = 0.5$$

sepal length
< 6

TRUE

FALSE

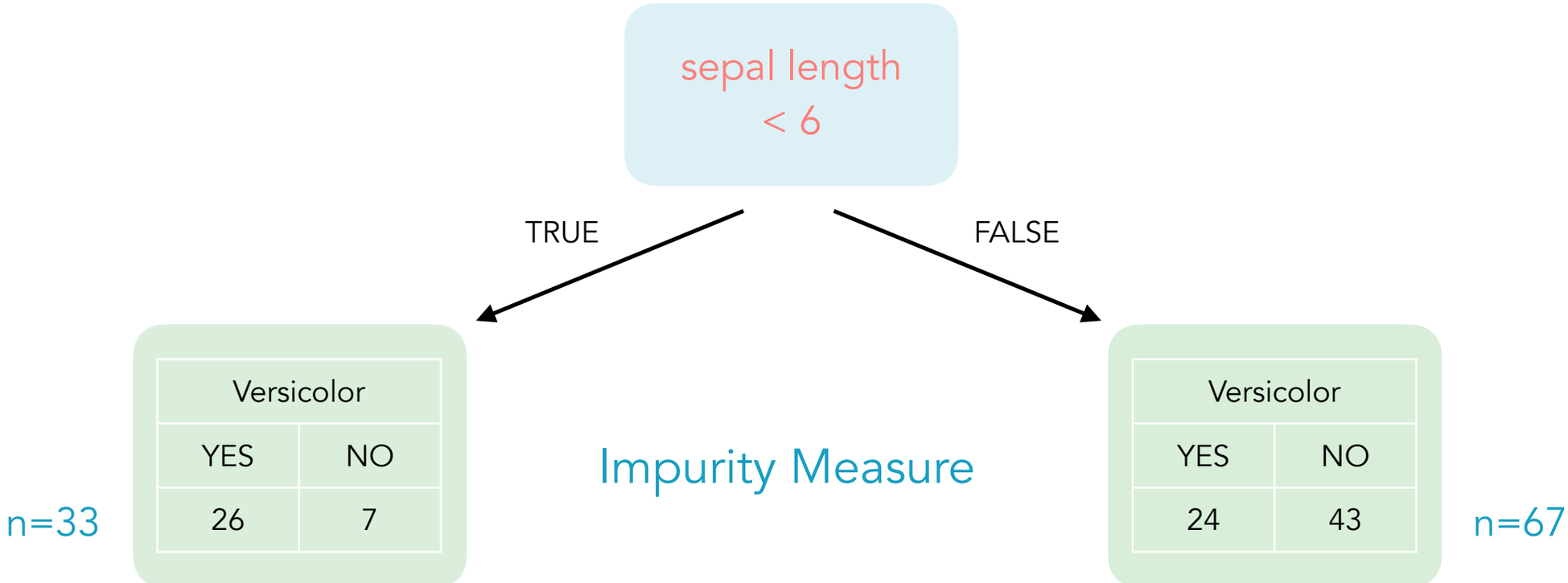
| Versicolor | |
|---|---|
| YES | NO |
| 26 | 7 |

Impurity Measure

| Versicolor | |
|---|---|
| YES | NO |
| 24 | 43 |

$$gini = 1 - \left(\frac{26}{26+7}\right)^2 - \left(\frac{7}{26+7}\right)^2 = 0.334$$

$$gini = 1 - \left(\frac{24}{24+43}\right)^2 - \left(\frac{43}{24+43}\right)^2 = 0.460$$




$$gini = 1 - \left(\frac{26}{26+7}\right)^2 - \left(\frac{7}{26+7}\right)^2 = 0.334$$

$$gini = 1 - \left(\frac{24}{24+43}\right)^2 - \left(\frac{43}{24+43}\right)^2 = 0.460$$

$$\text{weighted mean} = \frac{33}{33+67} * 0.334 + \frac{67}{33+67} * 0.460 = \underline{\underline{0.418}}$$

GDC Genetic Diversity Centre Zurich

sepal length
< 6

gini impurity = 0.418

sepal width
< 2.8

gini impurity = 0.477

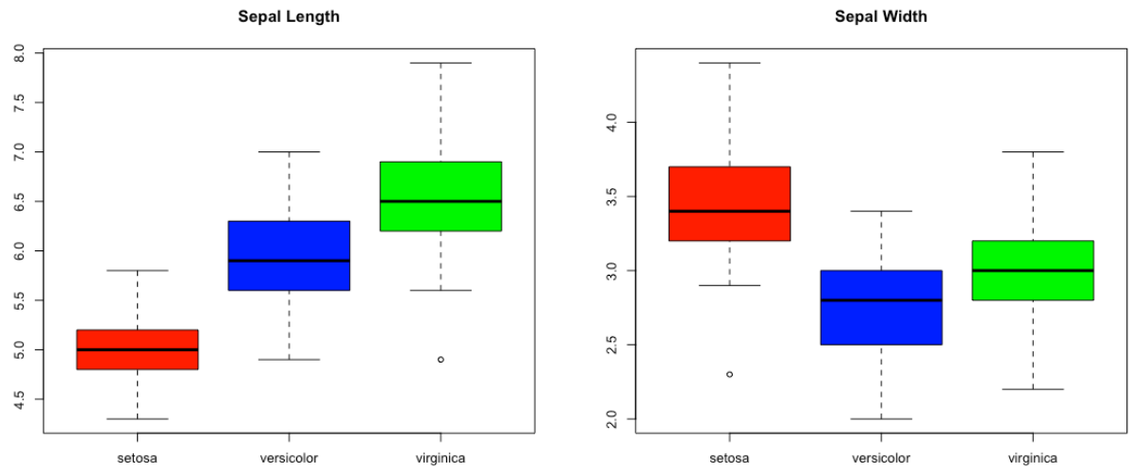
petal length
< 5

gini impurity = 0.145

petal width
< 1.5

gini impurity = 0.249

## Variable importance



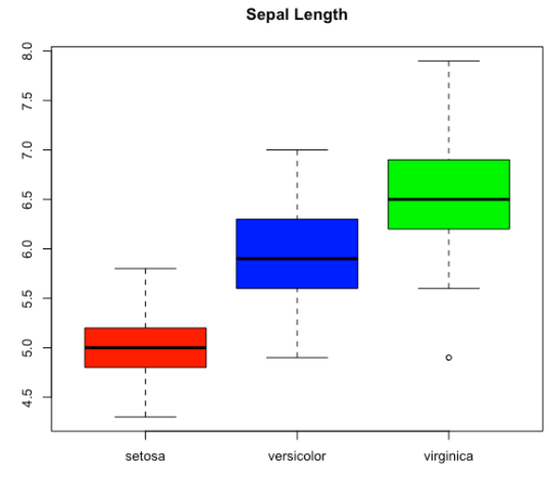
GDC Genetic Diversity Centre Zurich

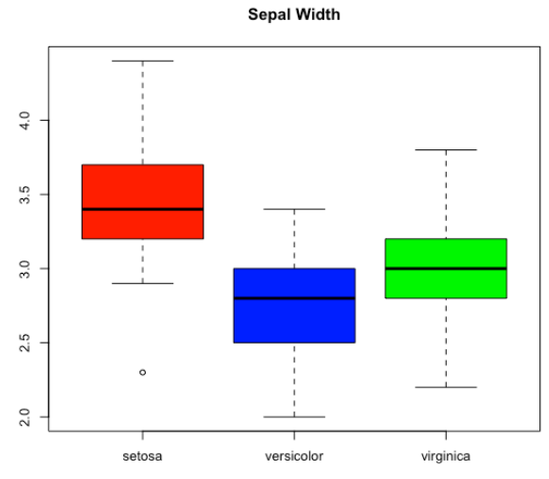
sepal length
< 6

gini impurity = 0.418

sepal width
< 2.8

gini impurity = 0.477



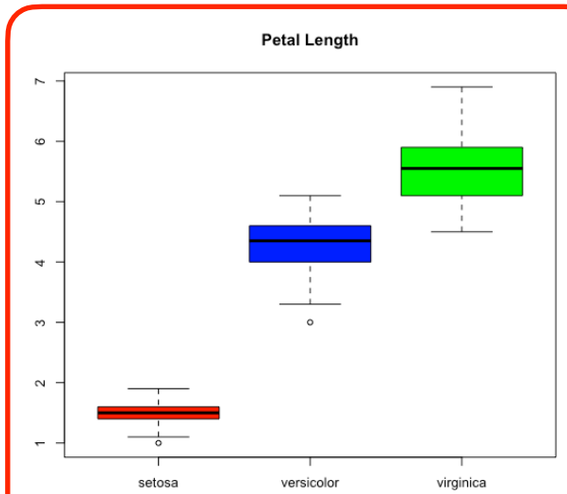



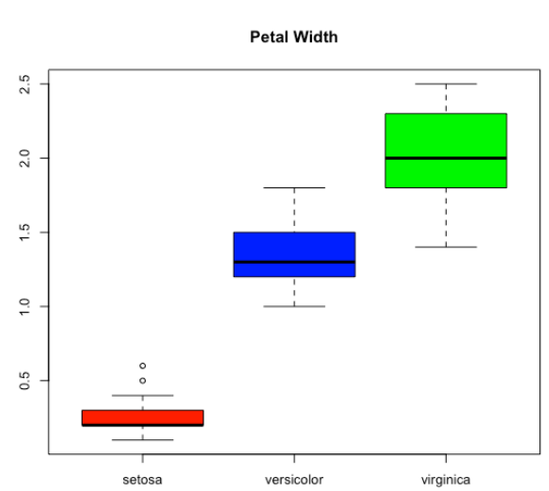

petal length
< 5

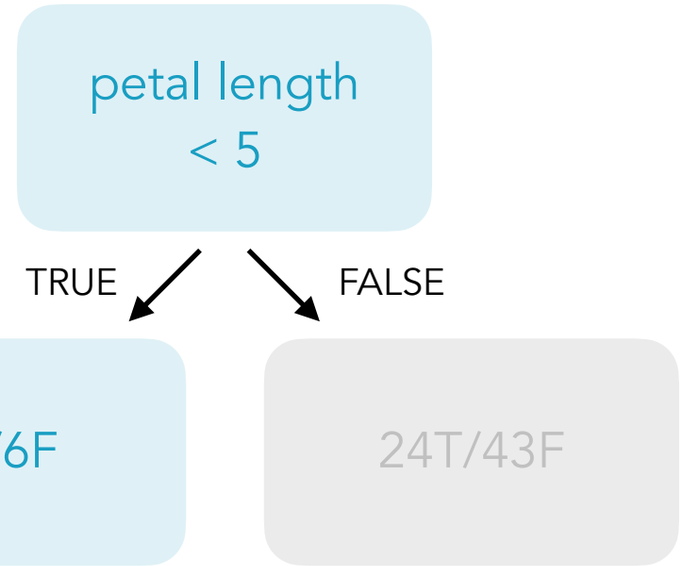
gini impurity = 0.145

petal width
< 1.5

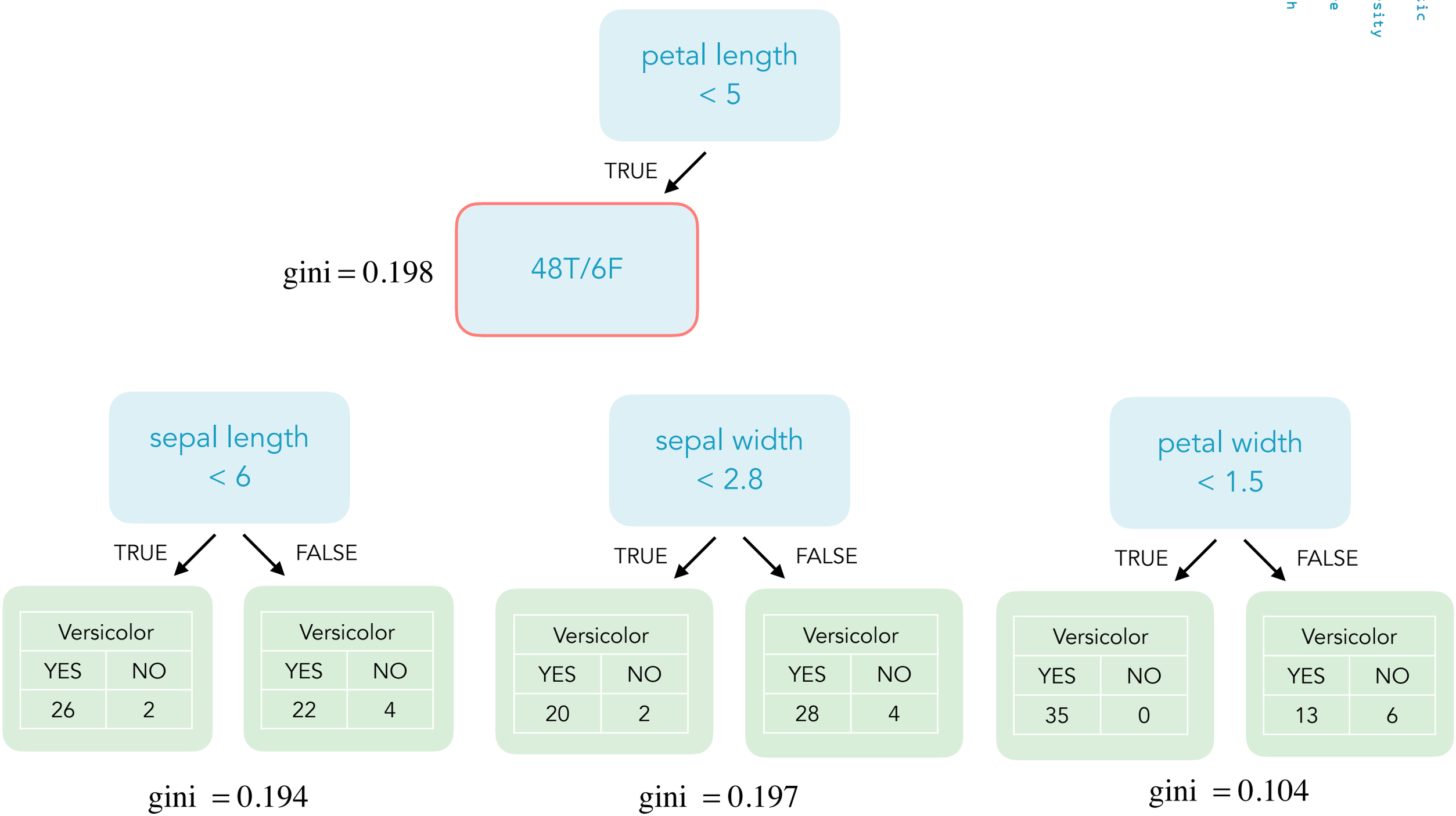
gini impurity = 0.249






$$\text{gini} = 1 - \left(\frac{48}{48+6}\right)^2 - \left(\frac{6}{48+6}\right)^2 = 0.198$$


petal length
< 5
TRUE
gini = 0.198
48T/6F
sepal length
< 6
TRUE
FALSE
Versicolor
YES NO
26 2
Versicolor
YES NO
22 4
gini = 0.194
sepal width
< 2.8
TRUE
FALSE
Versicolor
YES NO
20 2
Versicolor
YES NO
28 4
gini = 0.197
petal width
< 1.5
TRUE
FALSE
Versicolor
YES NO
35 0
Versicolor
YES NO
13 6
gini = 0.104

petal length
< 5

TRUE

$\text{gini} = 0.198$

48T/6F

sepal length
< 6

TRUE FALSE

| Versicolor | |
|---|---|
| YES | NO |
| 26 | 2 |

| Versicolor | |
|---|---|
| YES | NO |
| 22 | 4 |

$\text{gini} = 0.194$

sepal width
< 2.8

TRUE FALSE

| Versicolor | |
|---|---|
| YES | NO |
| 20 | 2 |

| Versicolor | |
|---|---|
| YES | NO |
| 28 | 4 |

$\text{gini} = 0.197$

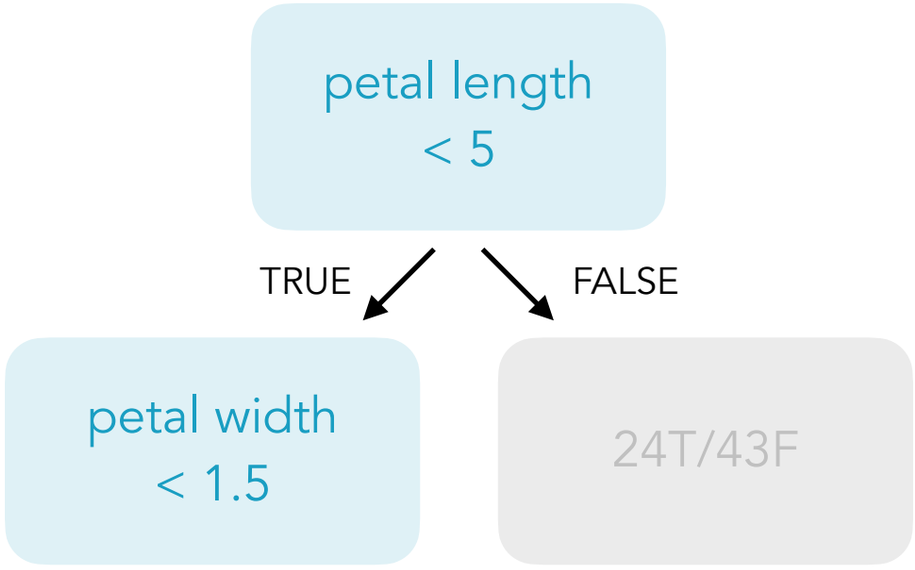
petal width
< 1.5

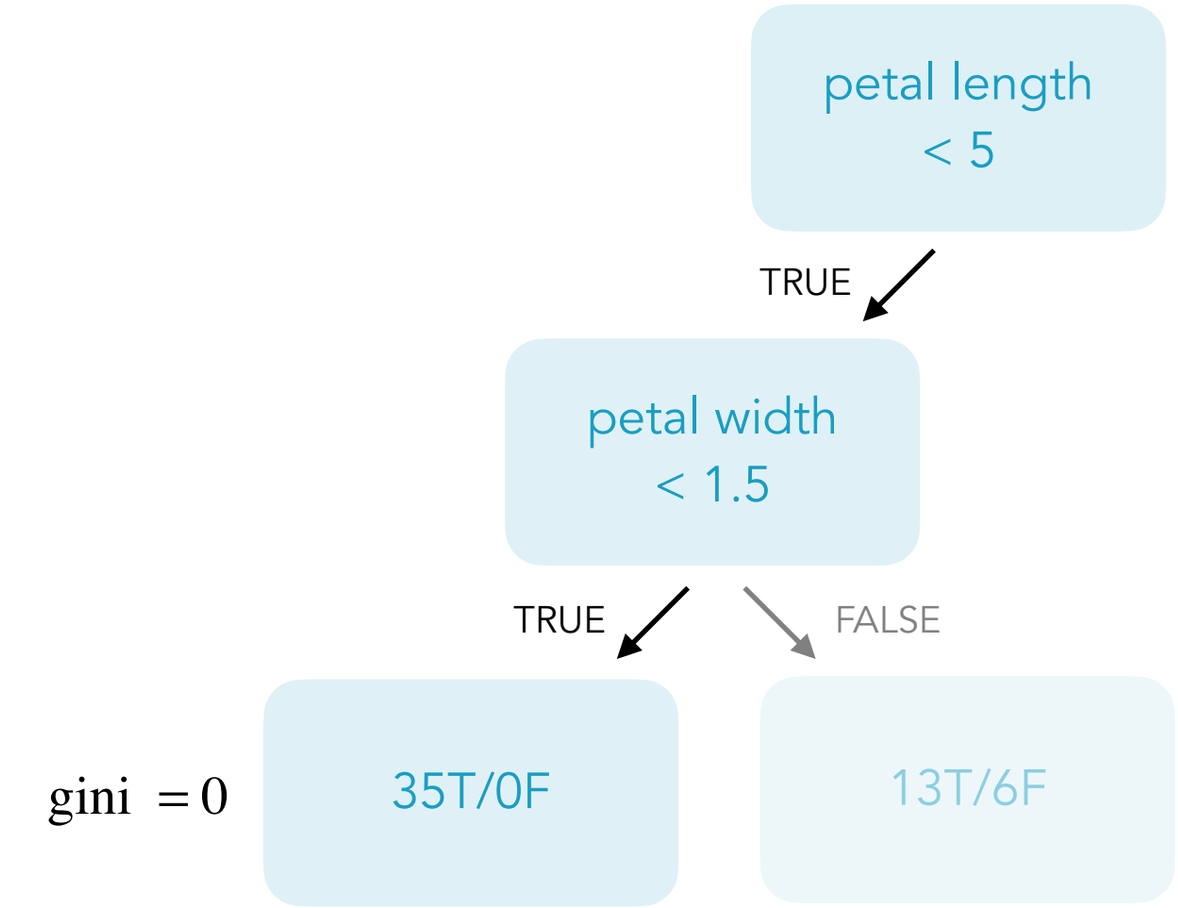
TRUE FALSE

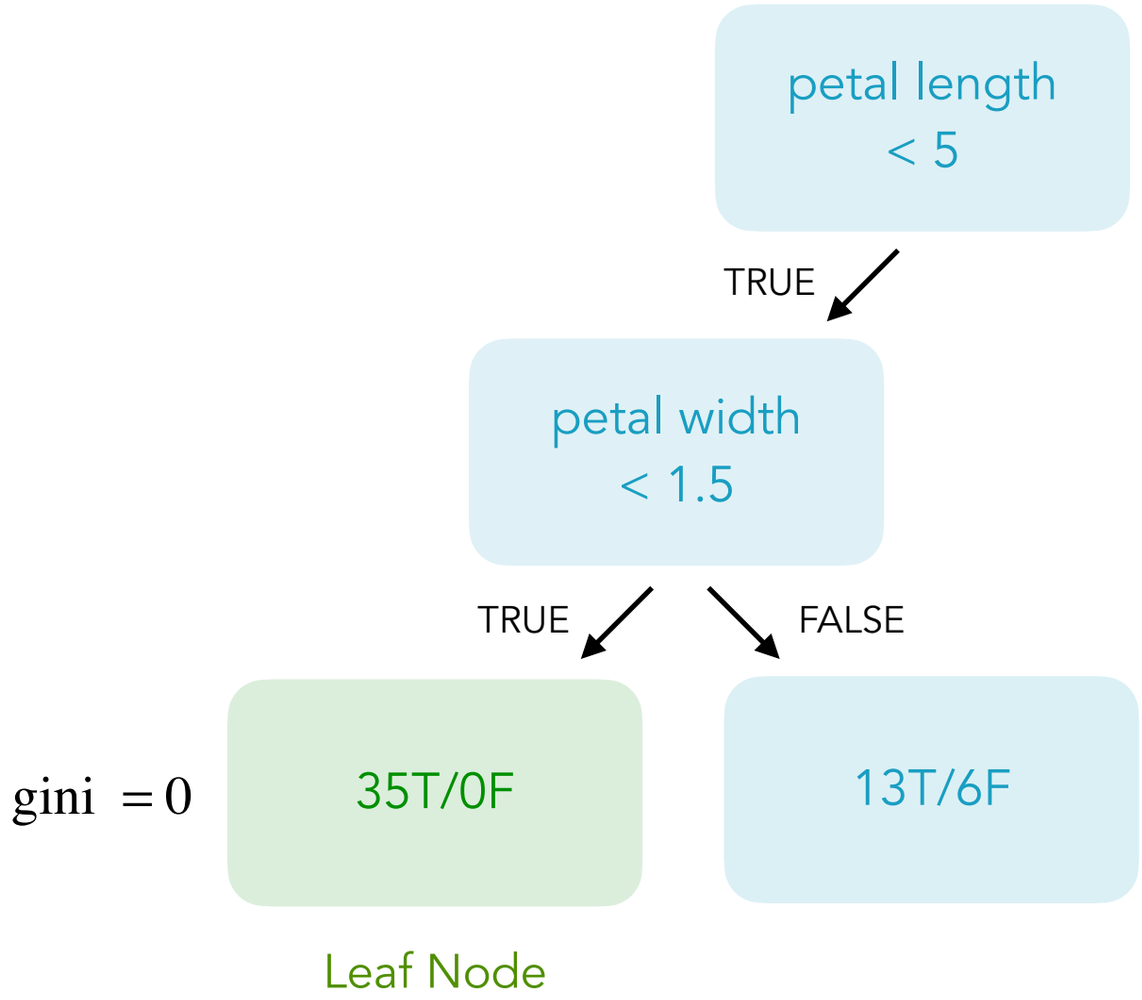
| Versicolor | |
|---|---|
| YES | NO |
| 35 | 0 |

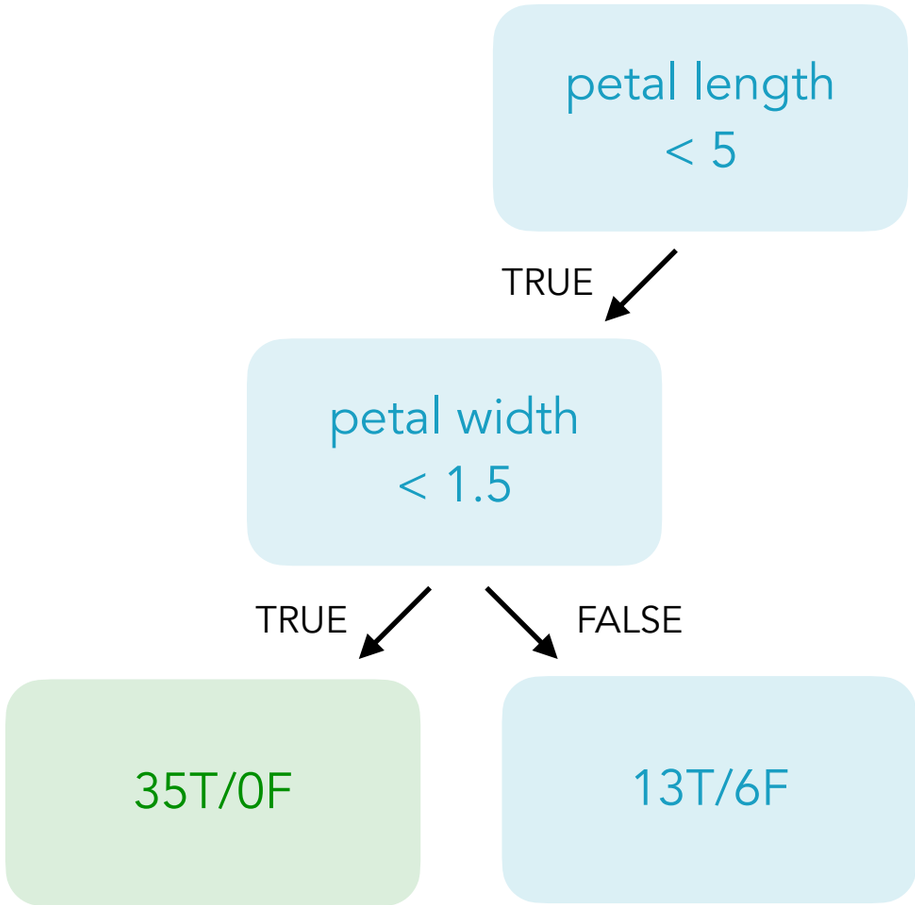
| Versicolor | |
|---|---|
| YES | NO |
| 13 | 6 |

$\text{gini} = 0.104$


petal length
< 5
TRUE
FALSE
petal width
< 1.5
24T/43F

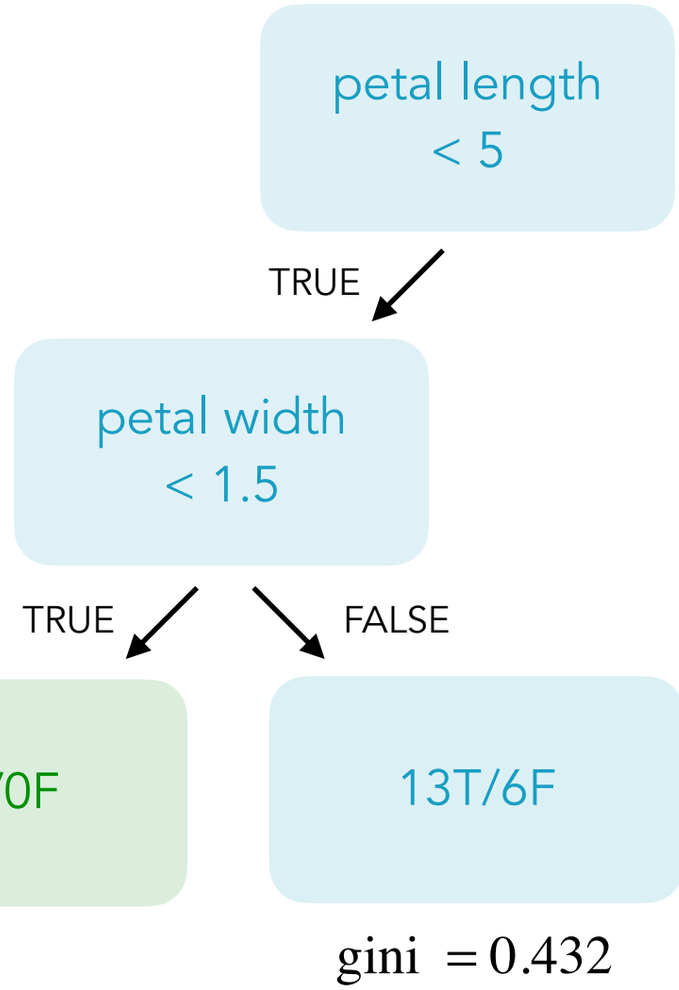

petal length
< 5
TRUE
petal width
< 1.5
TRUE
FALSE
gini = 0
35T/0F
13T/6F

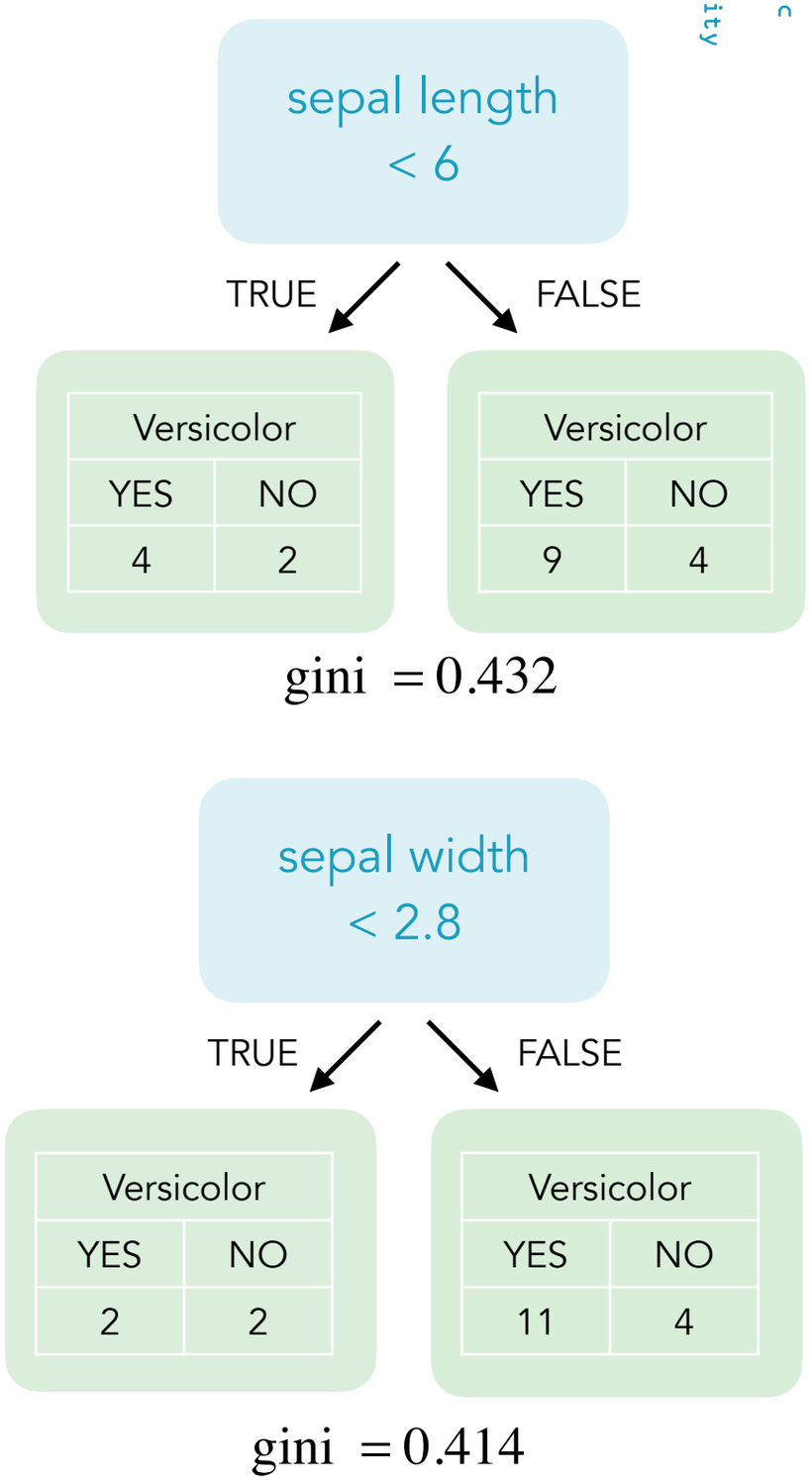

petal length
< 5
TRUE
petal width
< 1.5
TRUE
FALSE
gini = 0
35T/0F
13T/6F
Leaf Node

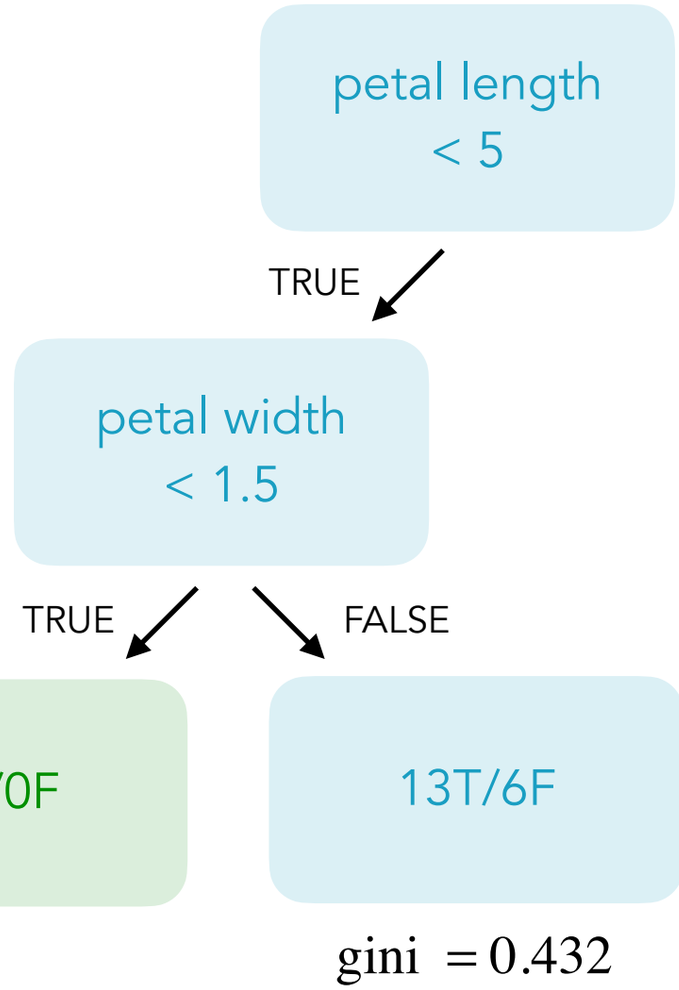


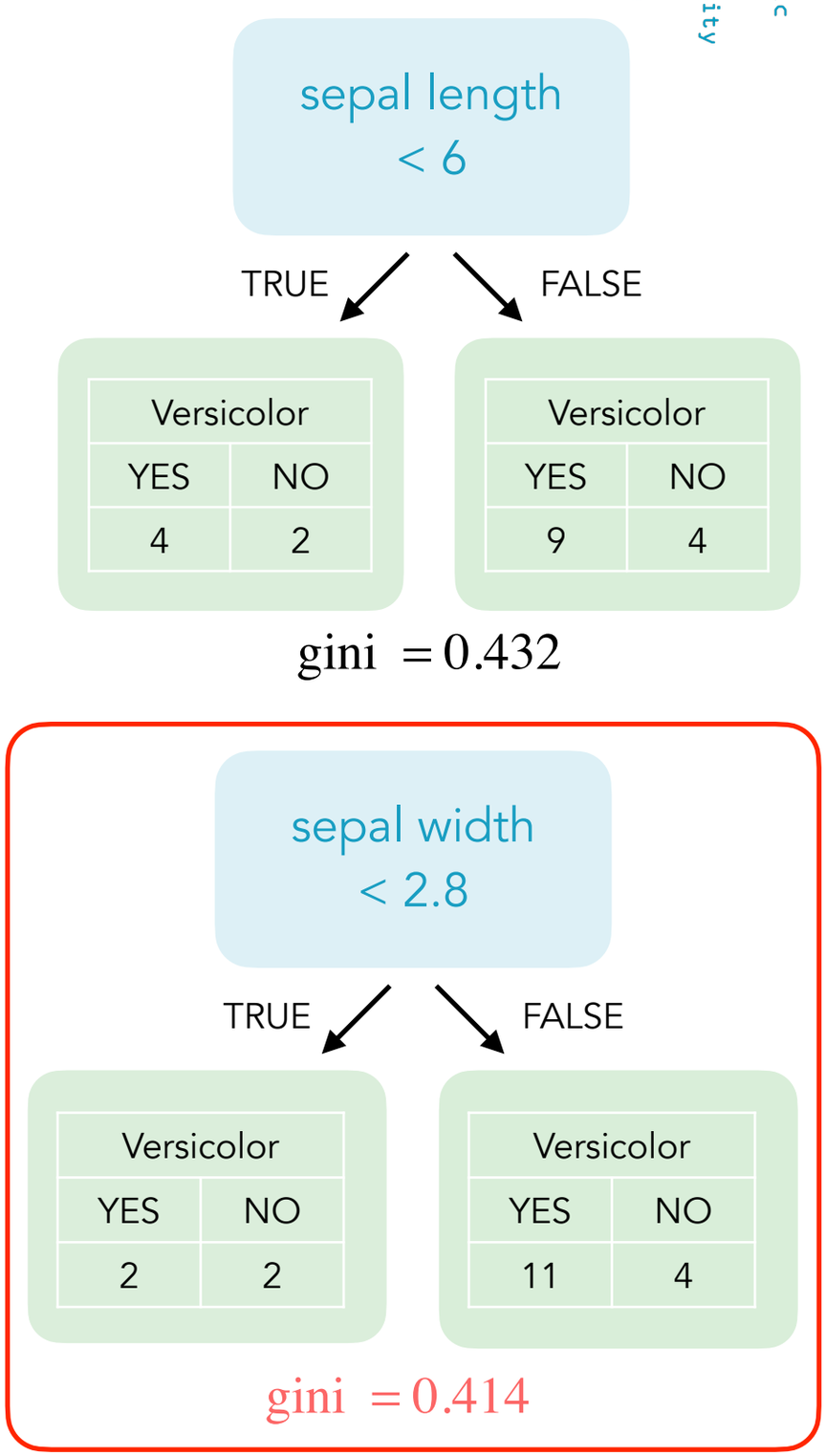

$$\text{gini} = 1 - \left(\frac{13}{13+6}\right)^2 - \left(\frac{6}{13+6}\right)^2 = 0.432$$

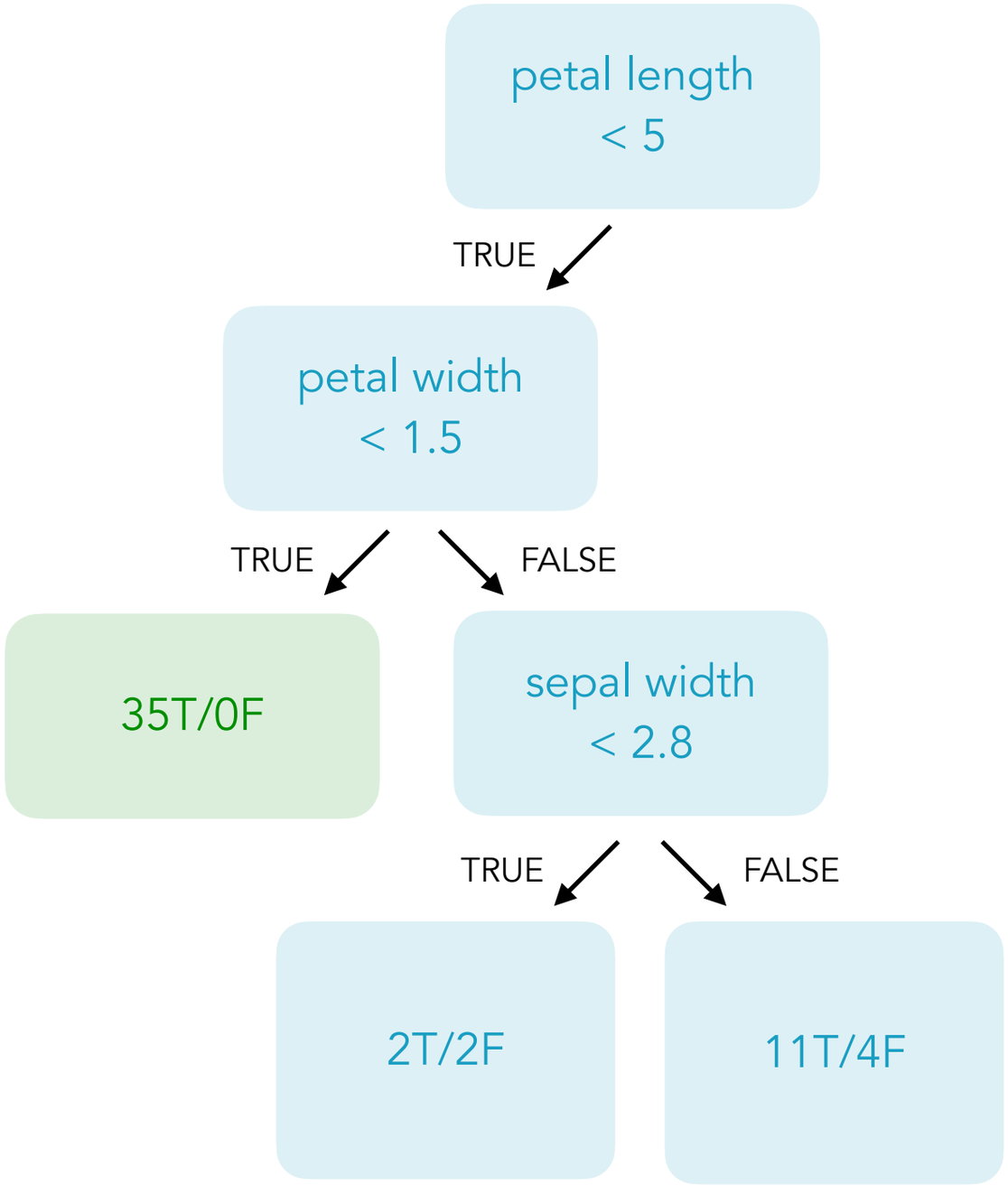
GDC Genetic Diversity Centre Zurich

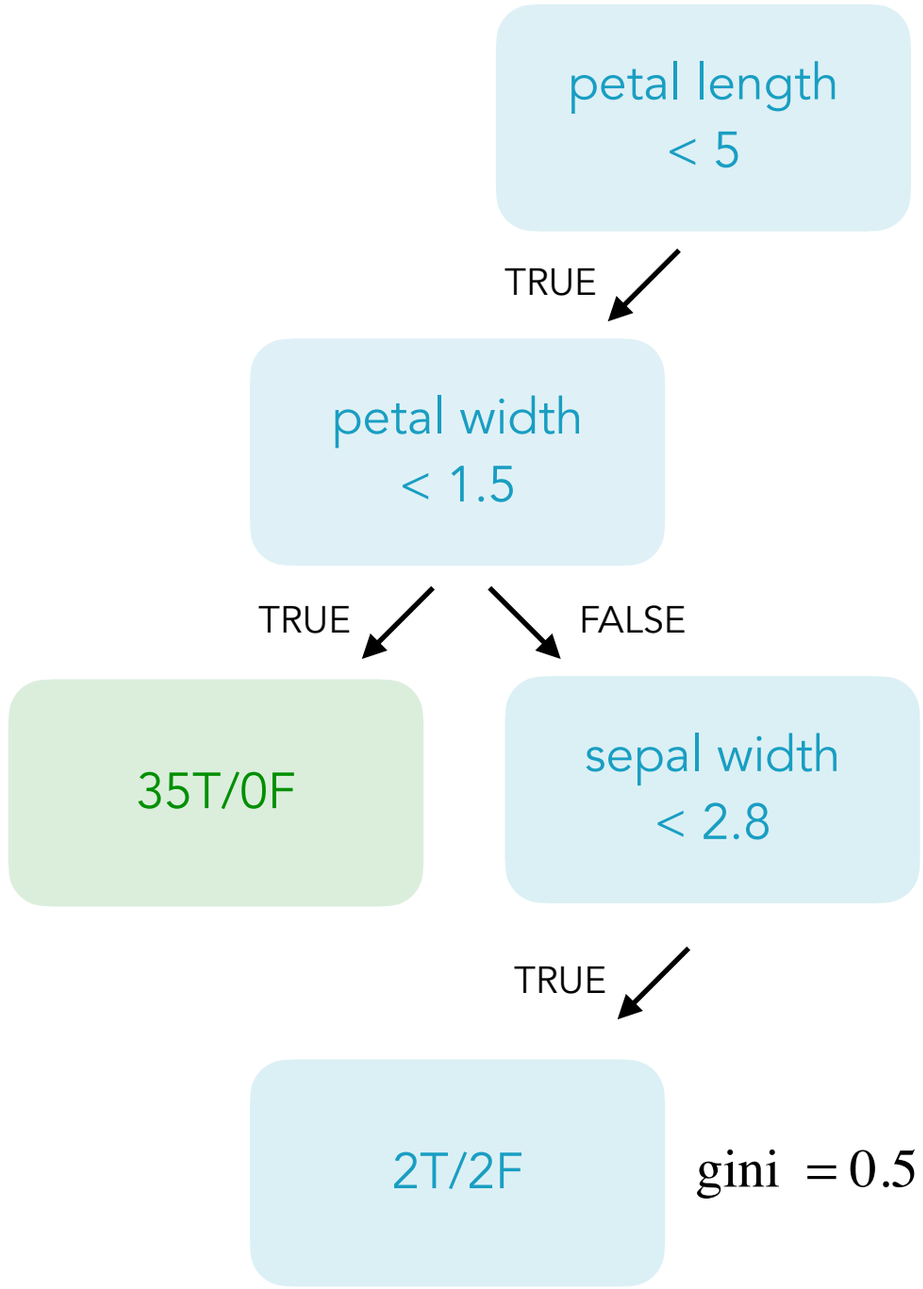


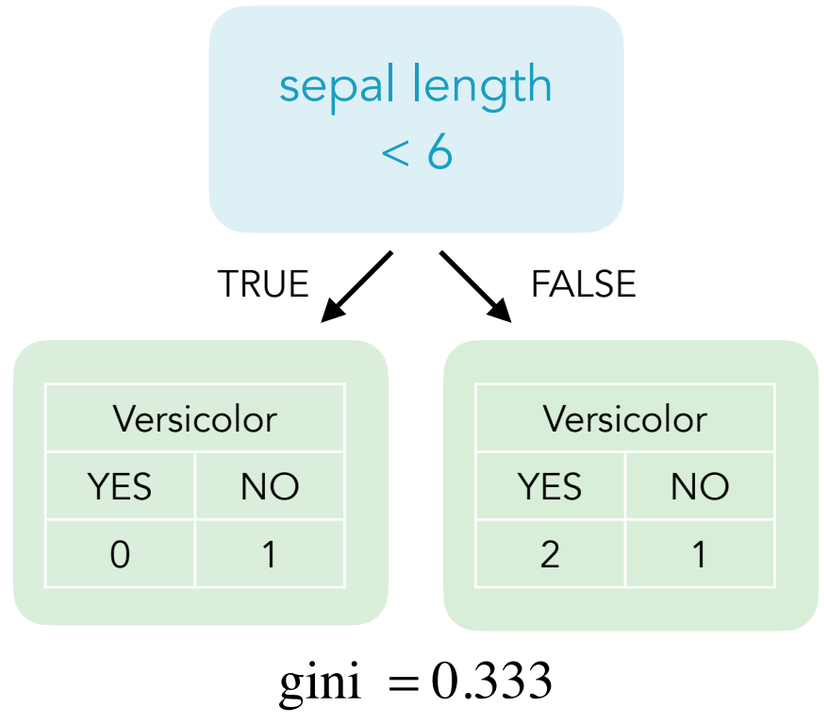



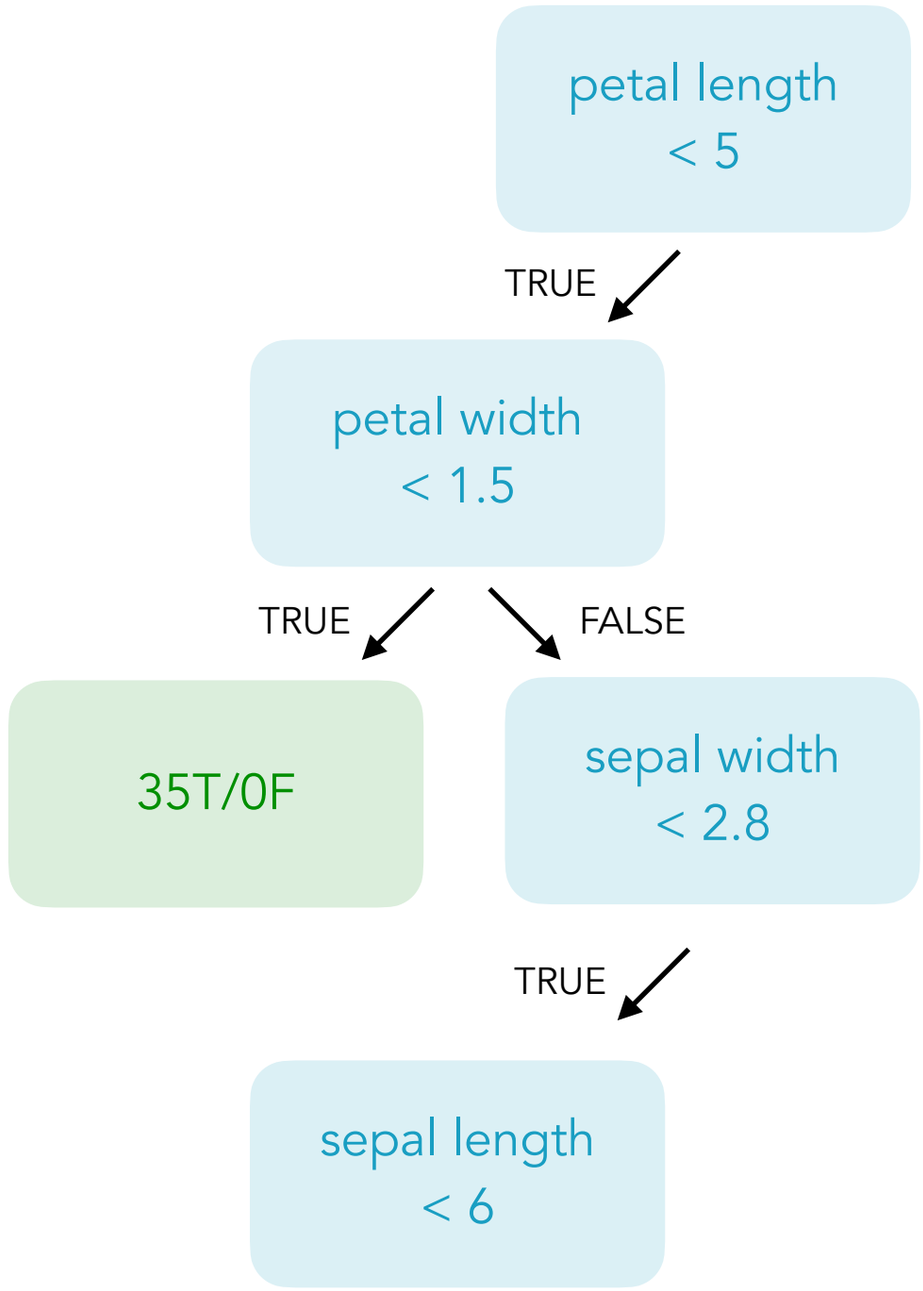

petal length
< 5
TRUE
petal width
< 1.5
TRUE
FALSE
35T/0F
13T/6F
gini = 0.432

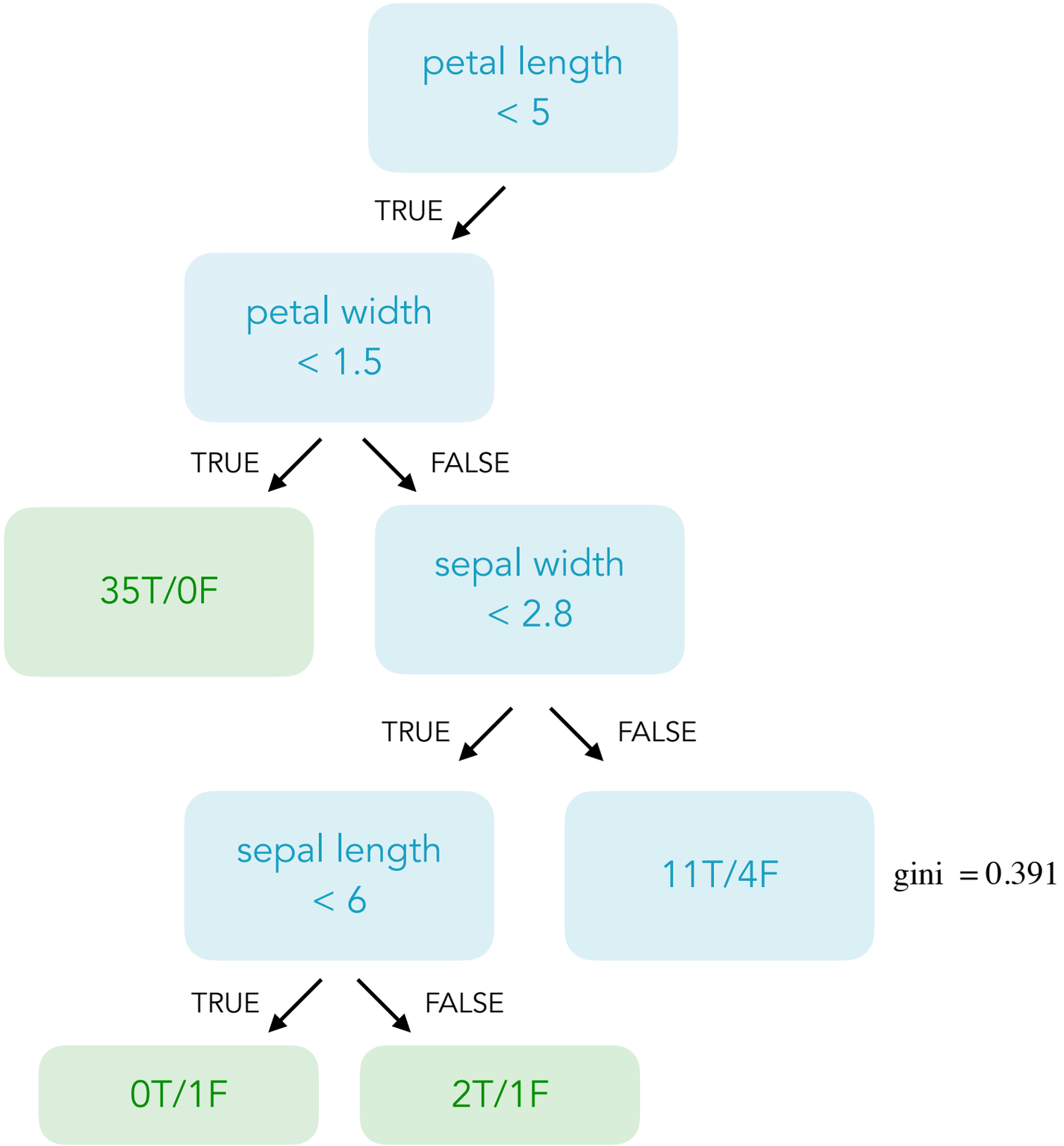

sepal length
< 6
TRUE
FALSE
Versicolor
YES
NO
4
2
Versicolor
YES
NO
9
4
gini = 0.432
sepal width
< 2.8
TRUE
FALSE
Versicolor
YES
NO
2
2
Versicolor
YES
NO
11
4
gini = 0.414


petal length
< 5
TRUE
petal width
< 1.5
TRUE
FALSE
35T/0F
sepal width
< 2.8
TRUE
FALSE
2T/2F
11T/4F


petal length
< 5
TRUE
petal width
< 1.5
TRUE
FALSE
35T/0F
sepal width
< 2.8
TRUE
2T/2F
gini = 0.5

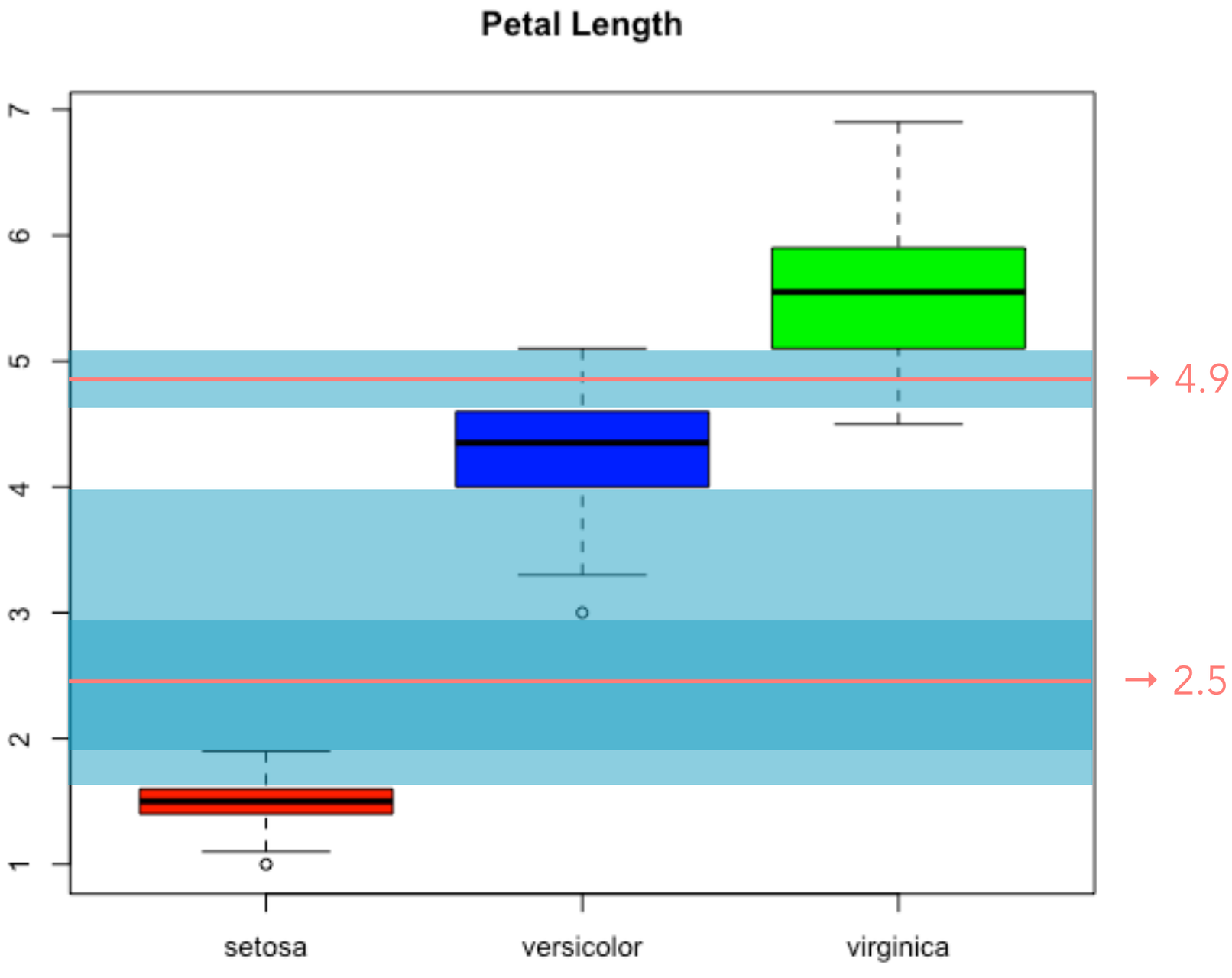

sepal length
< 6
TRUE
FALSE
Versicolor
YES
NO
0
1
Versicolor
YES
NO
2
1
gini = 0.333


petal length
< 5
TRUE
petal width
< 1.5
TRUE
FALSE
35T/0F
sepal width
< 2.8
TRUE
sepal length
< 6




$$\text{gini} = 0.391$$

Find good thresholds:



## Find good thresholds:

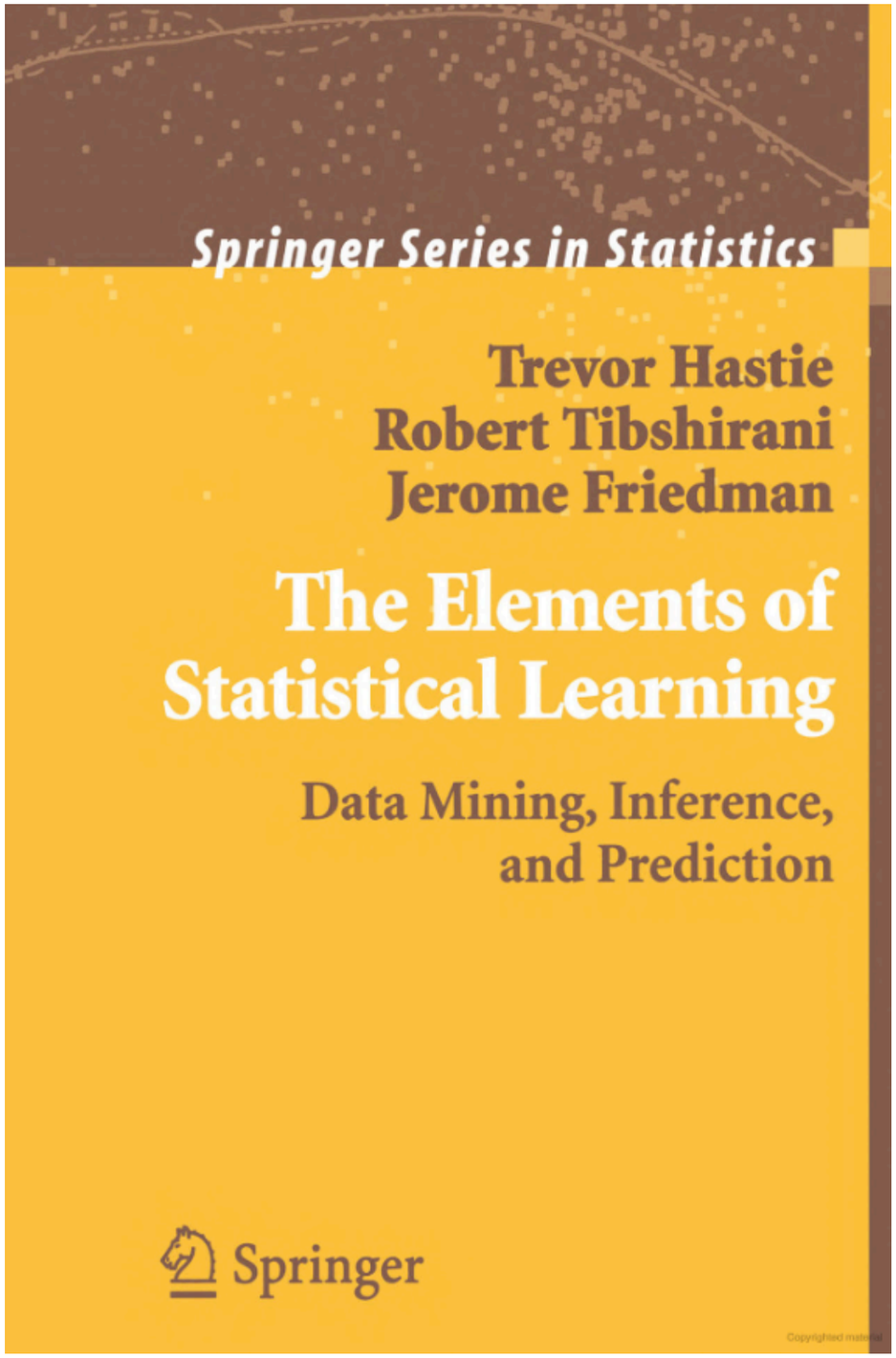
sort ↓

| | Sepal.Length | Sepal.Width | Species |
|---|---|---|---|
| 1 | 4.3 | 3.0 | setosa |
| 2 | 4.4 | 2.9 | setosa |
| 3 | 4.4 | 3.0 | setosa |
| 4 | 4.4 | 3.2 | setosa |
| 5 | 4.5 | 2.3 | setosa |
| 6 | 4.6 | 3.1 | setosa |
| 7 | 4.6 | 3.4 | setosa |
| 8 | 4.6 | 3.6 | setosa |
| 9 | 4.6 | 3.2 | setosa |
| 10 | 4.7 | 3.2 | setosa |
| 11 | 4.7 | 3.2 | setosa |
| 12 | 4.8 | 3.4 | setosa |
| 13 | 4.8 | 3.0 | setosa |
| 14 | 4.8 | 3.4 | setosa |
| 15 | 4.8 | 3.1 | setosa |
| 16 | 4.8 | 3.0 | setosa |
| 17 | 4.9 | 3.0 | setosa |
| 18 | 4.9 | 3.1 | setosa |
| … | … | … | … |

gini impurity for $4.35 = ?$

gini impurity for $4.4 = ?$

Calculate impurity values for each mean sepal length and chose the mean with the lowest Gini value as threshold.




"Trees have one aspect that prevents them from being the ideal tool for predictive learning, namely **inaccuracy**. They seldom provide predictive accuracy comparable to the best that can be achieved with the data a hand."

"Boosting or bagging decision trees can improves accuracy."

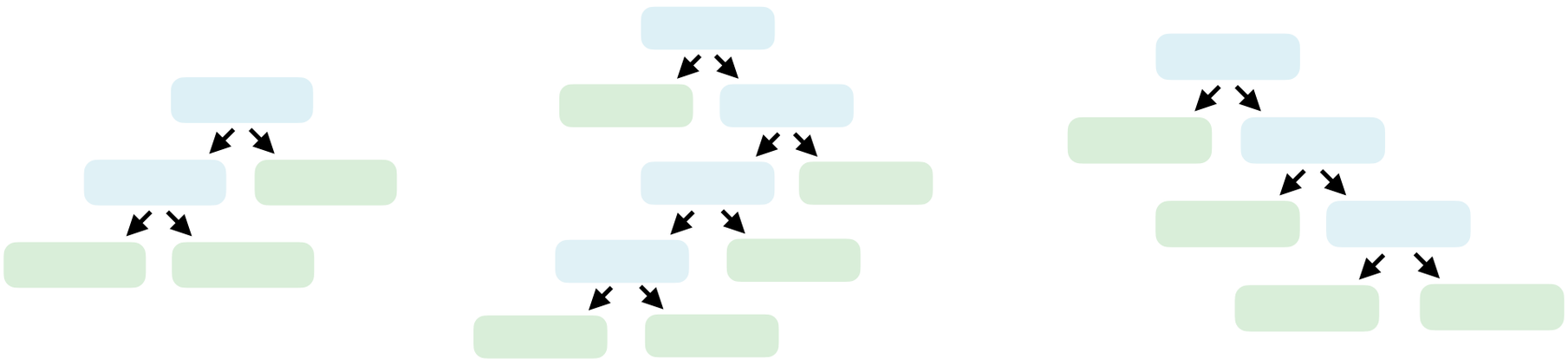
multiple versions of a prediction model

“wisdom of the crowd”

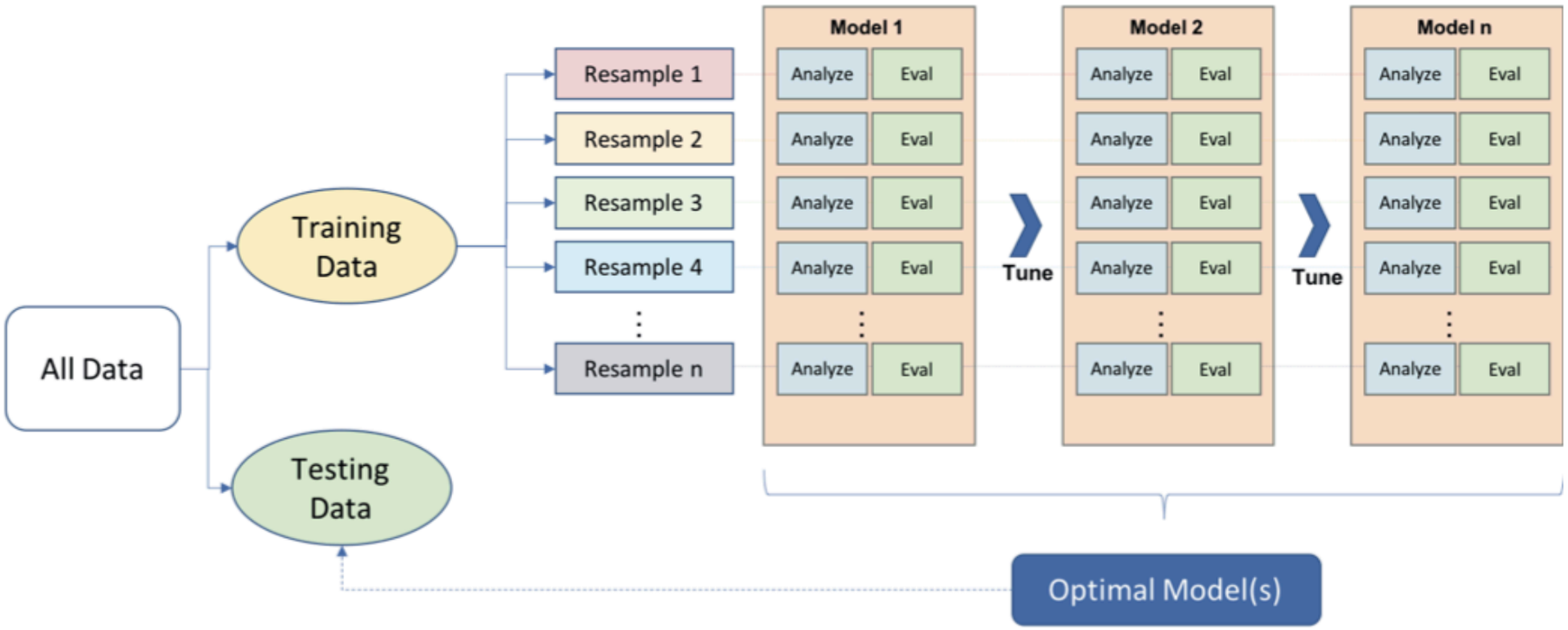
aggregated prediction

Bootstrap aggregating, or **bagging**, aims to enhance the precision and stability of classification and regression algorithms. Its purpose is to avoid overfitting the data by generating multiple models on different subsets of the original dataset. The final prediction is then made by combining the outcomes of each model. Bagging can help reduce variance and minimize overfitting, and it may be applied to any type of method, although it is most commonly used with decision tree methods.

# Random Forests



Random forests are a modification of bagged decision trees that build a large collection of trees to further improve predictive performance. They have become a very popular "out-of-the-box" or "off-the-shelf" learning algorithm that has good predictive performance with relatively little hyperparameter tuning.

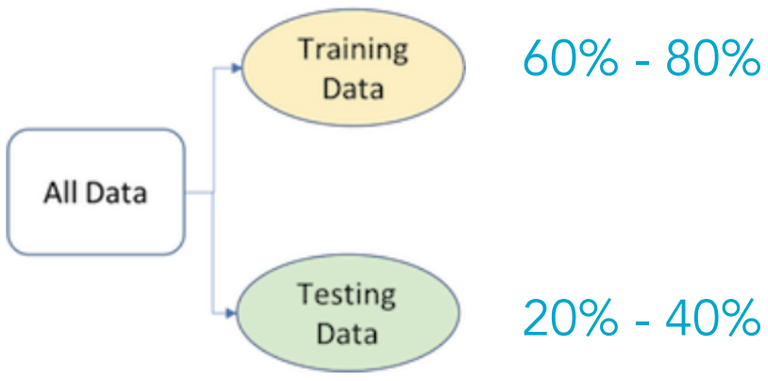



The iterative and heuristic nature of machine learning renders it challenging to determine the optimal approach for a specific problem or dataset with limited knowledge. Consequently, it is typical to employ, assess, and modify various machine learning techniques before achieving the ultimate ideal model. To guarantee precise and efficient ML modelling, strategic allocation of data towards learning and validation procedures is pivotal. Correctly processing feature and target variables, limiting data leakage, fine-tuning hyperparameters, and evaluating performance metrics are also essential.

Source: Boehmke & Grenwell (2020)

A significant aim of the machine learning procedure is to discover an algorithm $f(X)$ that predicts future values ($Y$) with optimum accuracy relying on a set of features ($X$). We require an algorithm which not only matches our past data but also predicts future results precisely to ensure **generalizability** of our algorithm.

To attain an exact understanding of the level of generalisability of our ultimate optimum model, we can segregate our data into **training** and **test sets**.




With an adequate sample size, the distribution of $Y$ should be comparable between the training and testing data sets.

The training set is intended for developing feature sets, algorithm training, hyperparameter refinement, model comparison, and all other necessary actions to choose a final model.

After deciding on the final model, the test dataset is used to impartially evaluate the model's accuracy, known as the generalisation error.

Source: Boehmke & Grenwell (2020)

GDC Genetic Diversity Centre Zurich

# Resampling Method: **Bootstrap**

## Original Dataset (N=10)

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 7.0 | 3.2 | 4.7 | 1.4 | versicolor |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 6.3 | 3.3 | 6.0 | 2.5 | virginica |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |
| 6.5 | 3.0 | 5.8 | 2.2 | virginica |

## 1. Create a bootstrap dataset

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |

A **bootstrap** sample is a randomly selected sample of data that is taken **with replacement**. The size of the bootstrap sample is identical to the original dataset used to construct it. Additionally, the distribution of values in the bootstrap sample is expected to closely resemble that of the original dataset. As samples are drawn with replacement, it is probable that each bootstrap sample will contain duplicate values. In fact, approximately 63.21% of the initial sample is included in each individual bootstrap sample on average.

GDC Genetic Diversity Centre Zurich

## Original Dataset (N=10)

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 7.0 | 3.2 | 4.7 | 1.4 | versicolor |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 6.3 | 3.3 | 6.0 | 2.5 | virginica |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |
| 6.5 | 3.0 | 5.8 | 2.2 | virginica |

## 1. Create a bootstrap dataset

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |

single values

GDC Genetic Diversity Centre Zurich

# Idea Bootstrap

## Original Dataset (N=10)

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 7.0 | 3.2 | 4.7 | 1.4 | versicolor |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 6.3 | 3.3 | 6.0 | 2.5 | virginica |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |
| 6.5 | 3.0 | 5.8 | 2.2 | virginica |

## 1. Create a bootstrap dataset

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |

duplicated values

GDC Genetic Diversity Centre Zurich

## Original Dataset (N=10)

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 7.0 | 3.2 | 4.7 | 1.4 | versicolor |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 6.3 | 3.3 | 6.0 | 2.5 | virginica |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |
| 6.5 | 3.0 | 5.8 | 2.2 | virginica |

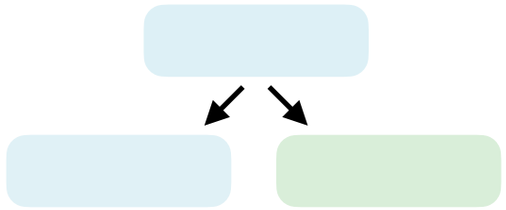
## 1. Create a bootstrap dataset

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |

The original observations not contained in a particular bootstrap sample are considered out-of-bag (OOB).

GDC Genetic Diversity Centre Zurich

## 2. Create a decision tree but only use a random subset of variables (mtry) at each node.



## 1. Create a bootstrap dataset

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |

mtry=2 (hyperparameter)

starting point: $\text{mtry} = \sqrt{N_{\text{variables}}}$

GDC Genetic Diversity Centre Zurich

3. Repeat (ntree)

1. Create a **bootstrap** dataset

2. Create a decision tree but only use a **random subset of variables** (mtry) at each node.

RF-Model

## Bagging

Bootstrapping means taking a sample of a population by drawing with replacement. It is one of the main ideas behind Bagging (which stands for Bootstrap AGGregatING).

Note: ntree and ntry are both hyperparameters.

GDC Genetic Diversity Centre Zurich

# Estimate accuracy of the random forest

## Original Dataset (N=10)

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 7.0 | 3.2 | 4.7 | 1.4 | versicolor |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 6.3 | 3.3 | 6.0 | 2.5 | virginica |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |
| 6.5 | 3.0 | 5.8 | 2.2 | virginica |

## 1. Create a bootstrap dataset

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica |

About 1/3 of the original data is not being used in the bootstrap dataset.

**> Out-of-Bag Dataset**

## Out-of-Bag (OOB) Dataset

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 7.0 | 3.2 | 4.7 | 1.4 | versicolor |
| 6.3 | 3.3 | 6.0 | 2.5 | virginica |
| 6.5 | 3.0 | 5.8 | 2.2 | virginica |

Estimate accuracy of the random forest

| Versicolor | |
|---|---|
| YES | NO |
| 601 | 399 |

| Versicolor | |
|---|---|
| YES | NO |
| 644 | 356 |

| Versicolor | |
|---|---|
| YES | NO |
| 468 | 532 |

OOB Error Rate ▸ mtry

New Sample

| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
|---|---|---|---|---|
| 5.70 | 2.15 | 3.98 | 1.31 | ??? |

RF-Model

| Versicolor | |
|---|---|
| **YES** | NO |
| **987** | 13 |

*Iris versicolor*

*OOB Error rate: x%*

GDC Genetic Diversity Centre Zurich

## Random Forests with R

```
library(randomForest); packageVersion("randomForest")
# 4.6.14
set.seed(190717)
new.iris.rf <- randomForest(Species ~ ., data = new.iris,
                            mtry = 2,
                            ntree = 1000)
print(new.iris.rf)
```

```
Call:
 randomForest(formula = Species ~ ., data = new.iris, mtry = 2, ntree = 1000)
               Type of random forest: classification
                     Number of trees: 1000
No. of variables tried at each split: 2

        OOB estimate of  error rate: 7%
Confusion matrix:
           versicolor virginica class.error
versicolor         47         3        0.06
virginica           4        46        0.08
```

## Predict Outcome

```
new.iris[c(1,100),]
#     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
# 1            7.0         3.2          4.7         1.4 versicolor
# 100          5.9         3.0          5.1         1.8  virginica

new.data      <- data.frame(new.iris[c(1,100),])
new.data.pred <- predict(new.iris.rf, new.data)
table(observed = new.data$Species, predicted = new.data.pred)
#            predicted
# observed    versicolor virginica
# versicolor           1         0
# virginica            0         1
```

# Advantages

# Limitations

It is a versatile method because it can handle both classification and regression problems.

Random forest classifier won't overfit the model.

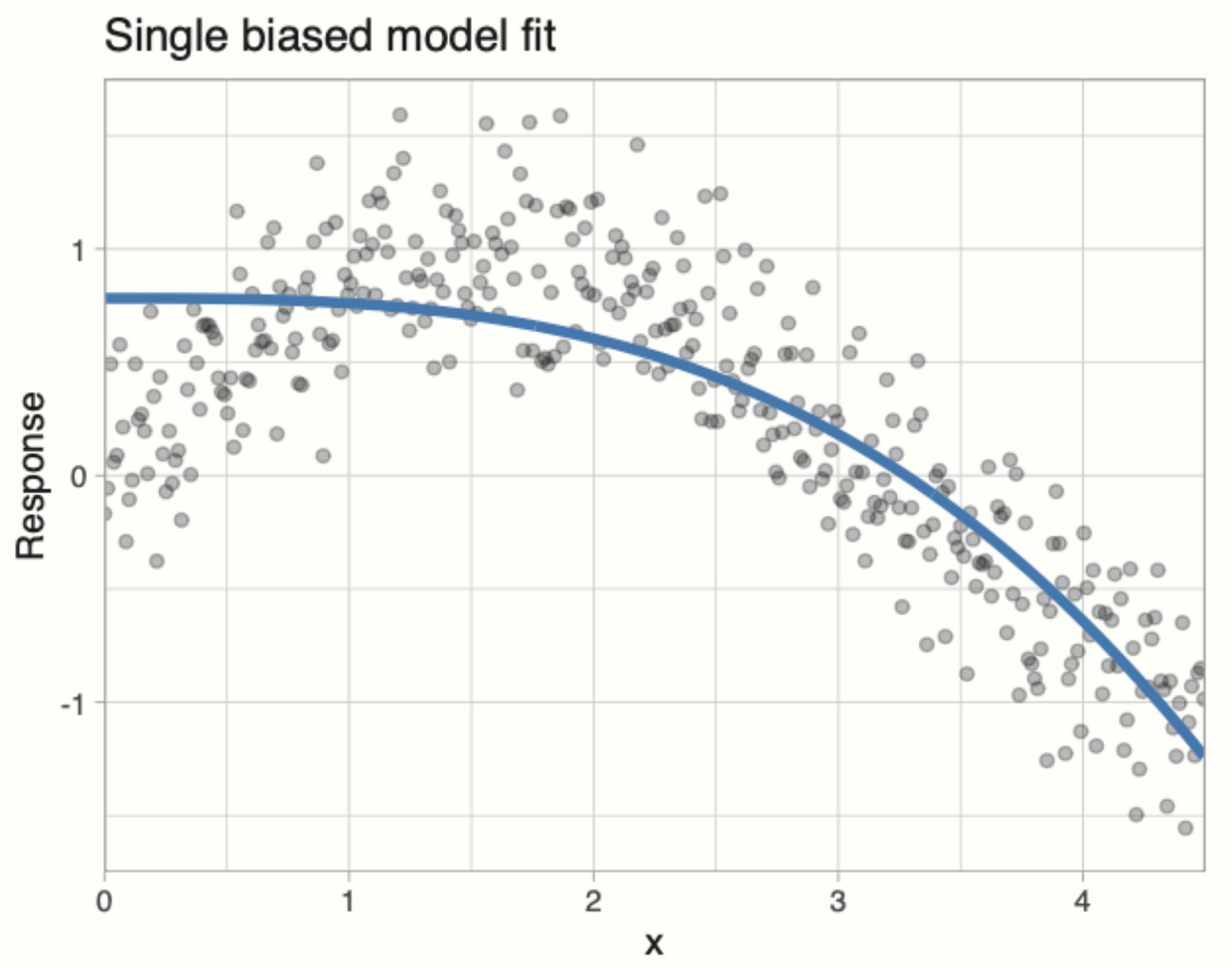
A considerable number of trees can impede the efficiency of the algorithm, rendering it unsuitable for real-time predictions.

This tool is designed for predictive modelling rather than descriptive purposes. It is recommended to use alternative approaches if you require a description of the relationships within your data.

# Extra

# Bias variance trade-off

**Prediction errors** can be divided into two subcomponents: "**bias**" and "**variance**". There is often a tradeoff between minimising bias and variance. By understanding how various error sources lead to bias and variance, we can enhance the data fitting process and achieve more accurate models.

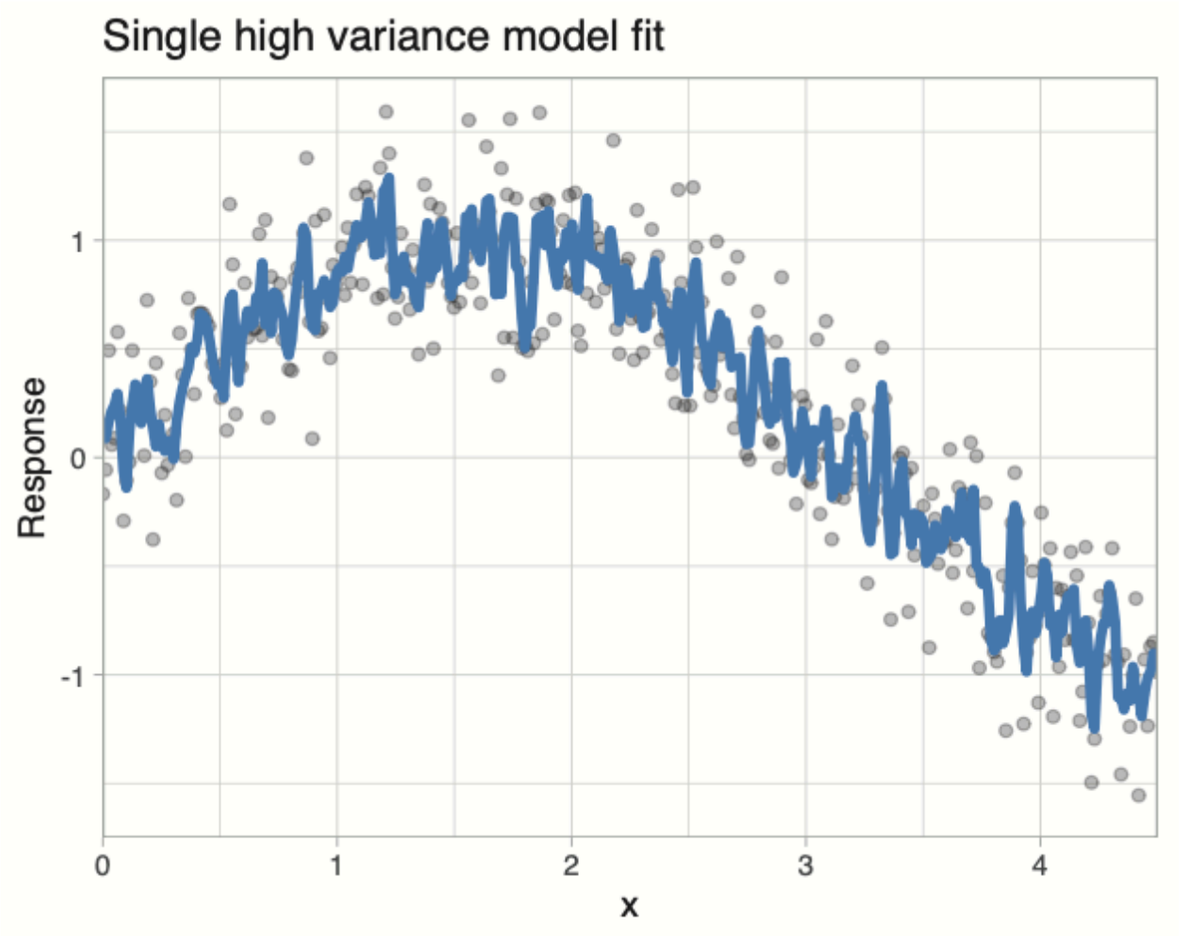



**Bias** is the divergence between our model's anticipated (or mean) prediction and the true value we aim to predict. Linear models are classical examples of high-bias models as they are less flexible and often fail to capture non-linear, non-monotonic relationships.

Source: Boehmke & Grenwell (2020)

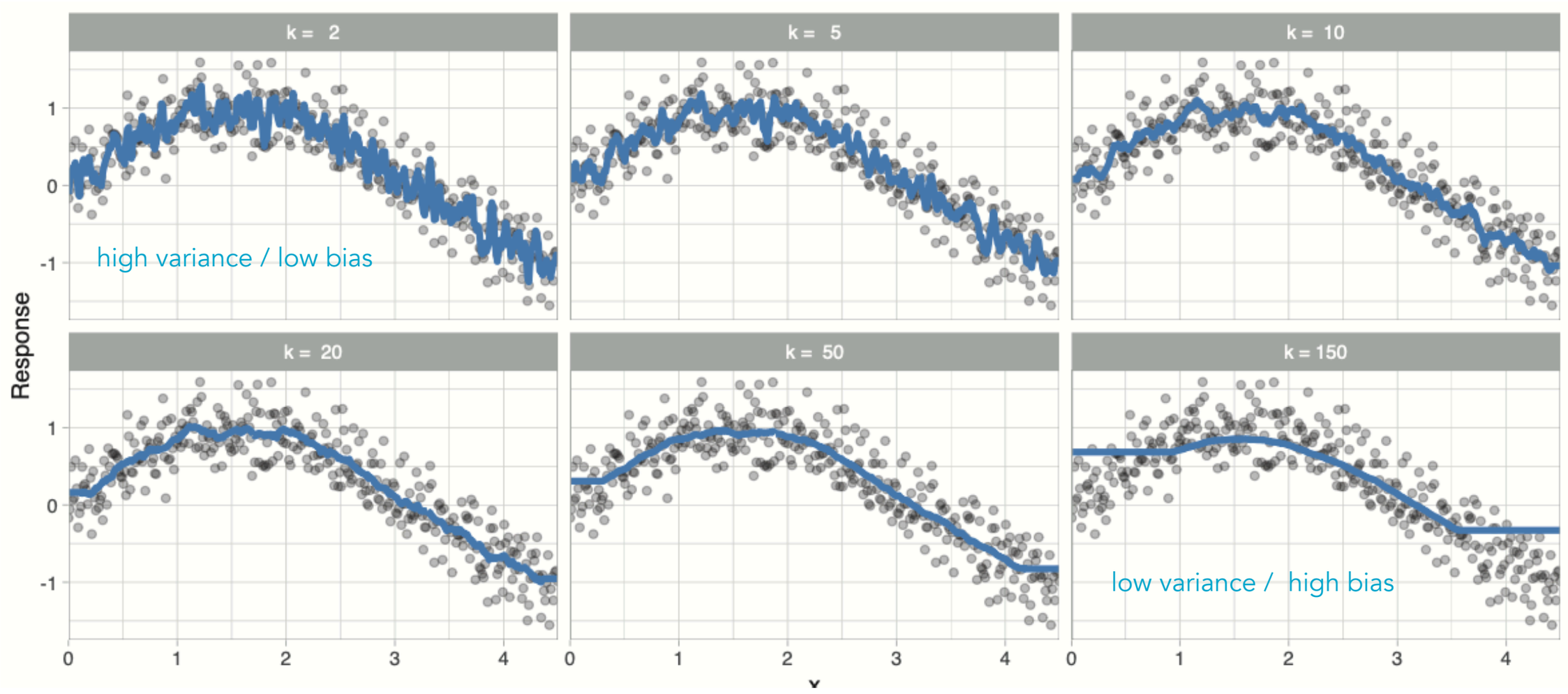
A biased polynomial model fit to a single data set does not capture the underlying non-linear, non-monotonic data structure.

Error due to **variance** is defined as the variability of a model prediction for a given data point.



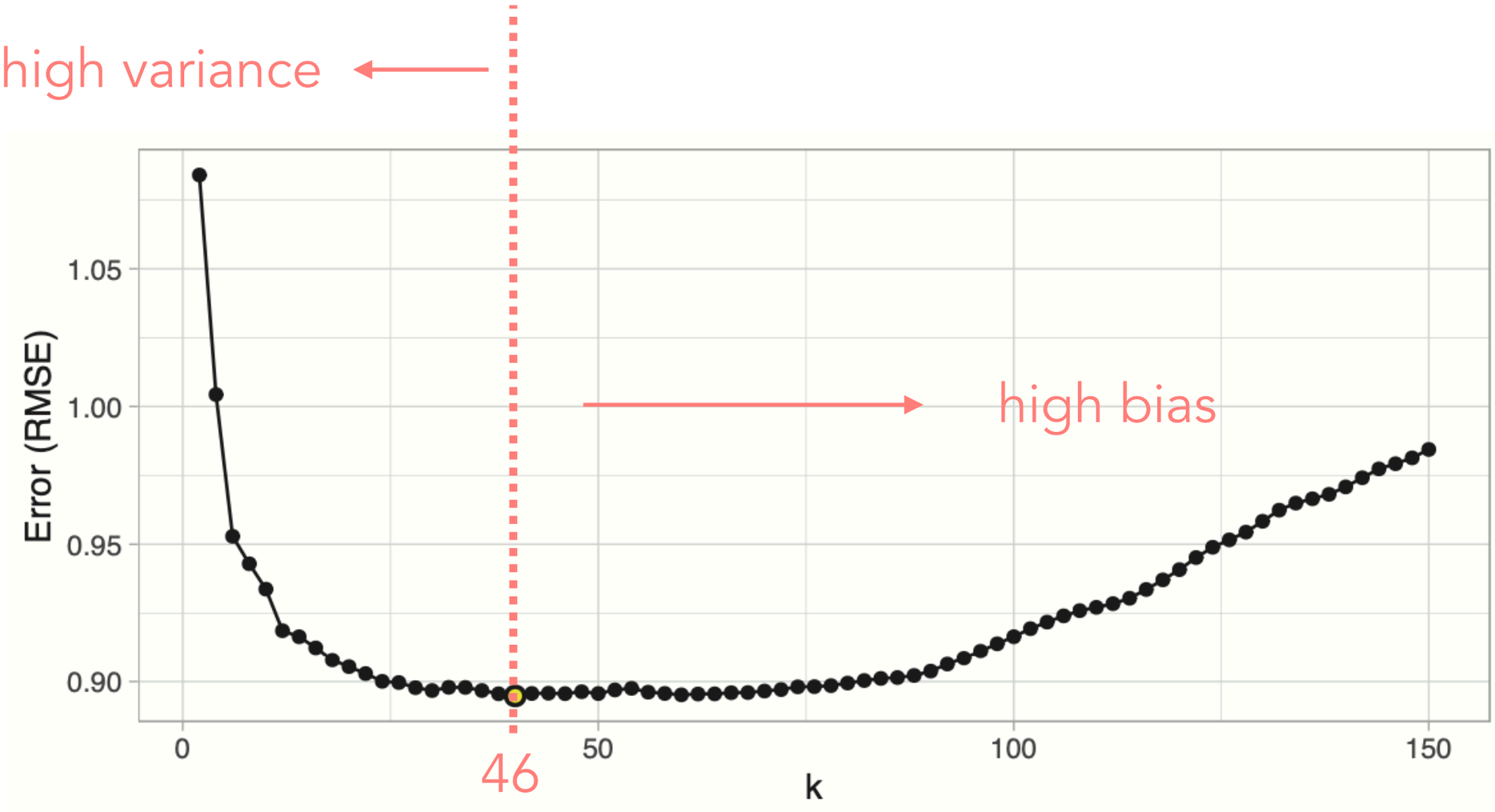

Source: Boehmke & Grenwell (2020)

A k-nearest neighbor model with high variance fitted to a single data set effectively captures the non-linear, non-monotonic data structure but may overfit to individual data points.




**K-nearest neighbour models** possess one hyperparameter (k) that dictates the predicted value derived from the k nearest observations to the one being forecasted in the training data. If the value of k is low, the model will predict the response value for a given observation by taking the average of response values for the k observations in the training data that are most similar to the observation being predicted.

Source: Boehmke & Grenwell (2020)




Results from a grid search involving the k-nearest neighbour model, where k values were evaluated between 2-150, indicate high error rates resulting from both high model variance for small k values and high model bias for large k values. The optimal model was identified at k = 46.

Source: Boehmke & Grenwell (2020)