Data Science

Topic: Data Mining

Karin Groothuis-Oudshoorn and Julia Mikhal

November 12th, 2024

Vandaag

- Organization Topic DM
- Basics of Data Mining (DM)
- Types of DM Methods
  - K-Nearest Neighbors (KNN)
  - Naive Bayes Method
  - Decision Trees (CART)
- Setting up a CRISP DM Analysis
- Summary

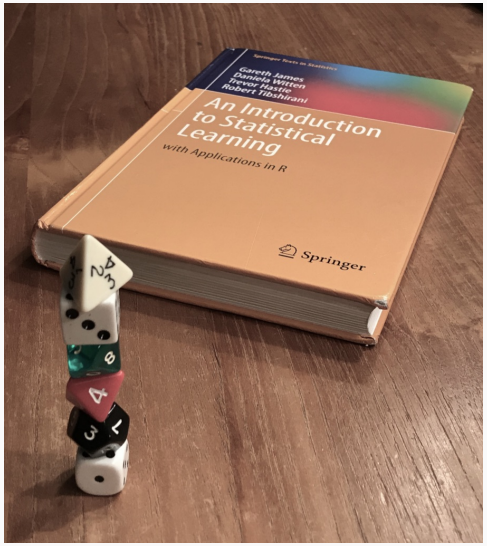
Organization Topic DM

Learning Goals

- Knowledge about some basic data mining (DM) methods
- Basic skills in data visualization, modelling using, e.g., R/Python
- Basic knowledge about the DM pipeline
- Basic skills in setting up a simple DM experiment using R/Python
- The role of the training and test set, cross-validation in the DM pipeline

Topic Data Mining: Material

- Book *Introduction to Statistical Learning with R*
- Freely available on the web (https://www.statlearning.com/)
- YouTube lectures by the authors themselves (see https://www.youtube.com/playlist?list=PLOg0ngHtcqbPTlZzRHA2ocQZqB1D_qZ5V)
- Chapter 1, 2, 3, 4, 5.1, 8.1, 10.3



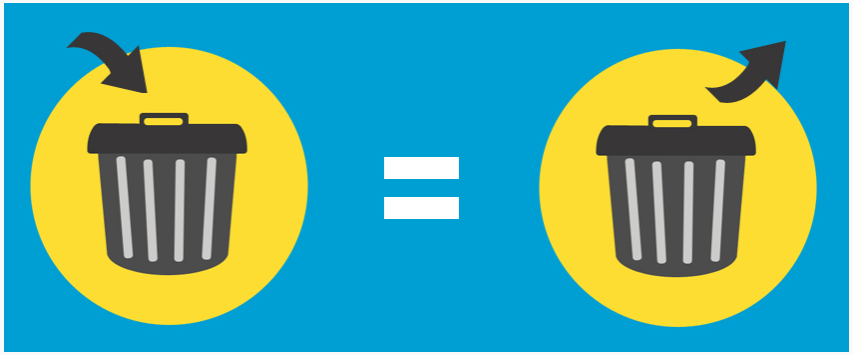
Basics of Data Mining (DM)

Data Mining and Data Science

- Data mining is **fundamental** in Data Science
- **What**: discovering patterns, correlations, anomalies, insights, trends from (large) datasets
- **Purpose**: to get insights of the data for decision-making, prediction and knowledge discovery
- Related to:
  - **Machine learning**: developing algorithms that enable computers to learn from data and make predictions or decisions
  - **Statistical learning**: providing a framework for understanding and analyzing data by modeling relationships and making predictions based on statistical principles and techniques
  - **Artificial Intelligence**: creating intelligent systems that can perform tasks autonomously

Data Mining

- Given lots of data
- Discover patterns and models that are:
  - **Valid**: hold on new data with some certainty
  - **Useful**: should be possible to act on the item
  - **Unexpected**: non-obvious to the system
  - **Understandable**: humans should be able to interpret the pattern



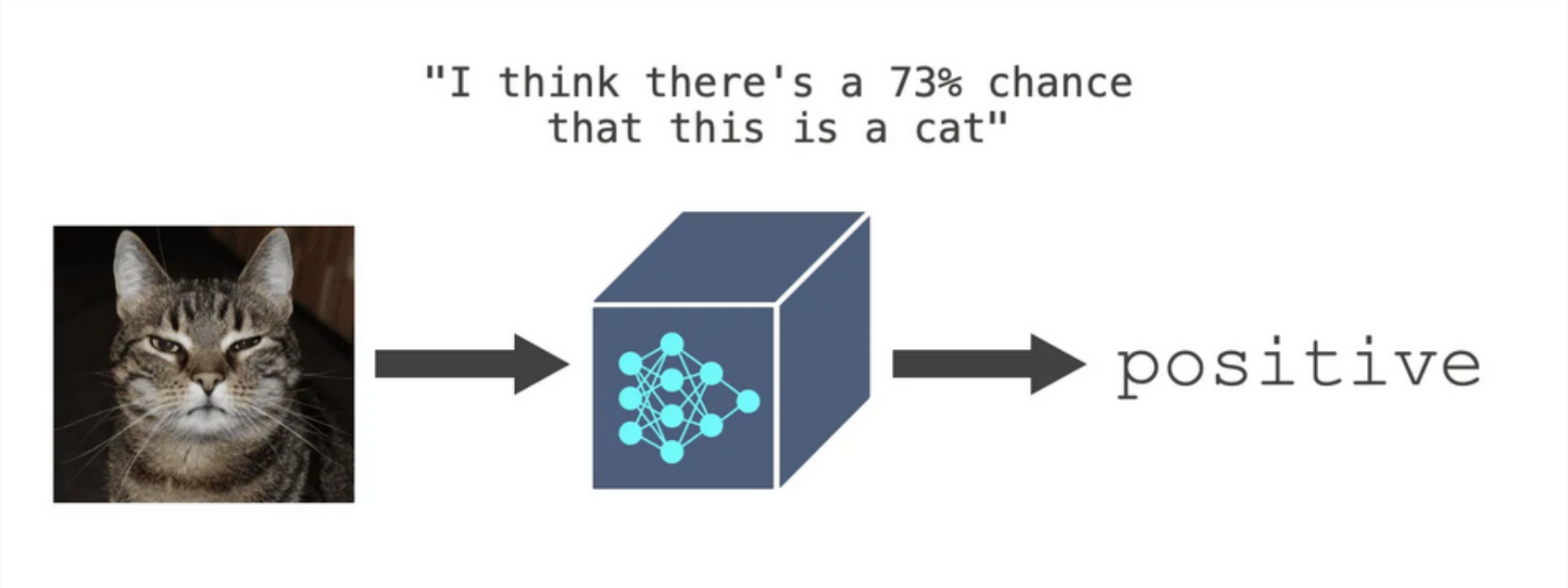
GARBAGE IN = GARBAGE OUT

Two Types of Data Mining Methods

- **Supervised** and **unsupervised** learning
- **Supervised** learning involves **training** a model for predicting or estimating (an output based on one or more inputs)
  - Training data includes desired outputs / labels
- With **unsupervised** learning the goal is to learn about relationships and structure of the data.
  - Training data does not include desired outputs / unlabeled

Supervised Learning

- **Regression** problem
  - Output is continuous
- **Classification** problem
  - Binary classification: two classes
  - Multi-class classification
  - Output is a binary or categorical value (based on a probability)



Examples Supervised / Unsupervised

- **Supervised**
  - prediction of credit card fraud (**classification**)
  - filtering out spam (**classification**)
  - convert hand-writing images into text (**classification**)
  - predicting house/property, stock market prices (**regression**)
- **Unsupervised**
  - identify groups of customers with a certain purchasing behavior (**clustering**)
  - identify patterns like: if a customer buys X then there is a tendency to buy Y also (**association**)

Applications in the Medical Domain (Supervised)

- Automatically composed advice for patients based on questionnaires, diagnostic information (**classification**)
- Automatic detection of atrial fibrillation (**classification**)
- Scheduling of OR: prediction of surgery duration (**regression**)
- Prediction of the time to fracture after the visit to osteoporosis poli (**regression**)
- Prediction of occurrence of a post-operative infectious complication (**classification**)
- Prediction of the length of stay after complex surgery (**regression**)

Classification or Regression?

Which of the following are classification problems? And which of them are regression problems?

- Predicting the gender of a person by his/her handwriting style
- Predicting house price based on area
- Predicting the nationality of a person
- Predicting the number of copies a music album will be sold next month
- Predicting whether the stock price of a company will increase tomorrow
- Predicting the probability of surviving after hip fracture surgery

Classification or Regression: Answers

Which of the following are classification problems? And which of them are regression problems?

- Predicting the gender of a person by his/her handwriting style: **classification**
- Predicting house price based on area: **regression**
- Predicting the nationality of a person: **classification**
- Predicting the number of copies a music album will be sold next month: **regression**
- Predicting whether the stock price of a company will increase tomorrow: **classification**
- Predicting the probability of surviving a after hip fracture surgery: **classification**

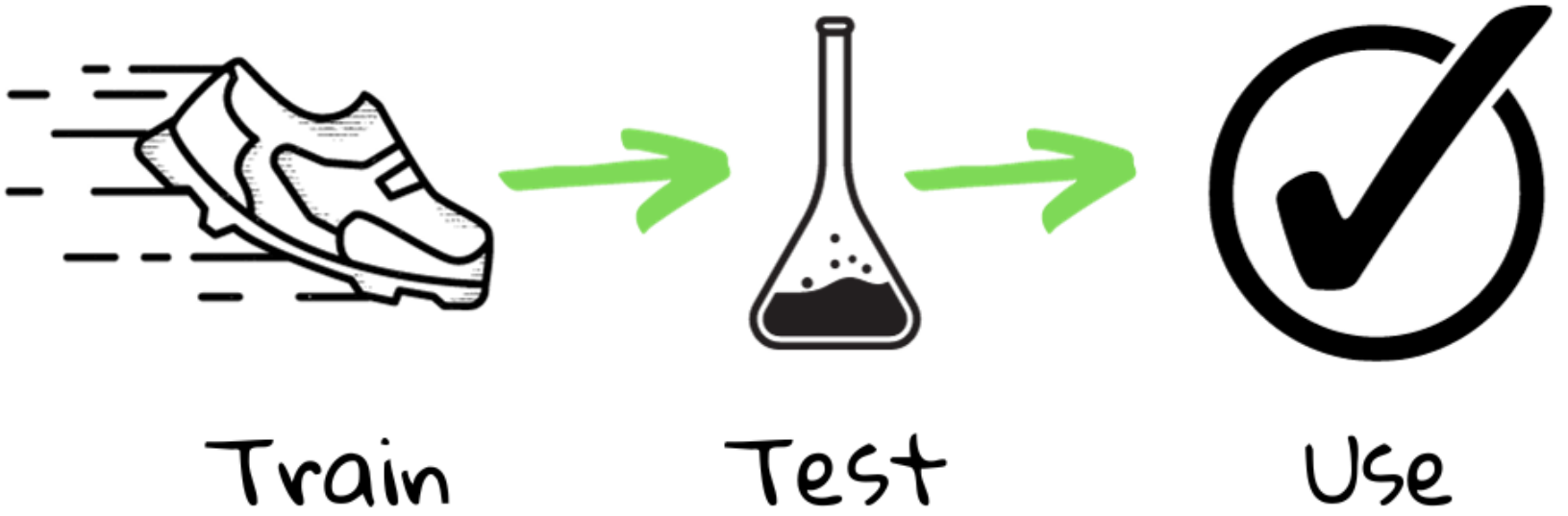
Terminology

- **Input**: feature, attribute, variable, covariate
- **Output**: dependent variable, response variable, label
- **Feature selection**: variable selection
- **Feature engineering**: variable transform, dummy coding
- **Method**: algorithm, approach or technique used to train a model on data (the estimator)
- **Model**: the trained outcome from applying a method to a dataset (the estimate)
- **Training**: process of teaching a model to make predictions or decisions by feeding it data
- **Learning**:the outcome of the training process

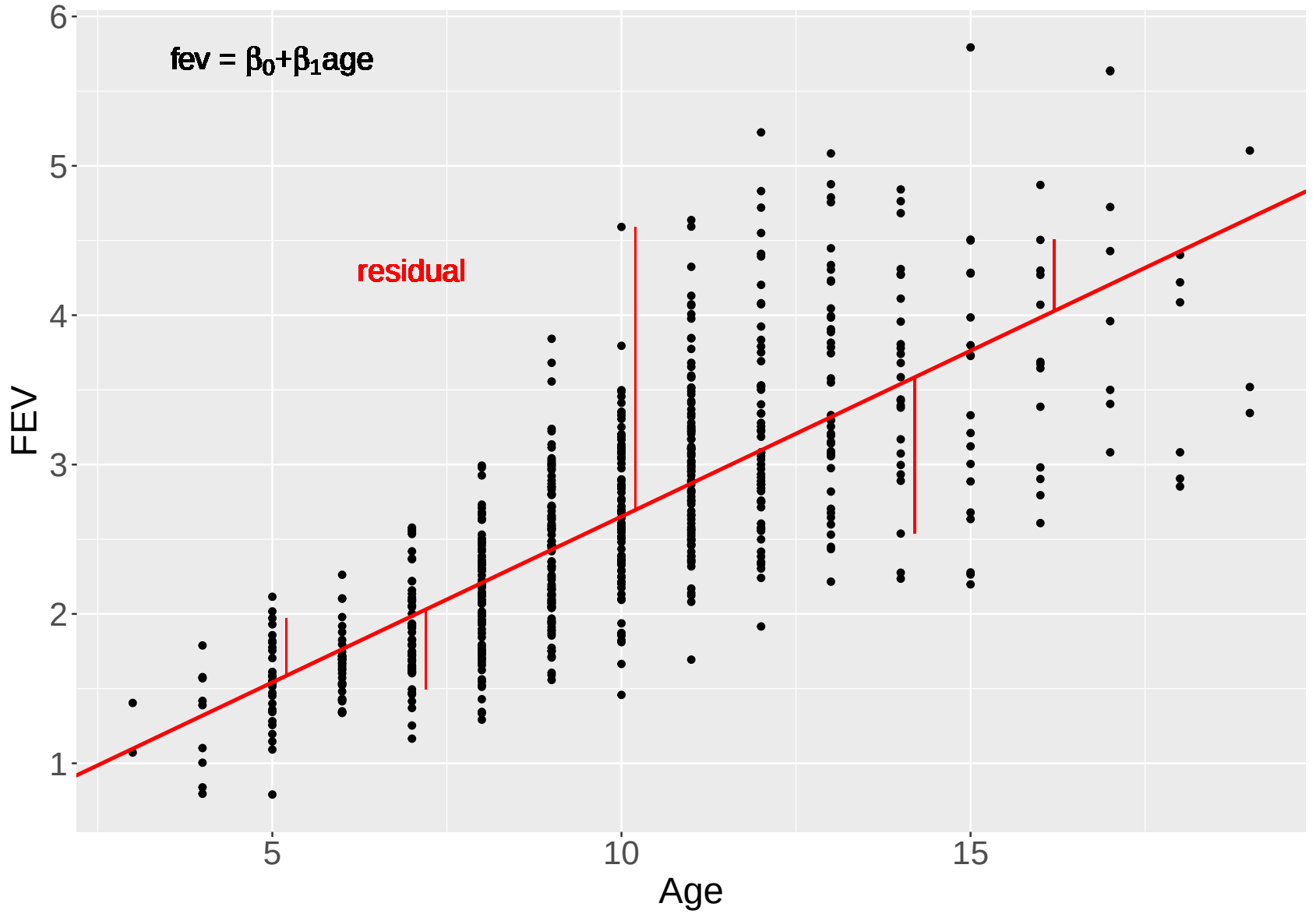
Training of a Model

- There are **many** models
- The more complex your model the better?
- How do we know how good your model is?
- How does your model perform on new data?

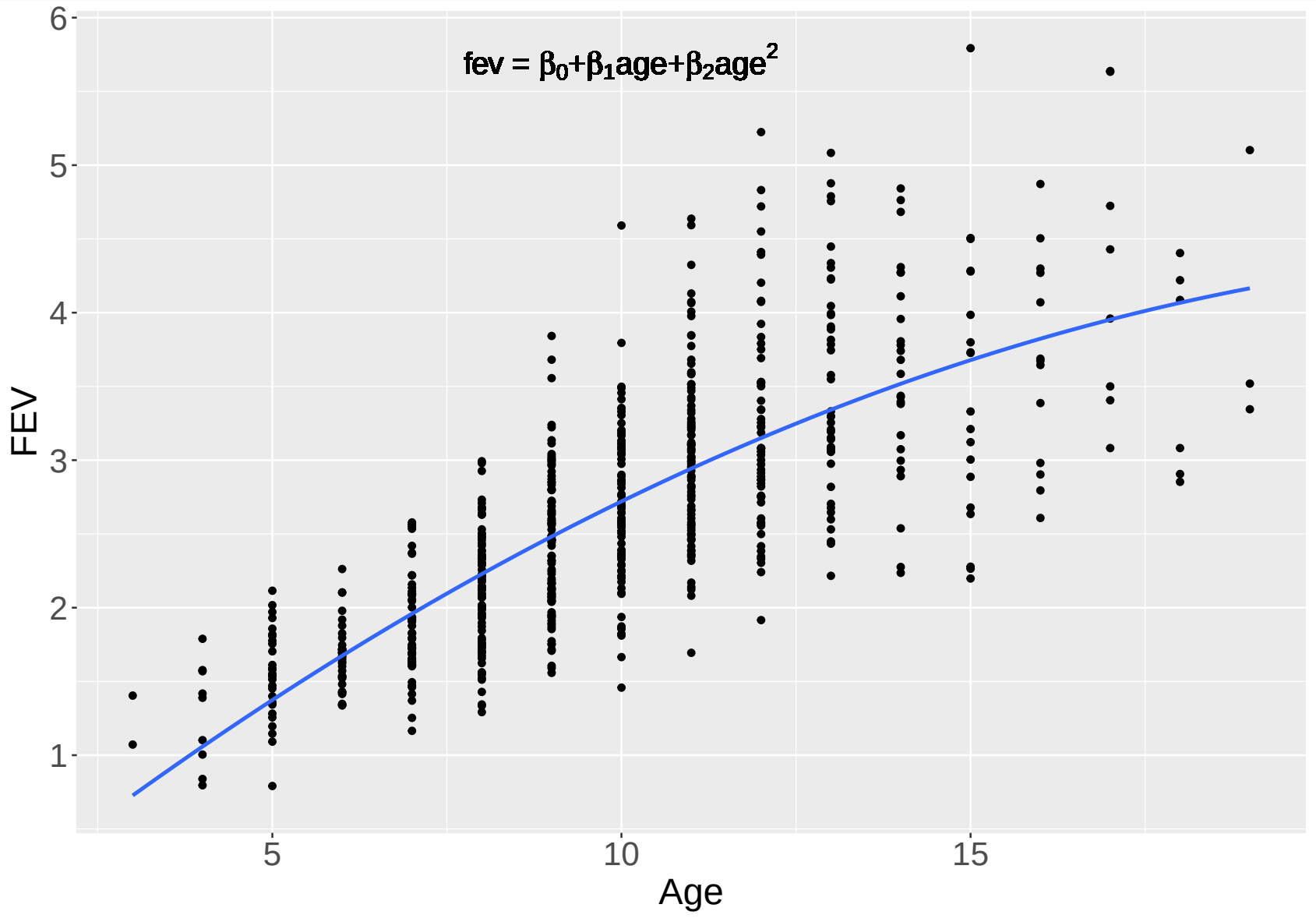
$\Rightarrow$ **Validation** of your model with unseen **test** data



Most Simple Method: Linear Regression (2 parameters)



Linear Regression (3 parameters)



Complexity

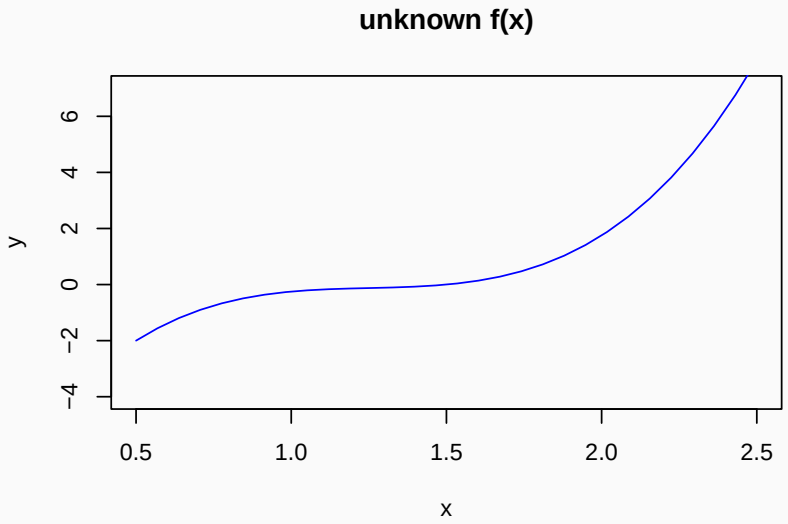
The number of parameters in a model reflects its **complexity** and **flexibility**. More parameters allow the model to capture finer details and nuances in the data.

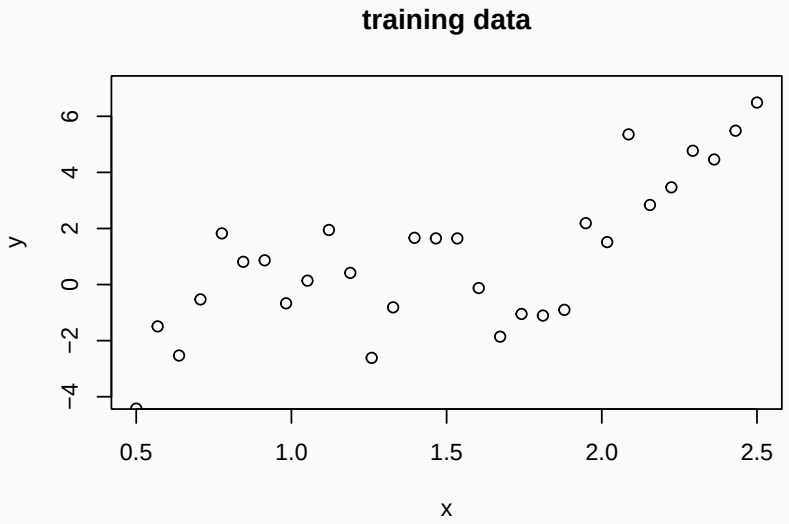
- The more complex a model the better?
  - Non linear terms (e.g., higher order polynomials)
  - More layers in your network
- The more features in your model the better?

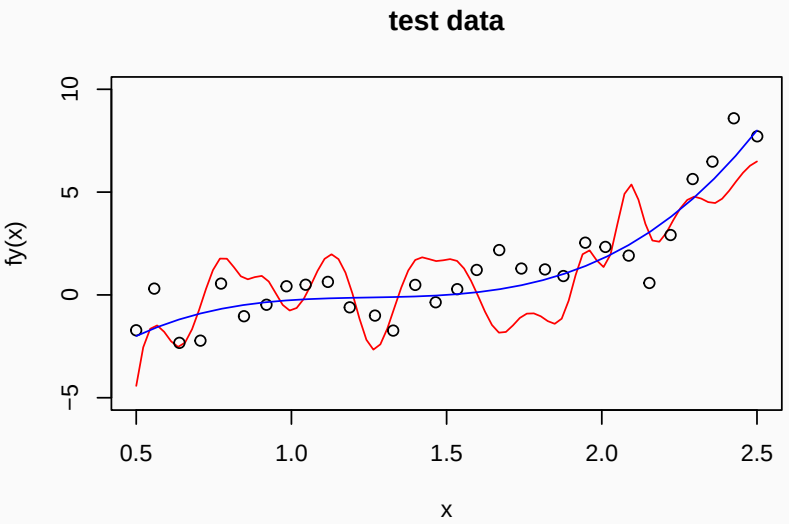
NO!

Beware of **OVERFITTING**!

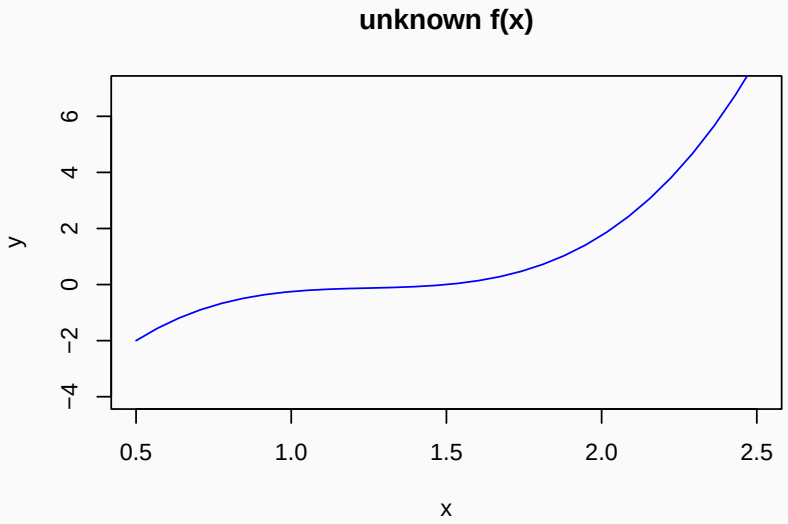
Overfitting

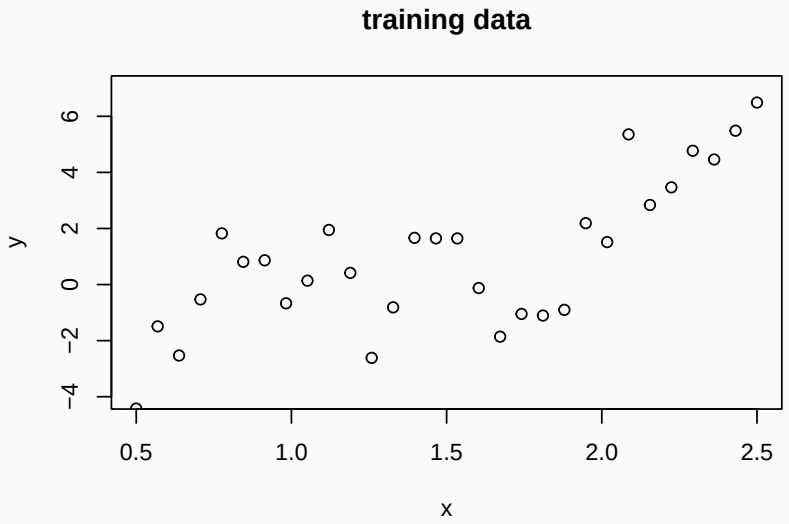


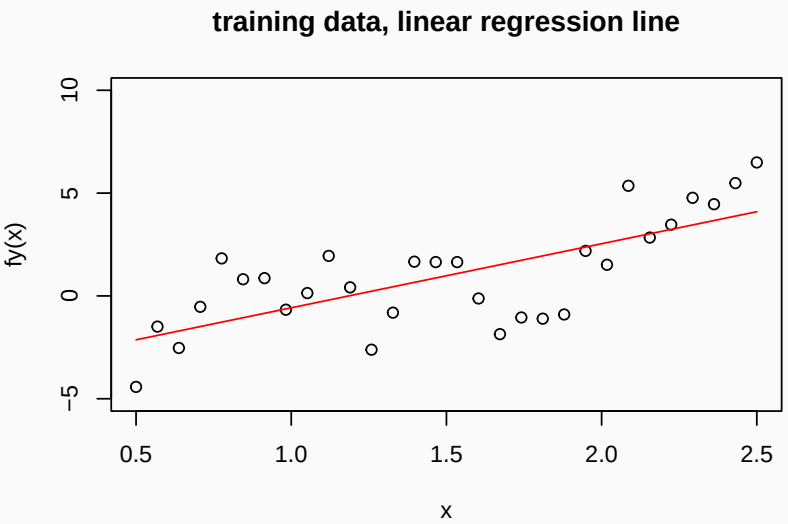


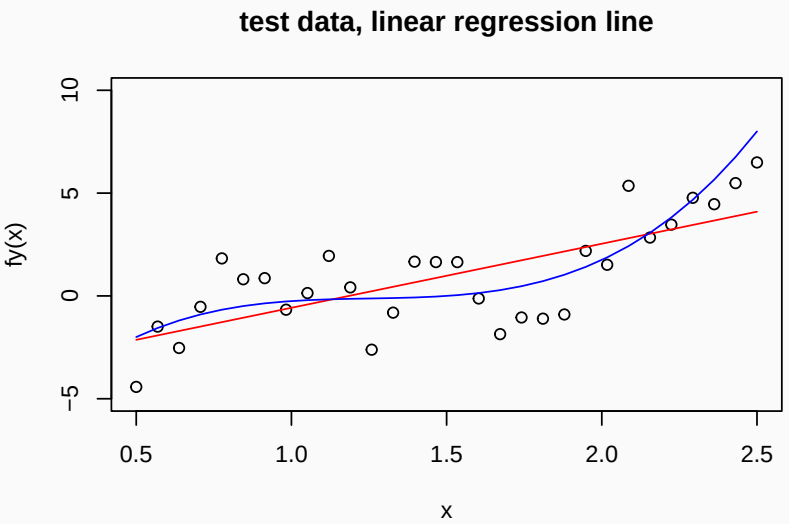


Underfitting

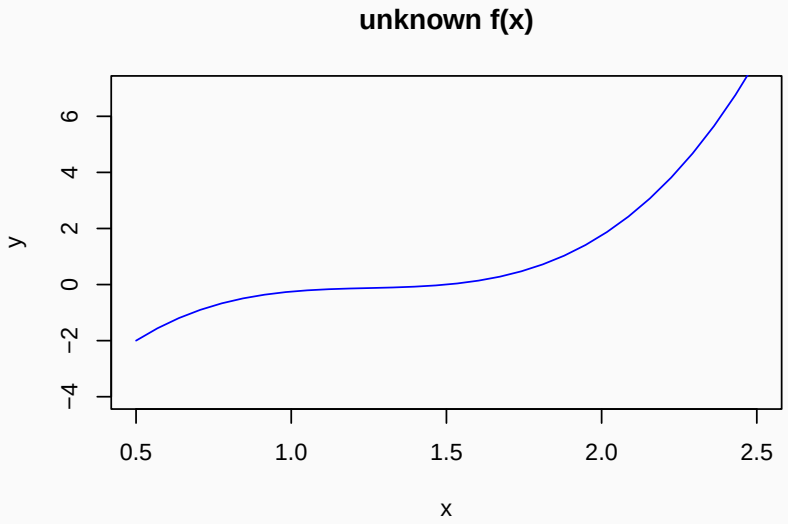


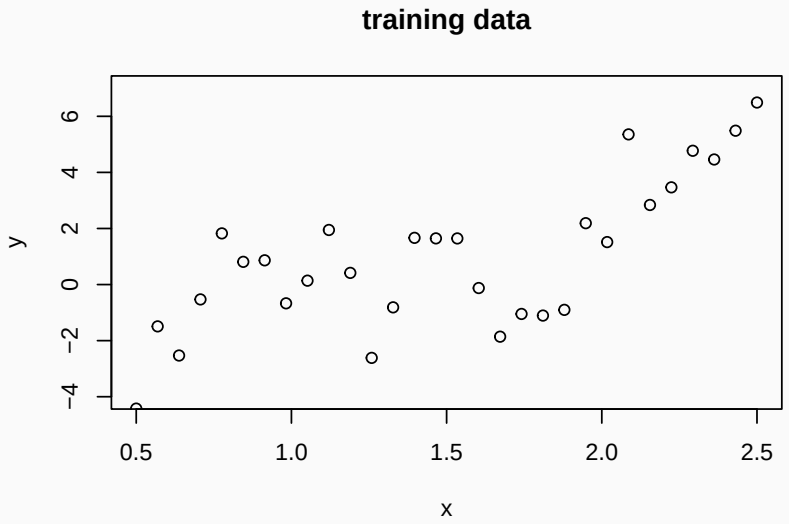




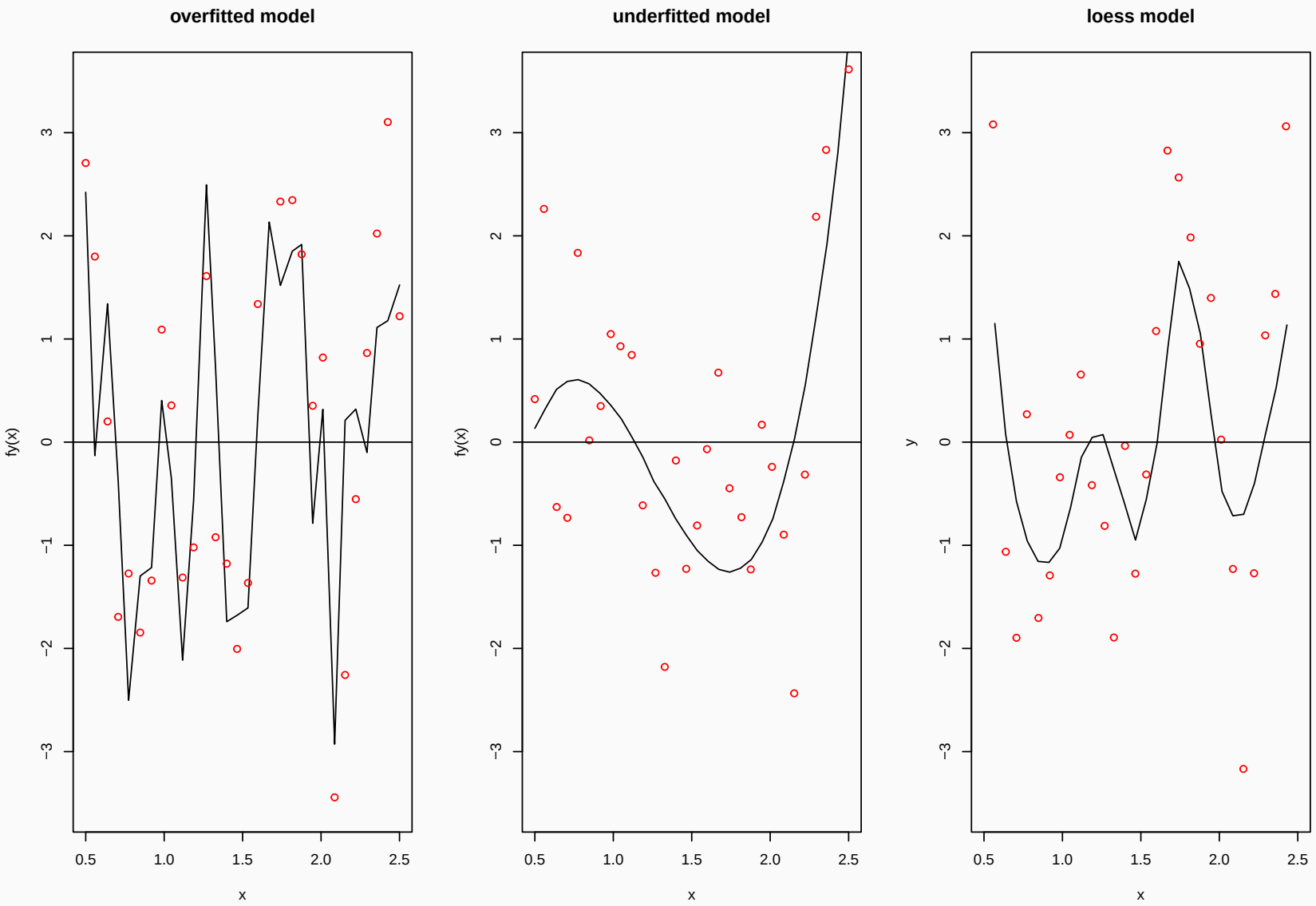


Balancing BIAS and VARIANCE: LOESS Model





Errors on Test Data



Overfitting / Underfitting

- **Overfitting**: a too complex model (large number of parameters) to capture random fluctuations in the training data $\Rightarrow$ poor performance on unseen data
- **Underfitting**: a too simple model to capture the underlying patterns in the data $\Rightarrow$ poor performance both on the training and unseen data
- The more complex a model the lower the **bias** (better fit to the training data) but higher the **variance** (i.e., sensitivity to variations in the training data)
- A simpler model has a higher bias but lower variance

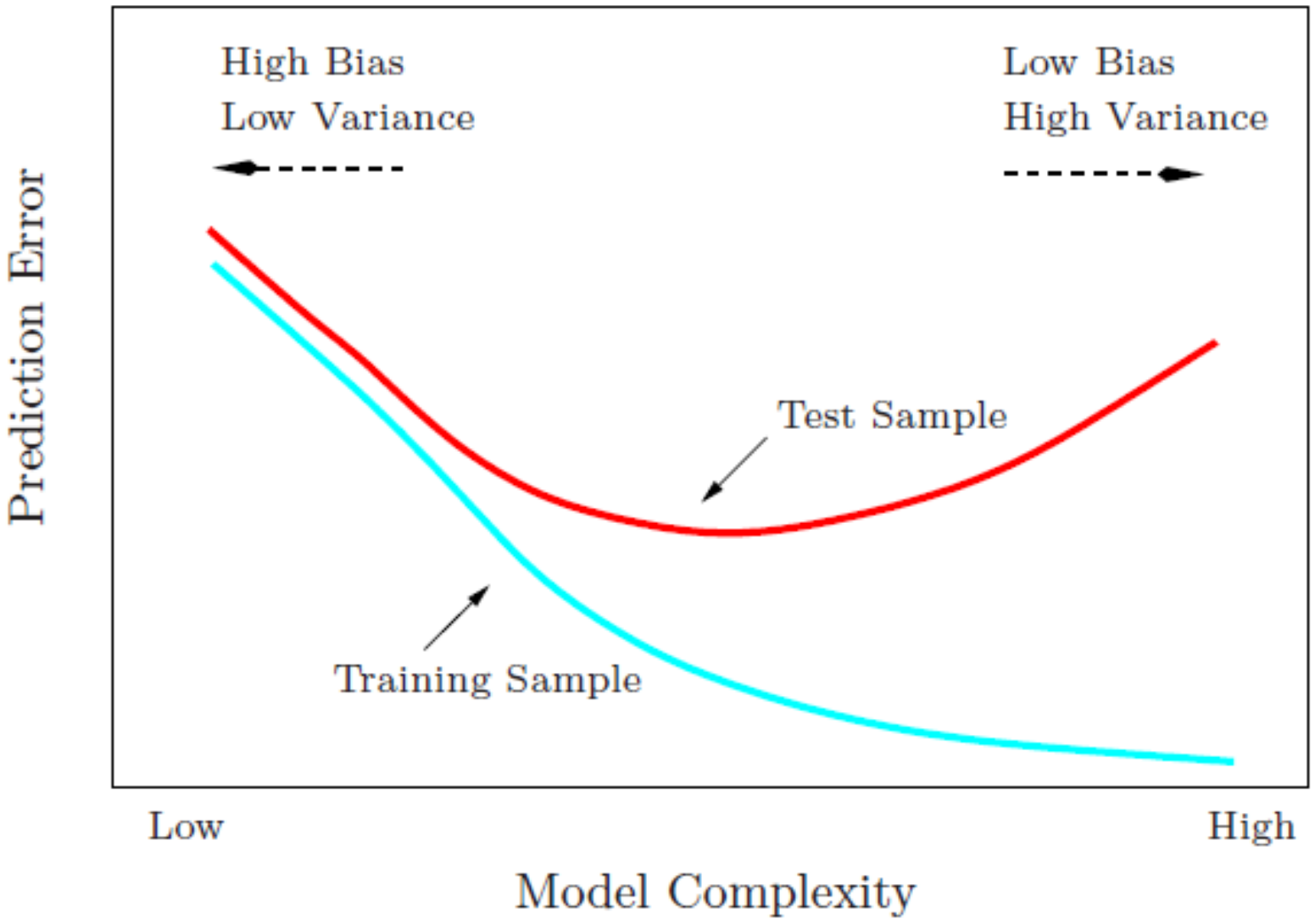
$\Rightarrow$ Balancing **Bias** and **Variance**

Bias-Variance Trade-off: Expected test prediction error

$$E[y_0 - \hat{f}(x_0)]^2 \quad = \quad Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

- $Var(\epsilon)$ irreducible error (= lower bound on test error)$\approx$ what we don't model, so can only be changed changing the information (the data)
- **Variance** $Var(\hat{f}(x_0))$: amount by which $\hat{f}$ would change with different training set
- **Bias**: Error in estimating f with approximating function
  - e.g., f is highly non linear and approximation function is linear
- Mean squared prediction error (= Bias + Variance) $\Rightarrow$ depends on how we model

Bias-Variance Trade-off



The more parameters a model has : the more complex a model is

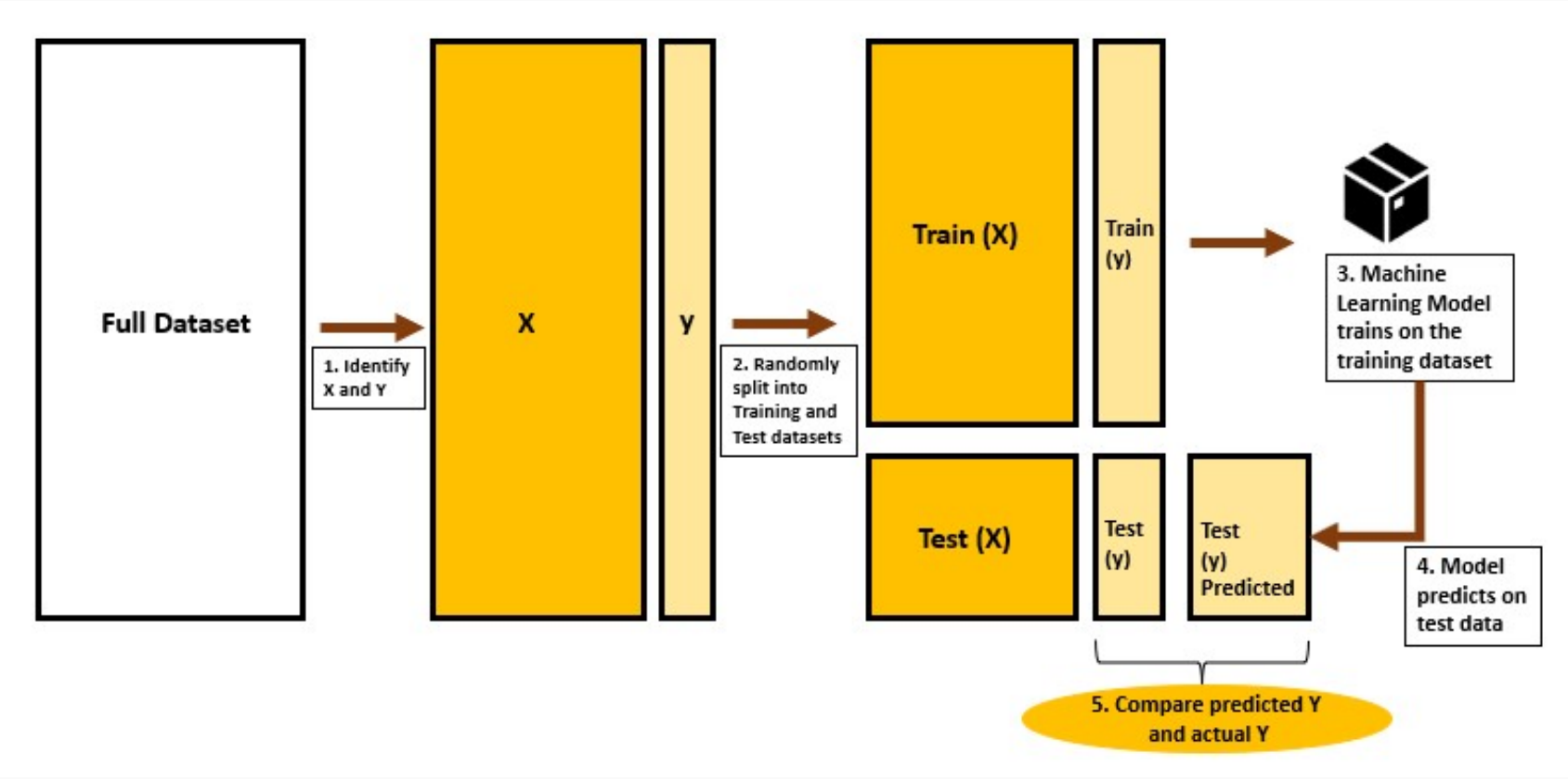
Different Ways of Model Validation

Important: Model **evaluation** / **performance** by **validation** on unseen **test** data to avoid overfitting

- external validation: new data
- internal validation: part of the data
  - **train/test split**
  - **cross-validation**
  - **bootstrap**
- **Performance measure**: test error, measure of quality of your model calculated on **test** data

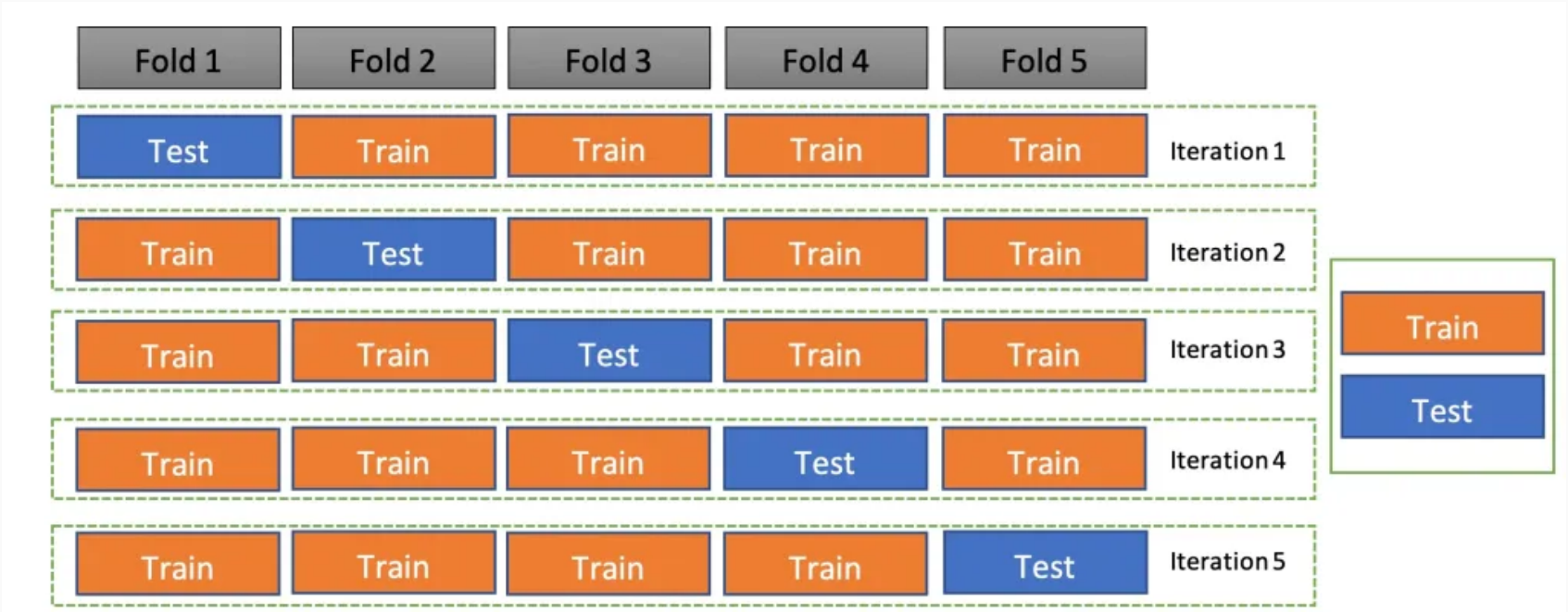
Train/Test Split

- Indication of performance on new data, future data
- Divide the data into training and test parts (e.g., 70%/30%)
- Train the model on training data and assess the performance of the model on test data
- Depends strongly on train/test split

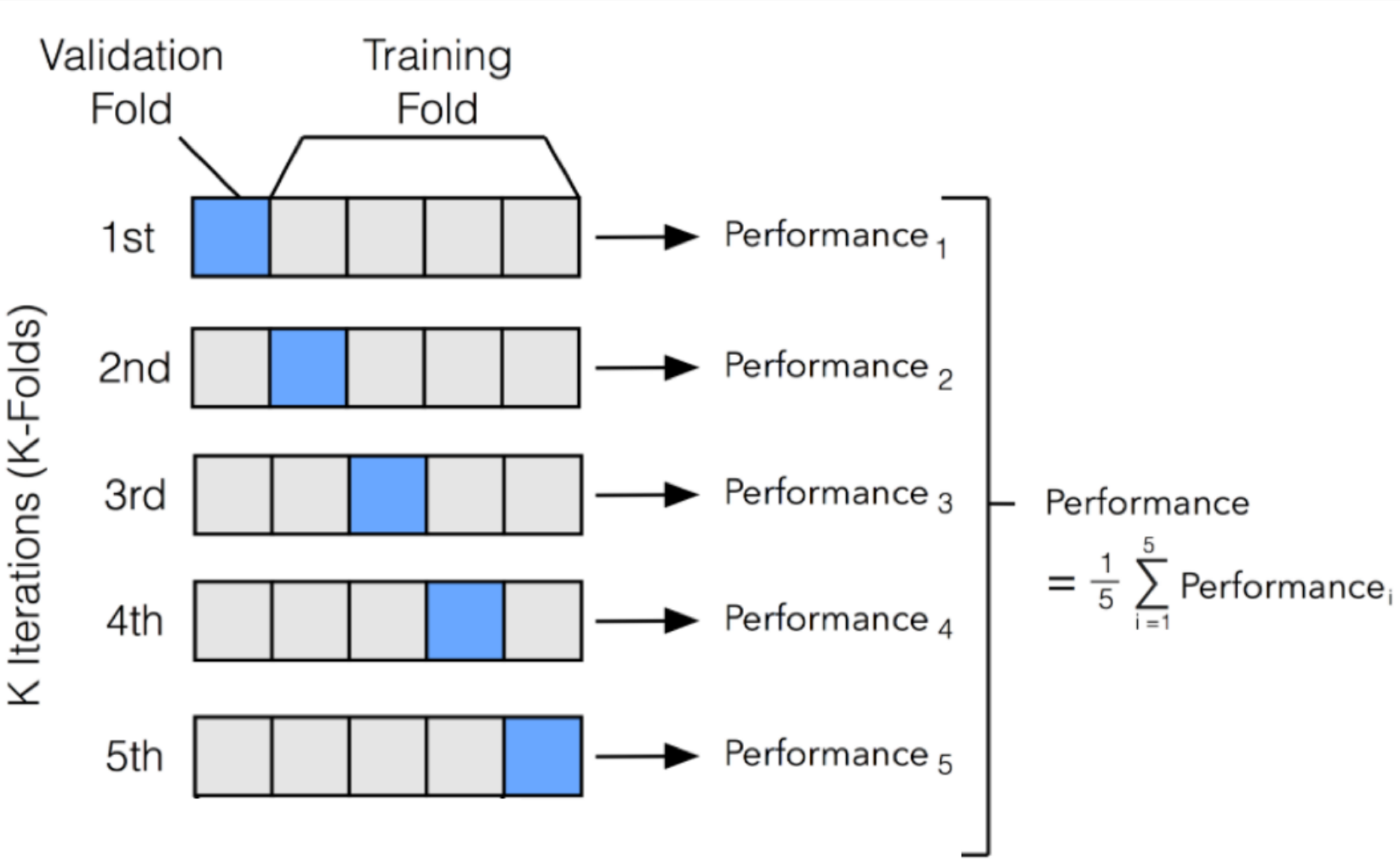


K-fold Cross-validation

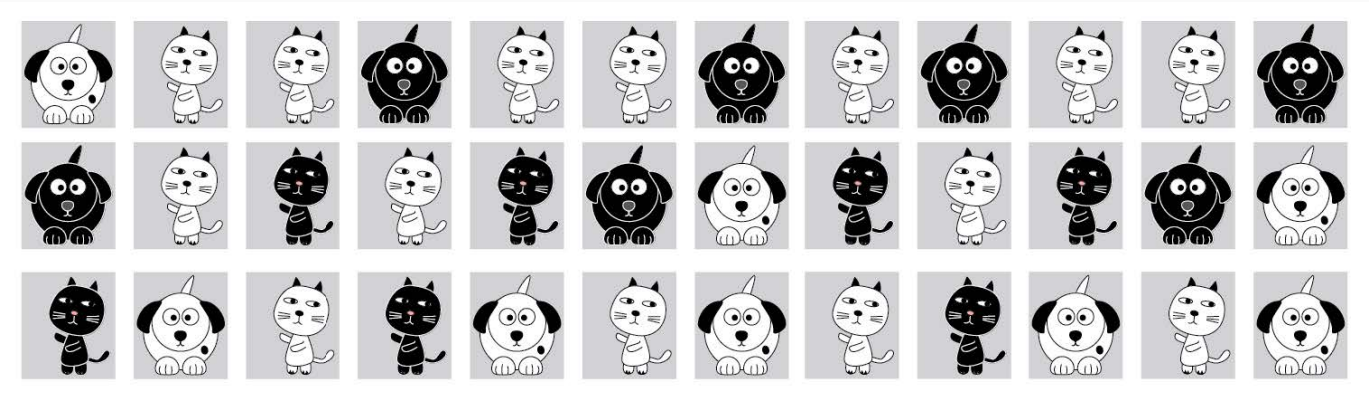
- Split available data in K equal pieces (e.g., $K = 5$ or $K = 10$)
- Train model on $K - 1$ piece of the data and test on 1 piece
- Repeat K times picking each time a different test piece
- Each data point is used $K - 1$ times for training and 1 time for testing.
- Calculate for each fold the performance measure(s)
- Average the K performances to obtain the final estimate

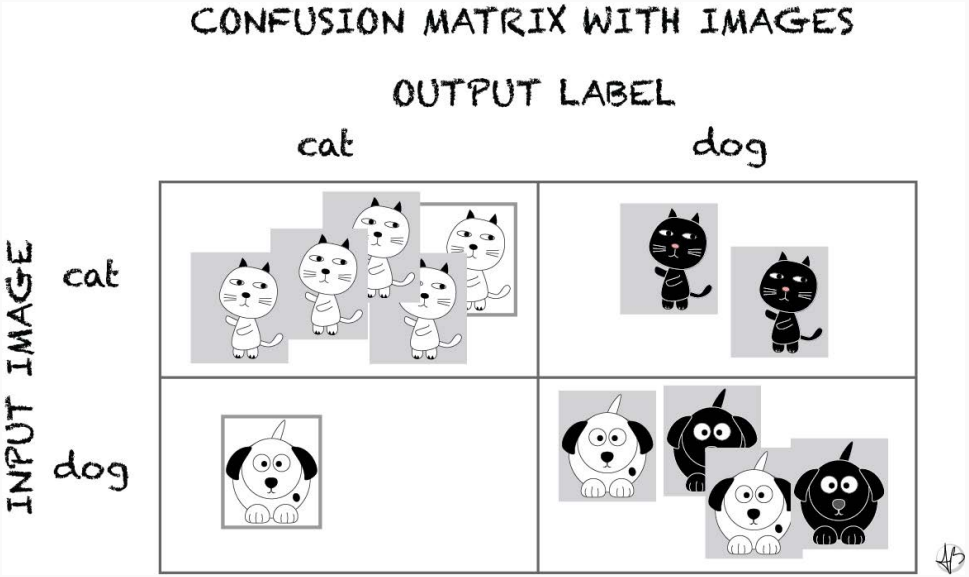


Cross-validation Performance



Performance Measure Classification: Confusion Matrix





Confusion Matrix

False Positives?

False Positive

False Negative

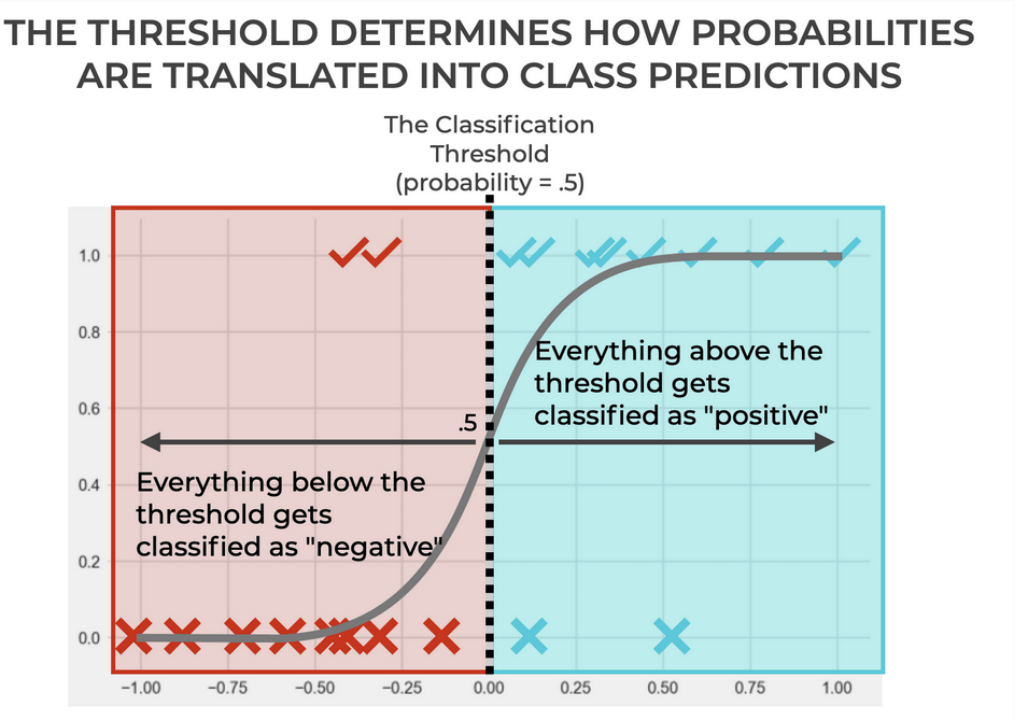
Performance Measures from Confusion Matrix

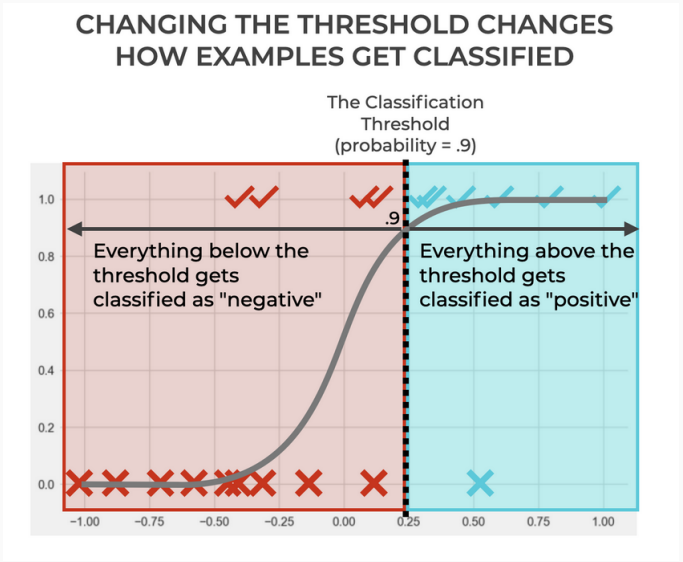
		Assigned class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

- Accuracy (percentage correctly classified) $= \frac{TP+TN}{TP+TN+FP+FN}$
- Sensitivity = True positive rate = Recall $= \frac{TP}{TP+FN}$
- Precision = Positive predictive value $= \frac{TP}{TP+FP}$
- Specificity = True negative rate $= \frac{TN}{TN+FP}$
- Combination of these: F_1-score $= 2\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$

All performances depend on the threshold (default value = 0.5).

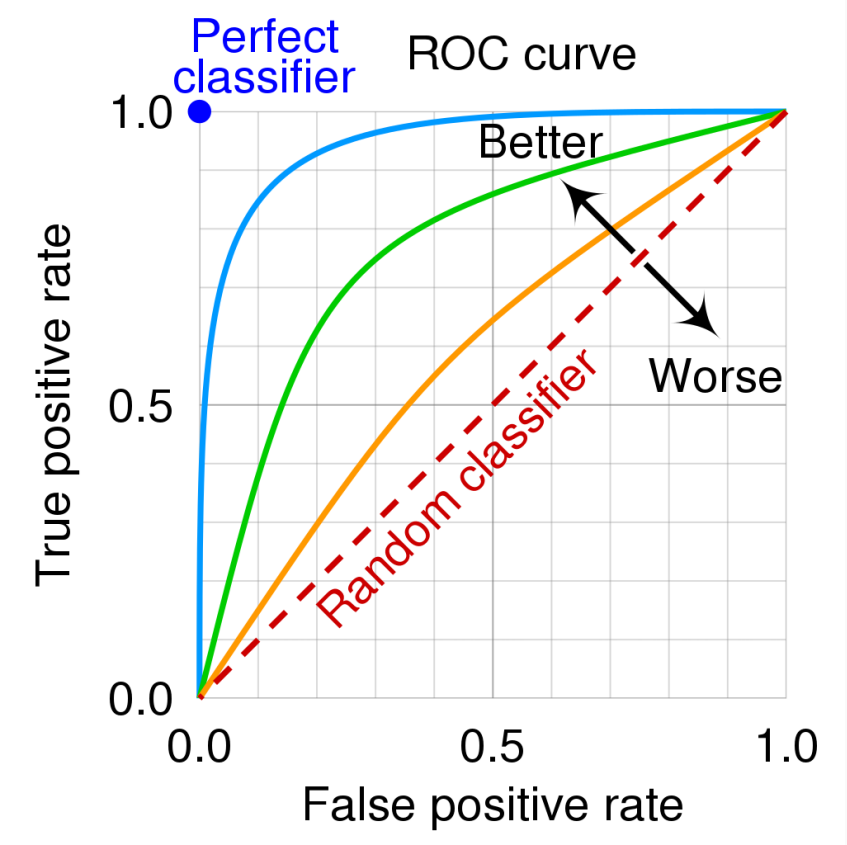
Changing the Threshold





ROC Curve

- Receiver Operating Characteristic (ROC) curve: plotting TPR against FPR at various threshold settings.
- Area under the curve (AUC): **independent** of the threshold



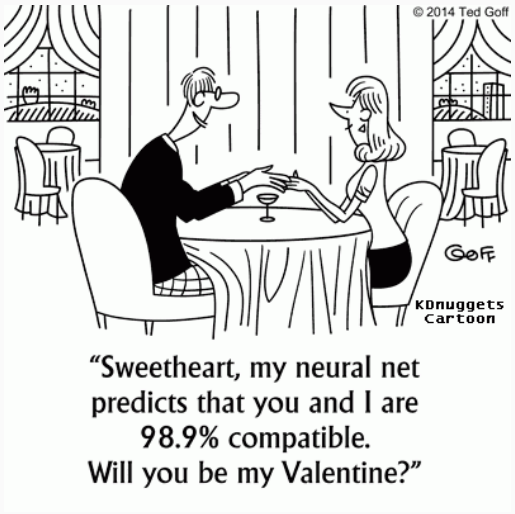
Performance Measures: Regression

- RMSE $= \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$ (Root Mean Square Error)
- MAE $= \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n}$ (Mean Absolute Error)
- $R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$ (Rsquared, Percentage variance explained)

Types of DM Methods

Different DM Methods

- Classification (binary / multi class):
  - Logistic regression
  - **K-Nearest Neighbor**
  - **Decision tree (CART)**
  - Random Forest
  - **Naïve Bayes method**
  - Support vector machine
  - Neural networks / Deep learning
  - ...
- Regression (outcome is continuous)
  - Linear regression
  - **K-Nearest Neighbor**
  - Random Forest
  - Regression trees
  - Neural networks / Deep learning
  - ...

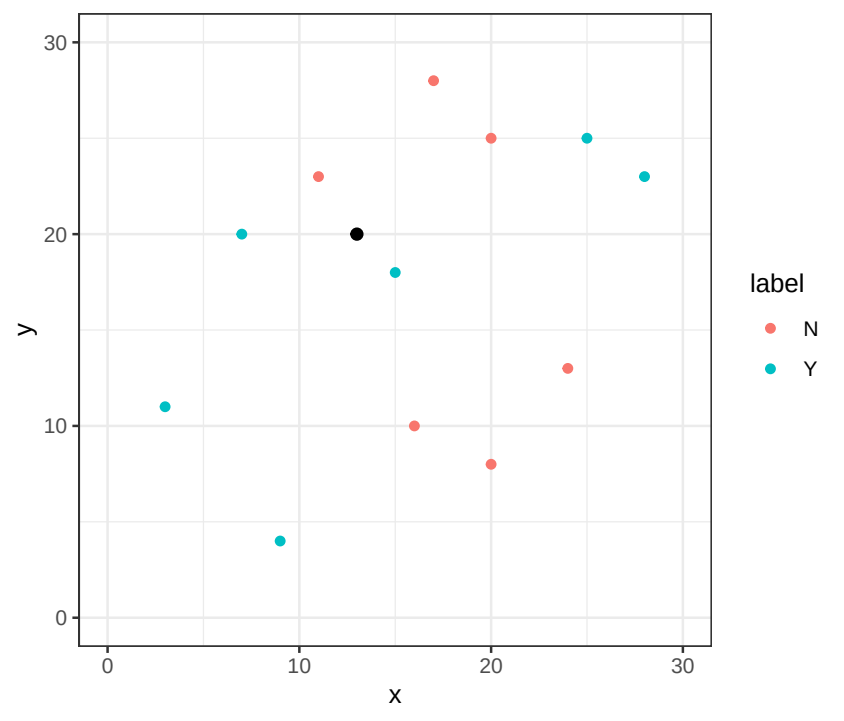


Unsupervised Learning Methods

- Principal Component Analysis
- Association rules
- Cluster methods
  - K-means clustering
  - Hierarchical clustering

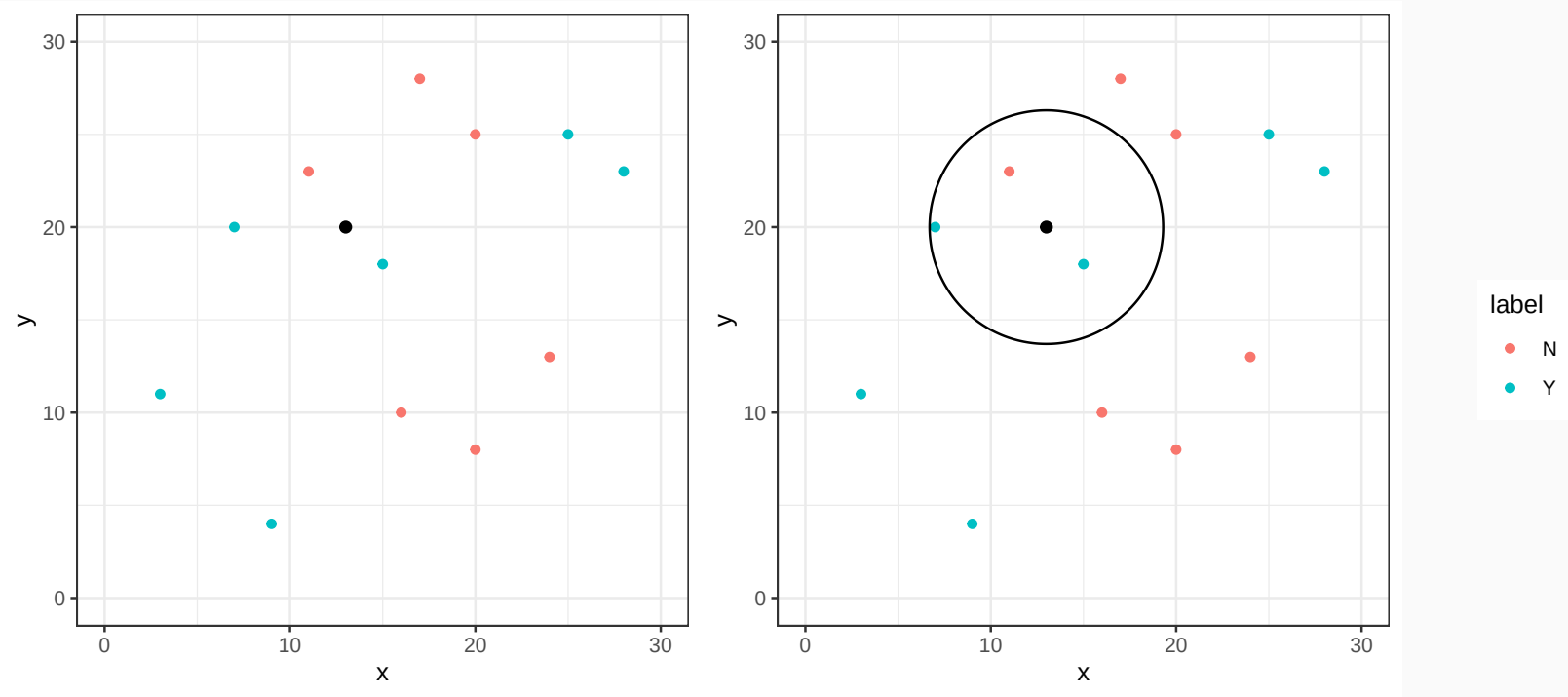
K-Nearest Neighbors (KNN)

Example K-Nearest Neighbors (KNN)

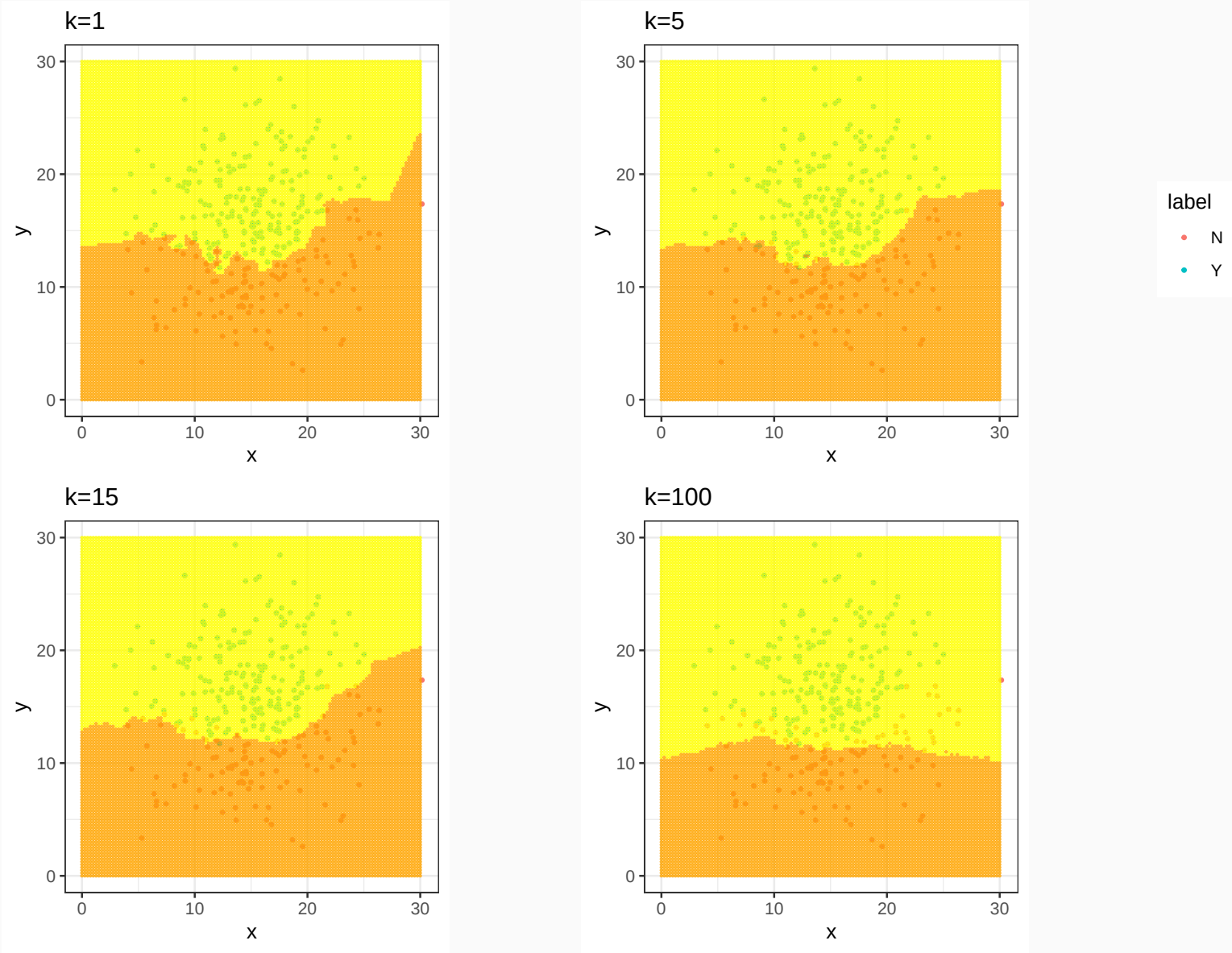


Principle KNN

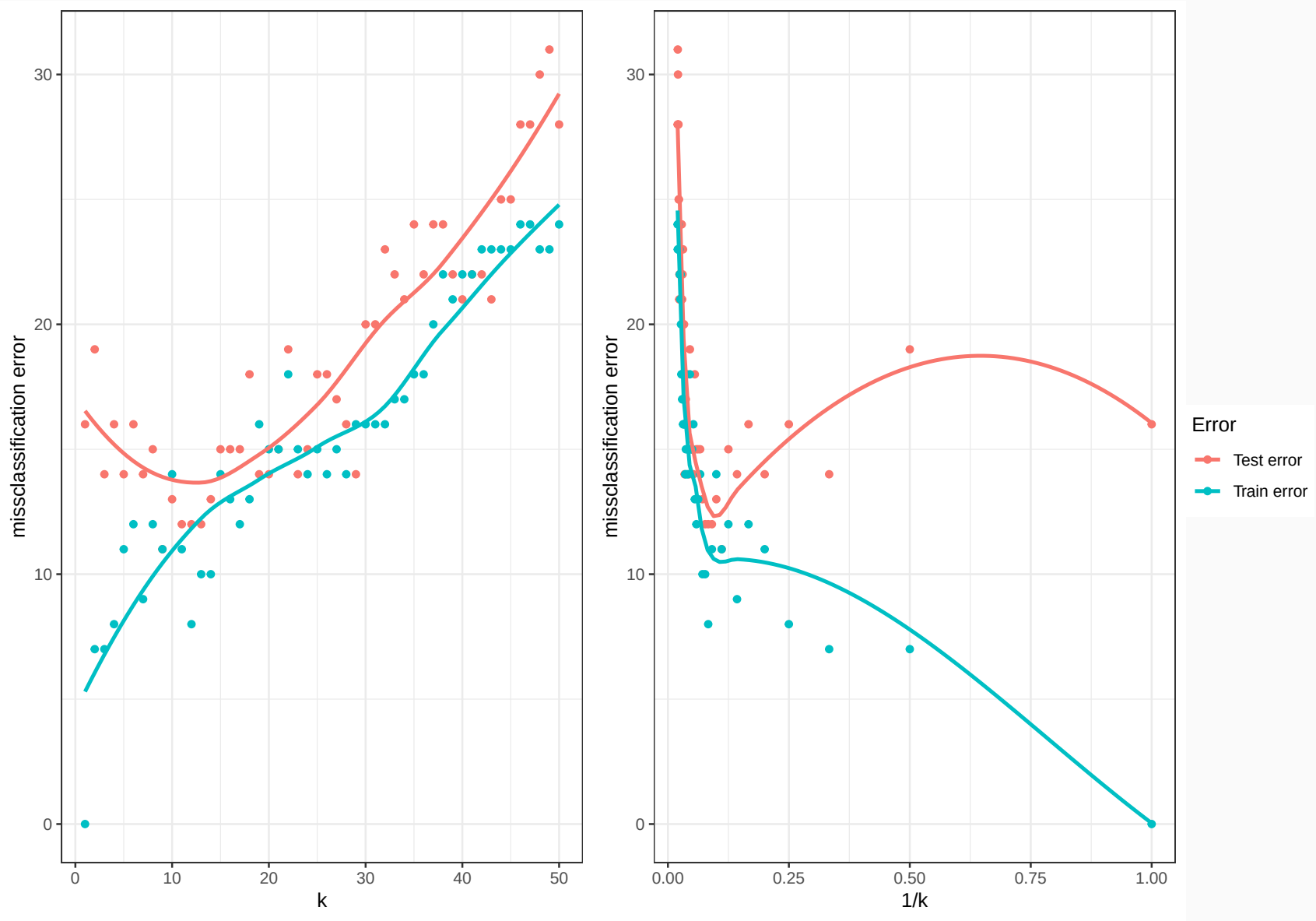
- Euclidean distance d between points (x_1, y_1) and (x_2, y_2): $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.
- Conditional Probability for class j: $P(Y = j|X = x_0) = \frac{1}{k} \sum_{i \in N_0} I(y_i = j)$



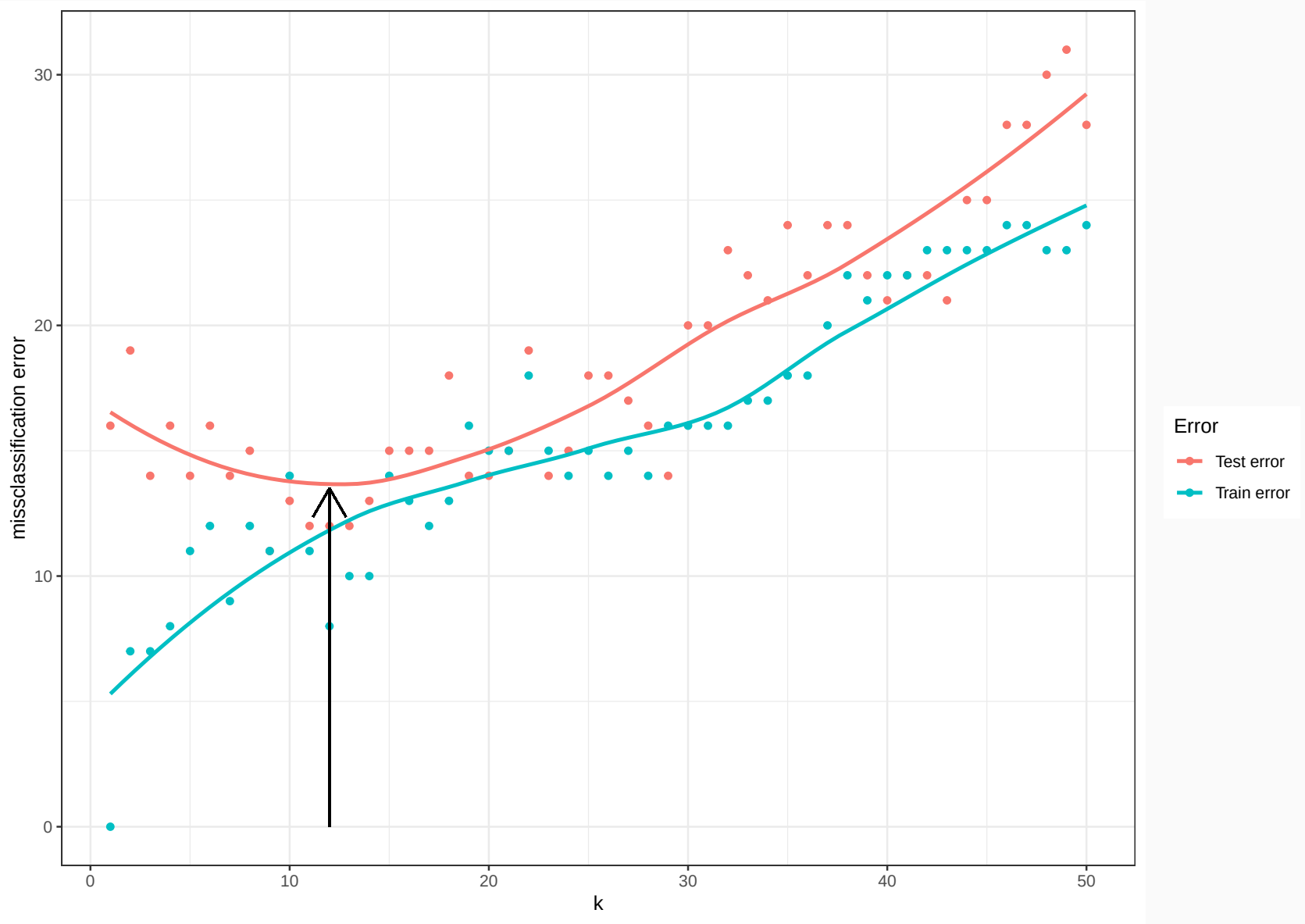
KNN: Tuning Parameter k



KNN: Choose Tuning Parameter

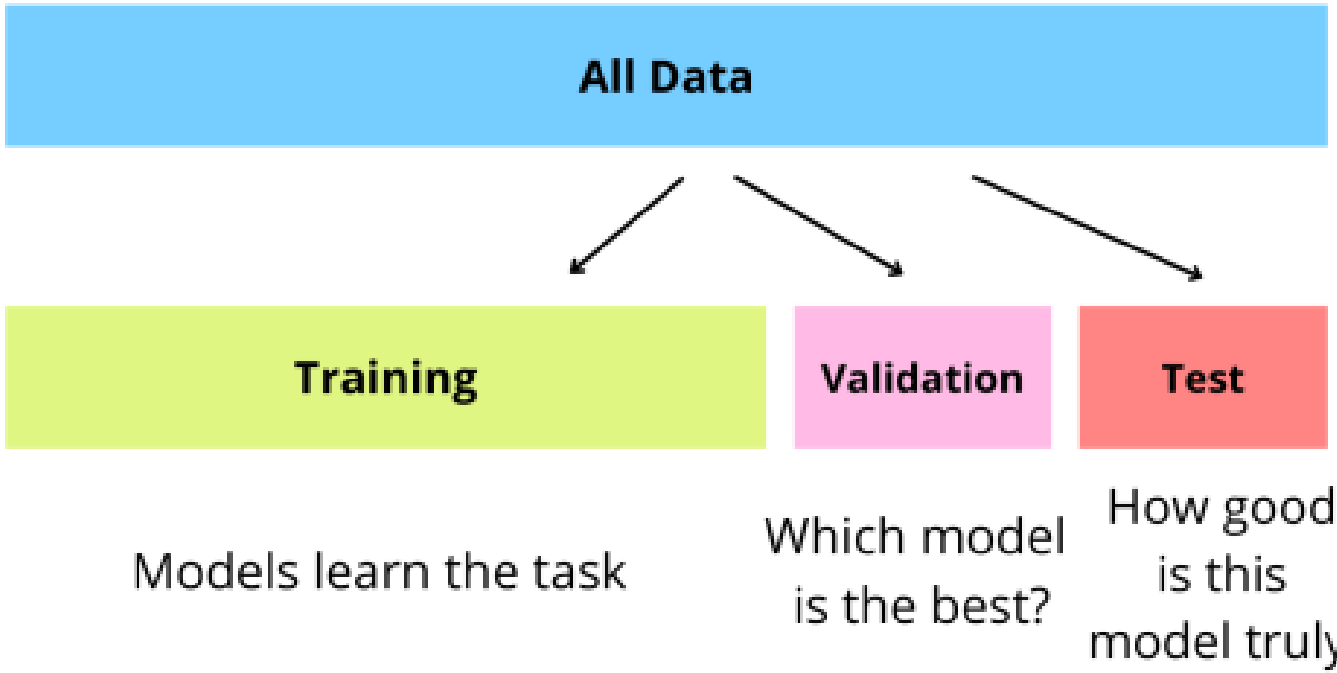


Optimal value of k = tuning the k parameter



Choice optimal k should not be based on test data set!

Tuning and Validation



KNN: Tuning and Validation Recipe

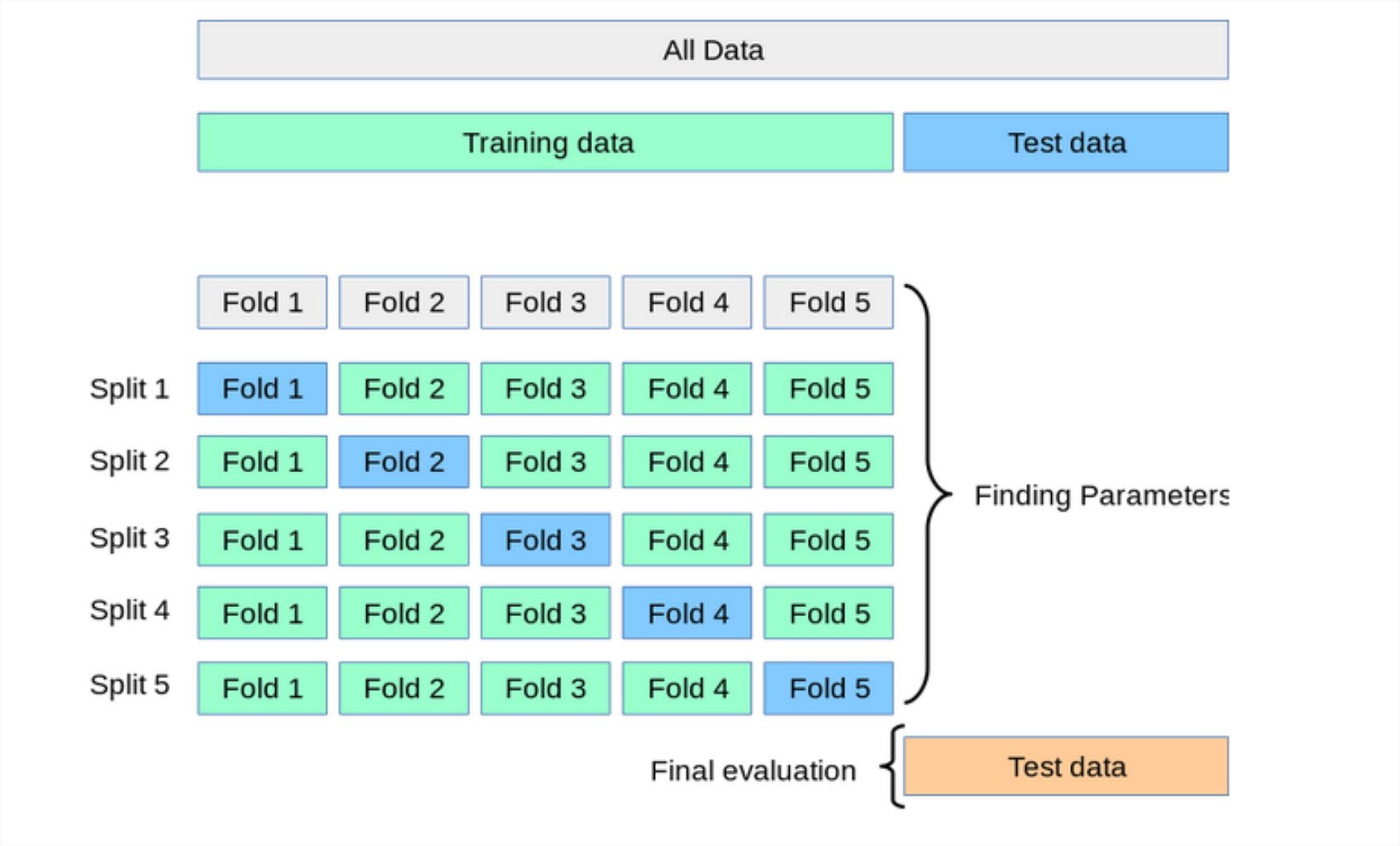
Recipe:

1. Define range for k (e.g. $k = 1, 2, \cdots, 50$)
2. Divide data into 3 sets (training, validation, test)
3. Train model for all k with training data
4. Calculate error with validation data
5. Find k_{opt} with smallest error
6. Re-fit model with k_{opt} on training and validation data

Then:

1. Estimate **test error** with test data
2. **FINAL model**: repeat **recipe** on all data

Tuning and Validation with CV



https://scikit-learn.org/stable/modules/cross_validation.html

Naive Bayes Method

Example: Naive Bayes Method

Age	Income	Student	Credit rating	Buys Computer
≤ 30	high	no	fair	no
≤ 30	high	no	excellent	no
31-40	high	no	fair	yes
> 40	medium	no	fair	yes
> 40	low	yes	fair	yes
> 40	low	yes	excellent	no
31-40	low	yes	excellent	yes
≤ 30	medium	no	fair	no
≤ 30	low	yes	fair	yes
> 40	medium	yes	fair	yes
≤ 30	medium	yes	excellent	yes
31-40	medium	no	excellent	yes
31-40	high	yes	fair	yes
> 40	medium	no	excellent	no
≤ 30	medium	yes	fair	??

Summarising Information: Marginal Distributions

Age	Buys Computer		Total
	Yes	No	
≤ 30	2	3	5
$31 - 40$	4	0	4
> 40	3	2	5
Total	9	5	14

Income	Buys Computer		Total
	Yes	No	
high	2	2	4
medium	4	2	6
low	3	1	4
Total	9	5	14

Student	Buys Computer		Total
	Yes	No	
No	3	4	7
Yes	6	1	7
Total	9	5	14

Credit Rating	Buys Computer		Total
	Yes	No	
fair	6	2	8
excellent	3	3	6
Total	9	5	14

Naive Bayes Method

- Recall **Bayes' Rule**:

$$P(hypothesis|evidence) = \frac{P(hypothesis) \cdot P(evidence|hypothesis)}{P(evidence)}$$

- $P(hypothesis|evidence)$: Posterior probability
- $P(hypothesis)$: prior
- $P(evidence|hypothesis)$: Likelihood
- $P(evidence)$: Marginal likelihood

and in terms of random variables:

$$P(Y = y|X = x) = \frac{P(Y=y) \cdot P(X=x|Y=y)}{P(X=x)}$$

Naive Bayes Method

$$P(+| \leq 30, \text{med}, \text{stu}, \text{fair}) = \frac{P(+) \cdot P(\leq 30, \text{med}, \text{stu}, \text{fair}|+)}{P(\leq 30, \text{med}, \text{stu}, \text{fair})}$$

- Prior: $P(+) = \frac{9}{14}$
- Likelihood: $P(\leq 30, \text{med}, \text{stu}, \text{fair}|+)$??

NAIVE Bayes: attributes are **conditionally independent**:

$$P(\leq 30, \text{med}, \text{stu}, \text{fair}|+) = P(\leq 30|+)P(\text{med}|+)P(\text{stu}|+)P(\text{fair}|+)$$

So: $P(\leq 30, \text{med}, \text{stu}, \text{fair}|+) = \frac{2}{9} \cdot \frac{4}{9} \cdot \frac{6}{9} \cdot \frac{6}{9} = 0.0439$

Naive Bayes Method

Remember: $P(B) = P(A)P(B|A) + P(notA)P(B|notA))$

Denominator $P(\leq 30, \text{med}, \text{stu}, \text{fair})$ equals:

$$P(+)P(\leq 30, \text{med}, \text{stu}, \text{fair}|+) + P(\text{-})P(\leq 30, \text{med}, \text{stu}, \text{fair}|\text{-}) =$$

$$\frac{9}{14} \cdot \frac{2}{9} \cdot \frac{4}{9} \cdot \frac{6}{9} \cdot \frac{6}{9} + \frac{5}{14} \cdot \frac{3}{5} \cdot \frac{2}{5} \cdot \frac{1}{5} \cdot \frac{2}{5} = 0.0351$$

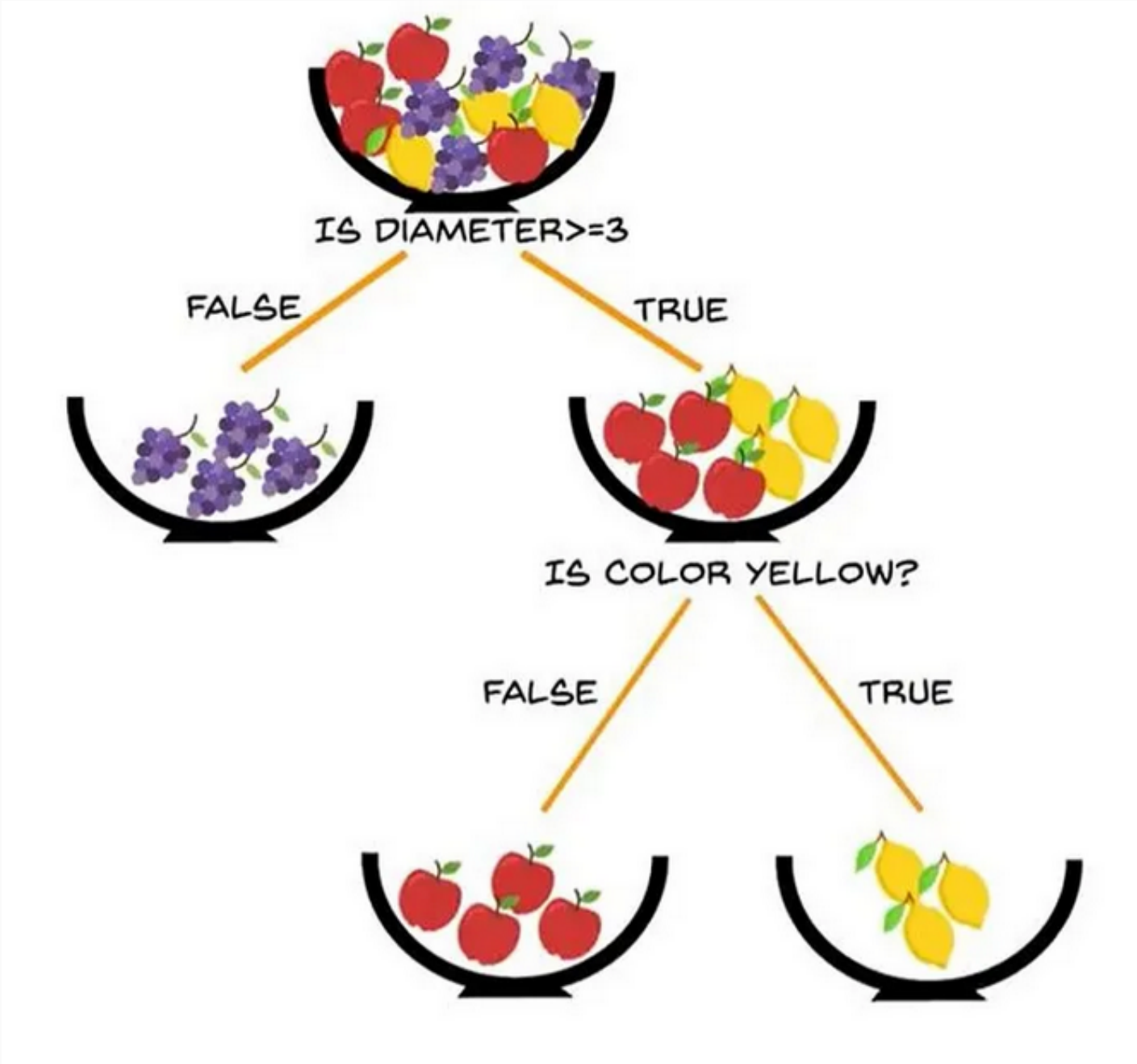
So

$$P(+| \leq 30, \text{med}, \text{stu}, \text{fair}) = \frac{\frac{9}{14} \cdot 0.0439}{0.0351} = 0.805 > 0.5$$

$\Rightarrow$ Prediction: **Buys Computer = Yes**

Decision Trees (CART)

Decision Trees



Decision Tree: What and Why?

Problem: given set of training cases / objects and their attribute values, try to determine label for new examples

- Classification
- Prediction

Why decision tree?

- Decision trees are powerful and popular tools for classification and prediction
- Decision trees represent rules, which can be understood by humans and used in knowledge systems like a database

Classification and Regression tree

- Split domain feature variables in rectangles
- Predictions in a certain rectangle get the same value

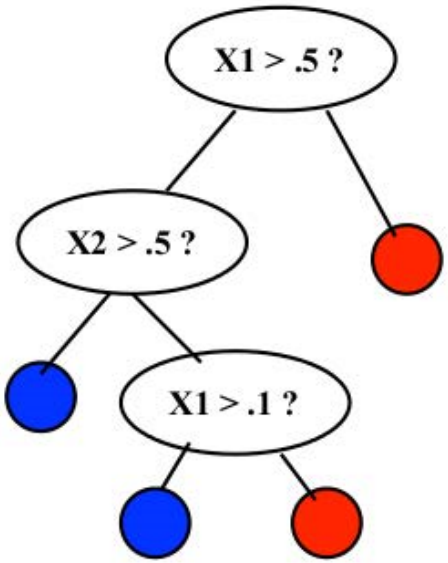
Regression tree

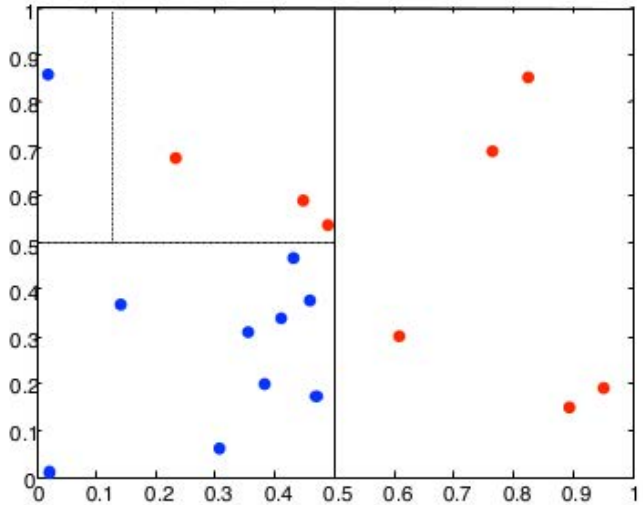
- Dependent variable is a continuous variable
- Prediction is the mean value
- Splits are obtained by minimizing the RSS (Residual Sum of Squares)

Classification tree

- Dependent variable is a **binary** variable
- Prediction is based on majority vote
- Splits are obtained by minimizing Gini index, entropy, (mis)classification error or other measures based on probabilities of the two classes

Decision Tree: continuous attributes





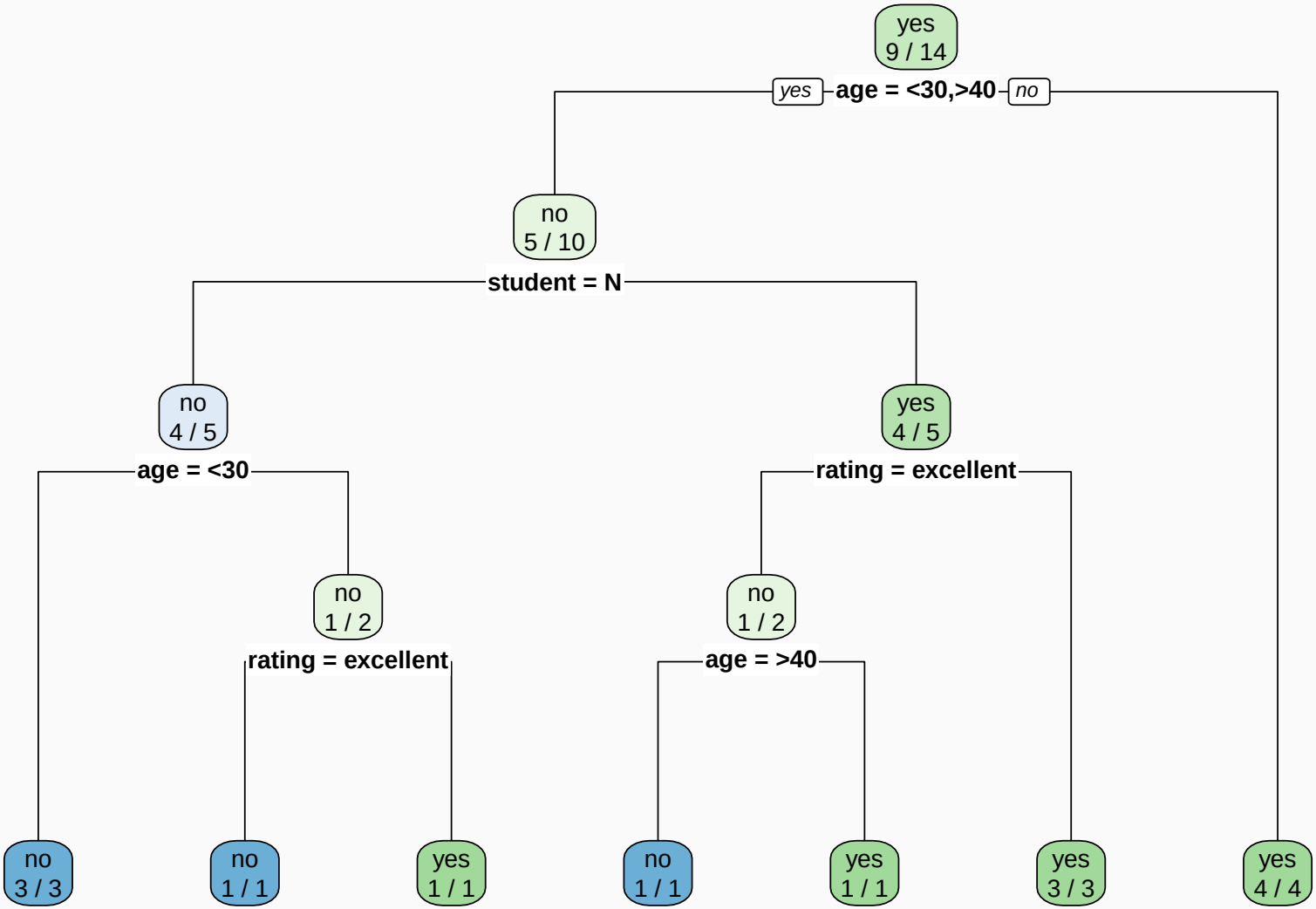
CART

- Many algorithms: ID3, C4.5, C5.0, CART, ...
- CART (Classification and Regression Trees) is introduced by Leo Breiman to refer to Decision Tree algorithms that can be used for classification or pregression predictive modeling problems
- CART algorithm is foundation for algorithms like bagged decision trees, random forest and boosted decision trees

CART Model Representation

- CART model is a binary tree
- Each root node presents a single input variable (x) and a split point on that variable (if x is numeric)
- The leaf nodes of the tree contain an output variable (y) which is used to make a prediction
- CART does not require any special data preparation
- The tree can be stored to a file as a graph or a set of rules

Decision tree example



CART for Classification

For classification the Gini Index is used which provides an indication of how 'pure' the leaf nodes are (how mixed the training data assigned to each node is).

$$\text{gini}(t) = 1 - \sum_j [P(j|t)]^2$$

where $P(j|t)$ is the relative frequency of class j at node t.

Example calculation GINI impurity

Age	Buys computer		Total
	Yes	No	
≤ 30	2	3	5
$31 - 40$	4	0	4
> 40	3	2	5
Total	9	5	14

$$\text{gini}_{\text{root}} = 1 - (\tfrac{9}{14})^2 - (\tfrac{5}{14})^2 = 0.46$$

$$\text{gini}_{\leq 30} = 1 - (\tfrac{2}{5})^2 - (\tfrac{3}{5})^2 = 0.48$$

$$\text{gini}_{31-40} = 1 - (\tfrac{4}{4})^2 - (\tfrac{0}{4})^2 = 0$$

$$\text{gini}_{>40} = 1 - (\tfrac{3}{5})^2 - (\tfrac{2}{5})^2 = 0.48$$

$$\text{gini}_{\text{age}} = \tfrac{5}{14} \cdot 0.48 + \tfrac{4}{14} \cdot 0 + \tfrac{5}{14} \cdot 0.48 = 0.343$$

$$\text{gini}_{\text{student}} = 0.367$$

$$\text{gini}_{\text{income}} = 0.440$$

$$\text{gini}_{\text{rating}} = 0.429$$

age has the highest information gain: $0.46 - 0.343 = 0.116$

Top node split with (mis)classification error

Age	Buys computer		Error
	Yes	No	
≤ 30	2	3	2
$31 - 40$	4	0	0
> 40	3	2	2
Total	9	5	4

Income	Buys computer		Error
	Yes	No	
high	2	2	2
medium	4	2	2
low	3	1	1
Total	9	5	5

Student	Buys computer		Error
	Yes	No	
No	3	4	3
Yes	6	1	1
Total	9	5	4

Credit Rating	Buys computer		Error
	Yes	No	
fair	6	2	2
excellent	3	3	3
Total	9	5	5

Top root split with attribute **Age** or **Student**

Next node

Age	Income	Student	Credit rating	Buys computer
≤ 30	high	no	fair	no
≤ 30	high	no	excellent	no
≤ 30	medium	no	fair	no
≤ 30	low	yes	fair	yes
≤ 30	medium	yes	excellent	yes
31-40	high	no	fair	yes
31-40	low	yes	excellent	yes
31-40	medium	no	excellent	yes
31-40	high	yes	fair	yes
> 40	medium	no	fair	yes
> 40	low	yes	fair	yes
> 40	low	yes	excellent	no
> 40	medium	yes	fair	yes
> 40	medium	no	excellent	no

Next node: age ≤ 30

Income	Buys computer: Yes	No	Total
low	1	0	1
medium	1	1	2
high	0	2	2
Total	2	3	5

$$\text{gini}_{\text{low}} = 1 - (\tfrac{1}{1})^2 - (\tfrac{0}{1})^2 = 0$$

$$\text{gini}_{\text{medium}} = 1 - (\tfrac{1}{2})^2 - (\tfrac{1}{2})^2 = \tfrac{1}{2}$$

$$\text{gini}_{\text{high}} = 1 - (\tfrac{0}{2})^2 - (\tfrac{2}{2})^2 = 0$$

$$\text{gini}_{\text{income}} = \tfrac{1}{5} \cdot 0 + \tfrac{2}{5} \cdot \tfrac{1}{2} + \tfrac{2}{5} \cdot 0 = 0.2$$

Rating	Buys computer: Yes	No	Total
fair	1	2	3
excellent	1	1	2
Total	2	3	5

$$\text{gini}_{\text{fair}} = 1 - (\tfrac{1}{3})^2 - (\tfrac{2}{3})^2 = \tfrac{4}{9}$$

$$\text{gini}_{\text{excellent}} = 1 - (\tfrac{1}{2})^2 - (\tfrac{1}{2})^2 = \tfrac{1}{2}$$

$$\text{gini}_{\text{rating}} = \tfrac{3}{5} \cdot \tfrac{4}{9} + \tfrac{2}{5} \cdot \tfrac{1}{2} = 0.467$$

Student	Buys computer: Yes	No	Total
No	0	3	3
Yes	2	0	2
Total	2	3	5

$$\text{gini}_{\text{student}} = 0$$

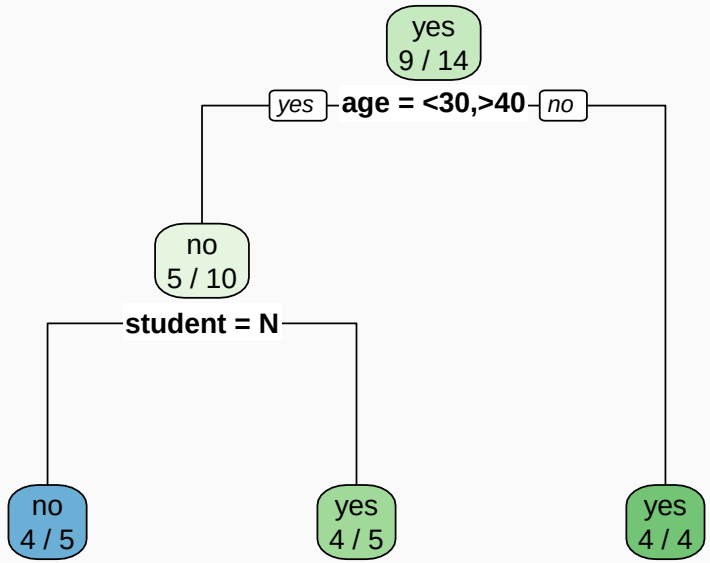
student has the lowest impurity: 0

Pruning Trees

- Reducing number of attributes in a tree $\rightarrow$ pruning
- Helps to avoid overfitting
- Pre-pruning:
  - Maximum number of leaf nodes
  - Maximum depth of the tree
  - Minimum number of training instances at a leaf node
- Post-pruning:
  - Prune full tree backward

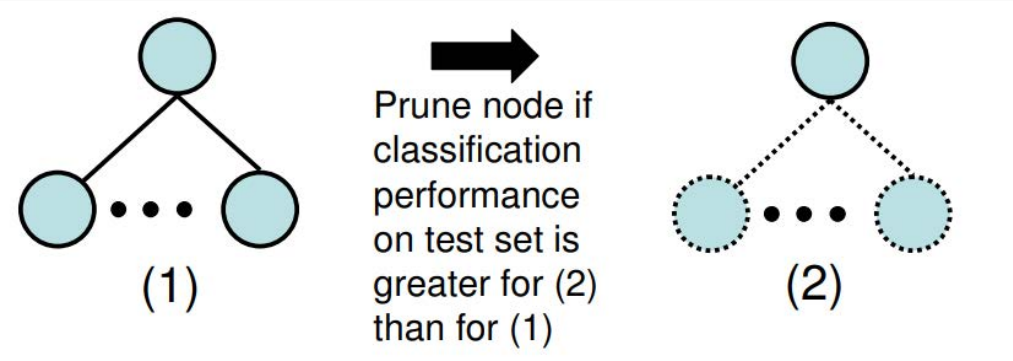
Example pre-pruning

```
model2 <- rpart(computer ~ age + income + student + rating, data = data,
                method = "class",
                control = rpart.control(minsplit = 3))
rpart.plot(model2, extra = 2)
```



Decision Tree Post Pruning

- Construct unpruned tree based on training data
- Start at the leaves, recursively eliminate splits
  1. Performance of the tree on test data (validation data)
  2. Prune the tree if the classification performance increases by removing the split

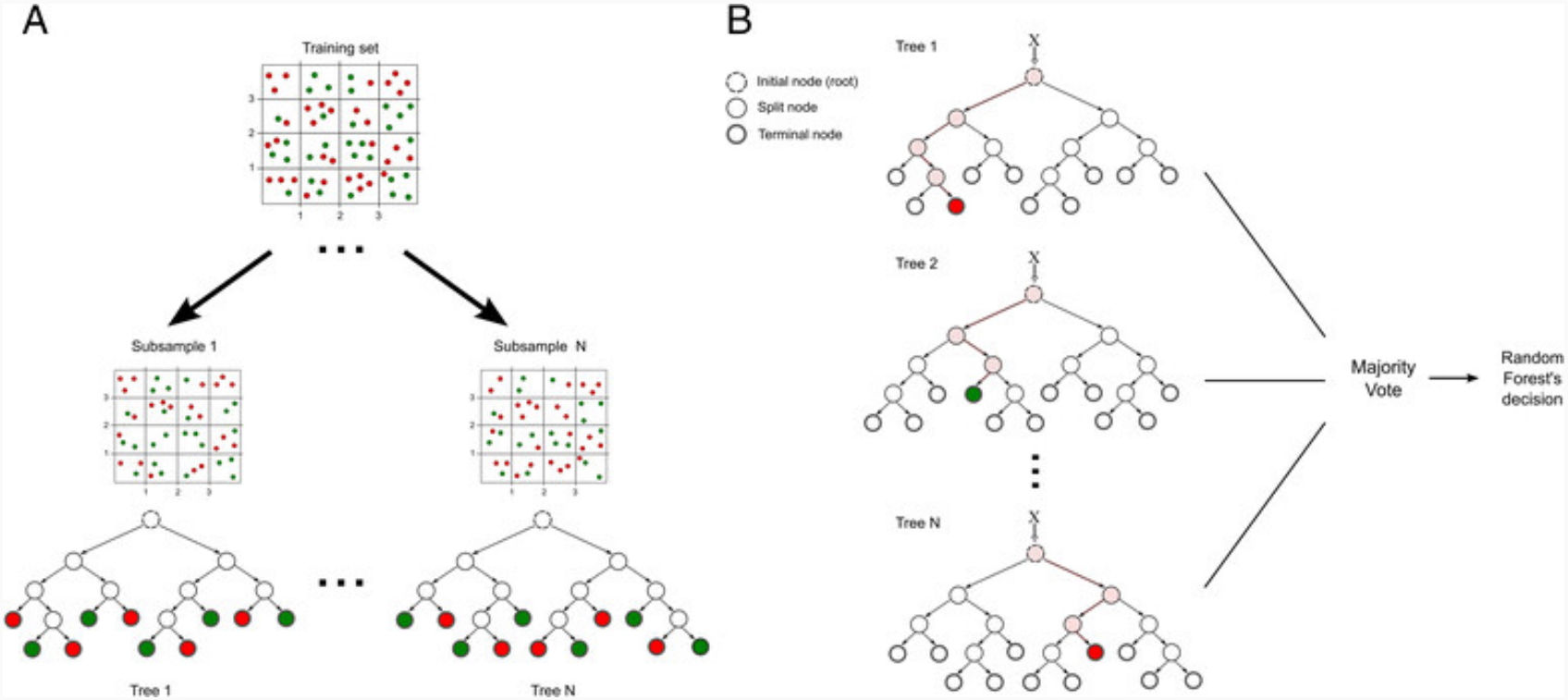


- Prune the tree by removing useless nodes based on:
  - Additional test data (not used for training)
  - Statistical significance tests (e.g., Chi-square criterion)

Summary: Decision Tree Learning

- One of the most widely used learning methods in practice
- Strengths:
  - Fast
  - Simple to implement
  - Easy interpretable rules
- Weaknesses:
  - Univariate splits at a time
  - Large trees are hard to understand
  - Performance not optimal

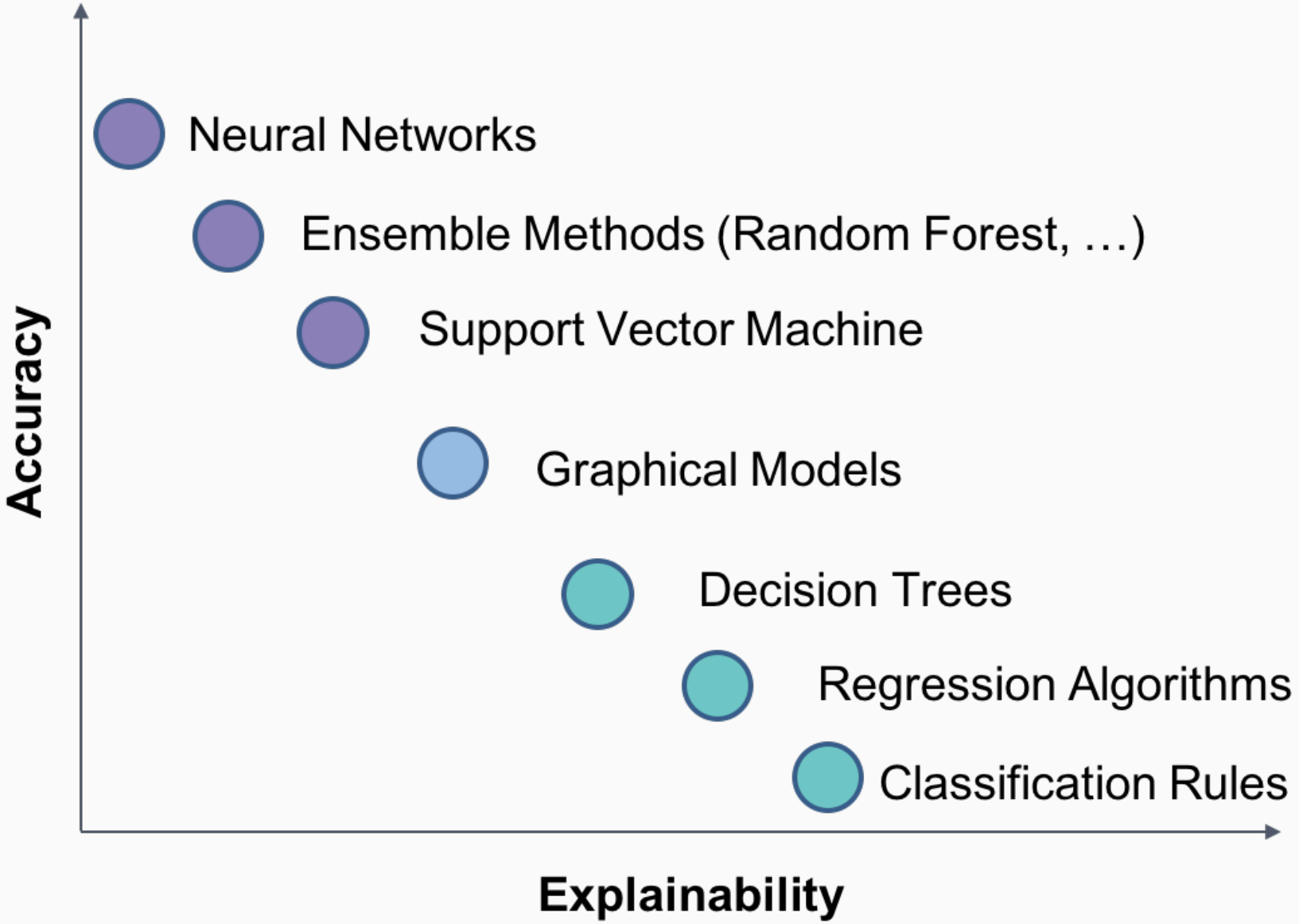
Random Forest (Ensemble Learning Method)



Hyperparameter Tuning for the Random Forest

- n_estimators = number of trees in the forest
- max_features = max number of features considered for splitting a node
- max_depth = max number of levels in each decision tree
- min_samples_split = min number of data points placed in a node before the node is split

Explainability of Models



Setting up a CRISP DM Analysis

CRISP-DM

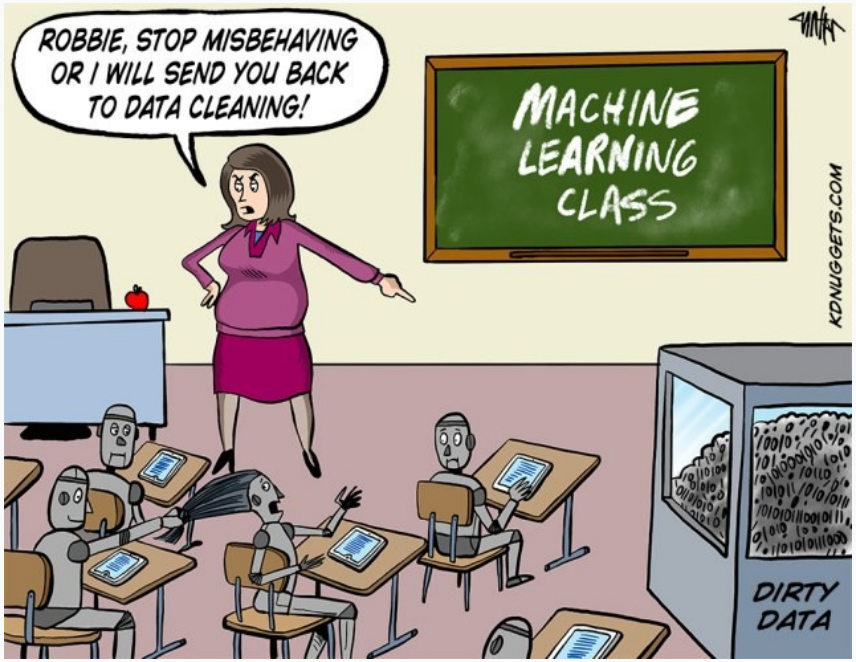
Cross-Industry Standard Process for Data Mining (CRISP-DM): it breaks the process of data mining into six major phases.

Data Understanding / Cleaning / Preparation

- Load data in a suitable Database (see DEP)
- Detect and remove outliers
- Deal with missing values (remove data, impute missing values: see, e.g., https://www.jstatsoft.org/article/view/v045i03)
- Enrich the data with data from external sources
- Transform feature values (make dummy variables from categorical variables; transform text to numerical)
- Deal with class imbalance (perform down- or up-sampling)

Data Cleaning / Preparation

Data cleaning / preparation takes most of the time



Modelling Steps

- **Model development**
  - Model specification
  - Feature selection
  - **Tuning** of parameters: cross-validation, grid search
- **Model evaluation**: validation $\Rightarrow$ Performance
  - external validation: new data
  - internal validation: part of the data
    - **train/test split**
    - **cross-validation**
    - **bootstrap**
- *All steps from model development should be included in model evaluation step*
- **Final model**: based on the whole data!

https://machinelearningmastery.com/train-final-machine-learning-model/

Feature selection

- Feasibility / Cost / Availability
  - Are features available in time when predictions are needed
- Domain knowledge
  - scientific literature
- Automatic feature selection with e.g. backward selection, forward selection

Summary

Summary

- Supervised versus unsupervised
- Important concepts: Bias-Variance trade-off, Overfitting, Validation
- Different methods: KNN, Naive Bayes, Decision trees

- What next...
  - Practical assignments
  - Projects

Project

- Identify what kind of DM problem it is
- Well-formulated research questions
- Design of a valid DM pipeline (including feature construction and selection)
- Comparison of (at least 3) different DM models
- Assessment of the performance of the constructed DM models in a sound way (with test/train set, Cross-validation)
- Critical reflection: strengths and weaknesses of the methodology, results: place the results in the context of the problem

End