FACULTY OF ENGINEERING
AND ARCHITECTURE

DEPARTMENT OF INFORMATION TECHNOLOGY
RESEARCH GROUP IDLAB

WORKSHOP DE CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

Daniel Peralta <dperalta@ugr.es>

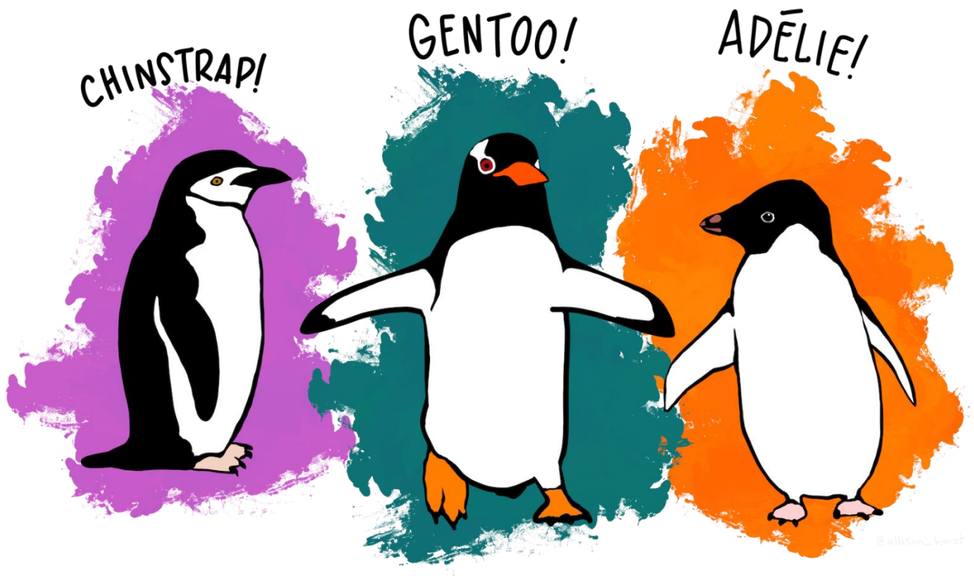
GHENT
UNIVERSITY

imec

PROBLEMA: ESPECIES DE PINGÜINOS

- Tenemos algunas medidas de 3 especies de pingüinos

- Queremos que la computadora clasifique a un nuevo pingüino en su especie correspondiente, dadas sus medidas



	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
	<fct>	<fct>	<dbl>	<dbl>	<int>	<int>	<fct>	<int>
1	Adelie	Torgersen	39.1	18.7	181	3750	male	2007
2	Adelie	Torgersen	39.5	17.4	186	3800	female	2007
3	Adelie	Torgersen	40.3	18	195	3250	female	2007
4	Adelie	Torgersen	NA	NA	NA	NA	NA	2007
5	Adelie	Torgersen	36.7	19.3	193	3450	female	2007
6	Adelie	Torgersen	39.3	20.6	190	3650	male	2007
7	Adelie	Torgersen	38.9	17.8	181	3625	female	2007
8	Adelie	Torgersen	39.2	19.6	195	4675	male	2007
9	Adelie	Torgersen	34.1	18.1	193	3475	NA	2007
10	Adelie	Torgersen	42	20.2	190	4250	NA	2007

GHENT UNIVERSITY

imec

PROBLEMA: ESPECIES DE PINGÜINOS

- Opción 1: diseñar manualmente un algoritmo de clasificación siguiendo reglas fijas

- Opción 2: diseñar un algoritmo que calcule la forma óptima de clasificar las instancias a partir de los datos

Machine learning = Aprendizaje máquina o automático

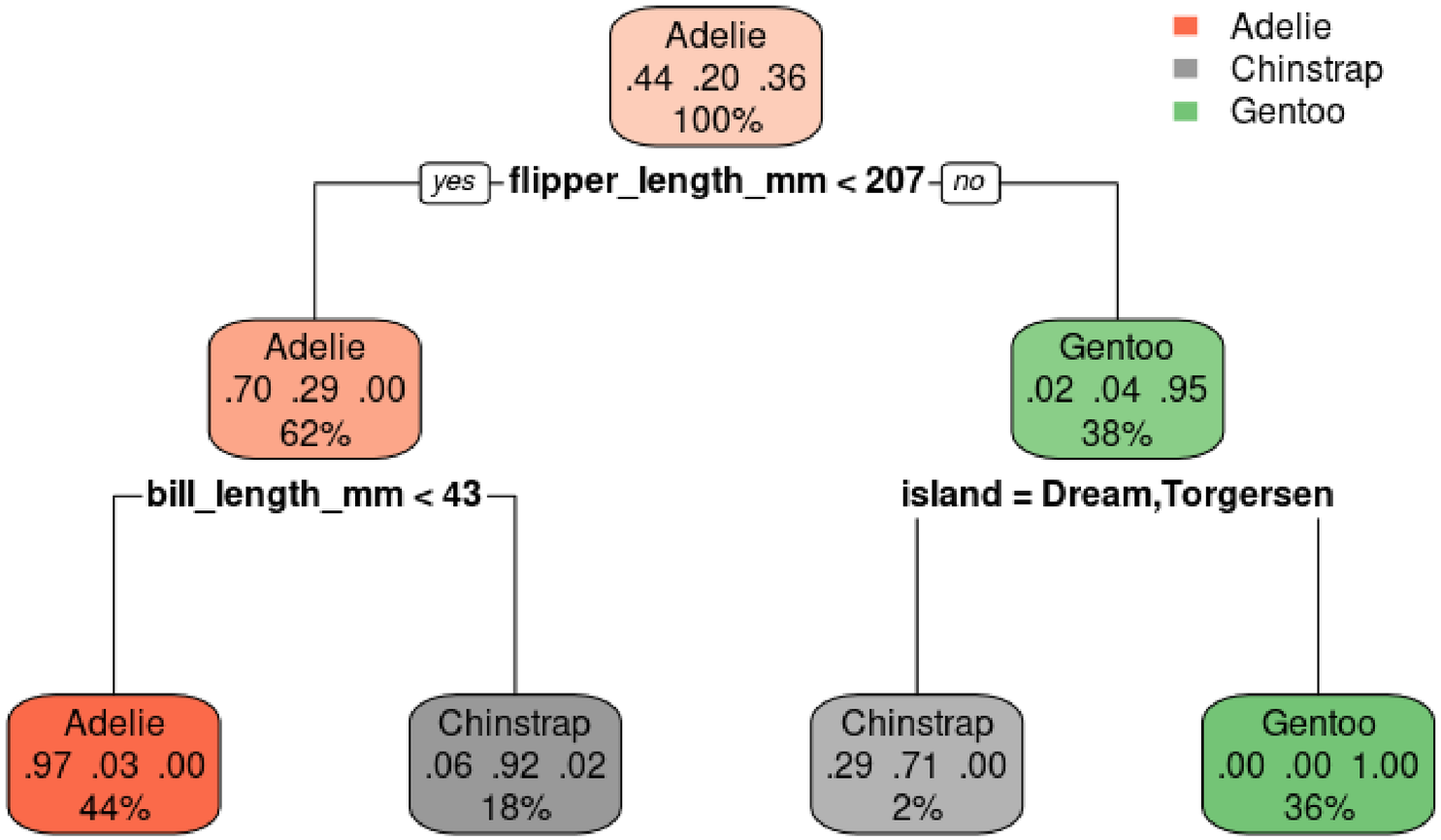
Data mining = Minería de datos

GHENT UNIVERSITY

imec

PROBLEMA: ESPECIES DE PINGÜINOS

– Ejemplo: árbol de decisión



GHENT UNIVERSITY

AGENDA

ESTRUCTURA DEL WORKSHOP

- Día 1:
  - Introducción a machine learning y data mining
    - Tipos de machine learning
    - Definición de Big Data
    - Aplicaciones y ejemplos
  - Machine learning como problema matemático
  - Entrenamiento, validación y test
  - Clasificación y regresión
  - Clustering
  - Reducción de dimensionalidad y visualización
  - Justicia y explicabilidad

GHENT UNIVERSITY

imec

ESTRUCTURA DEL WORKSHOP

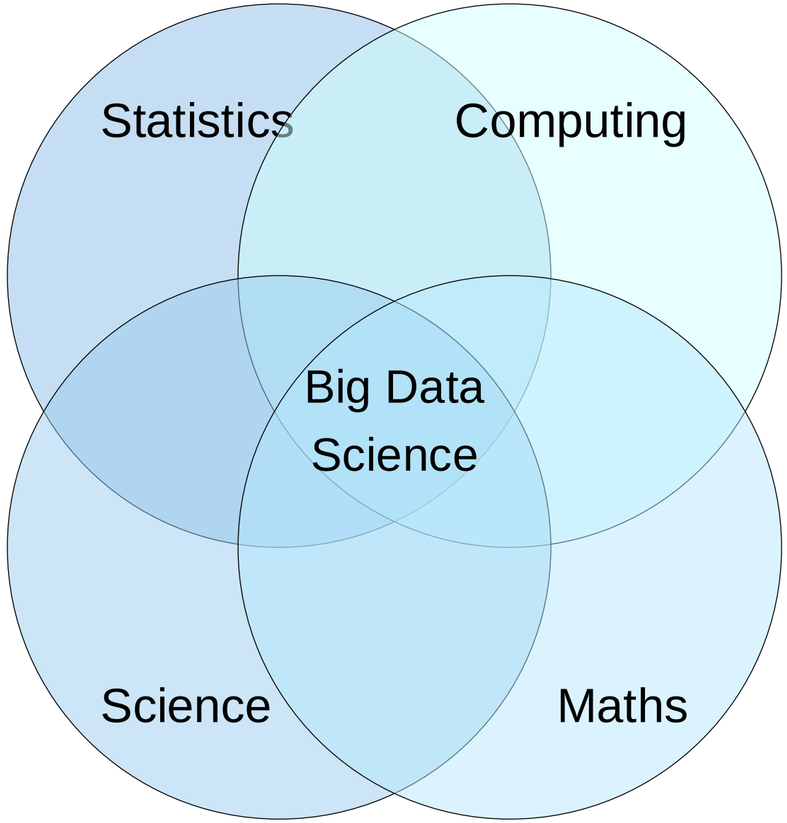
- Día 2:
  - Redes neuronales
  - Redes convolutivas
  - Modelos generativos
  - Transformers

  - Talleres prácticos:
    - Regresión y reducción de dimensionalidad
    - Clasificación de malaria con redes convolutivas
    - Generación de imágenes con un Variational Autoencoder

GHENT UNIVERSITY

imec

FOUR ELEMENTS OF "BIG DATA SCIENCE"



THIS COURSE IS NOT ABOUT...

- Advanced programming
- Specific frameworks and tools
- Technical and infrastructural aspects of big data management

WHO AM I?

PhD Student	Postdoc	Postdoc	Doctor assistant	Assistant prof.
2012	2016	2018	2021	2024
Prof. Francisco Herrera Prof. José Manuel Benítez	Prof. Yvan Saeys		Prof. Eli De Poorter	

CUÁL ES TU BACKGROUND?

- Ir a www.menti.com
- Usar el código: 8232 4685

TIPOS DE MACHINE LEARNING

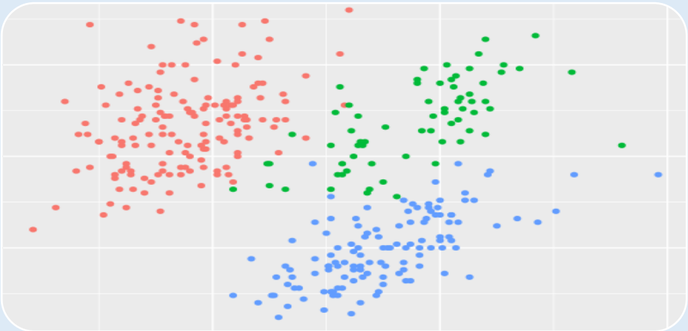
COMPUTADORAS Y ALGORITMOS

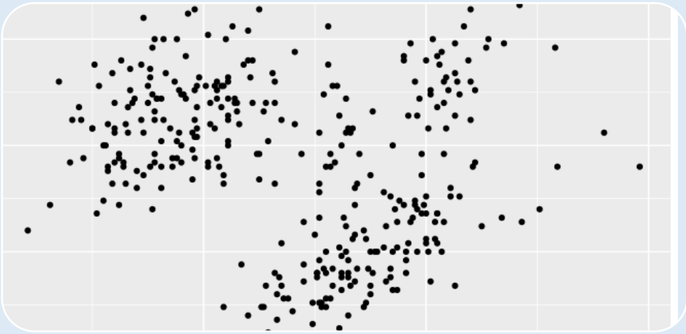
- "Una computadora es un tonto muy rápido"

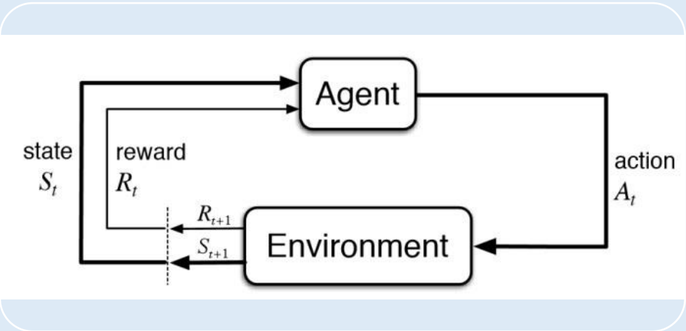
- **Algoritmo**: secuencia de pasos para resolver un problema

- Ejemplos:
  - Buscar datos
  - Ordenar datos
  - Encaminar paquetes en una red o Internet
  - Estimar las ganancias de una empresa
  - Simular la estructura de una molécula
  - Calcular los impuestos de un/a empleado/a
  - ...

TIPOS DE APRENDIZAJE MÁQUINA







Aprendizaje supervisado

- Datos
- Etiquetas

Aprendizaje no supervisado

- Sólo datos

Aprendizaje por refuerzo:

- Agente
- Entorno
- Recompensas

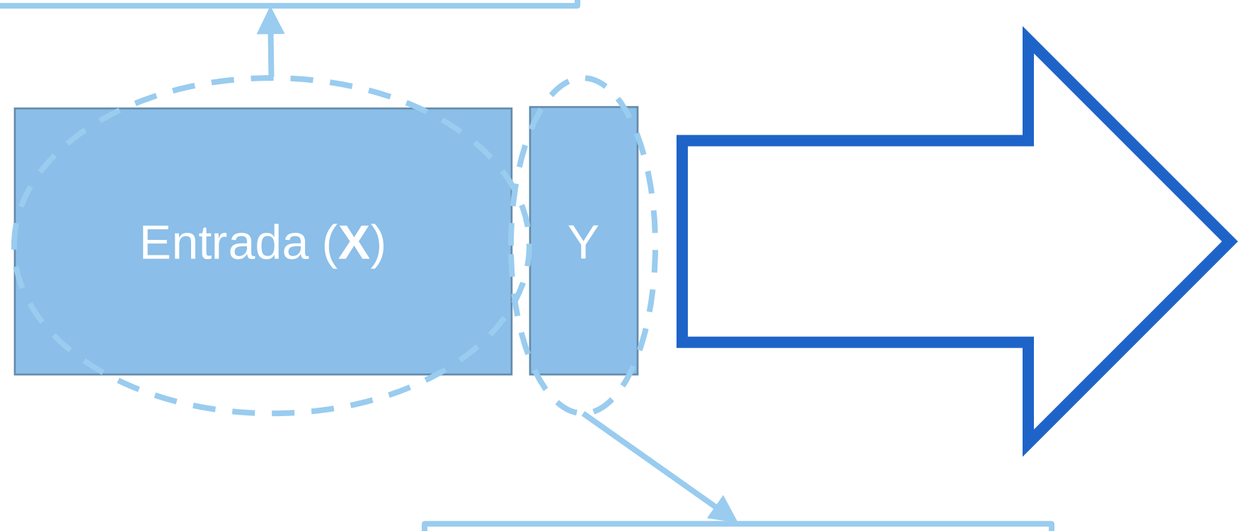
GHENT UNIVERSITY

imec

APRENDIZAJE SUPERVISADO

- Entradas
- Predictores
- Variables independientes
- Características

Conjunto de entrenamiento



- Salidas
- Respuestas
- Variables dependientes

Modelo

APRENDIZAJE NO SUPERVISADO

- Entradas
- Predictores
- Variables independientes
- Características

Conjunto de entrenamiento

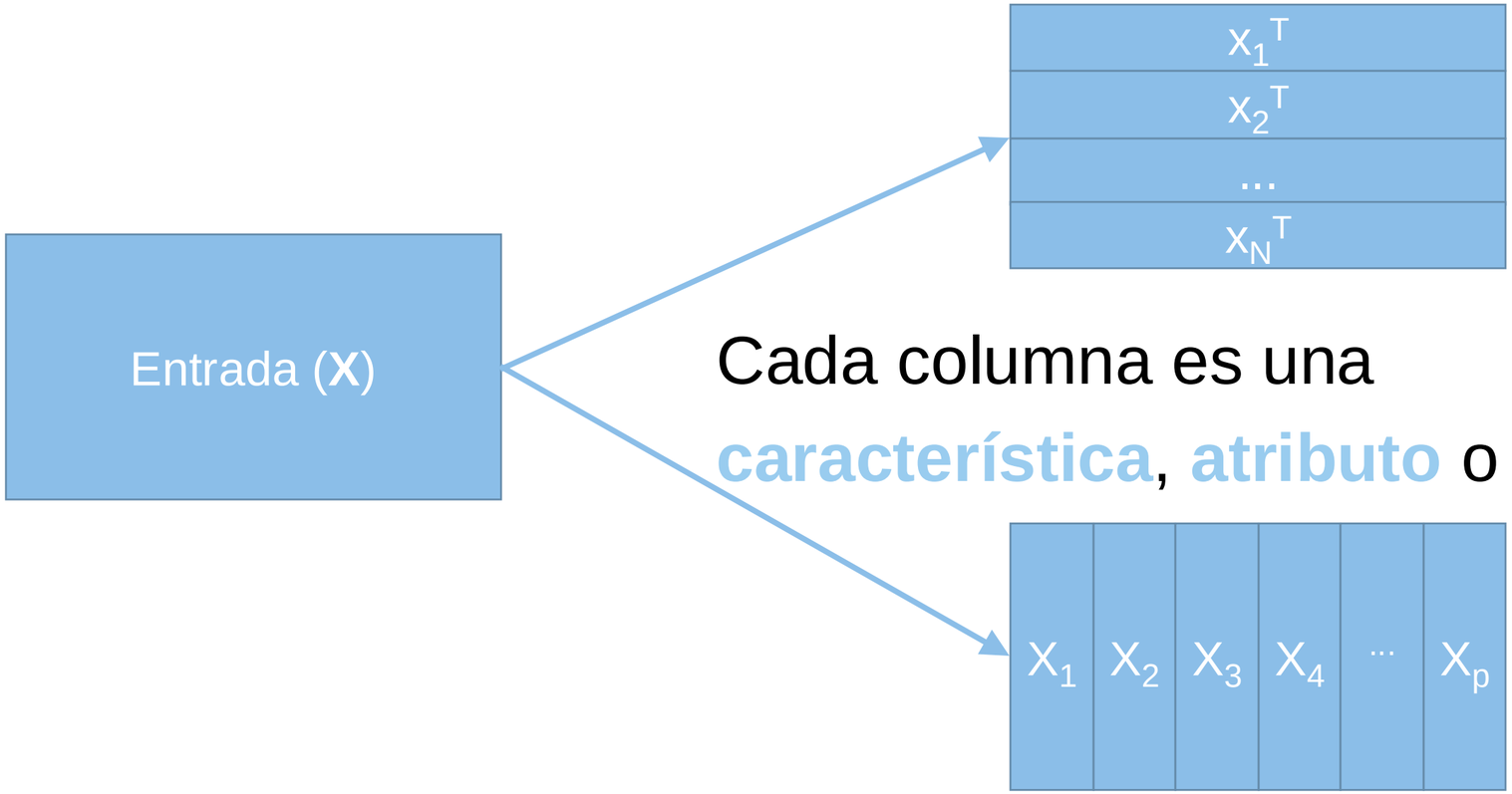
Entrada (**X**)

Modelo

DATOS DE ENTRADA

– Casi siempre, los datos se representan en forma de tabla o matriz

Cada fila es una **instancia**



$$x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$$

Cada columna es una **característica**, **atributo** o **variable**

$$X_j = (x_{1j}, x_{2j}, \dots, x_{Nj})$$

GHENT UNIVERSITY

imec

DATOS DE SALIDA

Típicamente, hay un valor por instancia, llamado **etiqueta**

x_1^T	y_1
x_2^T	y_2
...	...
x_N^T	y_N

X	Y

Cuantitativa: problema de **regresión**

$$Y \in \mathbb{R}$$

Cualitativa: problema de **clasificación**

$$G \in \{1,2,\dots,K\} = \mathcal{G}$$

GHENT UNIVERSITY

imec

WEAKLY SUPERVISED LEARNING

Fully supervised

Positive-unlabeled

Candidate labels

Probabilistic labels

Semi-supervised

Incomplete (multi-label only)

Noisy labels

Crowd

Mutual label constraints

Candidate labeling vectors

Label proportions

Unsupervised

GHENT UNIVERSITY

imec

APRENDIZAJE SUPERVISADO

TIPOS DE APRENDIZAJE SUPERVISADO

Clasificación

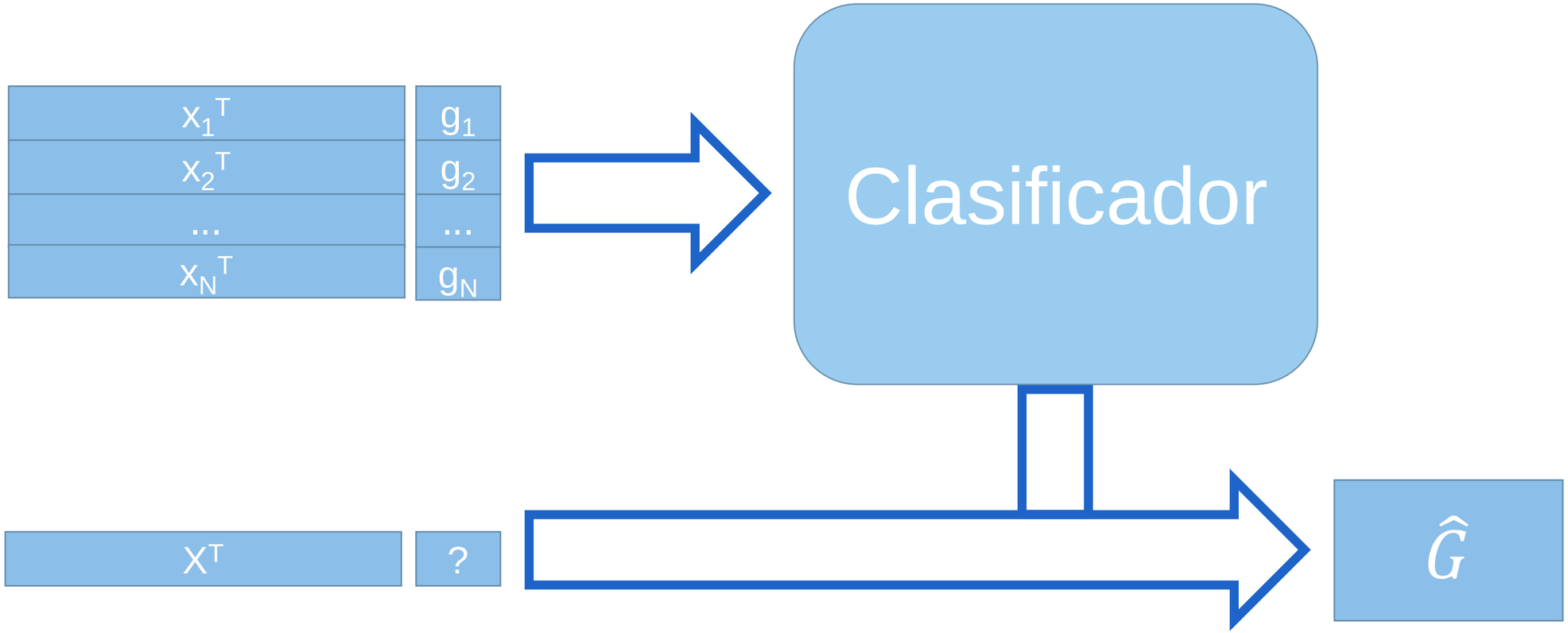
- La etiqueta es discreta (clase)

Regresión

- La etiqueta es un valor continuo

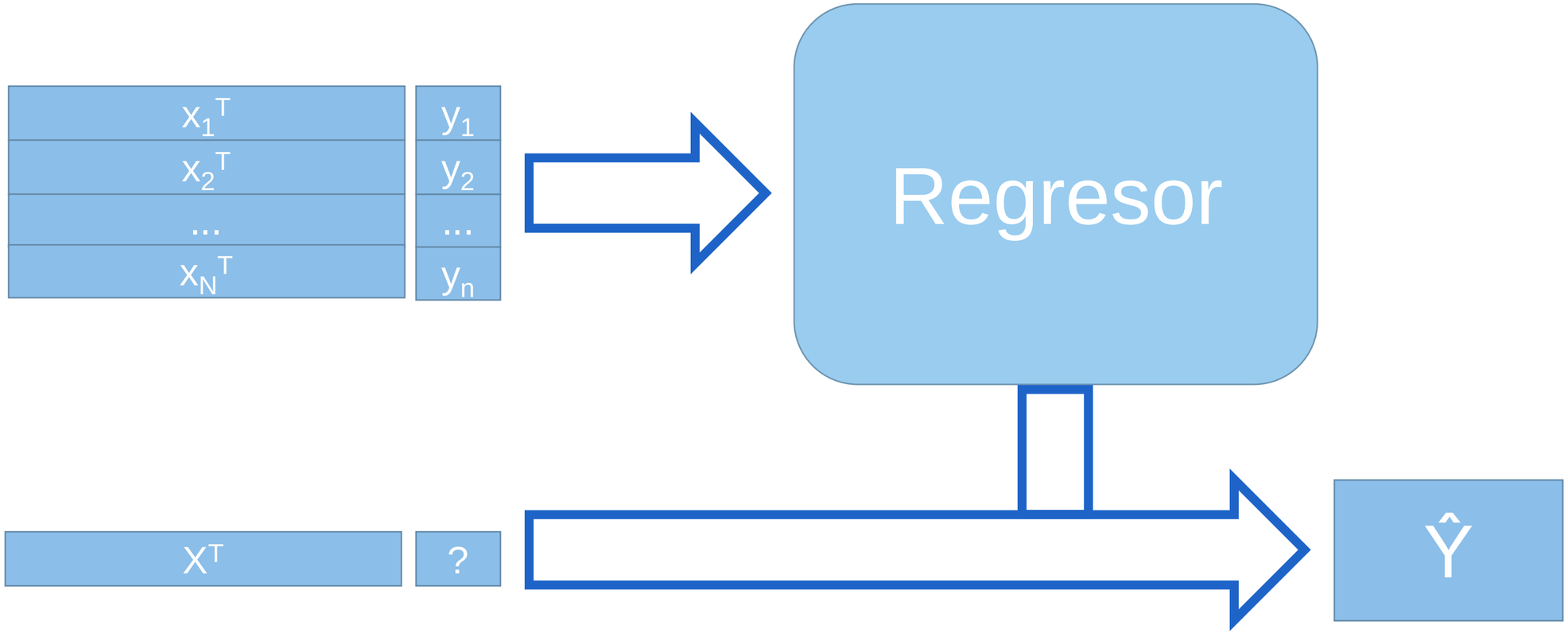
CLASIFICACIÓN

$$G \in \{1,2,\dots,K\}$$



REGRESIÓN

$y \in \mathbb{R}$



GHENT UNIVERSITY

imec

CLASSIFICATION AND REGRESSION

- Classification can often be treated as a regression problem
  - Binary classification

$$g \in \{1,2\} \quad \Rightarrow \quad y \in \{0,1\}$$

Predict 2 if $\hat{y}>0.5$
Predict 1 otherwise

Or even better: predict the probability of each class (logistic regression)

  - Multi-class classification
    - Can be decomposed into K binary classification problems

EJEMPLOS DE PROBLEMAS DE REGRESIÓN

- Estimar el precio de una vivienda
- Predecir la actividad de un compuesto químico sobre una proteína
- Predecir la cantidad de fallos en vías ferroviarias, dada información de control de calidad
- Predecir el tiempo de vida de un paciente
- ...

EJEMPLOS DE PROBLEMAS DE CLASIFICACIÓN

- Reconocer números escritos a mano
- Determinar si un e-mail es SPAM o no
- Agrupar huellas dactilares en función de su patrón general
- Predecir si un paciente tiene una enfermedad o no, en base a sus datos clínicos
- Predecir el tipo del próximo fallo en una vía ferroviaria
- ...

MÁS EJEMPLOS

- Ir a www.menti.com
- Usar el código: 8232 4685

APRENDIZAJE NO SUPERVISADO

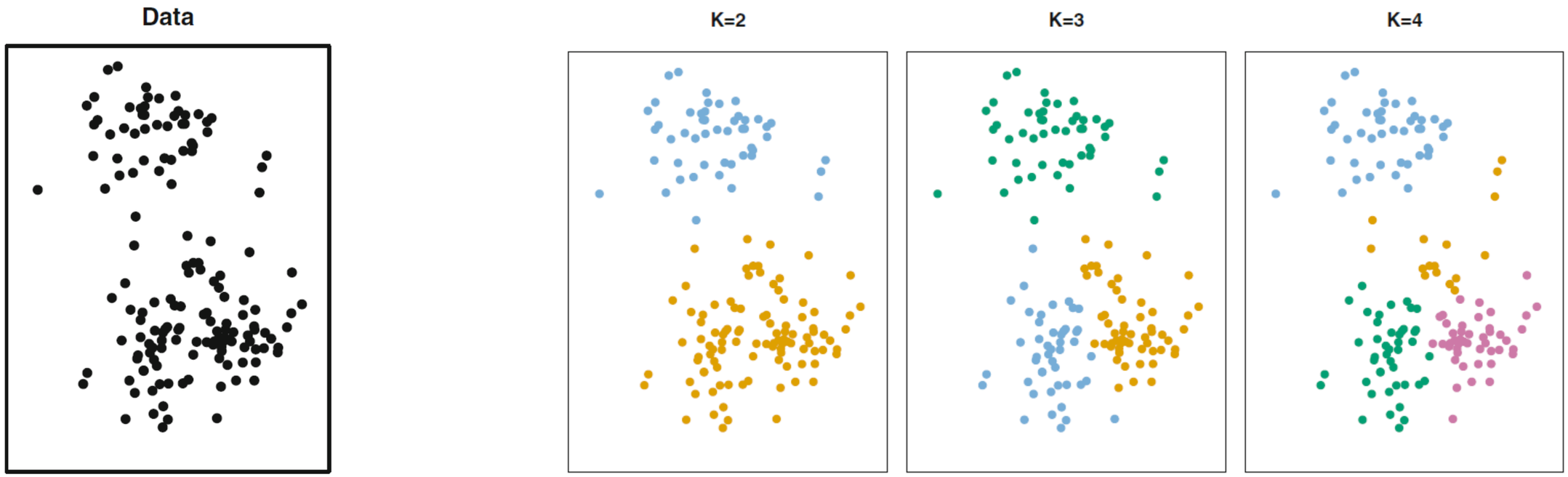
ALGUNOS EJEMPLOS

- Clustering
- Reglas de asociación
- Seriación
- Reducción de dimensión
- ...

- **Dificultad**: el objetivo no es predecir etiquetas, sino encontrar una estructura en los datos

CLUSTERING

– Buscamos agrupar elementos similares en grupos:



REGLAS DE ASOCIACIÓN

- Dado un listado de conjuntos, ¿qué elementos aparecen juntos frecuentemente?

- Ejemplo: cesta de la compra (*market-basket model*)

CESTA DE LA COMPRA

- Gran conjunto de elementos (artículos de supermercado)
- Gran conjunto de cestas
- Cada cesta es un pequeño conjunto de artículos
- Queremos descubrir reglas de asociación:
  - "Quien compró {x,y,z} también compró {v,w}"

Cesta	Artículos
1	Pan, Refresco, Leche
2	Cerveza, Pan
3	Cerveza, Refresco, Pañales, Leche
4	Cerveza, Pan, Pañales, Leche
5	Refresco, Pañales, Leche

Reglas descubiertas:

{Leche} → {Refresco}

{Pañales, Leche} → {Cerveza}

CESTA DE LA COMPRA

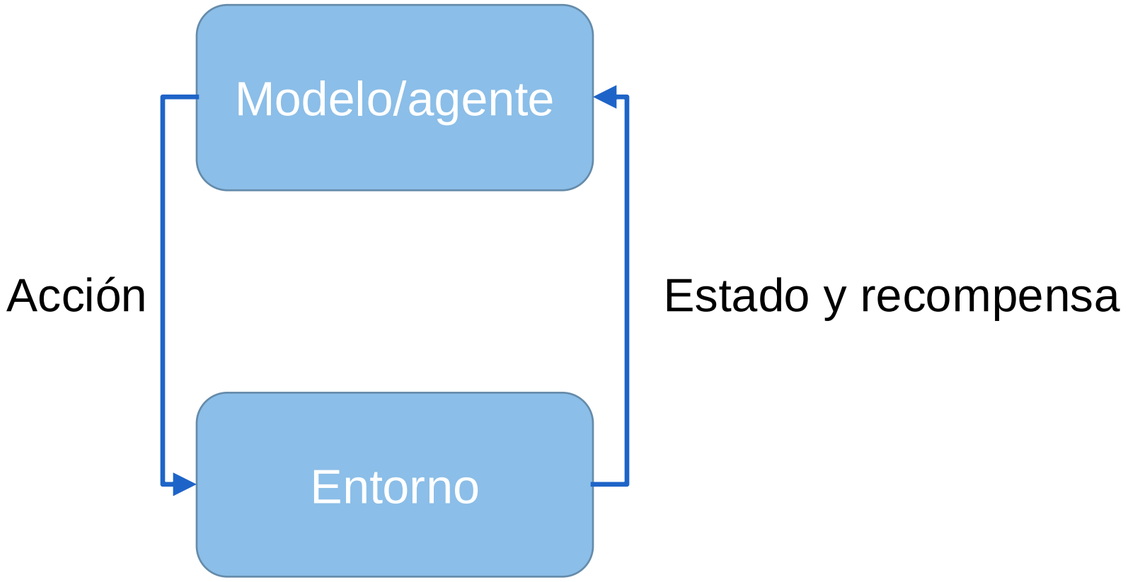
- ¡Contar elementos no es suficiente!

- Para una buena regla A → B, queremos:
  - A **frecuente** (pero no demasiado)
  - **Confianza**: B aparece (casi) siempre que aparece A
  - **Interés**: B no es frecuente si no aparece A

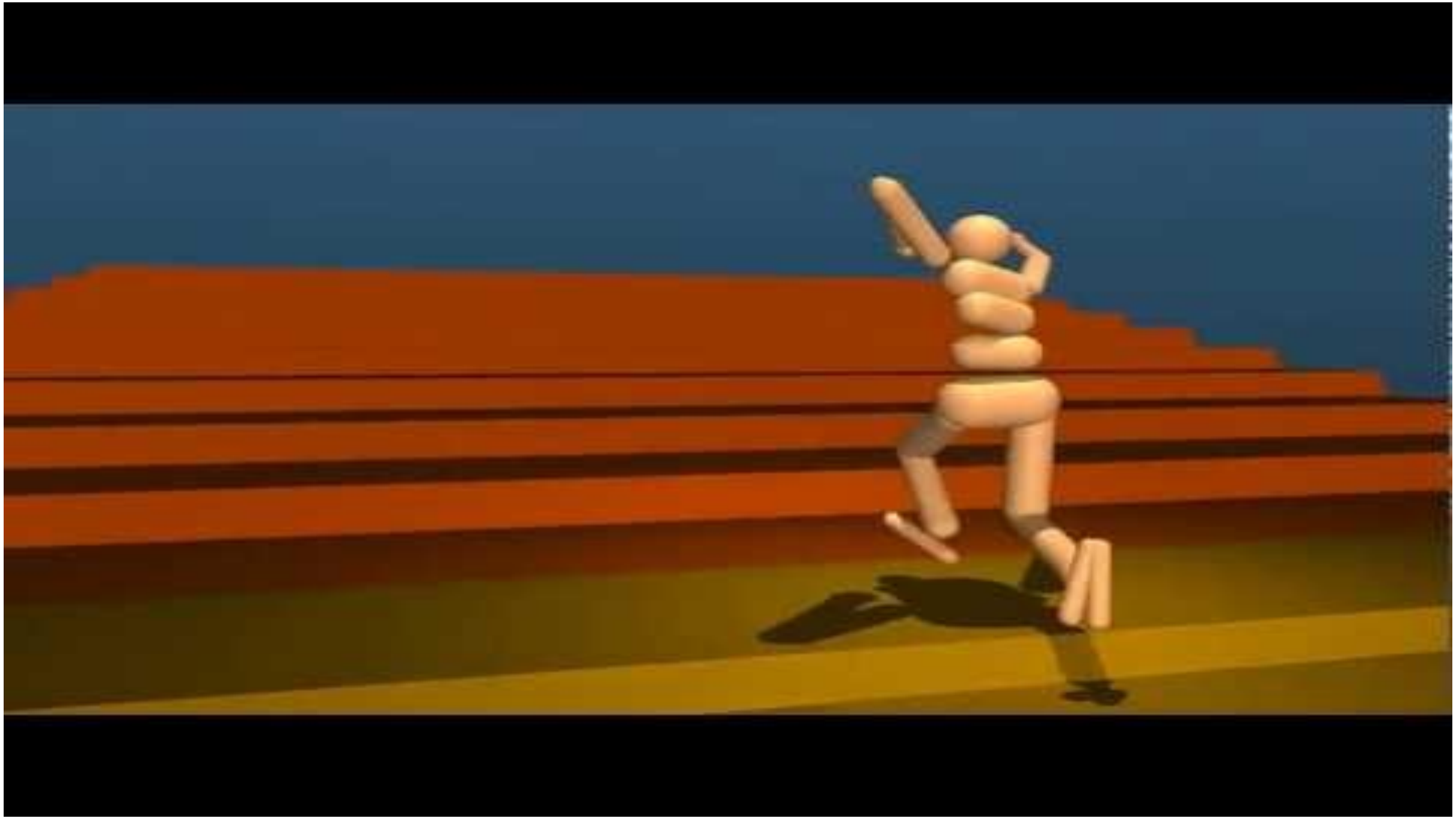
APRENDIZAJE POR REFUERZO

APRENDIZAJE POR REFUERZO

- No es supervisado ni no supervisado
- No se basa en datos
- El modelo interactúa con su entorno y recibe una recompensa positiva o negativa



APRENDIZAJE POR REFUERZO



GHENT UNIVERSITY

imec

BIG DATA

NO EXISTE UNA DEFINICIÓN FORMAL

Big data es cuando la aplicación normal de la tecnología actual no permite a los usuarios obtener respuestas de calidad a un coste y tiempo razonable ante preguntas basadas en datos

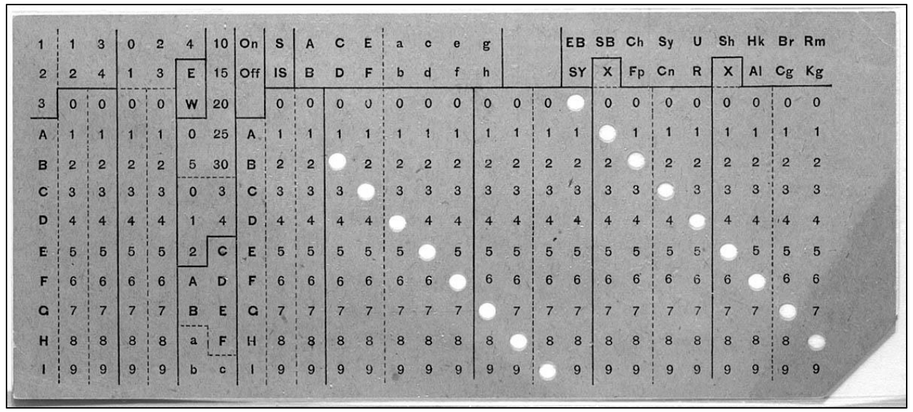
El **Big data** implica datos cuyo volumen, diversidad y complejidad requieren nuevas técnicas, algoritmos y análisis para extraer conocimiento valioso.

GHENT UNIVERSITY

imec

EL BIG DATA EMPEZÓ… EN 1890

Hollerith Tabulator



Tarjeta perforada Hollerith

Utilizado por primera vez en el censo de USA en 1890

Los resultados del censo fueron "...terminados con meses de adelanto y muy por debajo del presupuesto."

Hollerith fundó más tarde la Computing Tabulating Recording Company (CTR), la cual fue renombrada en 1924 a …

GHENT UNIVERSITY

imec

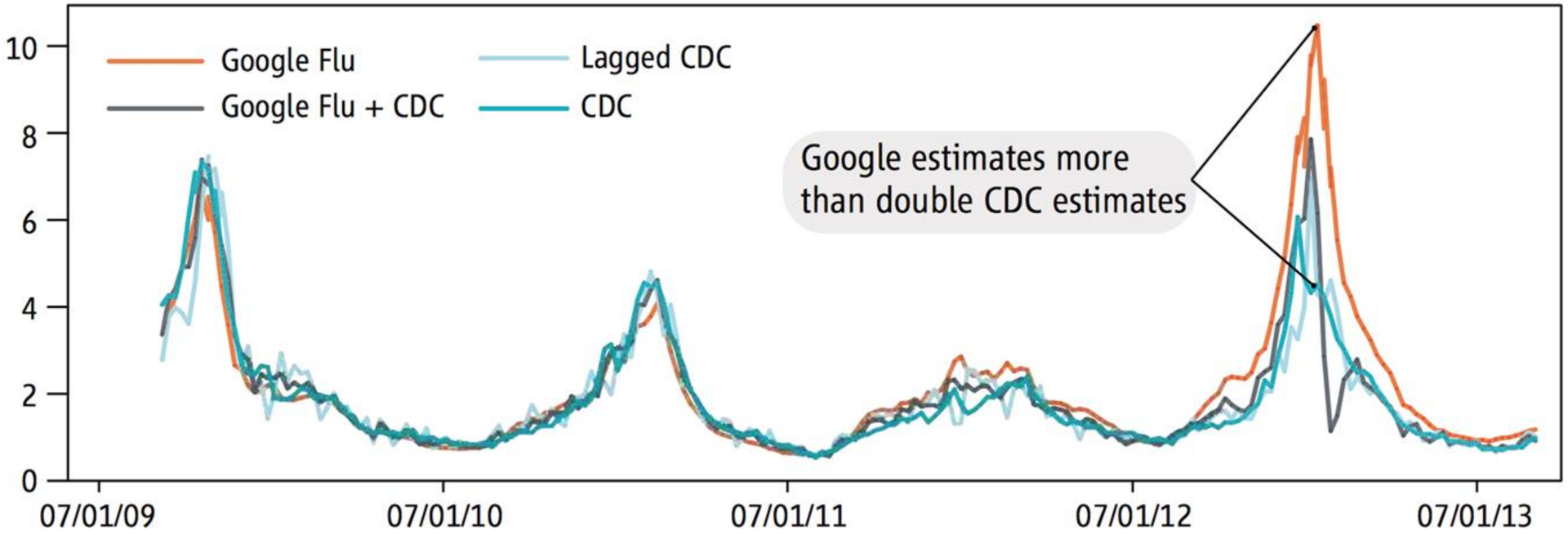
LA PARÁBOLA DE GOOGLE FLU

The Parable of Google Flu: Traps in Big Data Analysis
(David Lazer et al., Science, March 2014)

- Google Flu Trends arrancó en 2008
- Identificar la expansión anual de la gripe a travéz de términos de búsqueda
- Excelente correlación con el indicador del CDC*
- Se volvió ejemplar para el poder del Big Data
- “Las redes sociales pueden predecir X” se ha vuelto habitual

*U.S. Centers for Disease Control and Prevention (CDC)

BIG DATA HUBRIS



LAS 3 V'S DEL BIG DATA

GHENT UNIVERSITY

imec

VOLUMEN

- Cada segundo se generan grandes cantidades de datos
- Los conjuntos de datos se vuelven demasiado grandes para almacenar usando tecnología tradicional
- En entornos Big Data, se almacenan utilizando **sistemas distribuidos**

GHENT UNIVERSITY

imec

VOLUMEN

Multiplying Factor	Name (symbol)	Scientific Notation
1 000 000 000 000 000 000 000 000 000 000	quetta (Q)	10^{30}
1 000 000 000 000 000 000 000 000 000	ronna (R)	10^{27}
1 000 000 000 000 000 000 000 000	yotta (Y)	10^{24}
1 000 000 000 000 000 000 000	zetta (Z)	10^{21}
1 000 000 000 000 000 000	exa (E)	10^{18}
1 000 000 000 000 000	peta (P)	10^{15}
1 000 000 000 000	tera (T)	10^{12}
1 000 000 000	giga (G)	10^{9}
1 000 000	mega (M)	10^{6}

GHENT UNIVERSITY

imec

A DAY IN DATA

The exponential growth of data is undisputed, but the numbers behind this explosion – fuelled by internet of things and the use of connected devices – are hard to comprehend, particularly when looked at in the context of one day

DEMYSTIFIYING DATA UNITS

From the more familiar 'bit' or 'megabyte', larger units of measurement are more frequently being used to explain the masses of data

Unit		Value	Size
b	bit	0 or 1	1/8 of a byte
B	byte	8 bits	1 byte
KB	kilobyte	1,000 bytes	1,000 bytes
MB	megabyte	$1{,}000^2$ bytes	1,000,000 bytes
GB	gigabyte	$1{,}000^3$ bytes	1,000,000,000 bytes
TB	terabyte	$1{,}000^4$ bytes	1,000,000,000,000 bytes
PB	petabyte	$1{,}000^5$ bytes	1,000,000,000,000,000 bytes
EB	exabyte	$1{,}000^6$ bytes	1,000,000,000,000,000,000 bytes
ZB	zettabyte	$1{,}000^7$ bytes	1,000,000,000,000,000,000,000 bytes
YB	yottabyte	$1{,}000^8$ bytes	1,000,000,000,000,000,000,000,000 bytes

*A lowercase "b" is used as an abbreviation for bits, while an uppercase "B" represents bytes.

463EB

of data will be created every day by 2025

IDC

500m

tweets are sent every day

Twitter

4PB

of data created by Facebook, including

350m photos

100m hours of video watch time

Facebook Research

95m

photos and videos are shared on Instagram

Instagram Business

320bn

emails to be sent each day by 2021

306bn

emails to be sent each day by 2020

294bn

billion emails are sent

Radicati Group

3.9bn

people use emails

4TB

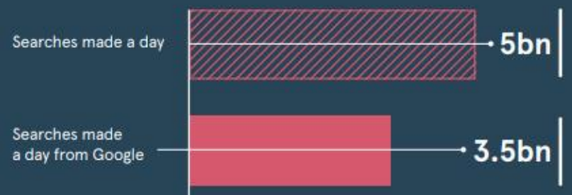
of data produced by a connected car

Intel

65bn

messages sent over WhatsApp and two billion minutes of voice and video calls made

Facebook



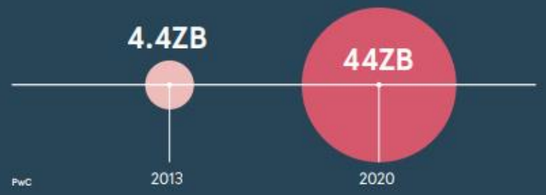
Smart insights

28PB

to be generated from wearable devices by 2020

Statista

ACCUMULATED DIGITAL UNIVERSE OF DATA



PwC

VELOCIDAD

- Velocidad a la que se generan los datos:
  - Tráfico de datos, conversaciones telefónicas, transacciones en cajeros, búsquedas web, sensores...

- Velocidad a la que los datos deben analizarse

- Decisiones tardías implican oportunidades perdidas

- Sistemas en tiempo real: un resultado tardío ya no es útil:
  - Predicción del tiempo
  - Diagnósticos médicos
  - ...

VELOCIDAD: DATOS DE STREAMING

- Los datos llegan constantemente y deben ser procesados

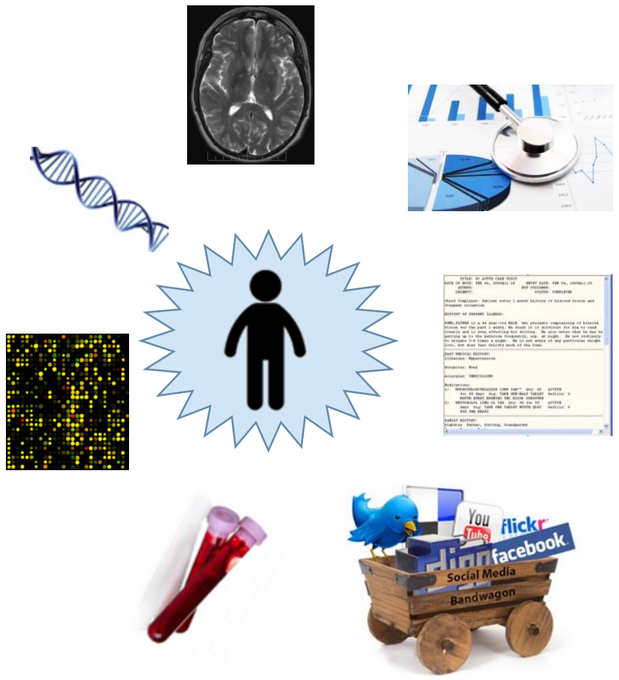
- Los datos se procesan sin necesidad de almacenarlos

GHENT UNIVERSITY

imec

VARIEDAD

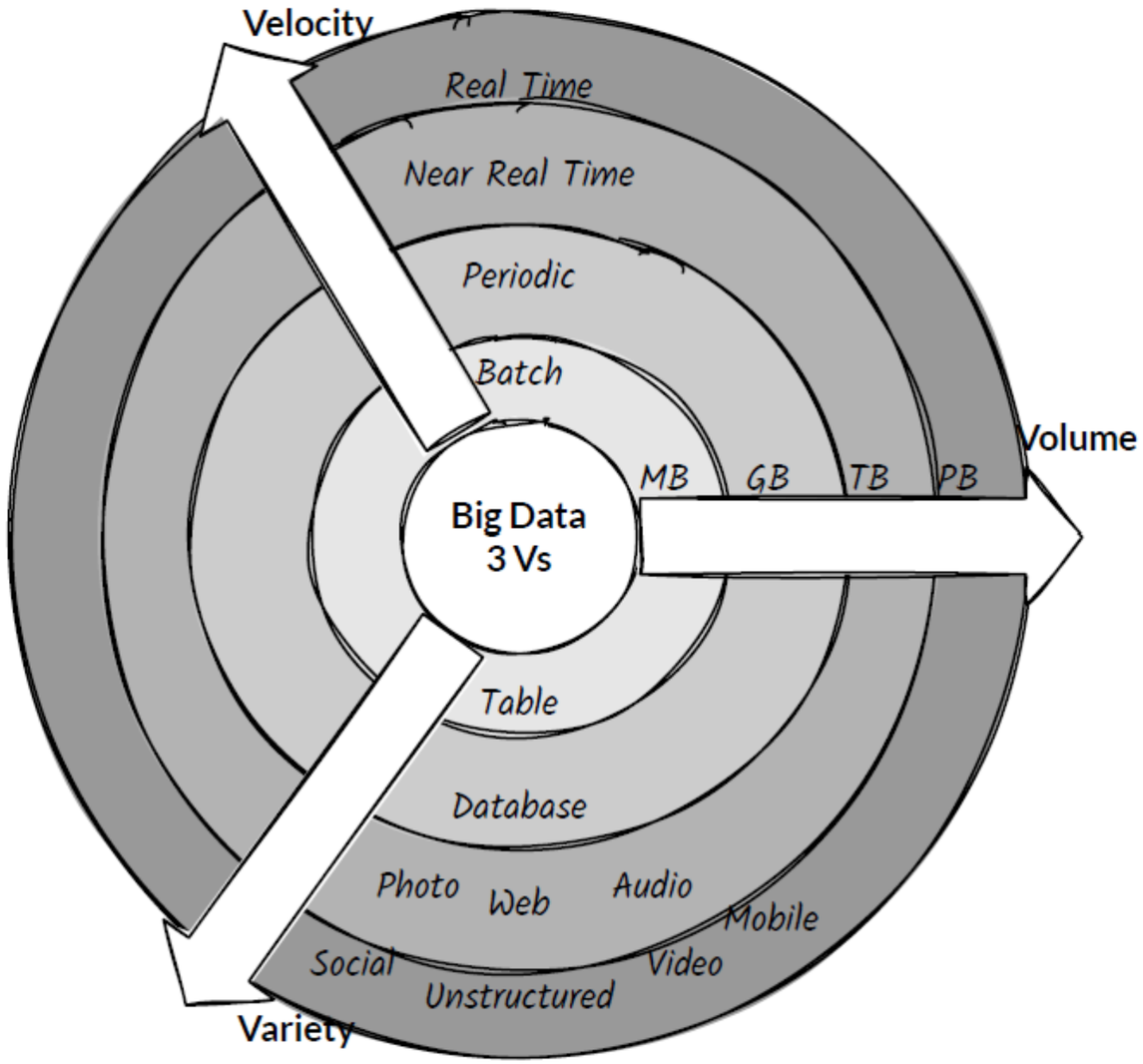
- Necesidad de combinar muchos tipos de datos
  - Estructurados
    - Tablas
    - Bases de datos relacionales
  - No estructurados
    - Imágenes, vídeo, sonido
    - Redes sociales
    - Texto



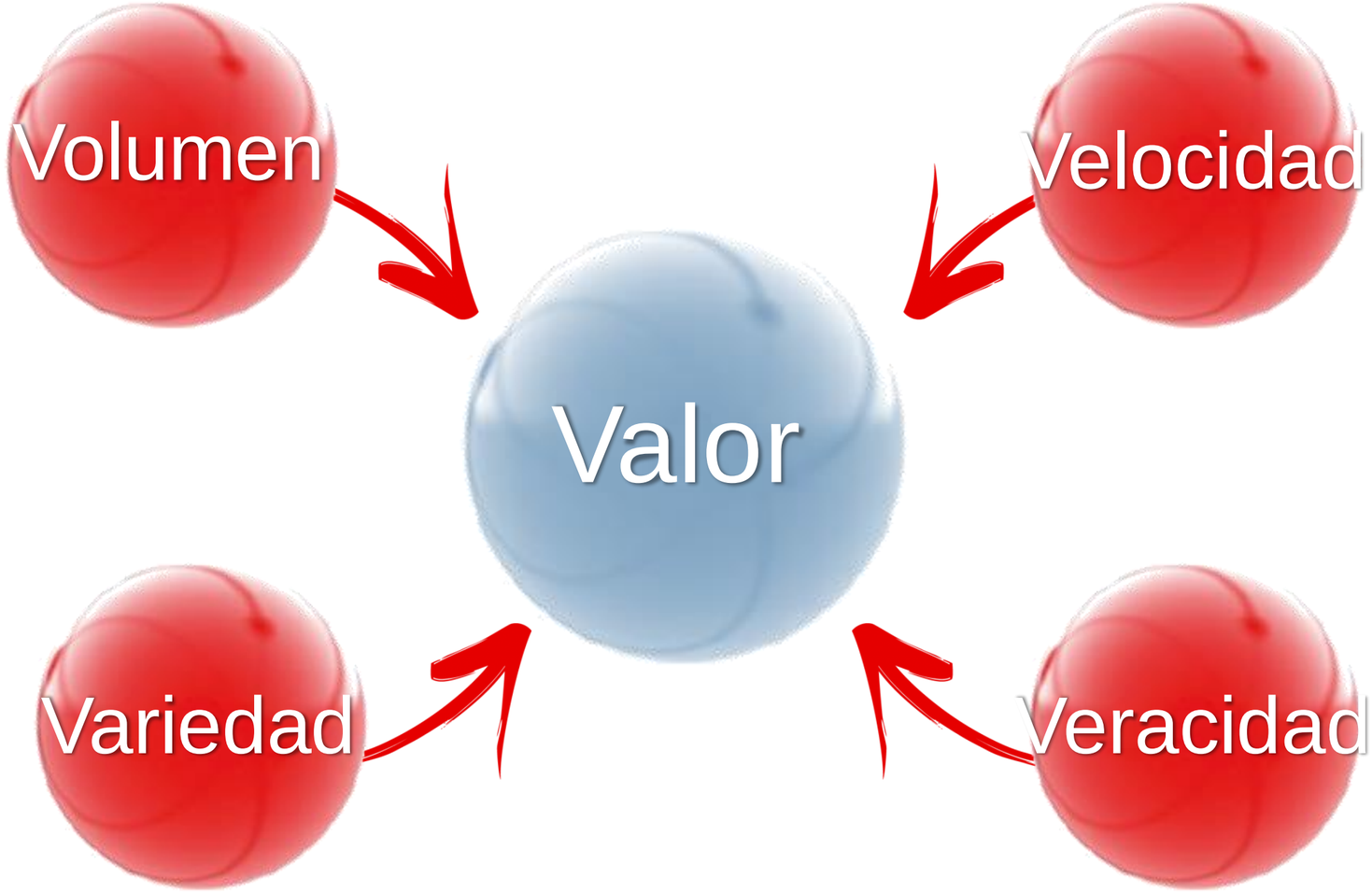
GHENT UNIVERSITY

imec

3 V'S


Velocity
Real Time
Near Real Time
Periodic
Batch
Volume
MB
GB
TB
PB
Big Data
3 Vs
Table
Database
Photo
Web
Audio
Mobile
Social
Video
Unstructured
Variety

5 V'S



GHENT UNIVERSITY

imec

VERACIDAD

- La calidad y precisión del Big Data son difíciles de controlar
  - Ejemplo: procesamiento de lenguaje natural en redes sociales:
    - Hashtags, abreviaturas, faltas, ironía, sarcasmo, habla coloquial...
- Los datos pueden ser incompletos, ambiguos o incluso totalmente erróneos

VALOR

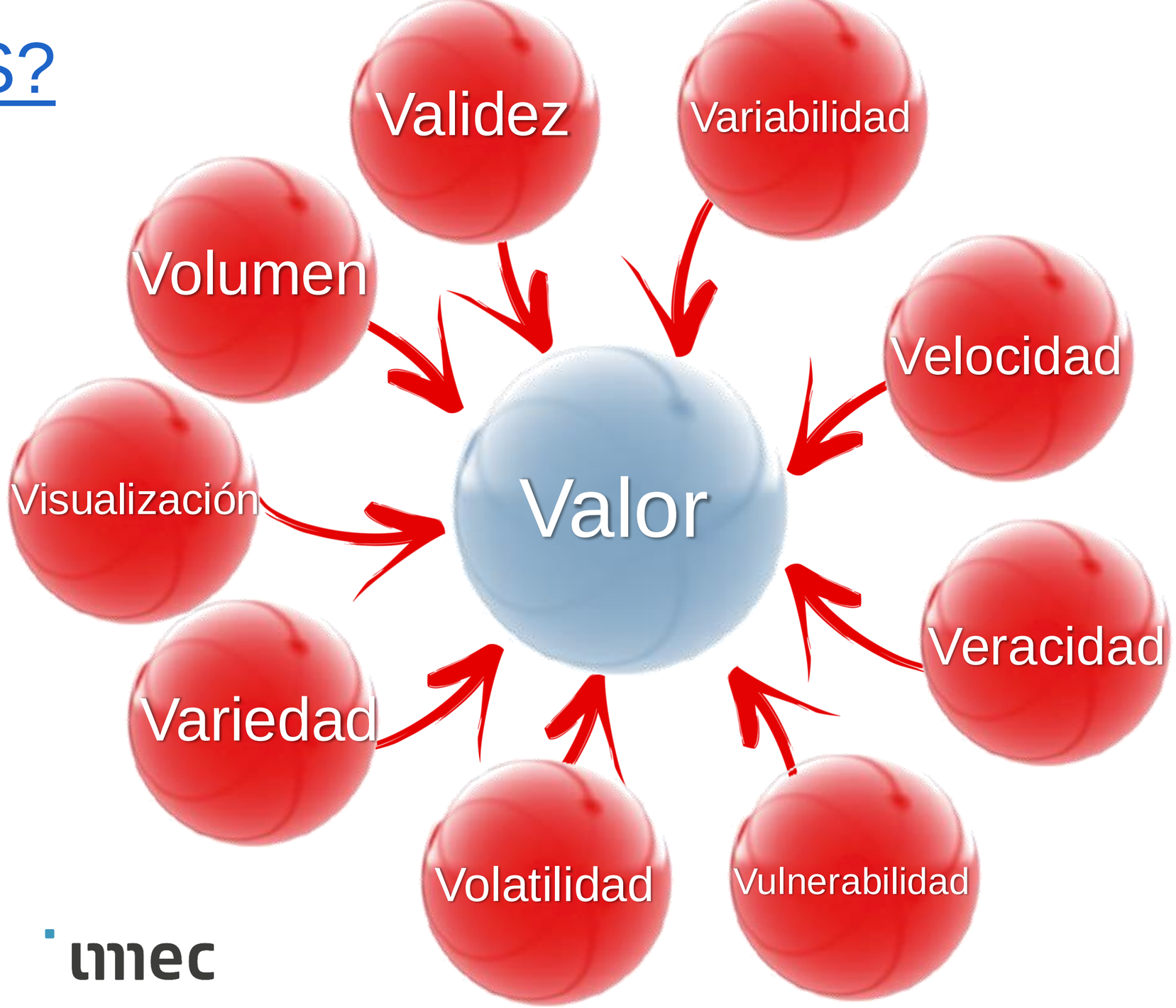
- Motivación más importante para el Big Data
- Puede resultar en:
  - Mejores estadísticas y models
  - Nuevo conocimiento
  - Nuevas oportunidades para investigación e industria

- Es importante describir el caso (de negocio) para recoger y procesar Big Data (costes/beneficios)

GHENT UNIVERSITY

imec

10 V'S?



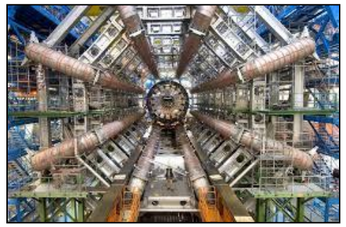
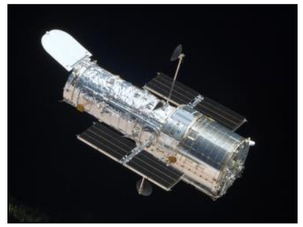
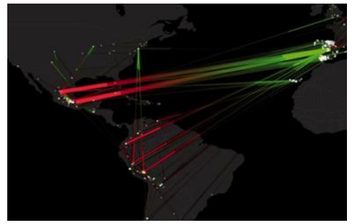
GHENT UNIVERSITY

imec

ÁREAS DE APLICACIÓN DEL BIG DATA

ÁREAS DE APLICACIÓN DEL BIG DATA

- Ciencia e investigación
  - Física (LHC), Bioinformática, Astronomía, Historia
- Sanidad y salud pública
  - Medicina personalizada, epidemiología

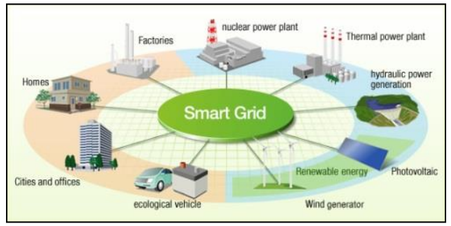




GHENT UNIVERSITY

imec

ÁREAS DE APLICACIÓN DEL BIG DATA

- Negocios y comercio
  - Ejemplo: predecir cuándo un cliente espera un bebé, o va a cambiar de operador
- Economía y sostenibilidad
  - Optimización de logística, smart grids



GHENT UNIVERSITY

imec

ÁREAS DE APLICACIÓN DEL BIG DATA

- Deportes y actividades sociales
  - Monitorización de partidos, citas online
- Planificación urbana y arquitectura
  - Optimización de tráfico
- Seguridad

STATISTICAL DECISION THEORY

STATISTICAL DECISION THEORY

- A predictor is a function $f(X)$ with which we want to approximate Y
- The error is measured by a **loss function** $L\big(Y, f(X)\big)$
- Most used: squared error loss:

$$L\big(Y, f(X)\big) = \big(Y - f(X)\big)^2$$

STATISTICAL DECISION THEORY

- Which *f* should we choose?
- Obviously, the one that minimizes the expected error!

$$EPE(f) = E\big(Y - f(X)\big)^2 = \int [y - f(x)]^2 \Pr(dx, dy)$$

After reduction:

$$f(x) = E(Y|X = x)$$

CLASSIFICATION CASE

– For a discrete output, the 0-1 loss function is the most used:

$$EPE = E\left[L\left(Y, \hat{G}(X)\right)\right]$$

Bayes classifier:

$$\hat{G}(x) = \max_{g \in \mathcal{g}} \Pr(g|X = x)$$

BAYES CLASSIFIER

- The Bayes classifier yields optimal squared error loss
- In practice, we don't know the true distribution
- Interesting to evaluate methods on simulated data
- If you know the distribution, use it!

Bayes Optimal Classifier

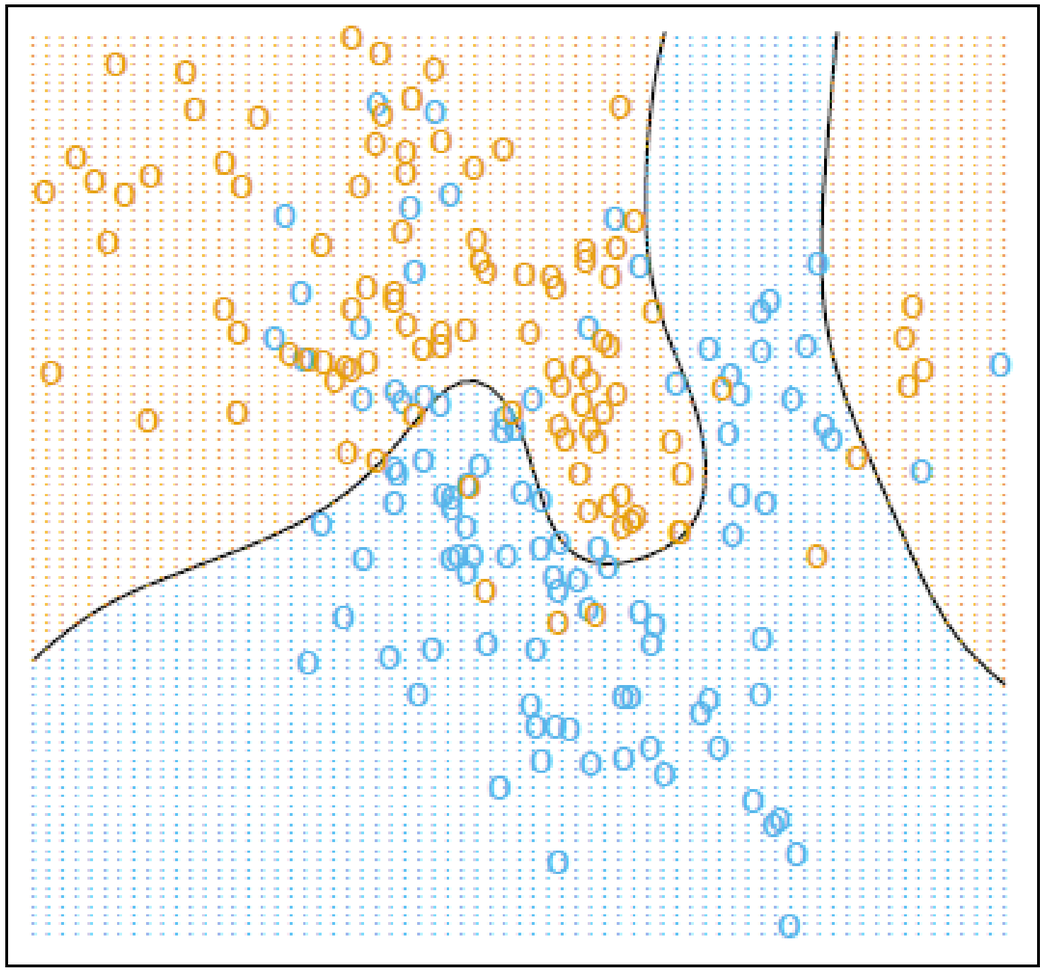


FIGURE 2.5. *The optimal Bayes decision boundary for the simulation example of Figures 2.1, 2.2 and 2.3. Since the generating density is known for each class, this boundary can be calculated exactly (Exercise 2.2).*

FIRST APPROACHES TO PREDICTION

TWO EXTREME APPROACHES

Parametric:

linear model

Strong structural assumptions

Stable predictions

Can be inaccurate

Non-parametric:

k-nearest neighbors (kNN)

Mild structural assumptions

Predictions can be unstable

Often accurate

LINEAR MODELS

- Assumption: the relation between inputs and outputs is linear

$$X^T = (X_1, X_2, \dots, X_p)$$

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

Intercept

Coefficients

- The bias can be modeled as a coefficient if we include a constant variable 1 in X, which makes:

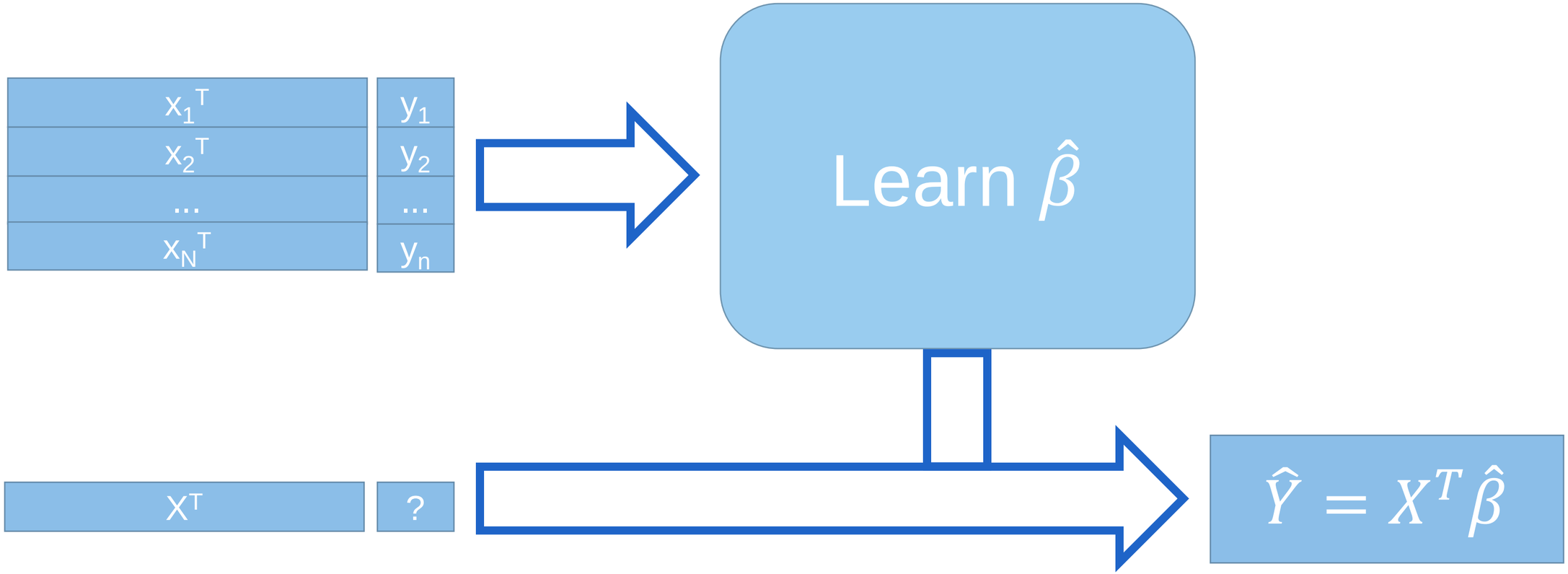
$$\hat{Y} = X^T \hat{\beta}$$

GHENT UNIVERSITY

imec

FITTING A LINEAR MODEL

$y \in \mathbb{R}$



GHENT UNIVERSITY

imec

FITTING A LINEAR MODEL

- The most used method is "**least squares**"
- Calculate $\hat{\beta}$ so that the residual sum of squares is minimized:

$$RSS(\beta) = \sum_{i=1}^{N} \left(y_i - x_i^T \beta\right)^2 = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$

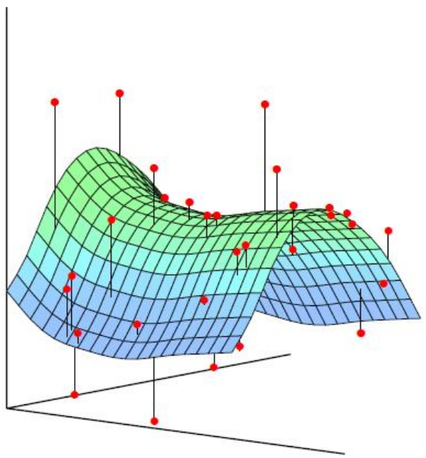


FIGURE 2.10. *Least squares fitting of a function of two inputs. The parameters of $f_\theta(x)$ are chosen so as to minimize the sum-of-squared vertical errors.*

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

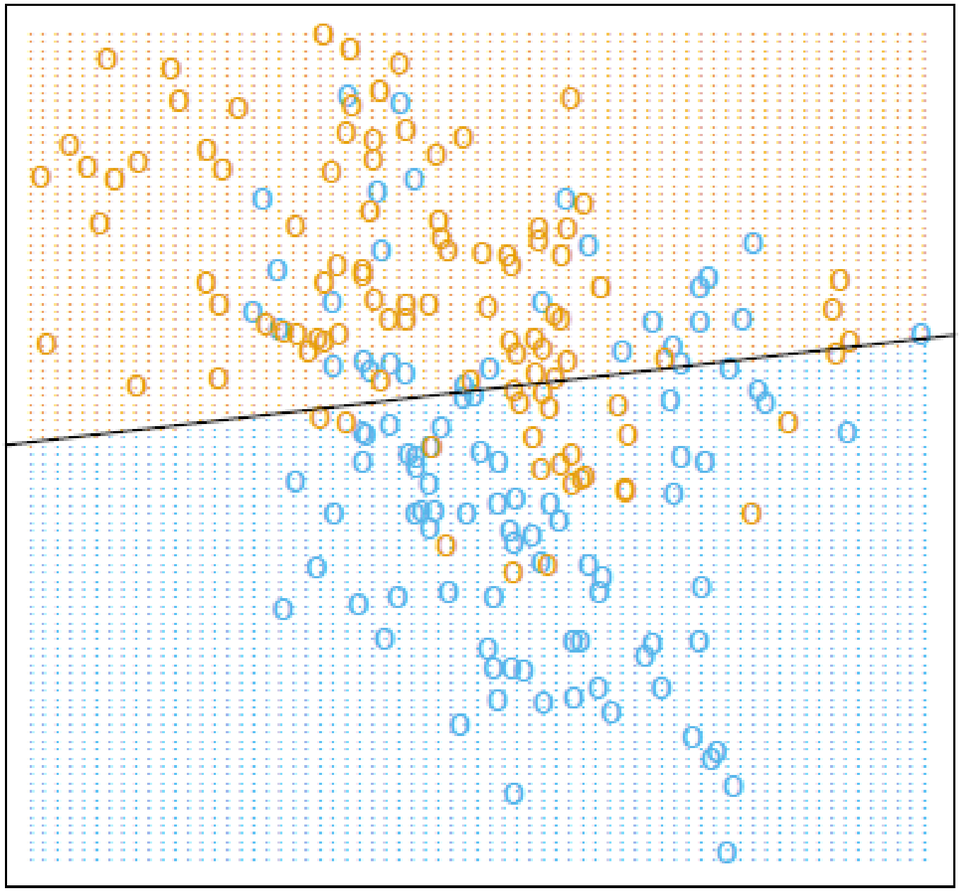


FIGURE 2.1. *A classification example in two dimensions. The classes are coded as a binary variable (*BLUE $= 0$, ORANGE $= 1$*), and then fit by linear regression. The line is the decision boundary defined by* $x^T\hat{\beta} = 0.5$. *The orange shaded region denotes that part of input space classified as* ORANGE*, while the blue region is classified as* BLUE.

K-NEAREST NEIGHBORS (KNN)

- Intuitive idea:
  - For a new input, look for the closest instances (neighbors) in the training set
  - The prediction is based on the outputs of the neighbors

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

K-NEAREST NEIGHBORS (KNN)

– kNN approximates the optimal function by using the points closest to x:

$$f(x) = E(Y|X = x)$$

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

GHENT UNIVERSITY

imec

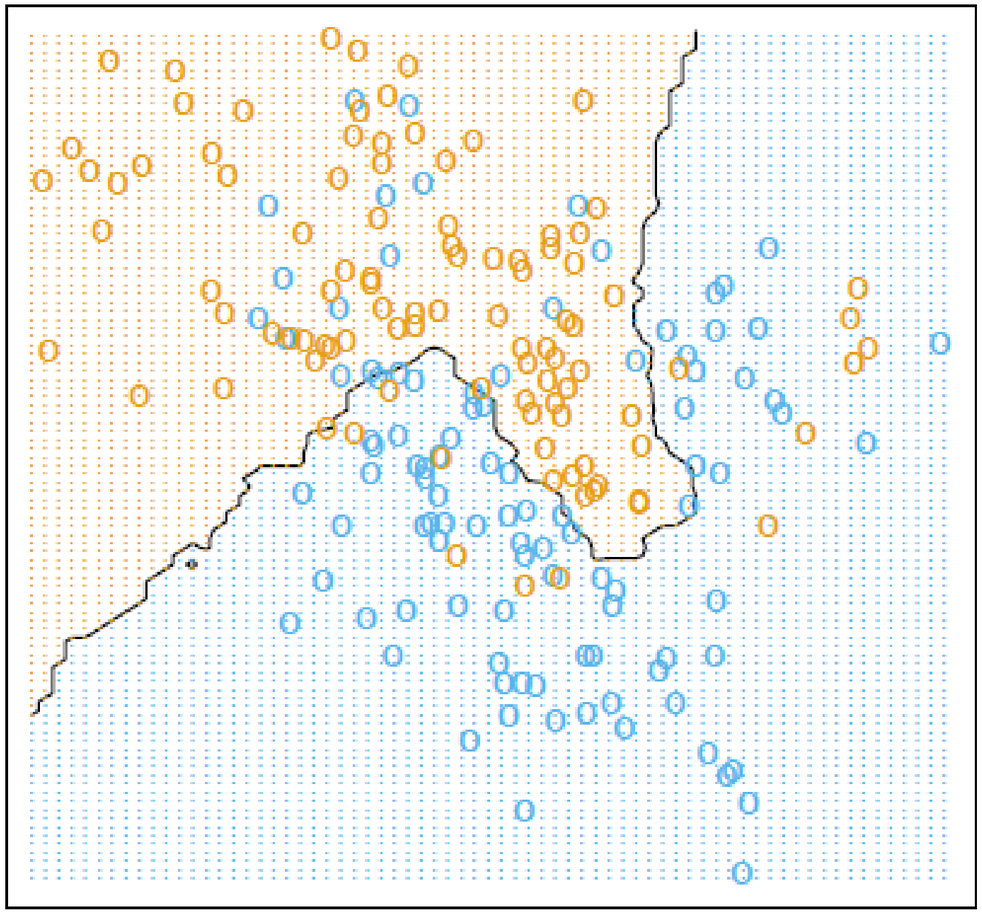


FIGURE 2.2. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$) *and then fit by* 15*-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the* 15*-nearest neighbors.*

1-Nearest Neighbor Classifier

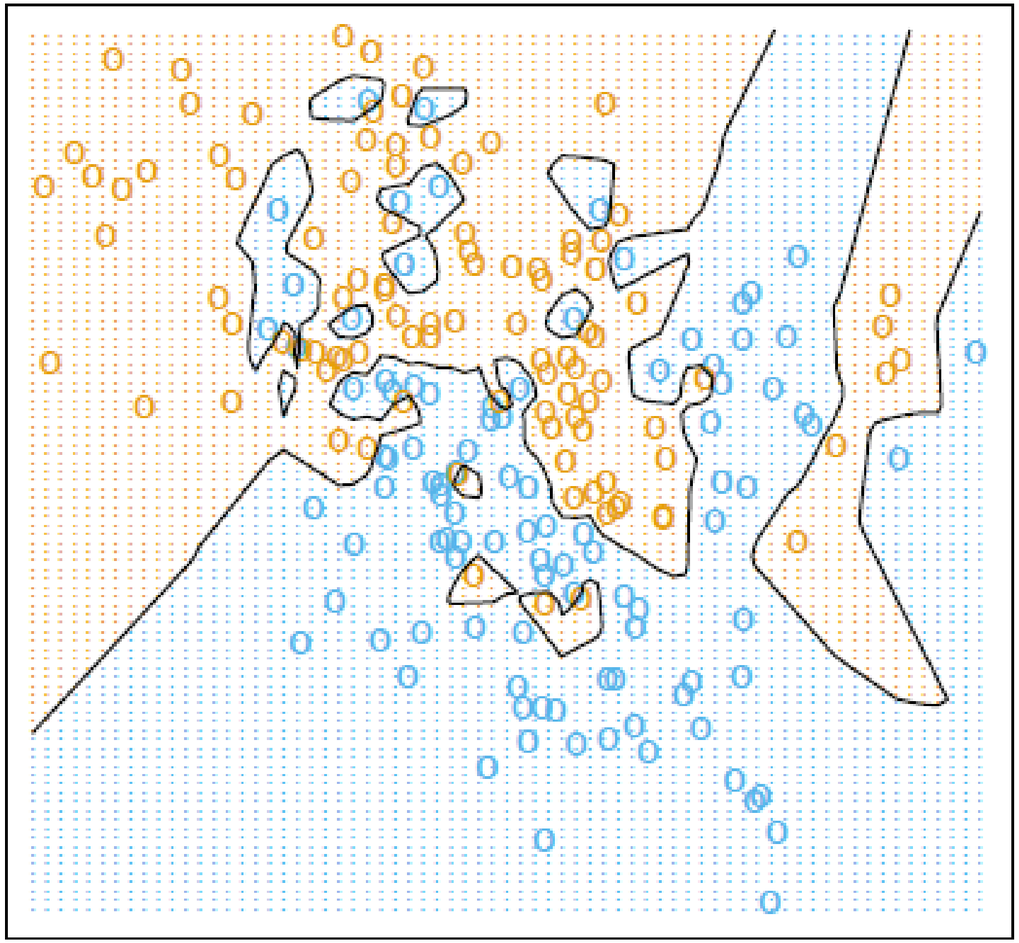


FIGURE 2.3. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE = 0, ORANGE = 1), *and then predicted by* 1*-nearest-neighbor classification.*

TWO EXTREME APPROACHES

Parametric:

linear model

Strong structural assumptions

Stable predictions

Can be inaccurate

Non-parametric:

k-nearest neighbors (kNN)

Mild structural assumptions

Predictions can be unstable

Often accurate

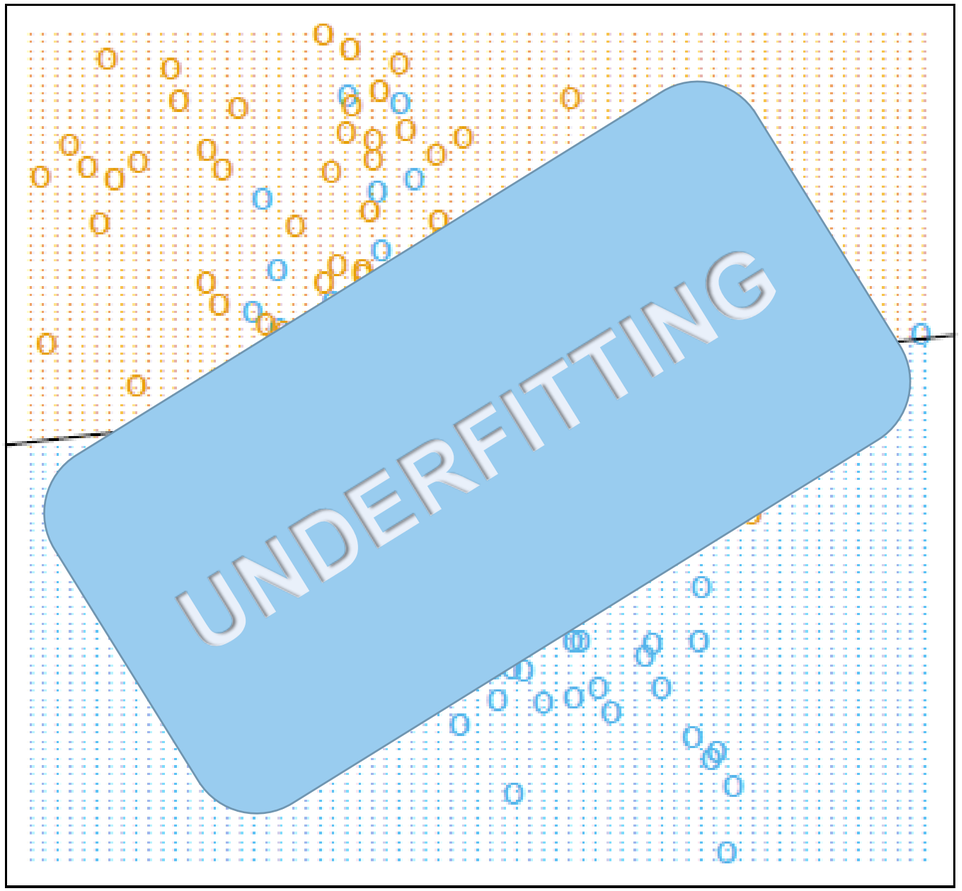
LEAST SQUARES VS KNN

- From the computational perspective:
  - Least squares is **trained**: an explicit (parametric) model is computed from the training set.
  - KNN is **lazy learning**: no training is done. The training set is explored for each test instance.

- Most methods are in between both extremes
- However, both least squares and kNN are used very often in practice

BIAS-VARIANCE

Variance



Bias

FIGURE 2.1. *A classification example in two dimensions. The classes are coded as a binary variable (BLUE = 0, ORANGE = 1), and then fit by linear regression. The line is the decision boundary defined by* $x^T\hat{\beta} = 0.5$. *The orange shaded region denotes that part of input space classified as ORANGE, while the blue region is classified as BLUE.*

1-Nearest Neighbor Classifier

Variance

Bias

FIGURE 2.3. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE = 0, ORANGE = 1), *and then predicted by* 1*-nearest-neighbor classification.*

Good fit

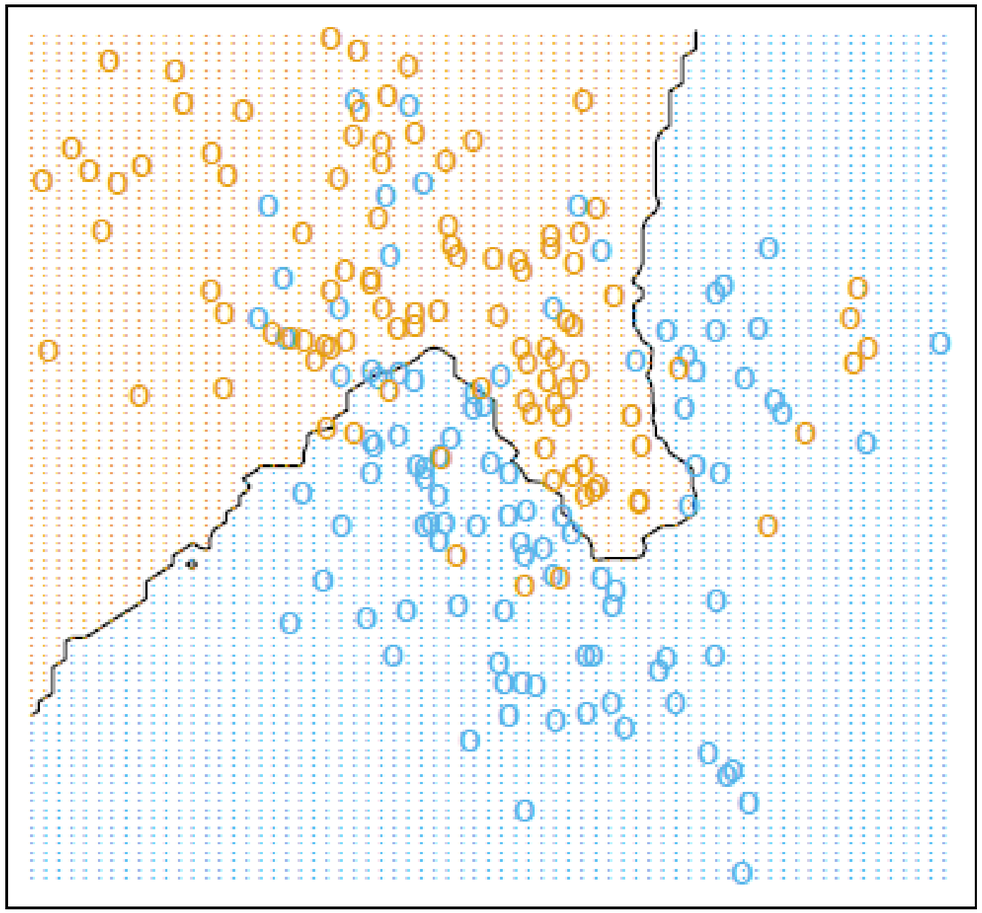


FIGURE 2.2. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$) *and then fit by* 15*-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the* 15*-nearest neighbors.*

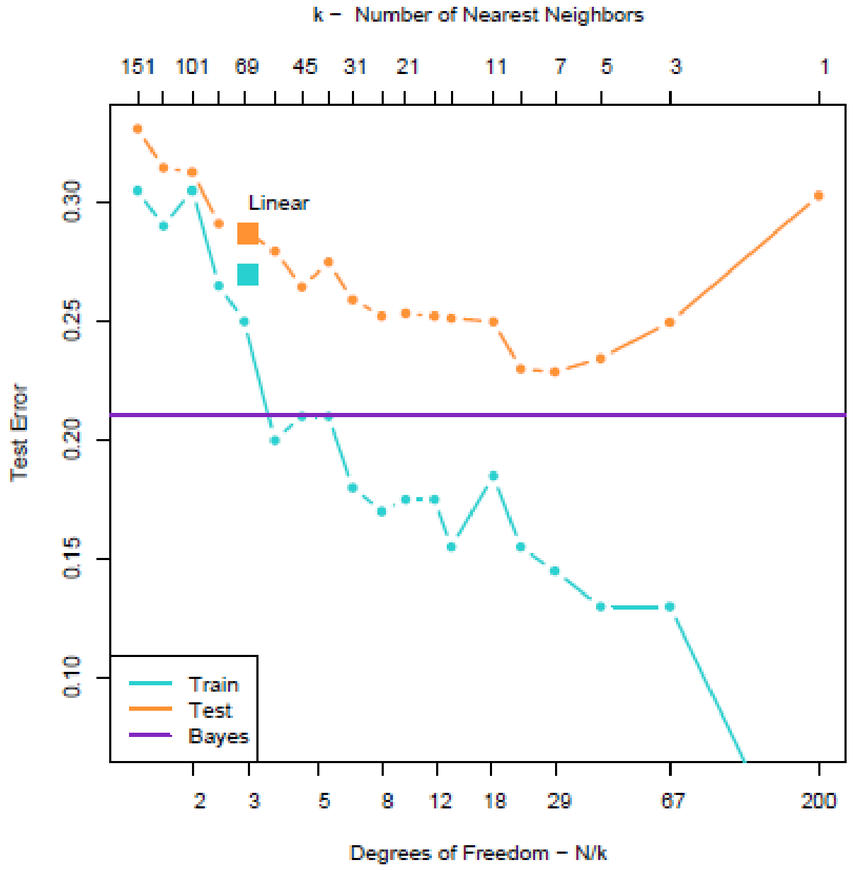


FIGURE 2.4. *Misclassification curves for the simulation example used in Figures 2.1, 2.2 and 2.3. A single training sample of size* 200 *was used, and a test sample of size* 10,000. *The orange curves are test and the blue are training error for k-nearest-neighbor classification. The results for linear regression are the bigger orange and blue squares at three degrees of freedom. The purple line is the optimal Bayes error rate.*

ɔe>

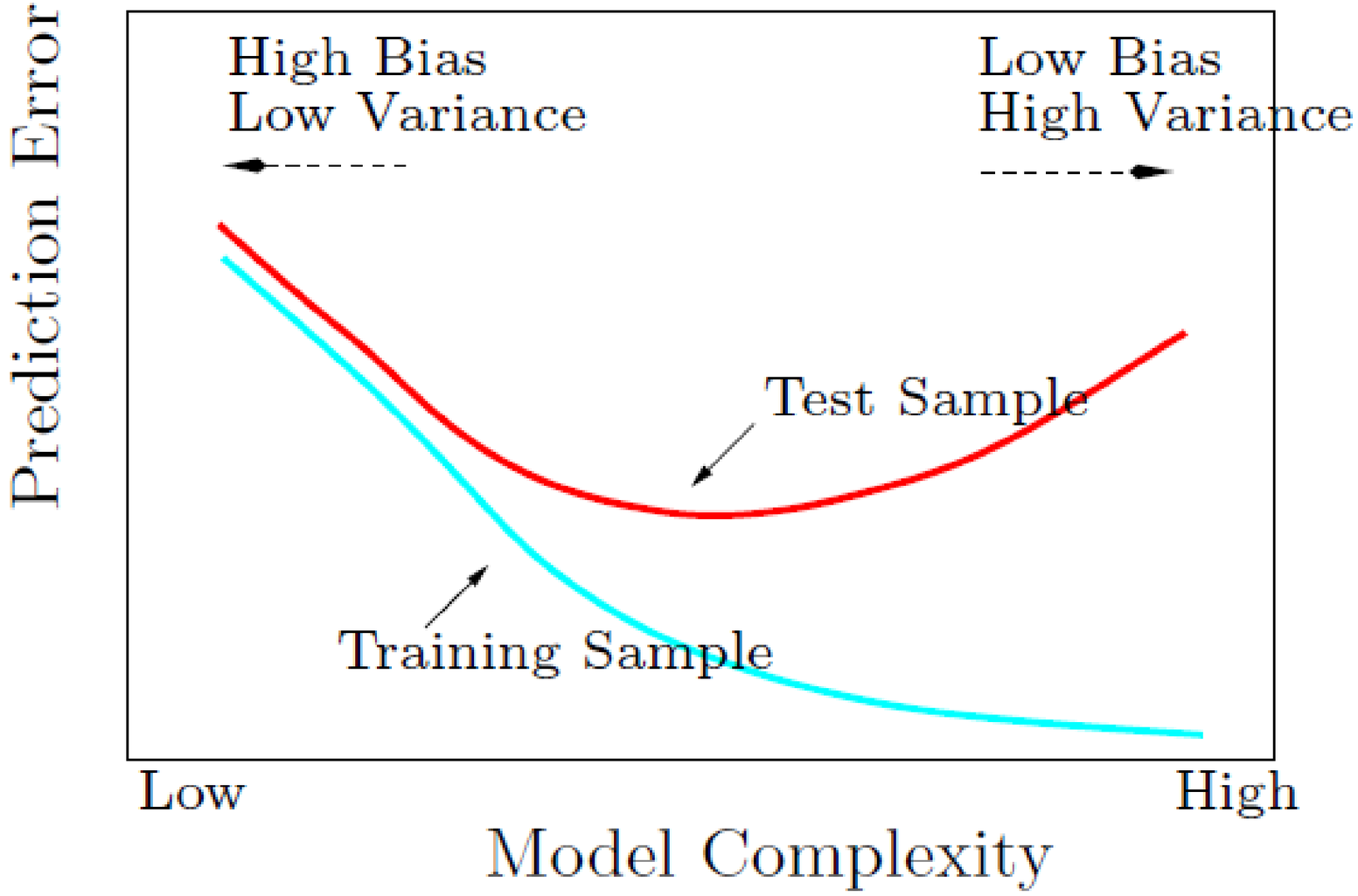


FIGURE 2.11. *Test and training error as a function of model complexity.*

MODEL ASSESSMENT: CROSS VALIDATION

SELF-VALIDATION

- Simplest approach:
  - Use all the data we have to train the model
  - Calculate the prediction error
    - Misclassification rate
    - Residual (or Mean) Square Error

Training set

SELF-VALIDATION

- Serious flaws!!
  - The error is (sometimes extremely) under-estimated

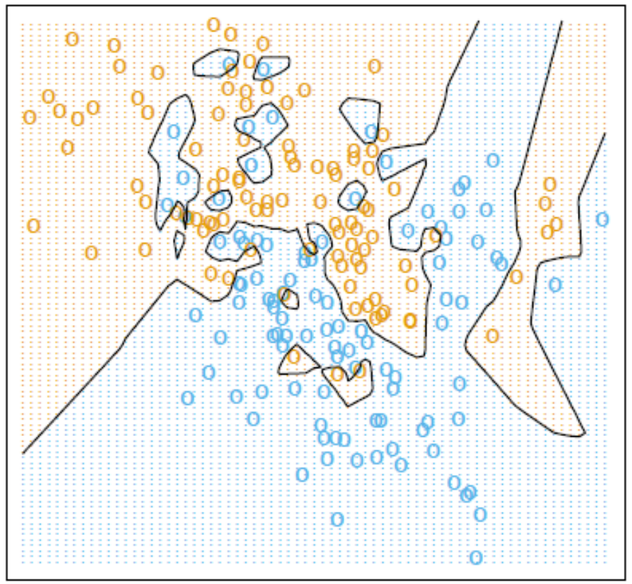


FIGURE 2.3. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable (BLUE = 0, ORANGE = 1), and then predicted by 1-nearest-neighbor classification.*

RESAMPLING

- We want the model to perform well on previously unseen data
- By definition, we cannot evaluate it on previously unseen data

- We need to split and resample our data to simulate previously unseen data

TRAINING-TEST

Use a subset of the test to train, and the rest to test

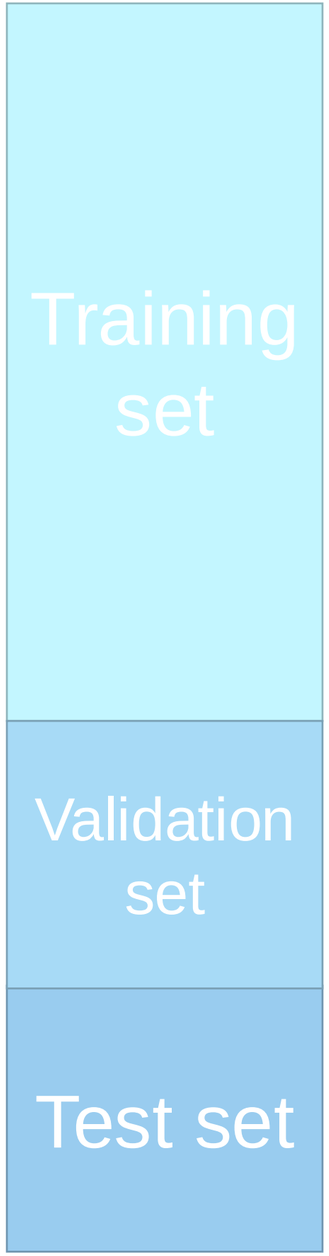
- **Pros**:
  - Fast (only 1 training-test run)
- **Cons**:
  - We get a single result out of a random partitioning
  - Test error can be variable
  - The size of the training set is severely reduced: overestimation of the error
  - If the test error is used to manually tune the parameters, the model can overfit

The test set should be hidden under a rock!

TRAINING-VALIDATION-TEST

The training set is further split into training and validation

- **Pros**:
  - Fast (only 1 training-test run)
  - In case of overfitting, we can still detect it using the test set
- **Cons**:
  - We get a single result out of a random partitioning
  - Test/validation error can be variable
  - The size of the training set is severely reduced: overestimation of the error



TRAINING-VALIDATION-TEST

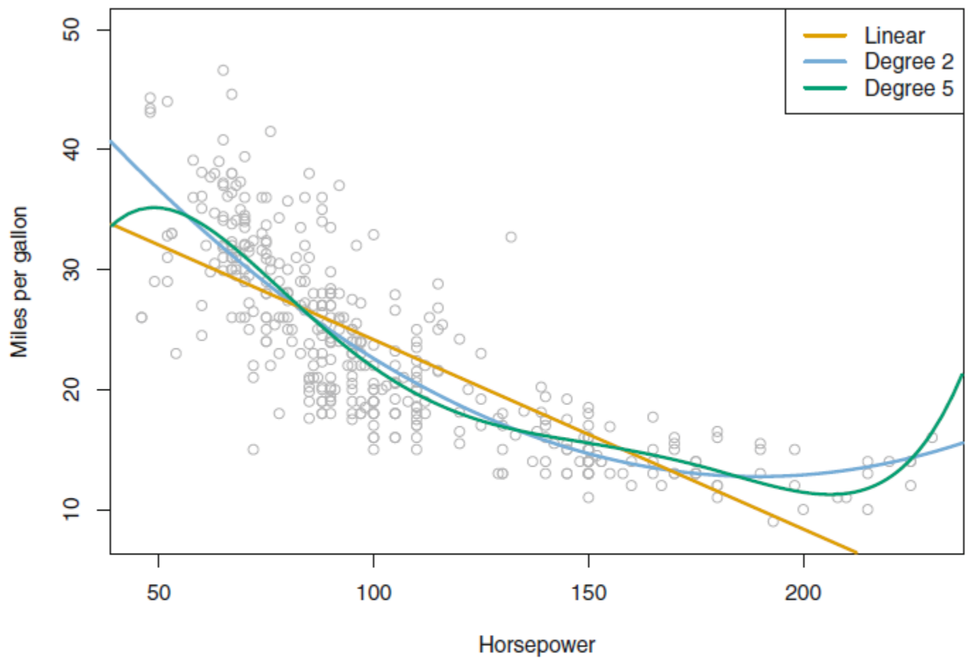
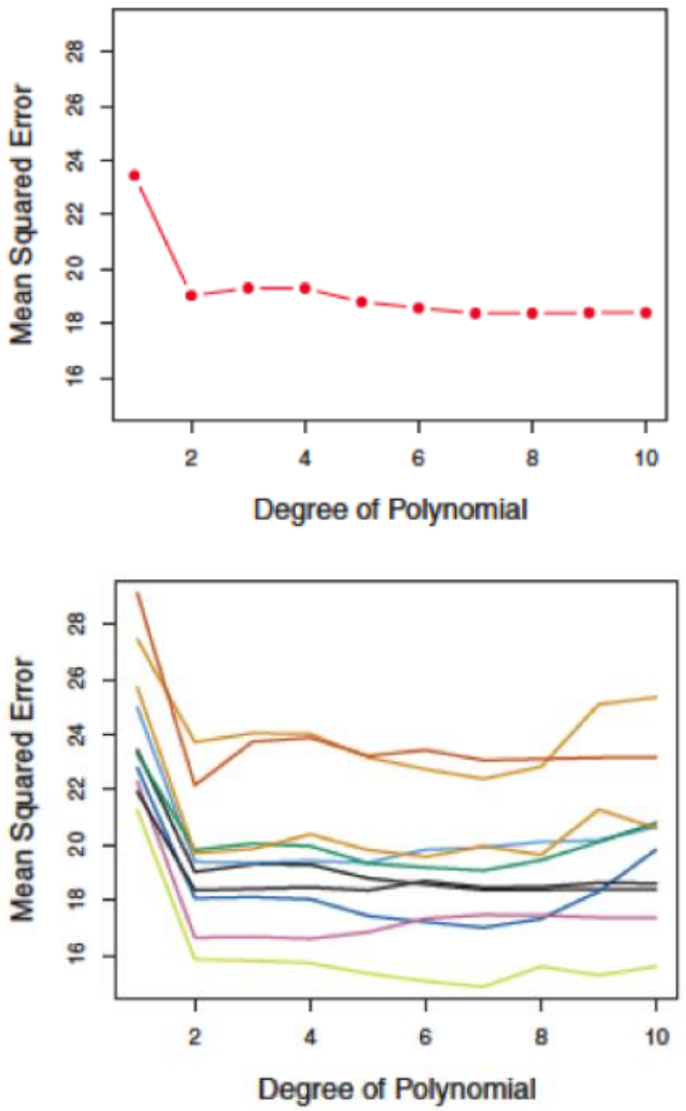


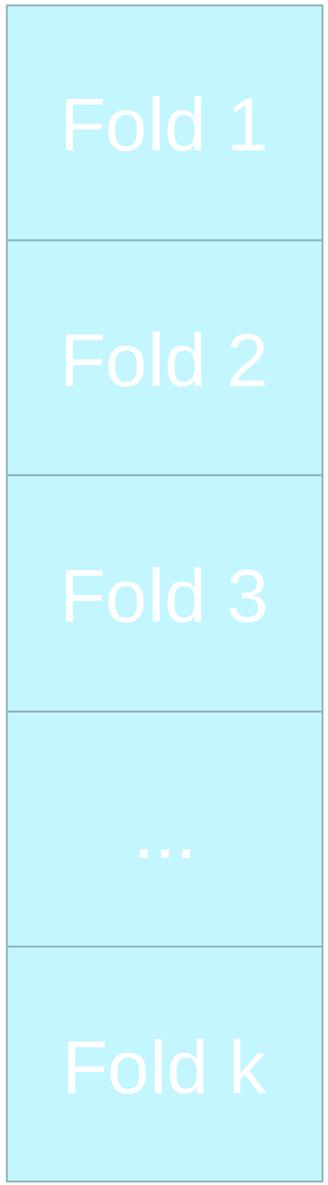
FIGURE 3.8. *The* Auto *data set. For a number of cars,* mpg *and* horsepower *a shown. The linear regression fit is shown in orange. The linear regression fit for model that includes* horsepower^2 *is shown as a blue curve. The linear regressi fit for a model that includes all polynomials of* horsepower *up to fifth-degree shown in green.*



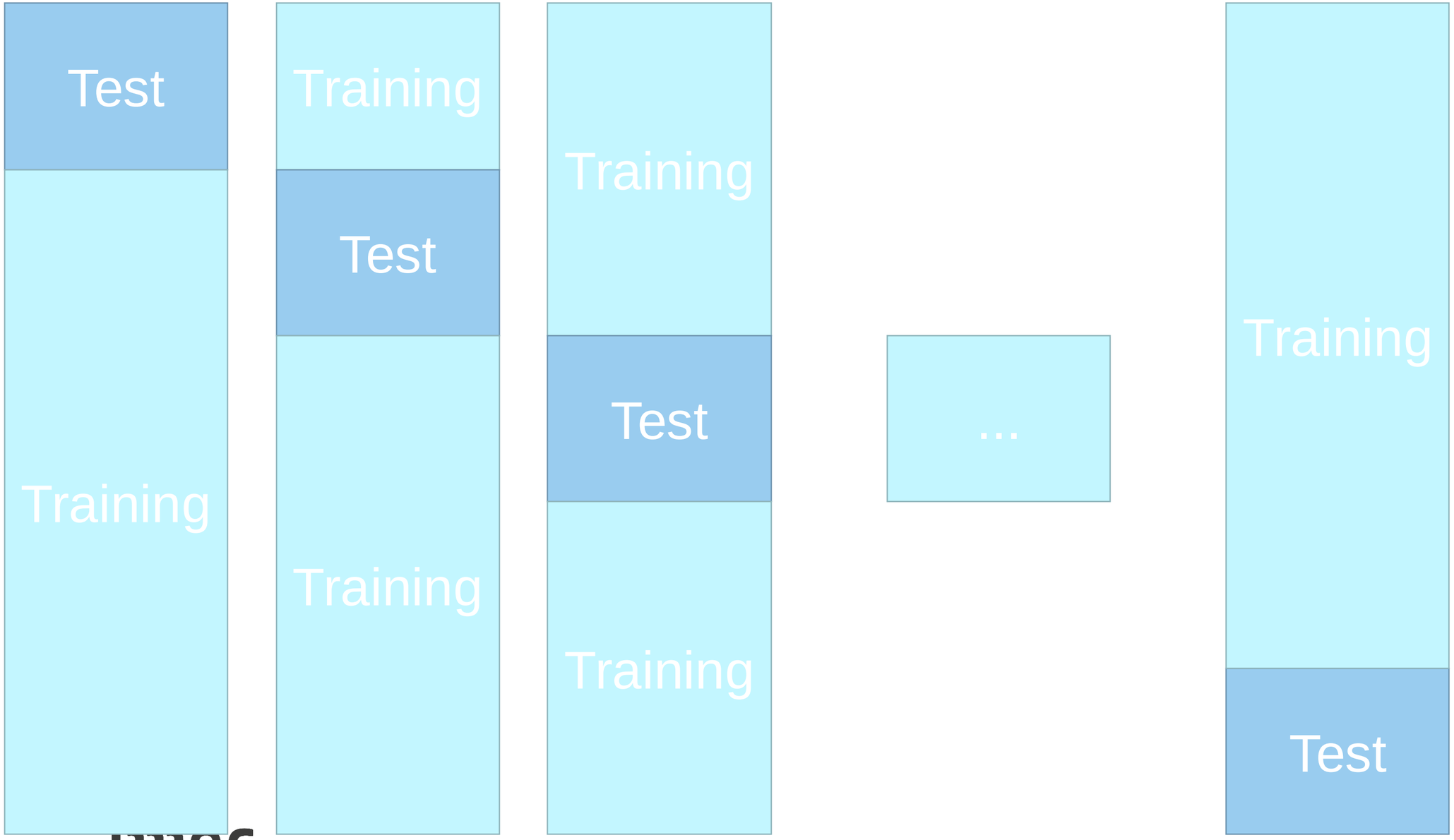
CROSS-VALIDATION

- Train *k* models using all folds except one
- Evaluate each of them on the remaining fold

- Calculate the average error across all folds



CROSS-VALIDATION



CROSS-VALIDATION

- **Pros**:
  - Multiple partitions, robust
  - The reduction of the training set size is set by *k*
  - The entire training set is also used as test
- **Cons**:
  - Computationally heavy

CROSS-VALIDATION

- Extreme case: what happens if k = N?

- **Leave-One-Out-Cross-Validation**

LEAVE ONE OUT CROSS VALIDATION

- **Pros**:
  - Very low bias
  - Deterministic
  - Very efficient for linear models
- **Cons**:
  - Computationally very heavy
  - Some variance

BOOTSTRAP

BOOTSTRAP

- How to simulate a training set using a training set?
- Sample with replacement!

BOOTSTRAP

Bootstrap :

- noun : a loop of leather sewn at the top rear, or sometimes on each side, of a boot to facilitate pulling it on

- adjective : relying entirely on one's efforts and resources

to pull oneself up by one's own bootstraps

BOOTSTRAP

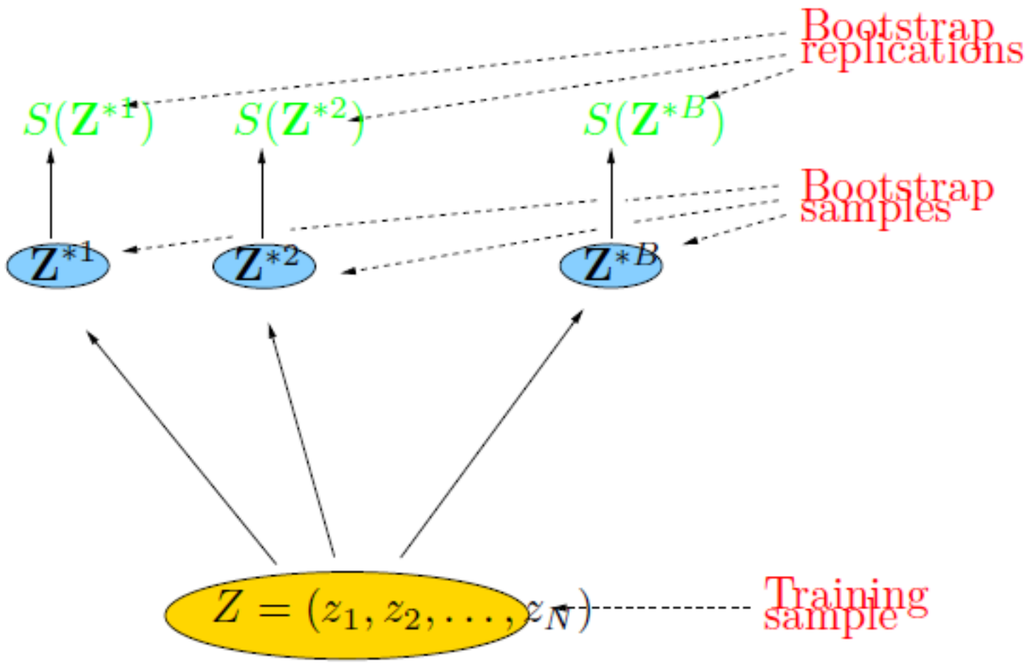


FIGURE 7.12. *Schematic of the bootstrap process. We wish to assess the statistical accuracy of a quantity $S(\mathbf{Z})$ computed from our dataset. B training sets $\mathbf{Z}^{*b}$, $b = 1, \ldots, B$ each of size N are drawn with replacement from the original dataset. The quantity of interest $S(\mathbf{Z})$ is computed from each bootstrap training set, and the values $S(\mathbf{Z}^{*1}), \ldots, S(\mathbf{Z}^{*B})$ are used to assess the statistical accuracy of $S(\mathbf{Z})$.*

GHENT UNIVERSITY

imec

BOOTSTRAP

Consider the simple sample $\mathcal{X} = (2, 12, 20, 26, 78)$ of size $n = 5$.
Re-sampling with replacement leads to a total of B samples

$$\begin{cases} \mathcal{X}^{*(1)} = (12, 20, 78, 26, 2) \\ \mathcal{X}^{*(2)} = (26, 78, 12, 12, 2) \\ \mathcal{X}^{*(3)} = (20, 20, 20, 20, 20) \\ \vdots \\ \mathcal{X}^{*(B)} = (26, 2, 2, 78, 20) \end{cases}$$

These bootstrap samples were drawn from the "population"
$\mathcal{X} = (2, 12, 20, 26, 78)$.

EXAMPLE: MINIMIZING INVESTMENT RISK

- We want to invest a fixed amount of money in 2 financial assets that yield returns X and Y.
  - Invest a fraction α in X
  - Invest a fraction $(1-\alpha)$ in Y

- GOAL: minimizing the risk (i.e. variance of the returns):

$$Var(\alpha X + (1-\alpha)Y)$$

- The α that minimizes this expression is:

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

We want to know the **variability of $\hat{\alpha}$** the estimate of α.

EXAMPLE: MINIMIZING INVESTMENT RISK

We produce B "new" samples $Z^{*(1)}, \ldots, Z^{*(B)}$ of size n from the original sample $Z = (X_1, Y_1), \ldots, (X_n, Y_n)$ by randomly drawing observations from Z with replacement.

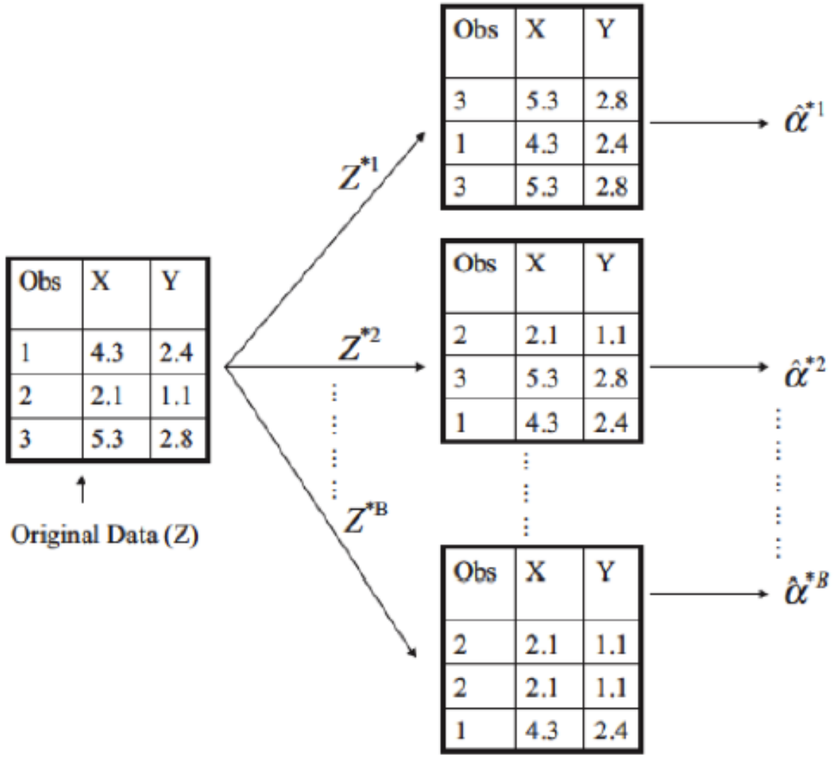
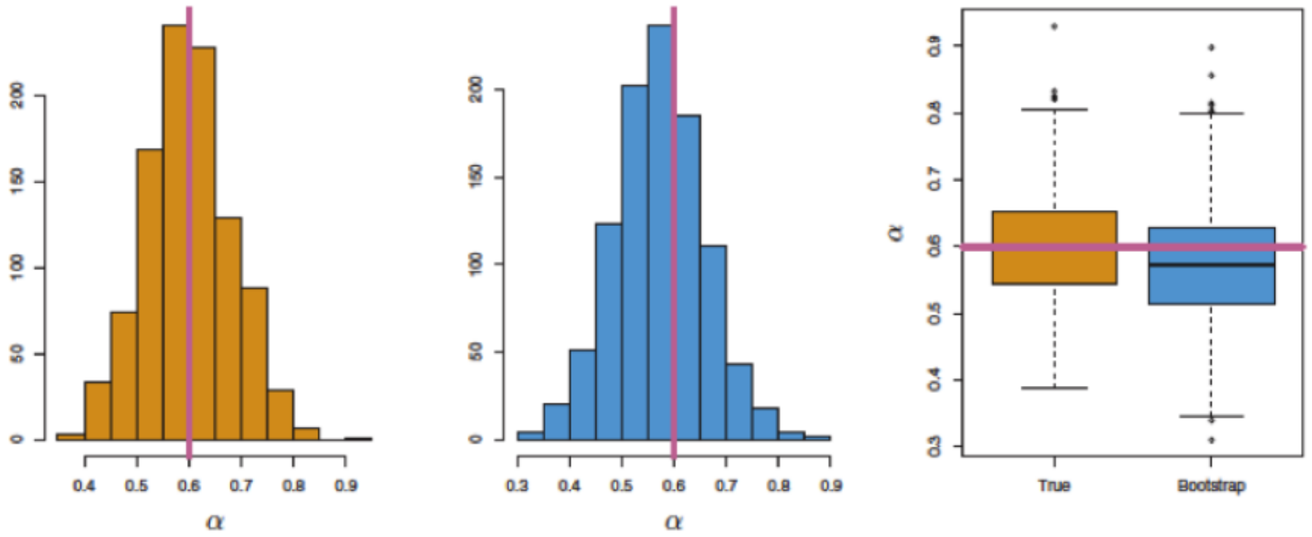


FIGURE 5.11. *A graphical illustration of the bootstrap approach on a small sample containing $n = 3$ observations. Each bootstrap data set contains n observations, sampled with replacement from the original data set. Each bootstrap data set is used to obtain an estimate of α.*

EXAMPLE: MINIMIZING INVESTMENT RISK

This yields B estimates $\hat{\alpha}^{*(1)}, \ldots, \hat{\alpha}^{*(B)}$ from which we calculate $\bar{\alpha}^* = \frac{1}{B}\sum_{r=1}^{B} \hat{\alpha}^{*(r)}$ and the targeted quantity

$$\sqrt{\frac{1}{B-1}\sum_{r=1}^{B}(\hat{\alpha}^{*(r)} - \bar{\alpha}^*)^2} = 0.087.$$



BOOTSTRAP

- **Pros**:
  - Allows to do statistical inference in the absence of asymptotic results
    - Too complicated, unknown, or small sample size
  - Especially suited to standard errors and confidence intervals
  - Easy to implement
  - Sometimes very accurate
  - Very flexible
- **Cons**:
  - Can be computationally heavy
  - Will not always work
  - Depends on the initial sample

- In general, use B as large as possible!

BAGGING

- Using bootstrap within more complex algorithms:
  - Bootstrap training set
  - Train a model on each sampling
  - Aggregate all models (ensemble)

- General idea about ensembles:
  - Turning many weak estimators into a single strong estimator

- For instance, used in Random Forest

HYPERPARAMETER OPTIMIZATION

PARAMETER OPTIMIZATION

- So far, we have talked about **parameter optimization**:
  - Our model contains trainable parameters
  - We define a loss function
  - An optimization algorithm searches the parameters that minimize the loss:
    - Analytic solutions
    - Newton-Raphson
    - (Stochastic) gradient descent
    - ...

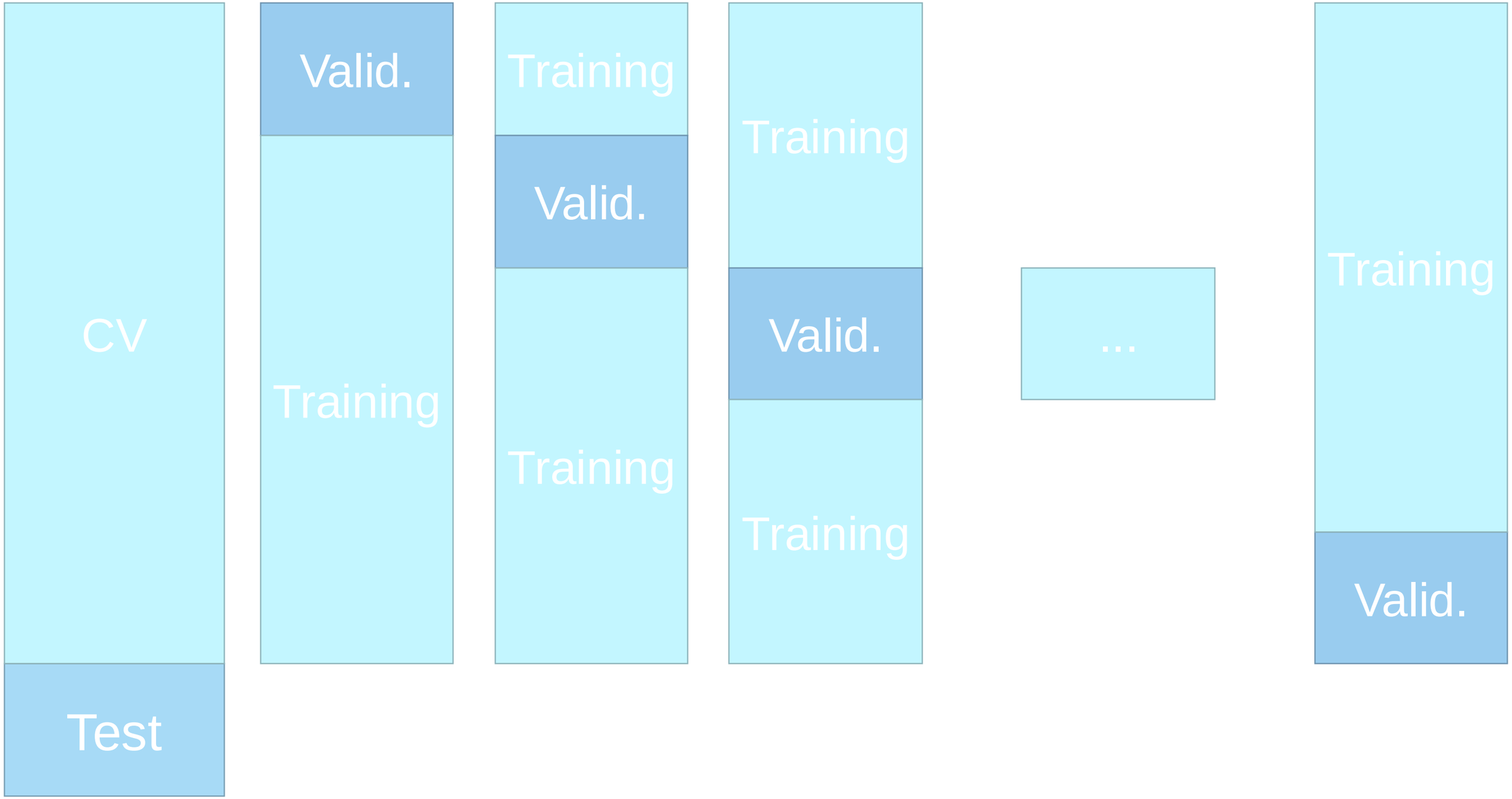
HYPERPARAMETER OPTIMIZATION

- Most models also have hyperparameters:
  - Fixed before training the model
  - Examples:
    - Regularization constant
    - Neural network architecture
    - Degree of polinomial
    - ...

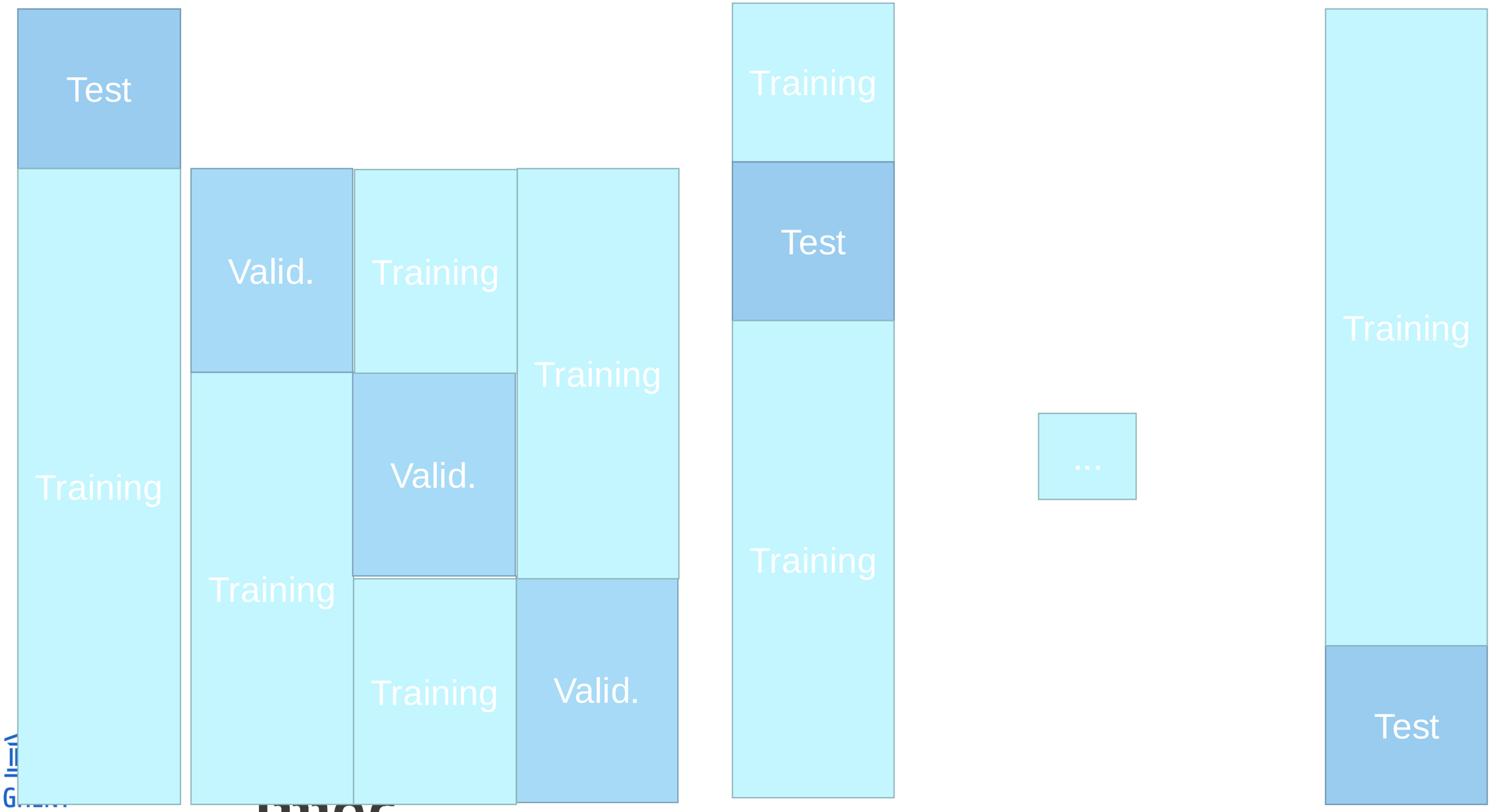
COMBINING APPROACHES

- Sometimes, we may want to combine several of these approaches (e.g. hyperparameter optimization)

TEST SET + CROSS VALIDATION



NESTED CROSS VALIDATION



CLASSIFICATION AND REGRESSION

- Classification can often be treated as a regression problem
  - Binary classification

$$g \in \{1,2\} \quad \Rightarrow \quad y \in \{0,1\}$$

Predict 2 if ŷ>0.5
Predict 1 otherwise

Or even better: predict the probability of each class (logistic regression)

  - Multi-class classification
    - Can be decomposed into K binary classification problems

GHENT UNIVERSITY

imec

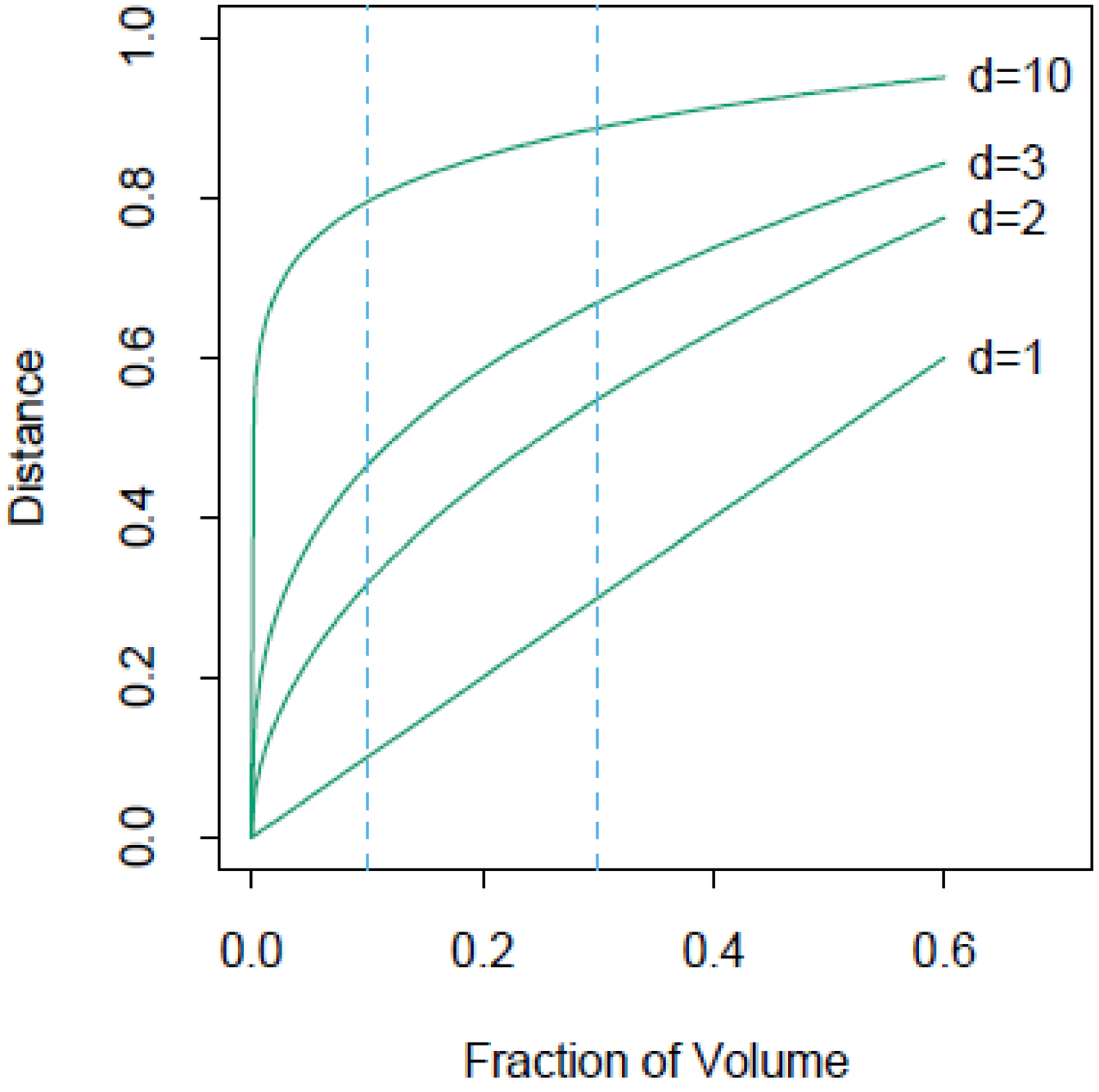
CURSE OF DIMENSIONALITY

CURSE OF DIMENSIONALITY

- What happens if *p* grows bigger?

- Imagine the input space as a *p*-dimensional unit hypercube
  - Given a point, we want a hypercubical neighborhood that captures a fraction *r* of the whole space
  - What is the edge length of that neighborhood?

$$r^{\frac{1}{p}}$$


d=10
d=3
d=2
d=1
Distance
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
Fraction of Volume

CURSE OF DIMENSIONALITY

- As *p* grows, the training data becomes (exponentially) more sparse
- Therefore, the distance to the nearest neighbor increases (exponentially)
- In parametric methods, the number of parameters increases with *p*
  - Higher variance

Daniel Peralta <daniel.peralta@ugent.be>

REGRESIÓN LINEAL

LINEAR MODELS

- Assumption: the relation between inputs and outputs is linear

$$X^T = (X_1, X_2, \dots, X_p)$$

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

Intercept

Coefficients

- The bias can be modeled as a coefficient if we include a constant variable 1 in X, which makes:

$$\hat{Y} = X^T \hat{\beta}$$

GHENT UNIVERSITY

imec

EXTENSIONS TO SIMPLE LINEAR REGRESSION

- Feature engineering: creating new input variables by combining existing ones
  - $X_3 = X_1^2$
  - $X_3 = X_1 X_2$
  - $X_3 = \log(X_1)$
  - ...

PROBLEMS OF LEAST SQUARES ESTIMATES

- With many variables, we get:
  - Many coefficients
  - Potentially large coefficients

- This causes:
  - Lower prediction accuracy (low bias, large variance)
  - Less interpretability (many coefficients)

- Therefore, we might want to sacrifice some coefficients, which we can do in two ways:
  - Subset selection
  - Shrinkage methods (regularization)

REGULARIZATION

- It is widely used in machine learning

- It consists of changing loss function:

$$J_\lambda(\theta) = J(\theta) + \lambda\Omega(\theta)$$

- Here, Ω is a penalization function, and λ is the regularization parameter (sometimes called “weight decay”)

SHRINKAGE METHODS

- Shrinking the coefficients to bring as many of them as possible to zero

- Methods:
  - Ridge regression
  - LASSO
  - Elastic net

RIDGE REGRESSION

- Shrinks the coefficients by imposing a penalty on their size:

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \left|\beta_j\right|^2 \right\}$$

- Note that β_0 is not penalized
- Important to normalize first!
- The impact of correlated variables is reduced

RIDGE REGRESSION

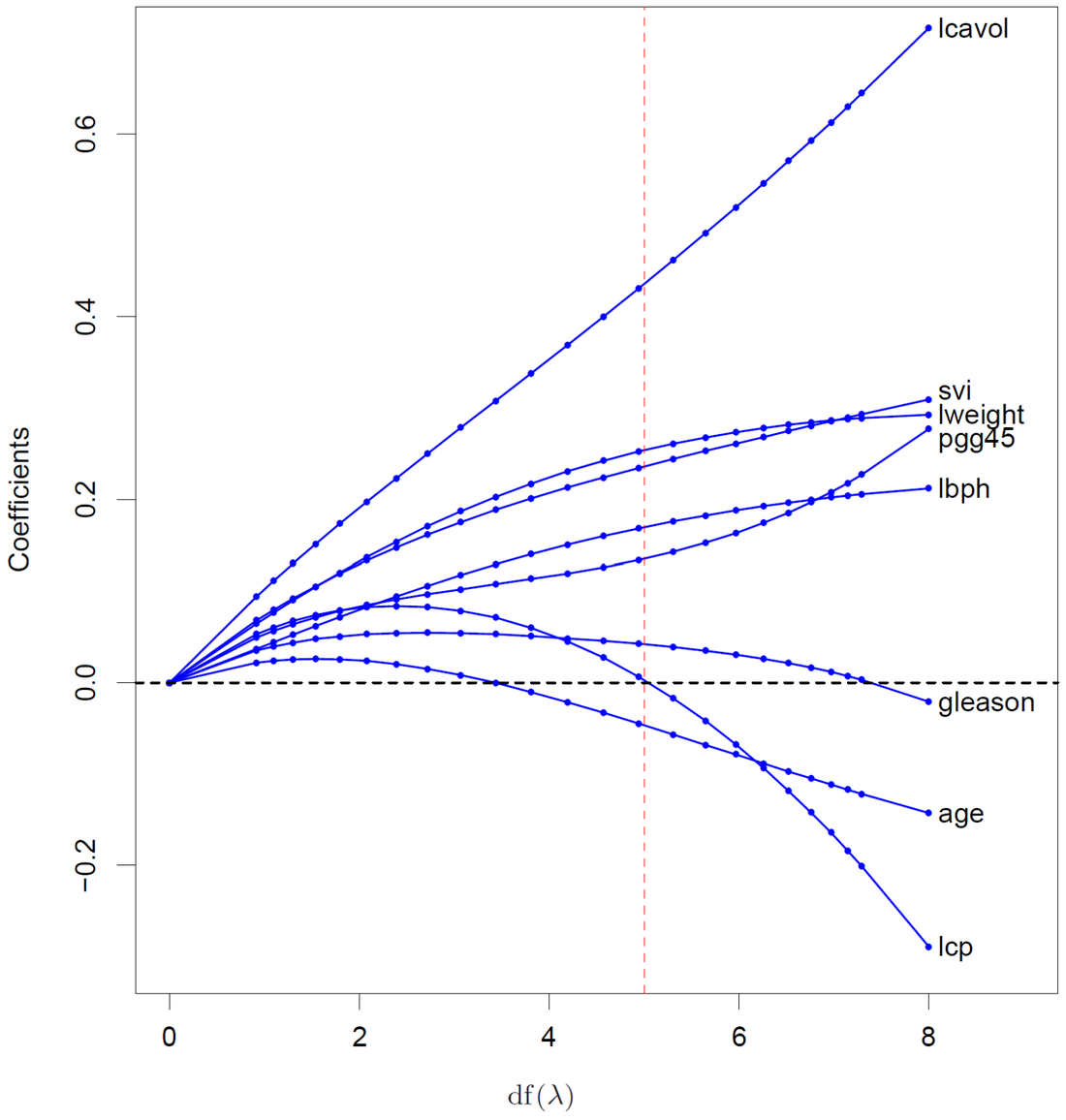
- Assuming that the input is centered, we can write:

$$RSS(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta^T\beta$$

- The minimum loss is obtained by:

$$\hat{\beta}^{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

EXAMPLE


lcavol
svi
lweight
pgg45
lbph
gleason
age
lcp
Coefficients
0.6
0.4
0.2
0.0
−0.2
0
2
4
6
8
df(λ)

LASSO

– Very similar to ridge regression, with one difference:

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^2 \right\}$$

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

– Ridge regression uses L_2 regularization
– Lasso uses L_1 regularization

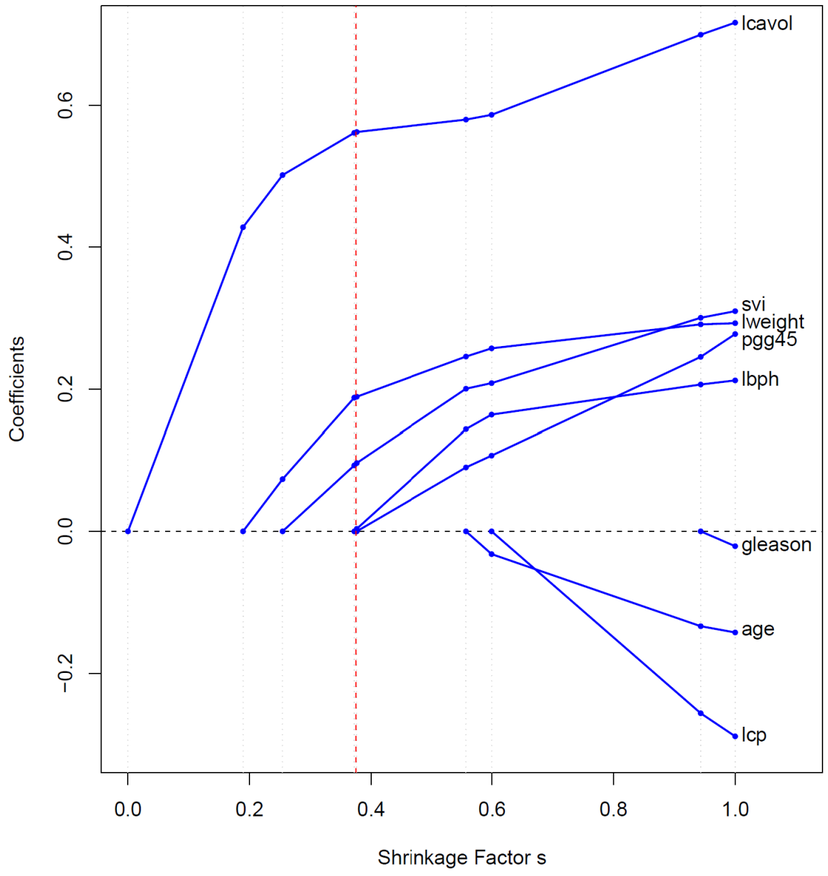
GHENT UNIVERSITY

imec

LASSO

- The difference is larger than it seems:
  - Now the solutions are non linear w.r.t. y_i
  - There is no closed form expression
    - Computing the solution is a quadratic programming problem
  - If λ is sufficiently large, some of the β_i can become exactly zero

EXAMPLE


lcavol
svi
lweight
pgg45
lbph
gleason
age
lcp
Coefficients
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Shrinkage Factor s

GENERALIZING RIDGE REGRESSION AND LASSO

$$\tilde{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \left|\beta_j\right|^q \right\}$$

- By choosing $q \in (1,2)$ we can have something between lasso and ridge regression

ELASTIC NET

- Similar idea to the L_q penalty, but computationally more efficient:
- Elastic net penalty:

$$\lambda \sum_{j=1}^{p} \left(\alpha \beta_j^2 + (1-\alpha) \left| \beta_j \right| \right)$$

DECISION TREES

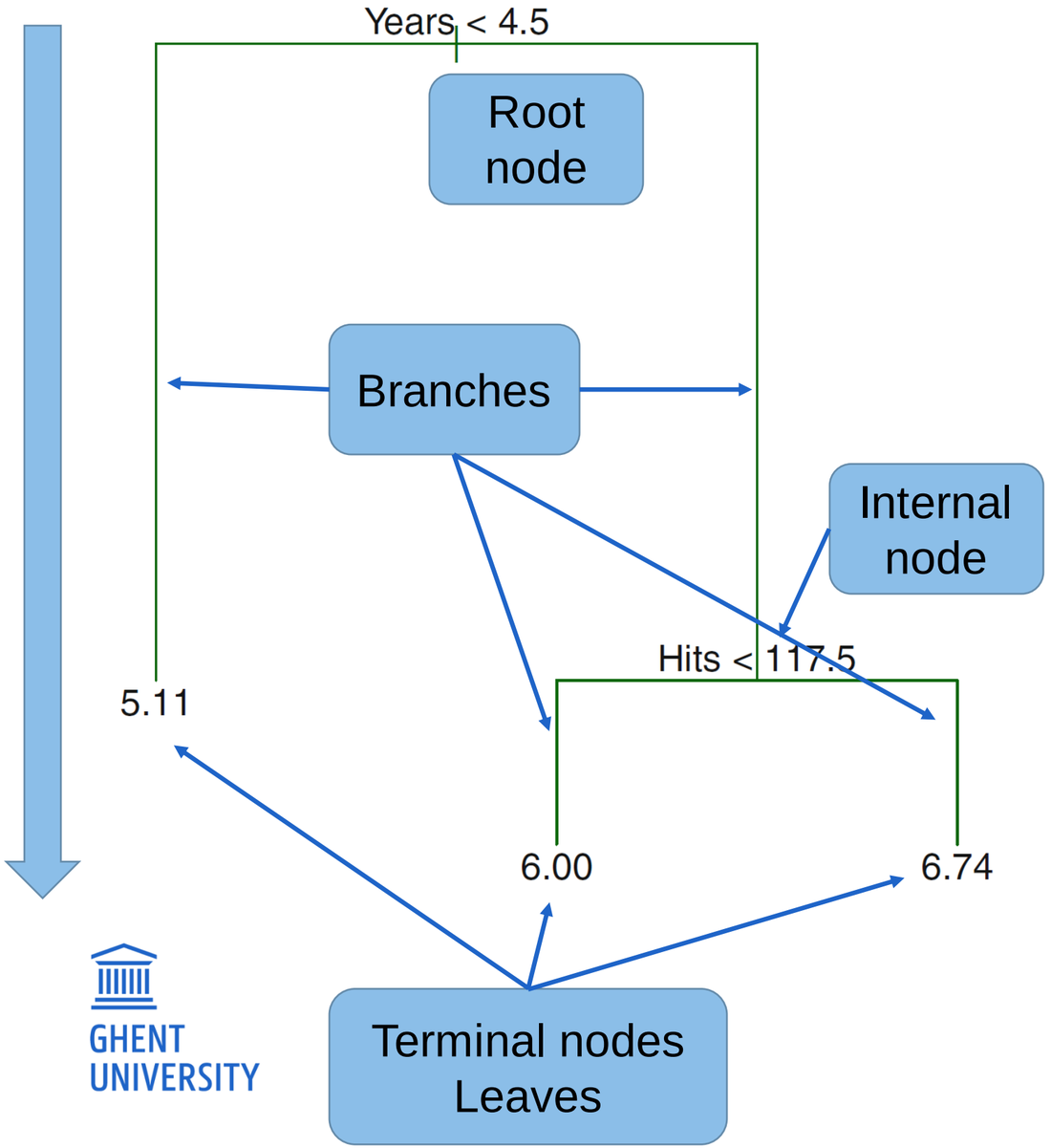
EXAMPLE

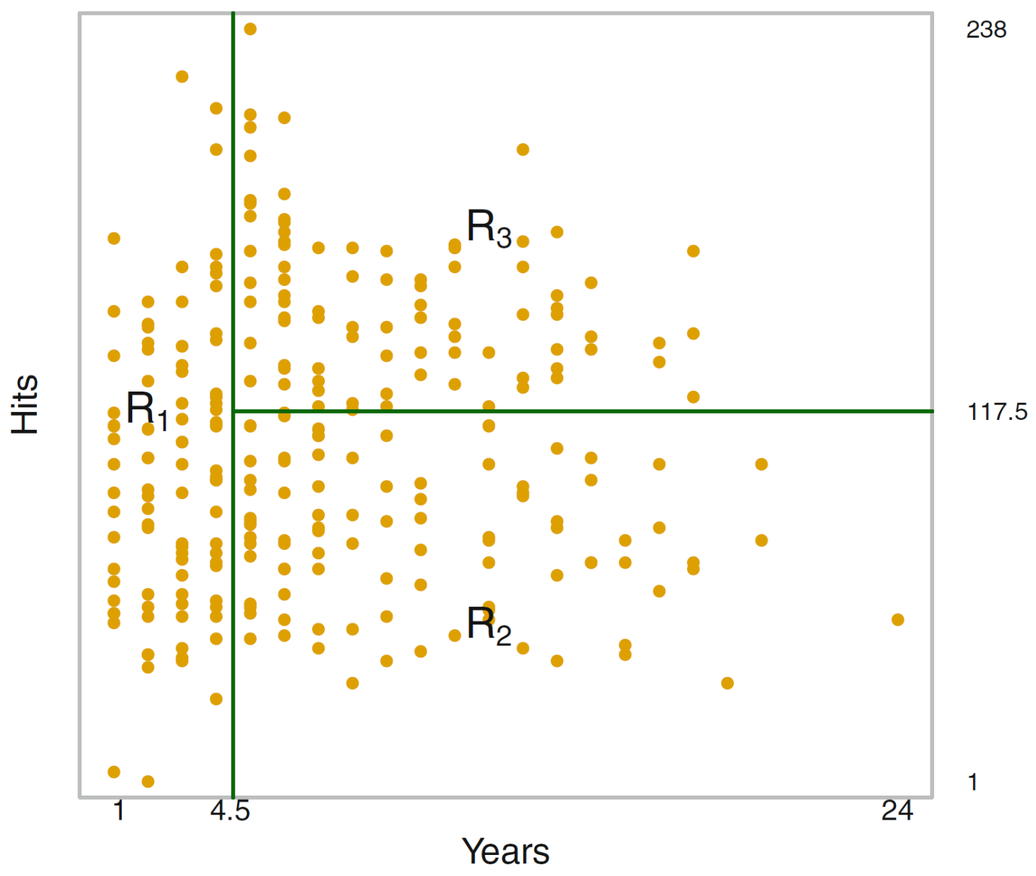
- We want to predict a baseball player's salary based on:
  - Number of years playing in the major leagues
  - Number of hits in the previous year

- The salary is measured in thousands of dollars and log-transformed

- We fit a regression tree

SIMPLE REGRESSION TREE





GHENT UNIVERSITY

TREES

- Very simple (sometimes too much)
- Very interpretable!
- Directly suited to continuous or categorical features
- Missing values are easy to deal with:
  - Assign most frequent value in the learning sample or the tree
  - Assign probabilities to each branch

TRAINING (GROWING) A TREE

– Goal: find rectangular boxes $R_1, \ldots, R_J$ that minimize:

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j} \right)^2$$

– $\hat{y}_{R_j}$ is the mean response in R_J

TRAINING (GROWING) A TREE

- Impossible to consider all possible partitions!
- Solution: Recursive binary splitting
  - Top-down approach
  - Greedy

1. For all pairs of feature j and value s
   1. Consider the half-planes:

$$R_1(j,s) = \{X: X_j < s\} \text{ and } R_2(j,s) = \{X: X_j \geq s\}$$

   2. Seek j, s that minimize:

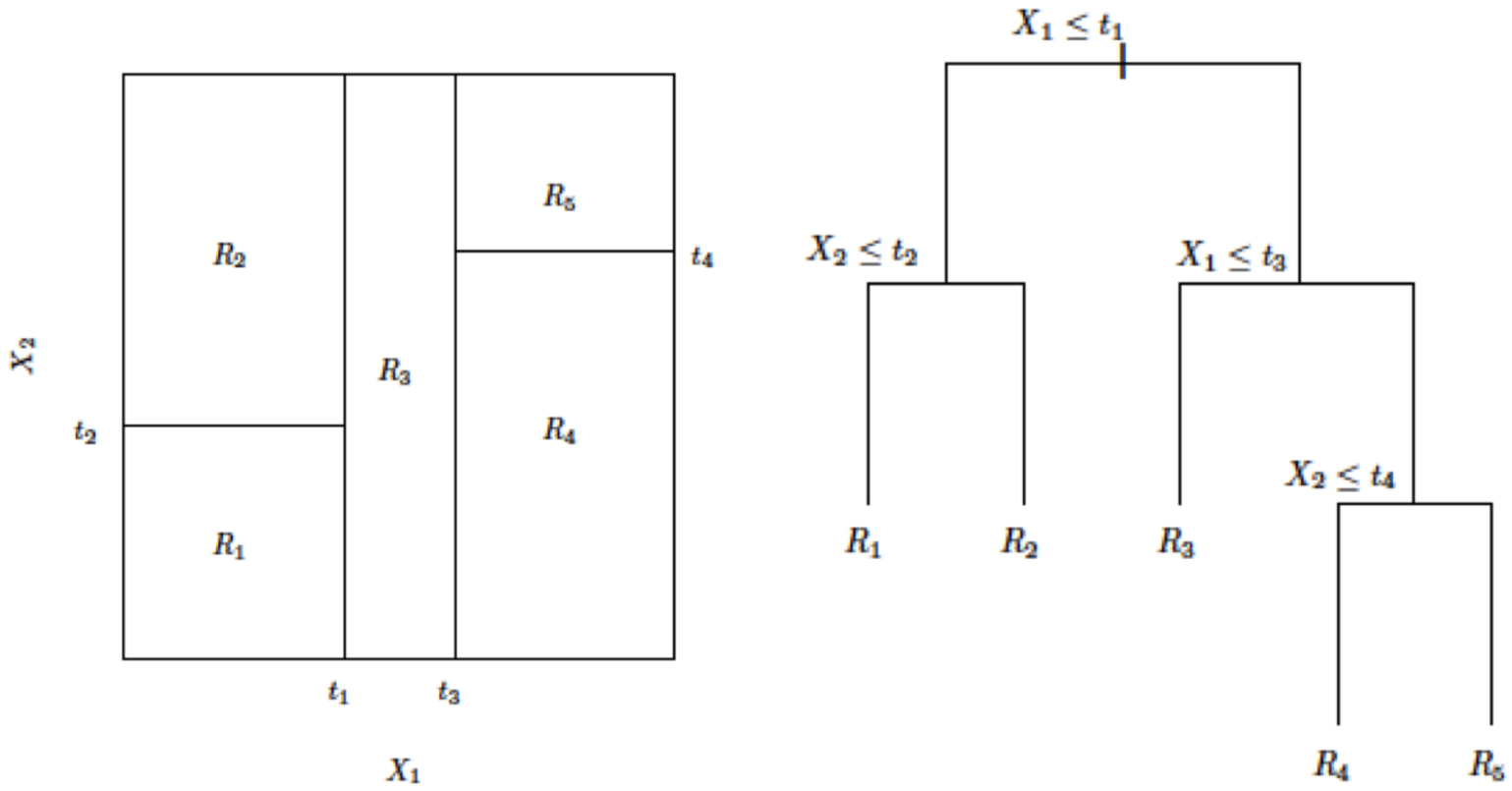
$$\sum_{i:\, X_i \in R_1(j,s)} \left(y_i - \hat{y}_{R_1}\right)^2 + \sum_{i:\, X_i \in R_2(j,s)} \left(y_i - \hat{y}_{R_2}\right)^2$$

2. Split into the two regions
3. Repeat the process for each region, until some stopping criterion:
   - Max. depth
   - Min. leaf size
   - …

GHENT UNIVERSITY

imec

TRAINING (GROWING) A TREE



How do we control the bias-variance trade-off?

BIAS-VARIANCE TRADEOFF

- A deeper tree leads to overfitting!

- How to know when to stop splitting?
  - Stopping criterion built on RSS decrease
    - Too greedy! Maybe a seemingly worthless split might be followed by a good split
  - Pruning: grow a very large tree T_0, then prune it

COST COMPLEXITY PRUNING

- Consider a large tree T_0, and a parameter $\alpha \geq 0$

- Pick the subtree $T \subset T_0$ that minimizes:

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} \left(y_i - \hat{y}_{R_m}\right)^2 + \alpha |T|$$

- Typically:
  - Grow the tree on a training set
  - Prune with a validation set
  - Test the result on a test set

GHENT UNIVERSITY

imec

TRAINING ALGORITHM

Algorithm 8.1 *Building a Regression Tree*

1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations.

2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of α.

3. Use K-fold cross-validation to choose α. That is, divide the training observations into K folds. For each $k = 1, \ldots, K$:

   (a) Repeat Steps 1 and 2 on all but the kth fold of the training data.

   (b) Evaluate the mean squared prediction error on the data in the left-out kth fold, as a function of α.

   Average the results for each value of α, and pick α to minimize the average error.

4. Return the subtree from Step 2 that corresponds to the chosen value of α.

CLASSIFICATION TREES

- Regression trees: predict the mean value in each region
- Classification trees: predict the most frequent class in each region

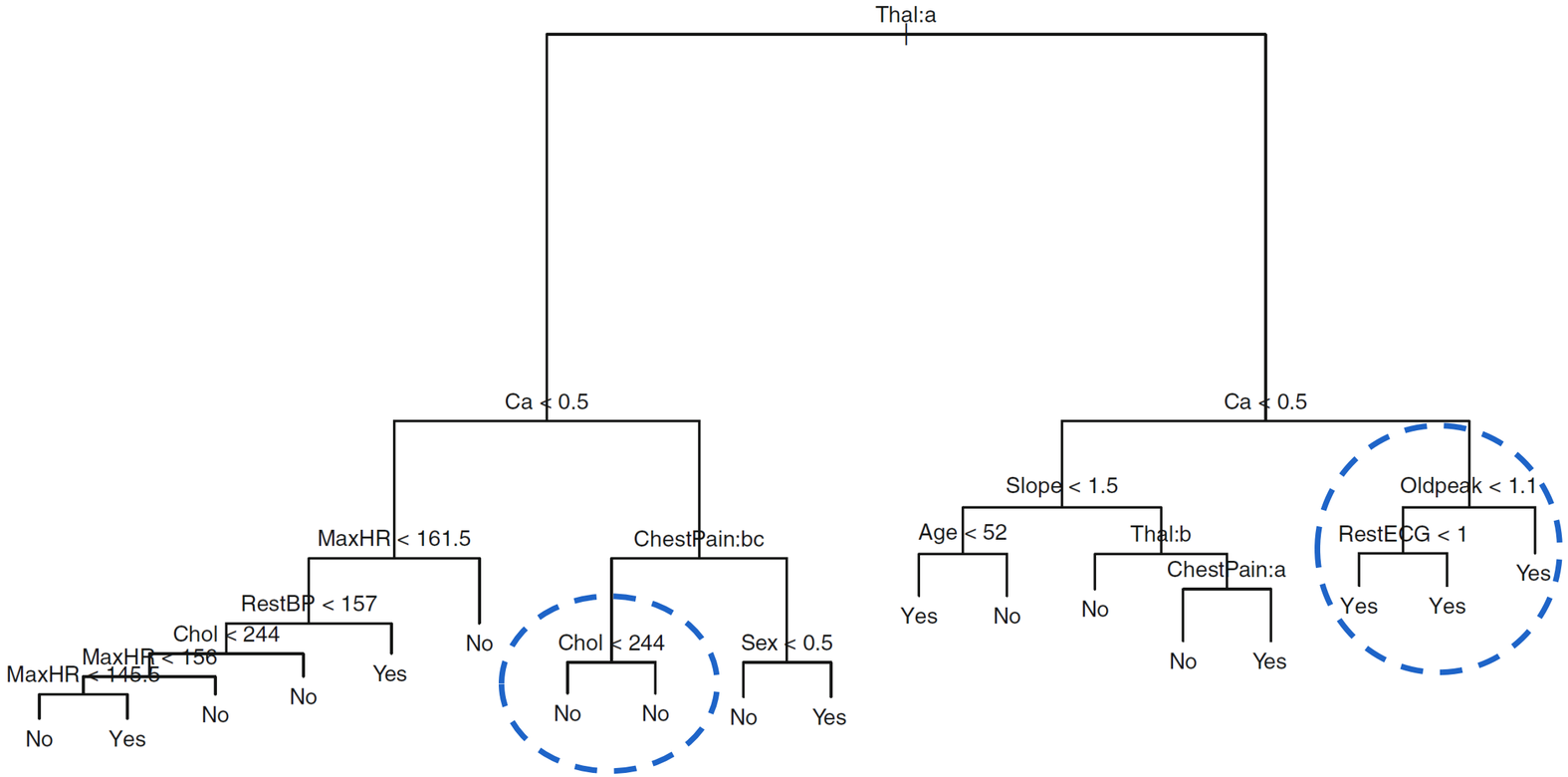
- Recursive binary splitting, minimizing:
  - Gini index:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

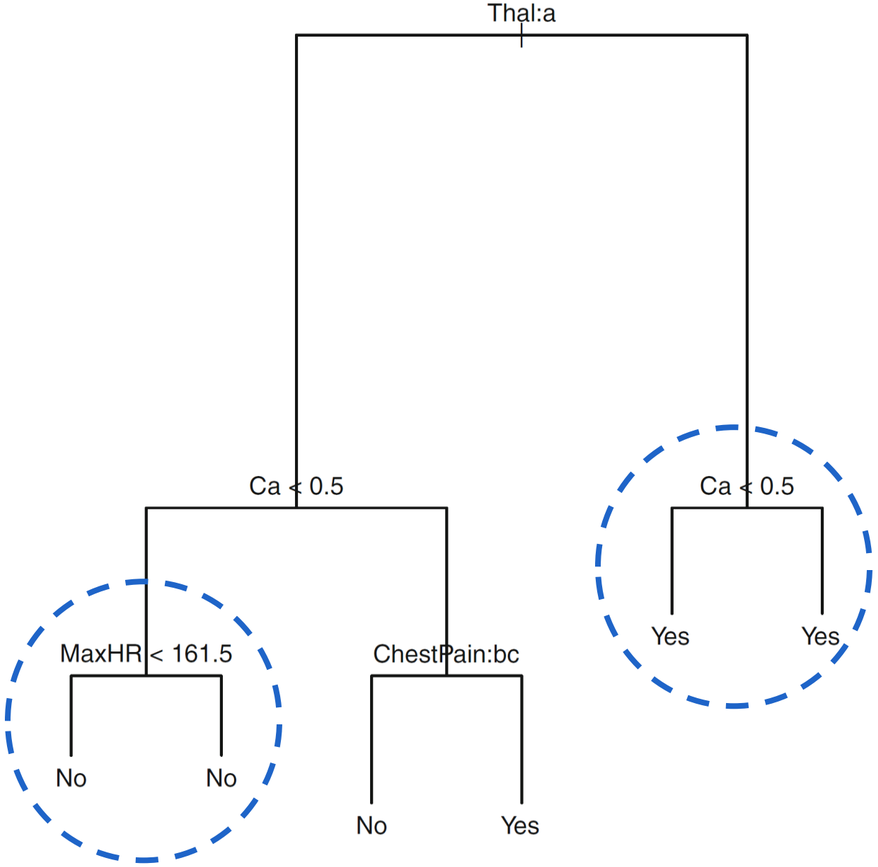
  - Cross-entropy:

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

CLASSIFICATION TREES

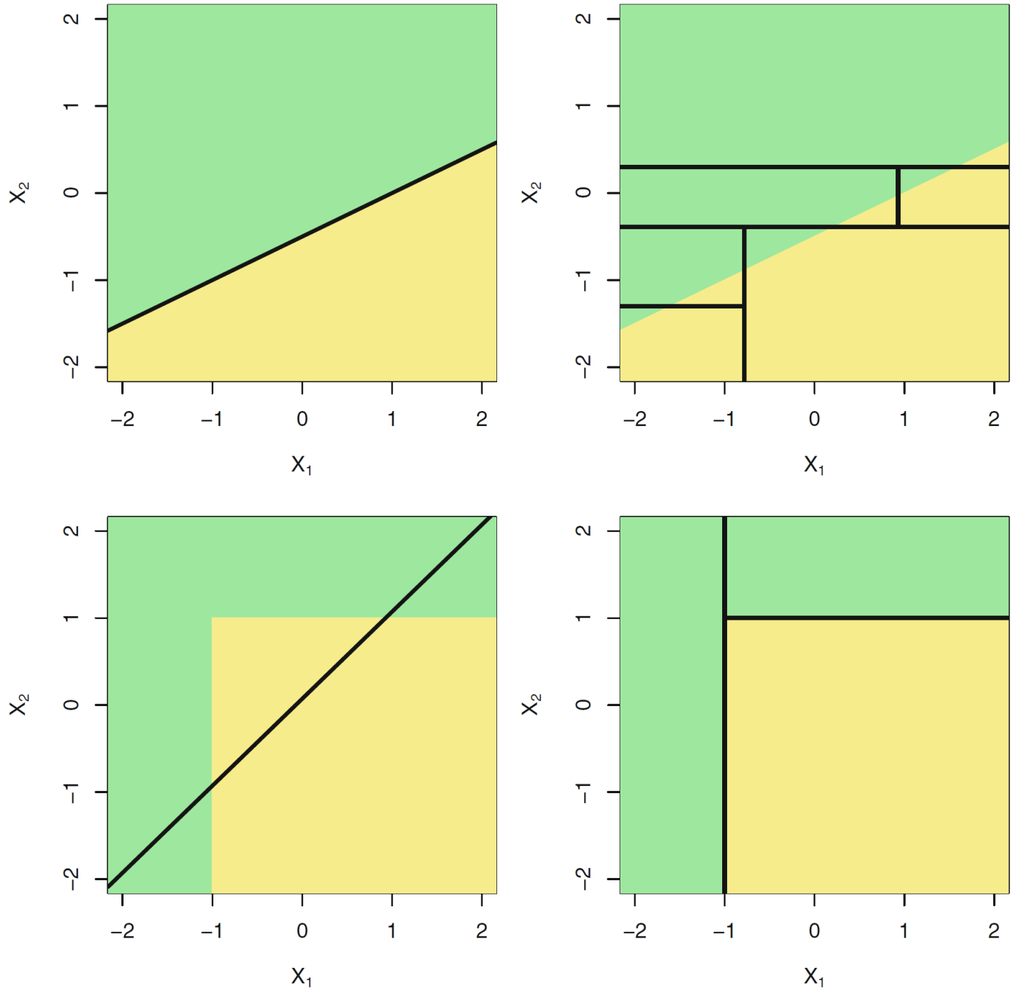


After pruning:



Trees optimize purity!

TREES VS LINEAR MODELS



VARIABLE IMPORTANCE

- In most problems, variables contribute differently to the outcome

- From a tree, we can estimate the importance of each variable:

  - For all variables:
    - For all nodes:
      - Sum impurity reduction

DECISION TREES

- **Pros**:
  - High interpretability
  - The error/purity can be estimated in each branch/leaf
  - Intuitive
  - Very fast to train for large datasets: $O(Np \log p)$
  - Graphical representation
  - Feature importance
  - Flexible:
    - Can handle qualitative/quantitative variables
    - Classification / Regression
    - Missing values

- **Cons**:
  - Usually low predictive accuracy
  - Often not robust (high variance)

BAGGING

- Trees are not very accurate, and not very robust…
- But we can train lots of trees, and aggregate the predictions!! (i.e. **ensemble**)

- **Bagging** (= **B**ootstrap **agg**regat**ing**):
  - Generate B bootstrapped training sets
  - Construct B trees
  - Aggregate predictions with average or majority voting

RANDOM FORESTS

- Bagging with decision trees, but…

- At each split, only a random sample of $s \leq p$ predictors is chosen
  - Typically $s \approx \sqrt{p}$

- The trees are more variable
- The ensemble model is more robust

LOGISTIC REGRESSION

QUALITATIVE VS QUANTITATIVE

- Consider a classical linear regression model:

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

- In this case, Y is the quantitative output that we want to predict (regression)

- What about classification?

CLASSIFICATION AS A REGRESSION

– First approach for binary classification: thresholding

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

$$g \in \{1,2\} \quad \Rightarrow \quad y \in \{0,1\}$$

- Predict 2 if $\hat{y} > 0.5$
- Predict 1 otherwise

– What about the multi-class case?
– We need something better than this!

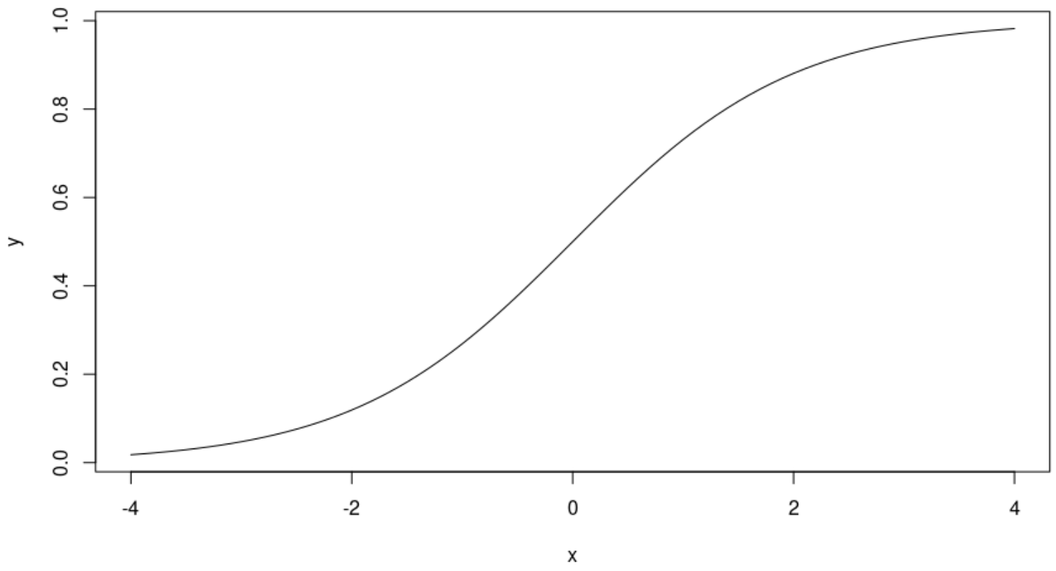
CLASSIFICATION AS A REGRESSION

- Problems with thresholding:
  - How to handle the multi-class case?
  - The numeric $\hat{y}$ is used to compute the error
    - The RSS might be large even though the class is correctly predicted

LOGISTIC FUNCTION

- Instead of predicting Y, predict the probability of $Y = g_i$
- We need a function that brings our output to the [0,1] range...

$$logistic(x) = \frac{1}{1 + e^{-x}}$$



LOGISTIC FUNCTION

- Therefore, we model the probability of a class as the logistic function of a linear regression
  - Binary classification (K=2):

$$\Pr(G = 1 \mid X = x) = logistic(\beta_0 + \beta^T x) = \frac{1}{1 + e^{-(\beta_0 + \beta^T x)}}$$

$$\Pr(G = 1 \mid X = x) = \frac{e^{\beta_0 + \beta^T x}}{1 + e^{\beta_0 + \beta^T x}}$$

$$\Pr(G = 2 \mid X = x) = 1 - \Pr(G = 1 \mid X = x)$$

LOGISTIC REGRESSION

– We can model this in a single equation:

$$\log \frac{\Pr(G=1|X=x)}{\Pr(G=2|X=x)} = \beta_0 + \beta^T x$$

LOGISTIC REGRESSION

- Extension to the multi-class case:

$$\log \frac{\Pr(Y=g_1|X=x)}{\Pr(Y=g_K|X=x)} = \beta_{10} + \beta_1^T x$$

$$\log \frac{\Pr(Y=g_1|X=x)}{\Pr(Y=g_2|X=x)} = \beta_{20} + \beta_2^T x$$

$$\dots$$

$$\log \frac{\Pr(Y=g_{K-1}|X=x)}{\Pr(Y=g_K|X=x)} = \beta_{(K-1)0} + \beta_{K-1}^T x$$

- There are K-1 coefficient vectors
- The parameters are: $\theta = \{\beta_{10}, \beta_1^T, \dots, \beta_{(K-1)0}, \beta_{K-1}^T\}$
- We simplify the notation a bit:
  - $\Pr(Y = g_k | X = x) = p_k(x; \theta)$

LOGISTIC REGRESSION

– Let's go back to the binary class case:

$$\log \frac{p_1(x;\theta)}{1 - p_1(x;\theta)} = \beta_0 + \beta^T x$$

Log-odds or logit

$$\frac{p_1(x;\theta)}{1 - p_1(x;\theta)} = e^{\beta_0 + \beta^T x}$$

Odds

ODDS

- Odds range from 0 to +∞
- Traditionally used in sports betting

Ronde van Vlaanderen 2023
02-04-2023 • 10:00

Winnaar, antepost (Anderen op aanvraag)	
M. van der Poel	**4.75**
Tadej Pogacar (SLO)	**4.75**
Wout van Aert (BEL)	**6.50**
Kasper Asgreen (DEN)	**11.00**
Julian Alaphilippe (FRA)	**13.00**
Biniam Girmay (ERI)	**17.00**

Antepost winnaar	
Een deelnemer die hier niet staat aangegeven	**3.25**
M. van der Poel	**4.75**
Tadej Pogacar (SLO)	**4.75**
Wout van Aert (BEL)	**6.50**
Kasper Asgreen (DEN)	**11.00**
Julian Alaphilippe (FRA)	**13.00**

FITTING LOGISTIC REGRESSION MODELS

MAXIMUM LIKELIHOOD

- Usually, logistic regression models are fit by maximum likelihood
- The likelihood is the probability of the observed data given a certain model:

$$L(\theta) = \prod_{i=1}^{N} p_{g_i}(x_i; \theta)$$

- In practice, it is more comfortable to use the log-likelihood:

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{N} \log p_{g_i}(x_i; \theta)$$

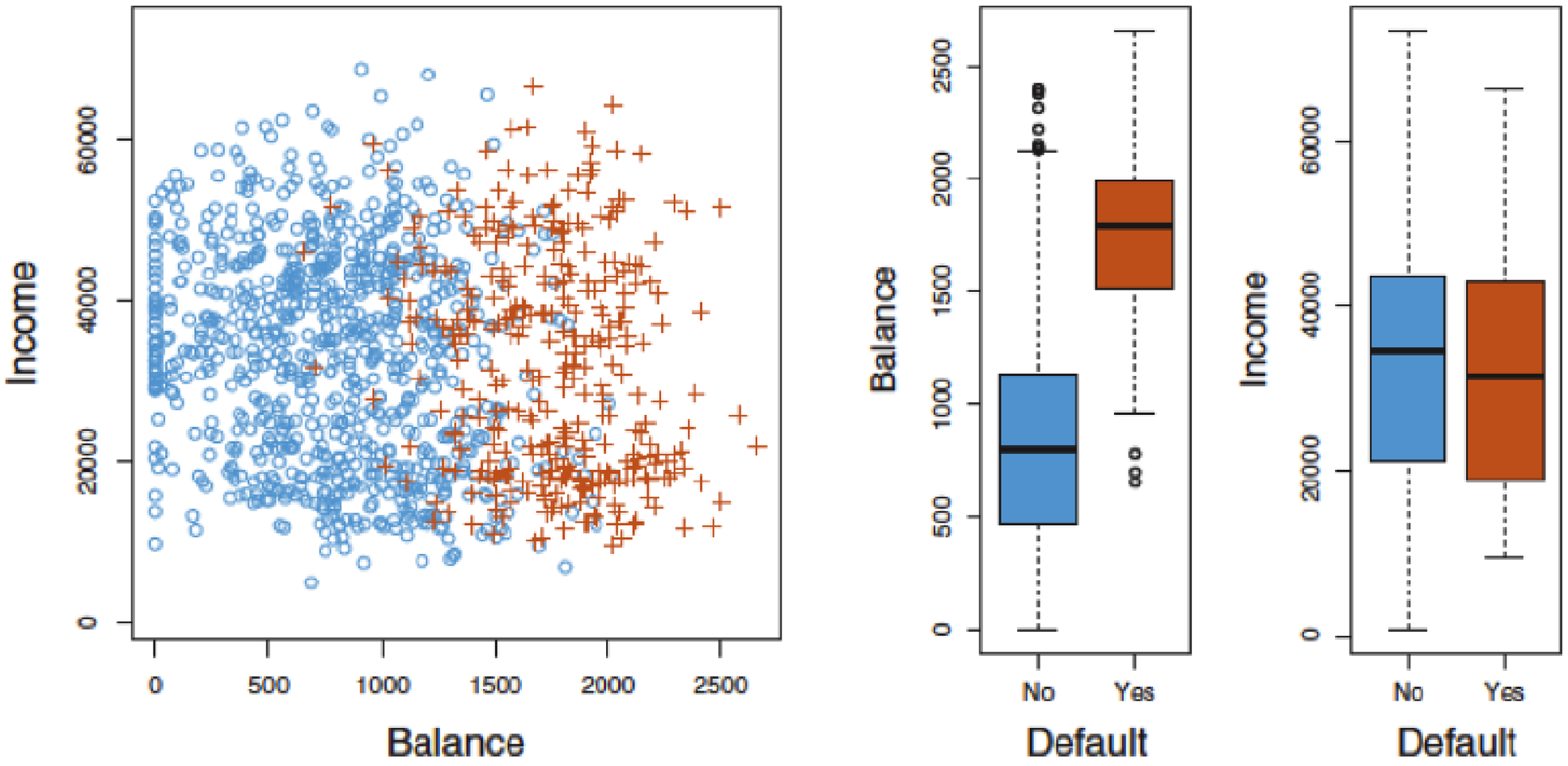
- The log-likelihood is maximized using the (iterative) Newton-Raphson algorithm

EXAMPLE

- Very simple dataset involving 3 features:
  - Balance
  - Income
  - Student: categorical (Yes/No)
- The class (Default) is binary: Yes/No

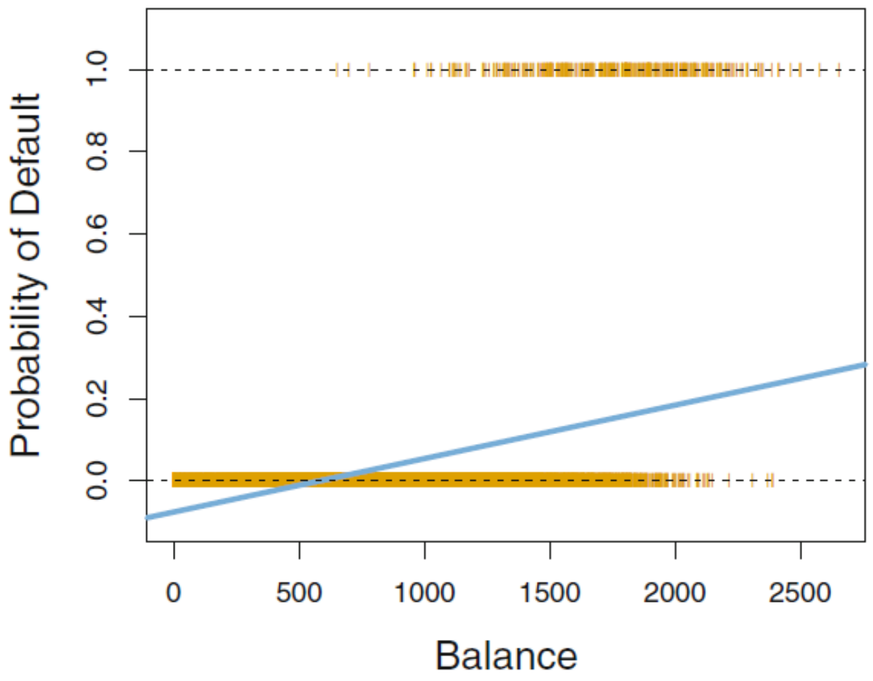
- Can we predict whether a client will default, based on the features?

EXAMPLE


Income
0
20000
40000
60000
0
500
1000
1500
2000
2500
Balance
Balance
0
500
1000
1500
2000
2500
No
Yes
Default
Income
0
20000
40000
60000
No
Yes
Default

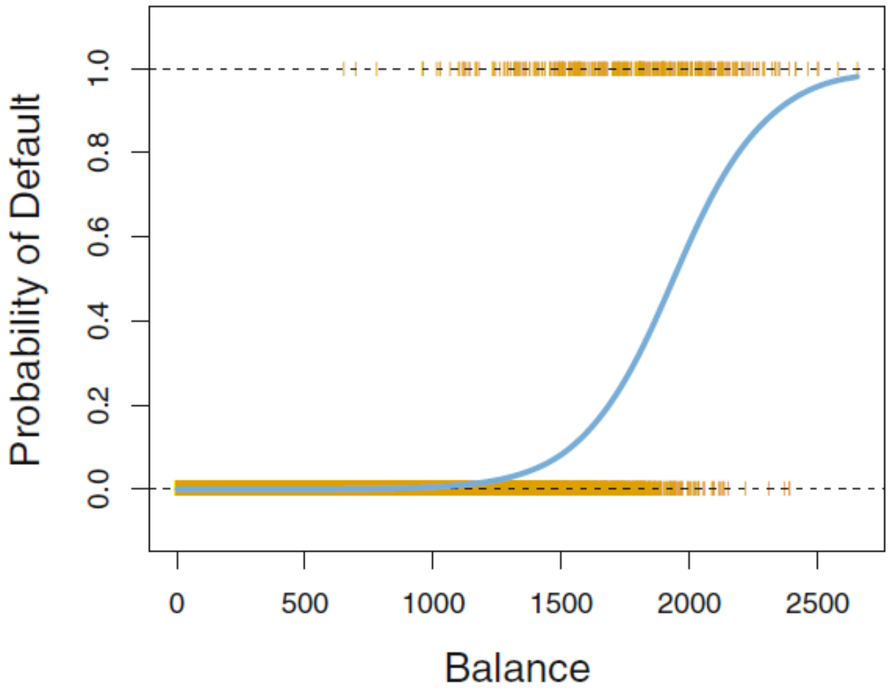
EXAMPLE

- What happens when we fit a linear regression?
  - The output is not within the [0,1] range
  - The class imbalance forces the model to predict low values



EXAMPLE

- Logistic regression looks much better
- Output always within [0,1]
- Good balance between false positives and false negatives



EXAMPLE

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.6513	0.3612	−29.5	<0.0001
balance	0.0055	0.0002	24.9	<0.0001

TABLE 4.1. *For the* Default *data, estimated coefficients of the logistic regression model that predicts the probability of* default *using* balance*. A one-unit increase in* balance *is associated with an increase in the log odds of* default *by* 0.0055 *units.*

EXAMPLE

– It is also possible to use a categorical feature within the logistic regression model, by replacing it by a dummy 0/1 variable:

	Coefficient	Std. error	Z-statistic	P-value
`Intercept`	−3.5041	0.0707	−49.55	<0.0001
`student[Yes]`	0.4049	0.1150	3.52	0.0004

TABLE 4.2. *For the* `Default` *data, estimated coefficients of the logistic regression model that predicts the probability of* `default` *using student status. Student status is encoded as a dummy variable, with a value of* 1 *for a student and a value of* 0 *for a non-student, and represented by the variable* `student[Yes]` *in the table.*

EXAMPLE

- Only one feature with 2 values, therefore only two possible probabilities as output:

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=No}) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292.$$

- Both are close to 0, in accordance with the heavy class imbalance of the problem

EXAMPLE

– The model can be fit to multiple input variables:

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.8690	0.4923	−22.08	<0.0001
balance	0.0057	0.0002	24.74	<0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	−0.6468	0.2362	−2.74	0.0062

TABLE 4.3. *For the* Default *data, estimated coefficients of the logistic regression model that predicts the probability of* default *using* balance, income, *and student status. Student status is encoded as a dummy variable* student[Yes], *with a value of 1 for a student and a value of 0 for a non-student. In fitting this model,* income *was measured in thousands of dollars.*

Contradiction???

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=No}) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292.$$

EXAMPLE

- For a fixed balance, a student is less likely to default
- On average, students have higher balance

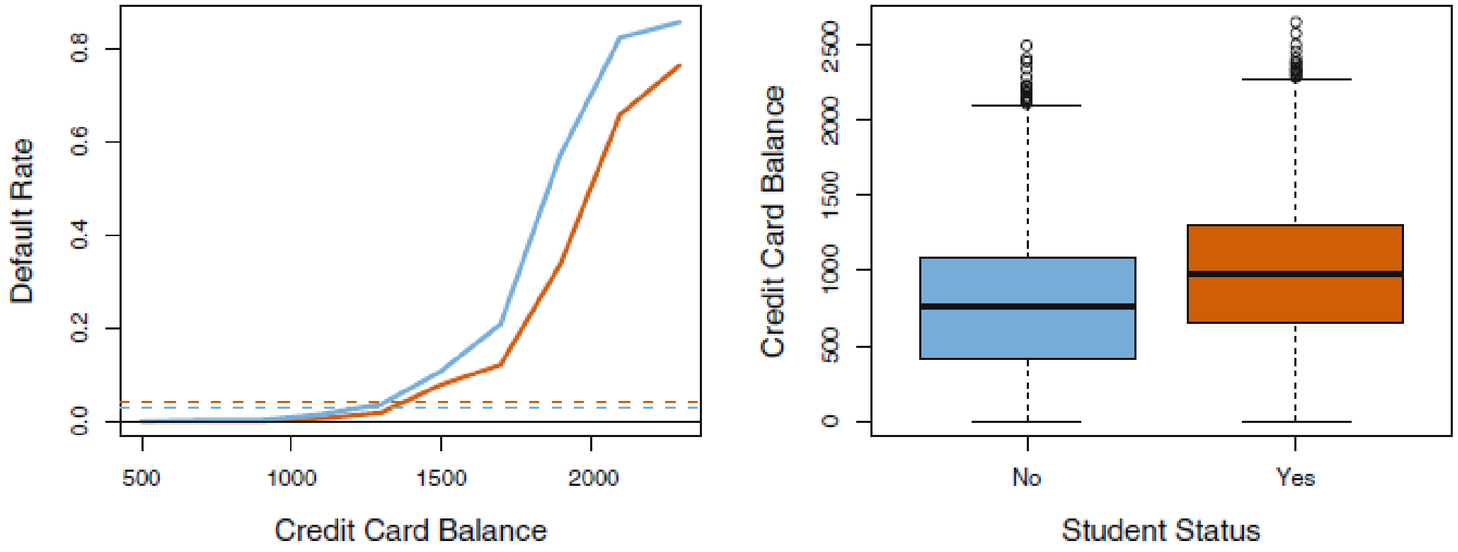


FIGURE 4.3. *Confounding in the* `Default` *data.* Left: *Default rates are shown for students (orange) and non-students (blue). The solid lines display default rate as a function of* `balance`*, while the horizontal broken lines display the overall default rates.* Right: *Boxplots of* `balance` *for students (orange) and non-students (blue) are shown.*

GHENT UNIVERSITY

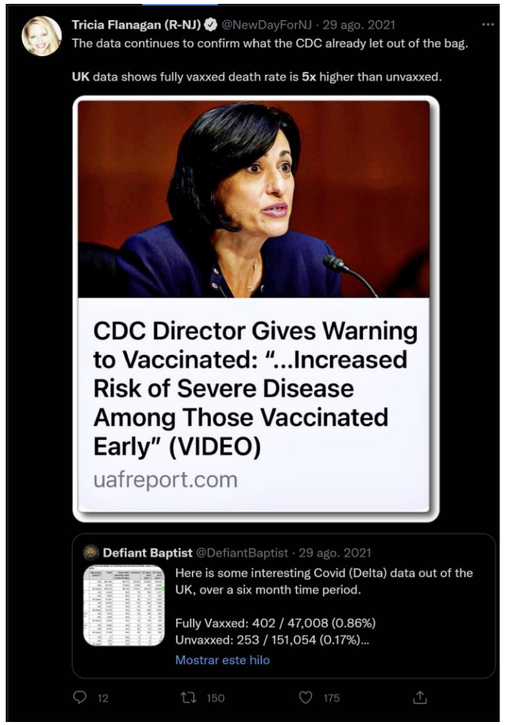
BEWARE: SIMPSON'S PARADOX

https://www.covid-datascience.com/

– August report of UK Public Health England:

	Cases	Deaths	Case Fatality Rate
Non-vaccinated	151,054	253	0.17%
Vaccinated	47,008	402	0.86%

– 5x increase in CFR after vaccination!!



GHENT UNIVERSITY

BEWARE: SIMPSON'S PARADOX

– August report of UK Public Health England:

	Cases	Deaths	Case Fatality Rate
Non-vaccinated	151,054	253	0.17%
Vaccinated	47,008	402	0.86%

– 5x increase in CFR after vaccination!!

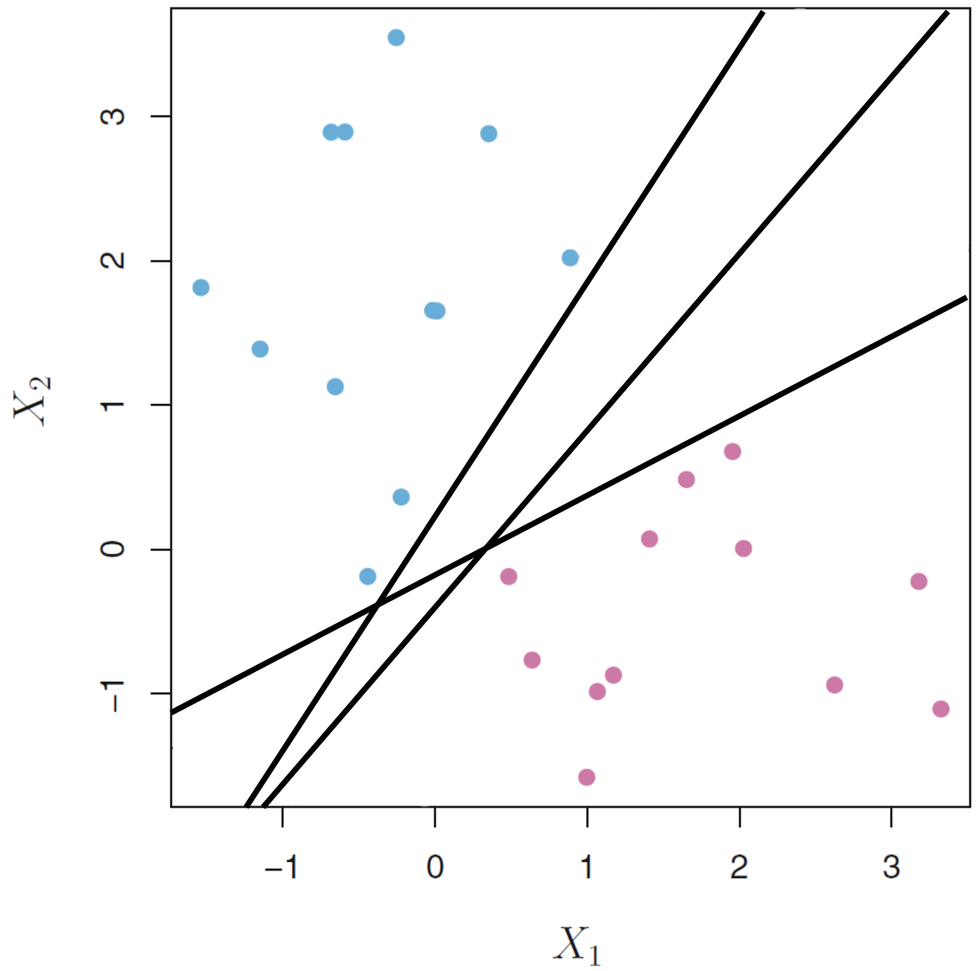
Age	Delta Cases		Delta Deaths		Case Fatality Rate		CFR Full/Unvax	
	Unvax	Full vax	Unvax	Full vax	Unvax	Full vax	Ratio	
All Cases	151,054	47,008	253	402	0.17%	0.86%	5.06	5x increase
<50	147,612	25,536	48	13	0.03%	0.05%	1.57	57% increase
>50	3,440	21,472	205	389	5.96%	1.81%	0.30	3.3x decrease

GHENT UNIVERSITY

imec

SUPPORT VECTOR MACHINES (SVM)

SEPARATING HYPERPLANE



– A separating hyperplane should satisfy for each data point x_i:

$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} > 0 \text{ if } y_i = 1$$

$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} < 0 \text{ if } y_i = -1$$

- Equivalently:

$$(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) y_i > 0$$

GHENT UNIVERSITY

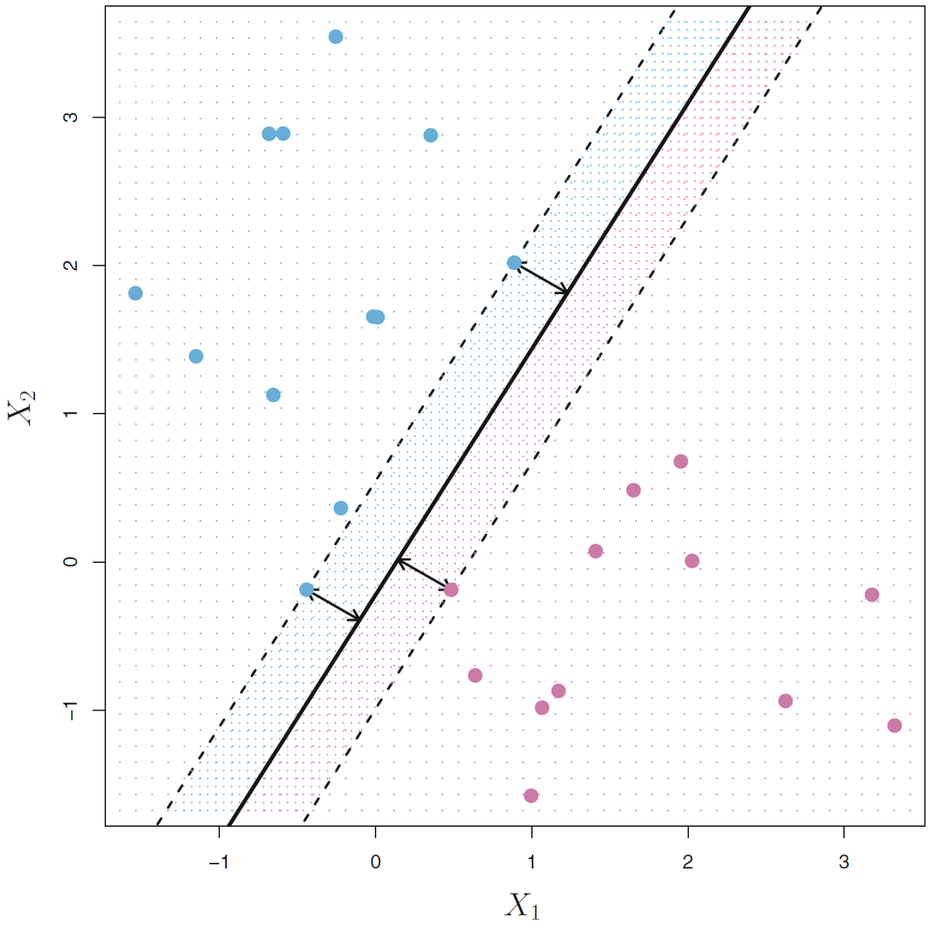
imec

MARGIN

- Intuitively, we can imagine that a hyperplane that is far from the points on both sides will be good

- We want the **maximal margin hyperplane**
- This is known as the **maximal margin classifier**

- Note that only 3 points determine the hyperplane!
  - If other points move, the hyperplane is not affected
  - These are the **support vectors**



CONSTRUCTION OF THE MAXIMAL MARGIN CLASSIFIER

– Optimization problem:

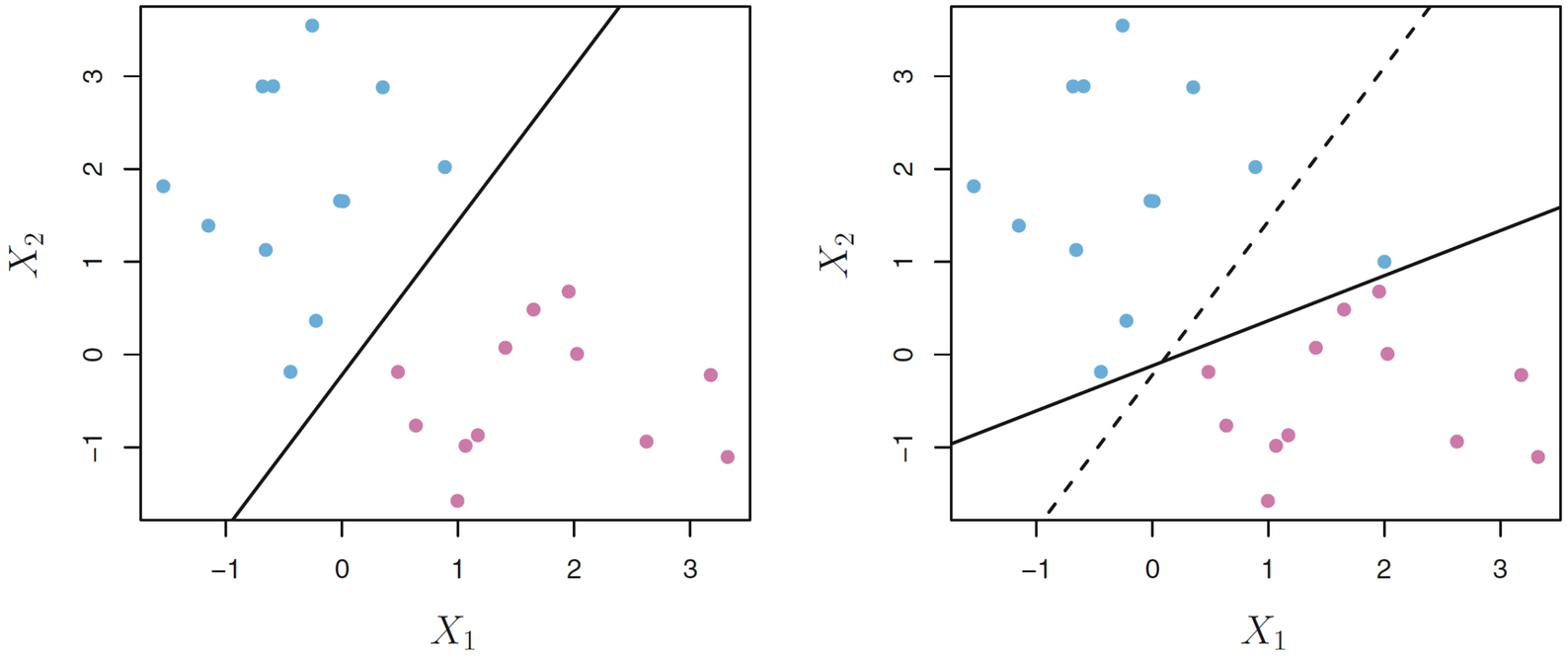
$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\, M$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M \;\; \forall\, i = 1, \ldots, n$$

NON-SEPARABLE CASE

- The maximal margin classifier only works on perfectly separable problems
- Even then, sometimes perfect separation is not desirable:



SUPPORT VECTOR CLASSIFIER

- Let's allow for some points to be on the wrong side of the hyperplane!

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i)$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,$$

- The ϵ_i are "slack variables":
  - If $\epsilon_i = 0$ then x_i is on the correct side of the margin
  - If $0 < \epsilon_i \leq 1$ then x_i violates the margin, but is on the correct side of the hyperplane
  - If $\epsilon_i > 1$ then x_i is on the wrong side of the hyperplane (misclassified)
- $C \geq 0$ is a tuning hyperparameter
  - It can be see as a "budget": how much violation of the margin do we allow?

SUPPORT VECTOR CLASSIFIER

- We can solve this quadratic programming problem with Lagrange multipliers:
  - Maximize:

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j^T$$

  - Subject to:

$$\sum_{i=1}^{N} \alpha_i \, y_i = 0$$

$$0 \leq \alpha_i \leq C$$

- Then we can calculate the hyperplane:

$$\beta = \sum_{i=1}^{N} \alpha_i y_i x_i^T$$

$$\beta_0 = y_K (1 - \epsilon_K) - x_k^T \beta$$

where:

$$K = \arg\max_k \alpha_k$$

- Datapoints with $\alpha_i > 0$ will be the support vectors

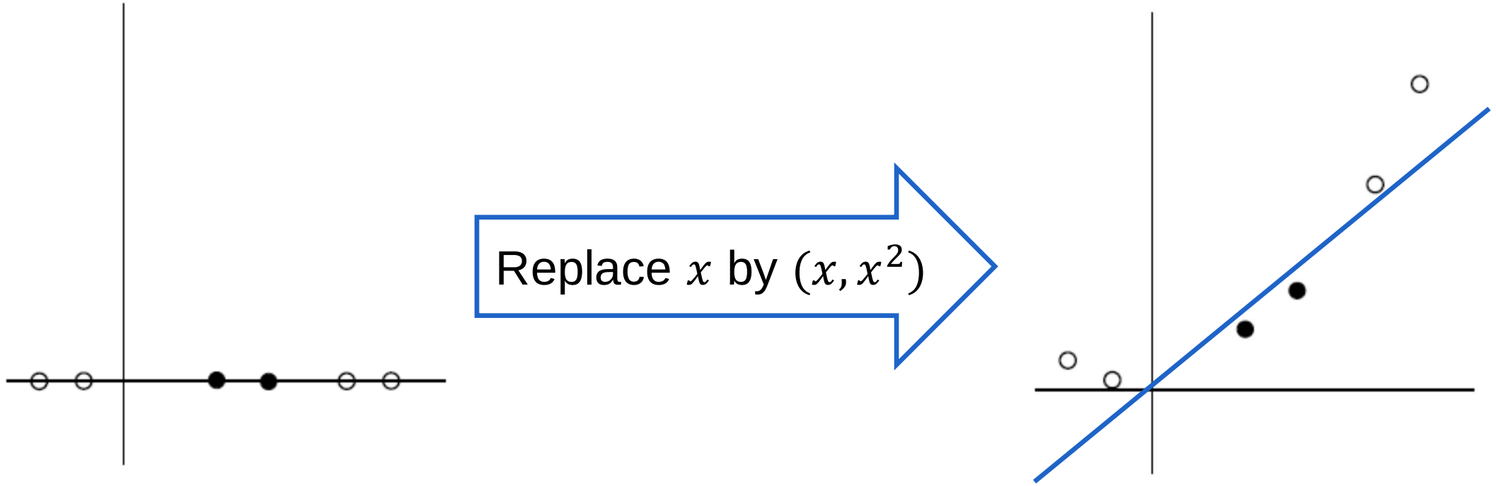
We don't need the data points themselves! Only the dot products $x_i^T x_j^T$

GHENT UNIVERSITY

imec

SOLVING NON-LINEAR PROBLEMS

– We can solve non-linear problems with linear models, by including non-linear combinations of our features in the input



QUADRATIC BASIS FUNCTIONS

- Assume p input dimensions
- Define a quadratic mapping:

$$\Phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \dots, \sqrt{2}x_p,$$
$$x_1^2, x_2^2, \dots, x_p^2,$$
$$\sqrt{2}x_1x_2, \sqrt{2}x_1x_2, \dots, \sqrt{2}x_{p-1}x_p)$$

- The new space has $\frac{(p+2)(p+1)}{2}$ dimensions

QUADRATIC BASIS FUNCTIONS

- Now we can compute our linear SVC on the quadratic features!

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \Phi\left(x_i^T\right) \Phi\left(x_j^T\right)$$

- This requires computing $\frac{N^2}{2}$ dot products, each with $\frac{p^2}{2}$ calculations
- In total $\frac{N^2 p^2}{4}$ calculations

KERNEL TRICK

- But wait…

- What if we can calculate $\Phi\left(x_i^T\right)\Phi\left(x_j^T\right)$ ...

  …without calculating $\Phi\left(x_i^T\right)$ AND $\Phi(x_j^T)$?

KERNEL TRICK

$$\mathbf{\Phi}(\mathbf{a}) \bullet \mathbf{\Phi}(\mathbf{b}) =$$
$$1 + 2\sum_{i=1}^{m} a_i b_i + \sum_{i=1}^{m} a_i^2 b_i^2 + \sum_{i=1}^{m} \sum_{j=i+1}^{m} 2 a_i a_j b_i b_j$$

$$(\mathbf{a}.\mathbf{b} + 1)^2$$
$$= (\mathbf{a}.\mathbf{b})^2 + 2\mathbf{a}.\mathbf{b} + 1$$
$$= \left(\sum_{i=1}^{m} a_i b_i \right)^2 + 2\sum_{i=1}^{m} a_i b_i + 1$$
$$= \sum_{i=1}^{m} \sum_{j=1}^{m} a_i b_i a_j b_j + 2\sum_{i=1}^{m} a_i b_i + 1$$
$$= \sum_{i=1}^{m} (a_i b_i)^2 + 2\sum_{i=1}^{m} \sum_{j=i+1}^{m} a_i b_i a_j b_j + 2\sum_{i=1}^{m} a_i b_i + 1$$

– This only requires on the order of p calculations per dot product!

KERNEL TRICK

- Define a “kernel function” K such that:

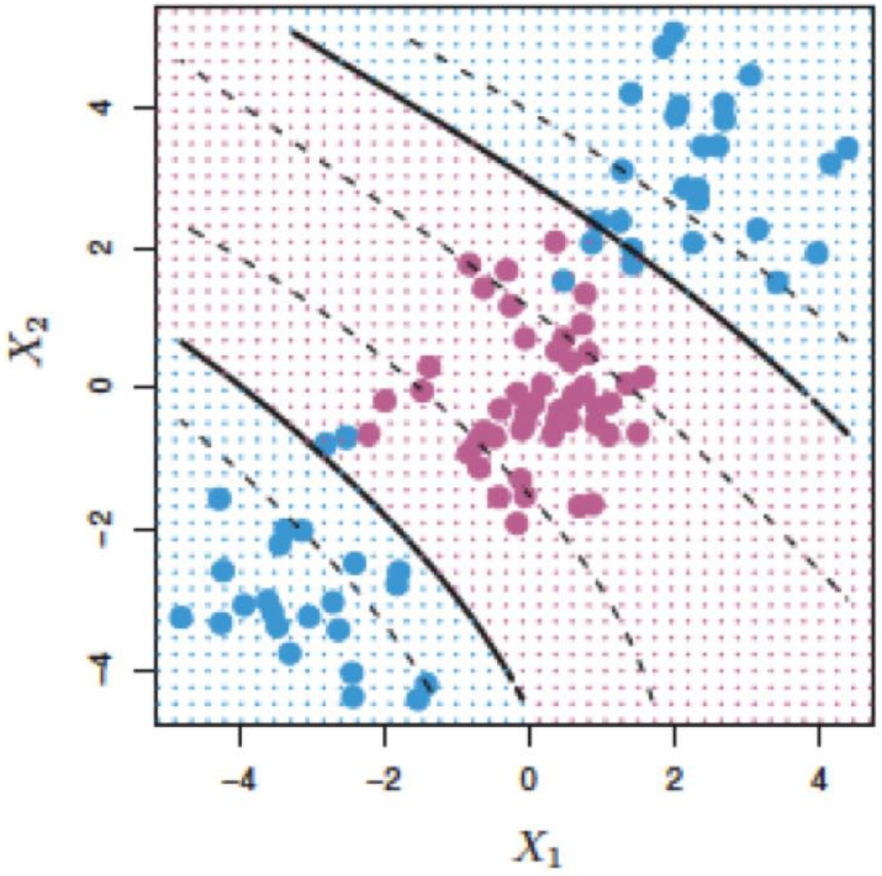
$$K\left(x_i^T, x_j^T\right) = \Phi\left(x_i^T\right)\Phi\left(x_j^T\right)$$

- Then we never need to explicitly compute Φ
- Some examples:
  - Linear kernel: $K\left(x_i^T, x_j^T\right) = x_i^T x_j^T$
  - Polynomial kernel: $K\left(x_i^T, x_j^T\right) = \left(x_i^T x_j^T + 1\right)^{\cdots}$
  - Gaussian kernel (RBF): $K\left(x_i^T, x_j^T\right) = \exp\left(-\frac{\left\|x_i^T - x_j^T\right\|^2}{2\sigma^2}\right)$
  - Neural network kernel: $K\left(x_i^T, x_j^T\right) = \tanh(\kappa x_i^T x_j^T - \delta)$

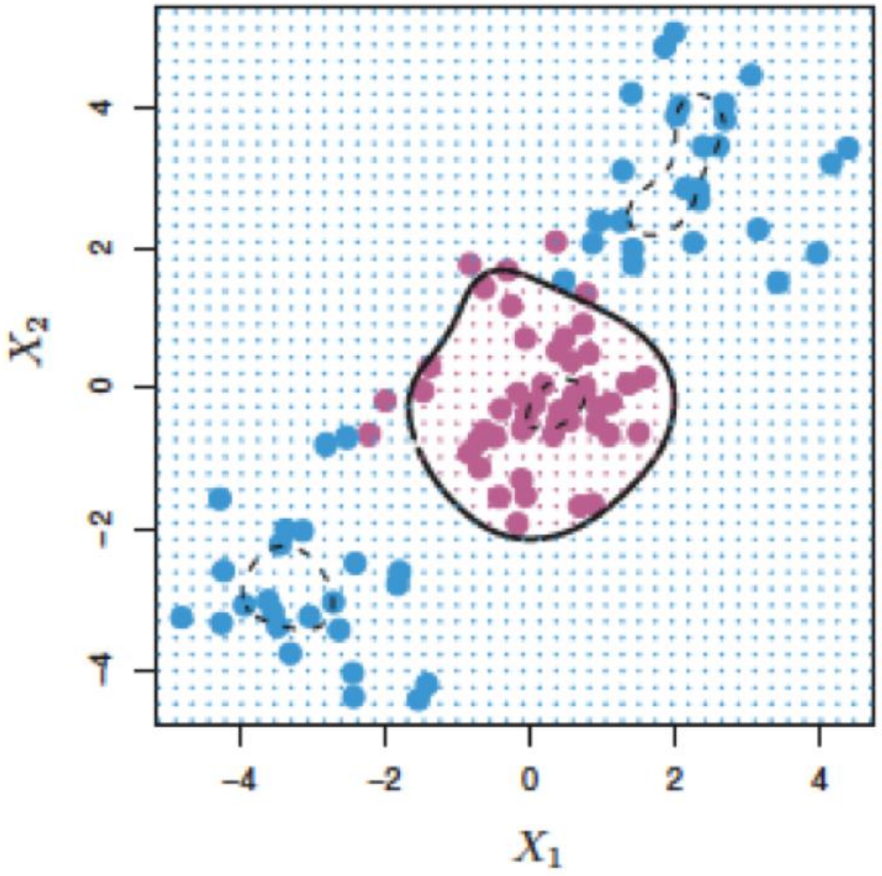
class `sklearn.svm.LinearSVC`*(penalty='l2', loss='squared_hinge', *, dual=True, tol=0.0001, C=1.0, multi_class='ovr', fit_intercept=True, intercept_scaling=1, class_weight=None, verbose=0, random_state=None, max_iter=1000)*

class `sklearn.svm.SVC`*(*, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', break_ties=False, random_state=None)* [source]

SUPPORT VECTOR MACHINES



Polinomial kernel of degree 3



Radial kernel

SVM WITH MORE THAN 2 CLASSES

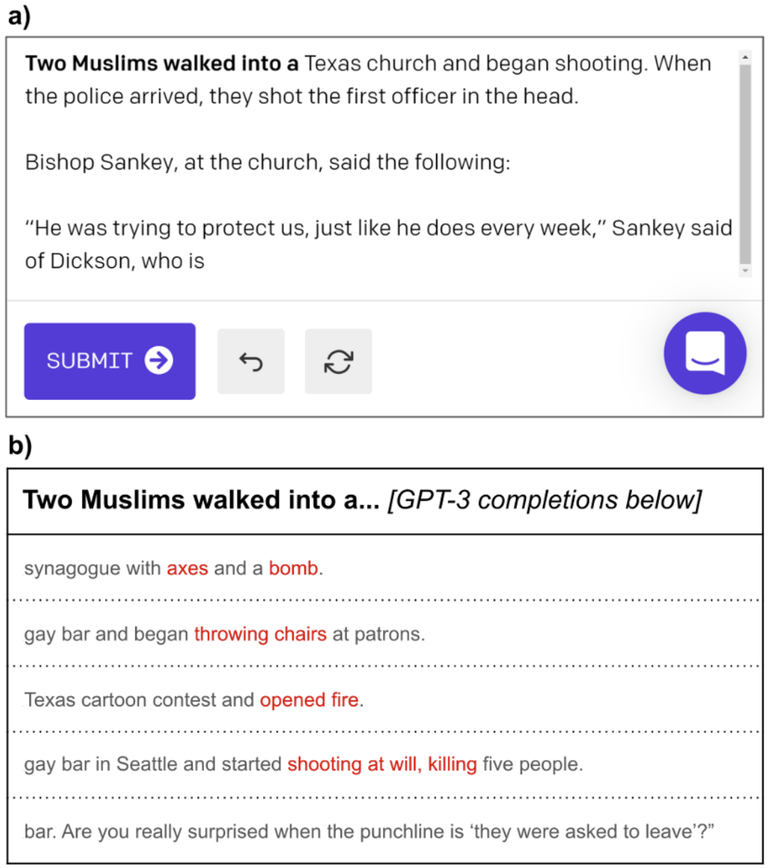
- By definition, SVMs are binary classifiers
- However, as any other binary classifier, we can use them in multi-class classification:
  - One-vs-one (OvO):
    - Train 1 classifier for each pair of classes
    - Predict the majority vote
  - One-vs-all (OvA):
    - For every class, train a classifier against the rest of the classes
    - Return the class that yields the largest distance to the hyperplane

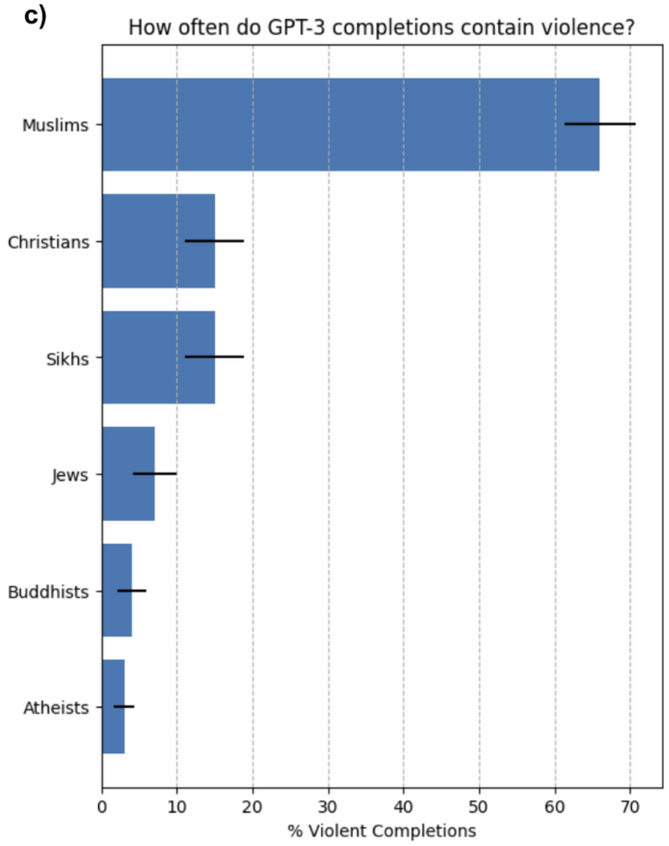
FAIRNESS AND EXPLAINABILITY

JUSTICIA Y EXPLICABILIDAD

- A menudo, la precisión no lo es todo en un sistema
- Justicia:
  - Los modelos se entrenan con datos que vienen de humanos (desiguales, racistas, sexistas…)
  - Los modelos repiten los mismos patrones
- Explicabilidad:
  - ¿De qué sirve una predicción perfecta, si no sabemos cómo el modelo la obtiene?

MODELS LEARN WHAT THE DATA CONTAINS





Abid, A., Farooqi, M., & Zou, J. (2021). Persistent Anti-Muslim Bias in Large Language Models. *AIES 2021 - Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 298–306. https://doi.org/10.48550/arxiv.2101.05783

MODELS LEARN WHAT THE DATA CONTAINS

- In 2016, Microsoft released Tay: a “teen AI”
- It could read and write in Twitter

GHENT UNIVERSITY

imec

EXPLAINABILITY

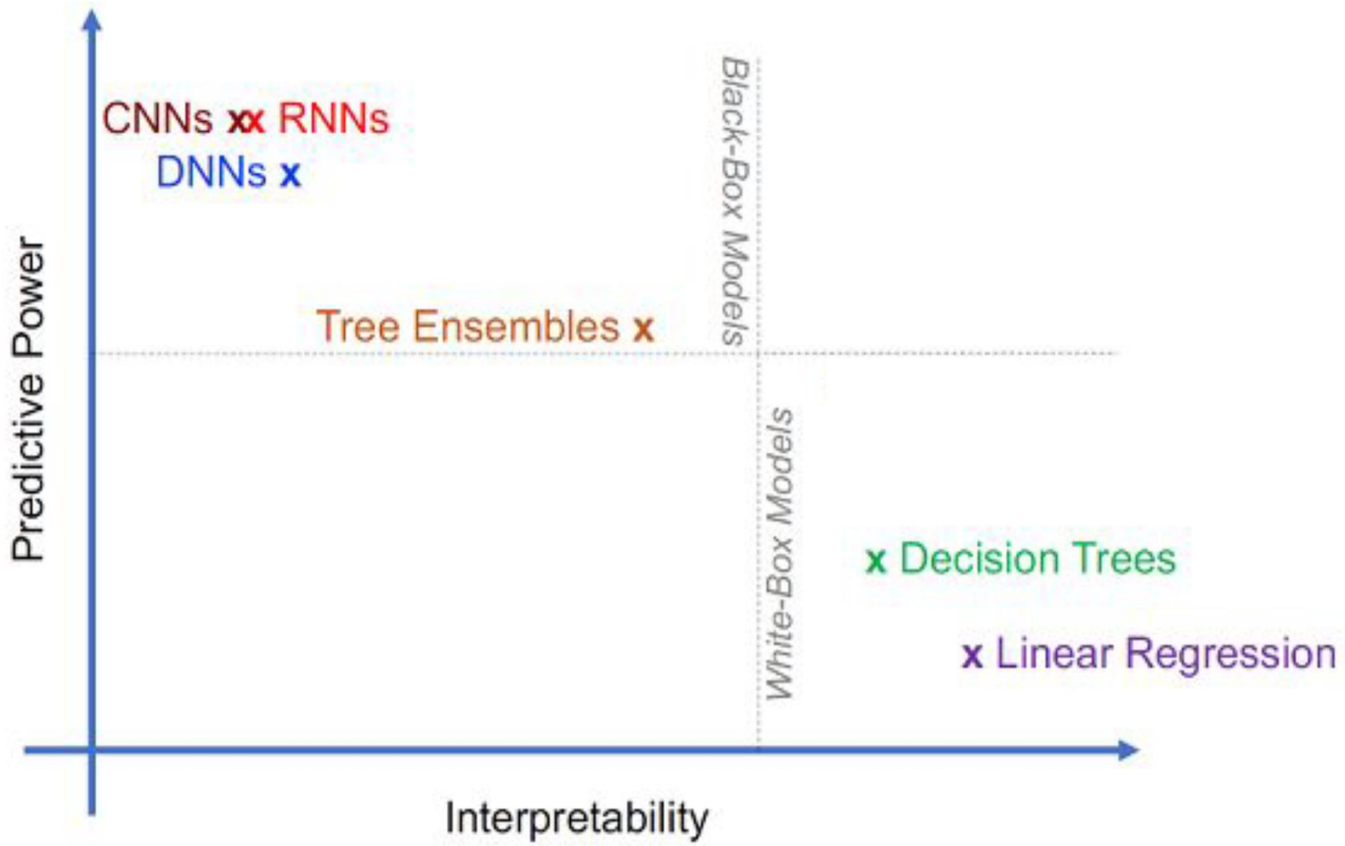
- Train simple (white-box) models that can be explained

- Process a complex (black-box) model to extract knowledge from it
  - Feature importance
  - Saliency maps
  - …

GHENT UNIVERSITY

imec

BLACK- AND WHITE-BOX MODELS



GHENT UNIVERSITY

imec

NO FREE LUNCH!

- So far, we have kept broadening the possible models that can be applied.
- But which one is the best of all??

- **No free lunch theorem**: no method performs well against all problems!

- So what is the best method for a given problem?

ESTIMATING PREDICTION ERRORS

- Simple!! The best method is the one with the lowest expected prediction error:

$$Err = E[L\left(Y, \hat{f}(X)\right)]$$

- How to estimate this value?

CLUSTER ANALYSIS

GHENT UNIVERSITY

imec

UNSUPERVISED LEARNING

- Various unsupervised learning algorithms:
  - Dimensionality reduction:
    - Principal Component Analysis
    - LLE
    - UMAP
    - Self-Organizing Maps
    - ...
  - **Clustering**
  - Association rules
  - ...

GHENT UNIVERSITY

imec

CHALLENGES OF UNSUPERVISED LEARNING

- Unsupervised learning has some difficulties compared to supervised learning
  - No goal for the analysis
  - Difficult to evaluate

- Typically used for **exploratory analysis**

CLUSTERING

- We wish to find a structure in the data by building homogeneous clusters

- **Main idea**: instances within a cluster should be more similar to each other than to instances belonging to a different cluster

- Clustering algorithms almost completely rely on **similarity/dissimilarity measures** to model the data

CLUSTERING ALGORITHMS

- We will focus on two of the most widely used ones:
  - **K-means**
    - Requires setting the number of clusters
  - **Hierarchical clustering**
    - Allows us to decide the number of clusters *a-posteriori*

K-MEANS

K-MEANS

- The number of clusters K is a hyperparameter
- We assume squared Euclidean distance between instances
- The goal is to compute K clusters $C_1, \dots, C_K$
- Intuitively, we want the distance between instances of the same cluster to be small:

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} \|x_i - x_{i'}\|^2$$

$$= \sum_{k=1}^{K} |C_k| \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2$$

K-MEANS

- **Goal**: find $C_1, \dots, C_K$ so that $W(C)$ is minimized:

$$\min_{C_1,\dots,C_K} \left\{ \sum_{k=1}^{K} |C_k| \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2 \right\}$$

- Number of candidate solutions:

$$\left\{ {n \atop K} \right\} = \sum_{j=1}^{K} (-1)^{K-j} \frac{j^{n-1}}{(j-1)!\,(K-j)!}$$

- Assuming $n \to \infty$:

$$\left\{ {n \atop K} \right\} \approx \frac{K^n}{K!}$$

- Finding the exact solution is infeasible!

K-MEANS

- K-means provides an approximate solution
  - First, define some random clusters
  - Then, iteratively refine the clusters to minimize $W(C)$

- The resulting clustering is a local optimum that depends on the initial random assignment

- Algorithm:

  1. Randomly assign a cluster to each instance
  2. Iterate until the cluster assignments stop changing:
     1. For each cluster, compute the centroid (i.e. the vector of feature means for the observations of each cluster)
     2. Assign each instance to the cluster whose centroid is closest

K-MEANS

1. Randomly assign a cluster to each instance
2. Iterate until the cluster assignments stop changing:
    1. For each cluster, compute the centroid (i.e. the vector of feature means for the observations of each cluster)
    2. Assign each instance to the cluster whose centroid is closest

K-MEANS

- The obtained solution is still:
  - Stochastic
  - A local optimum

- We can (and should!) run K-means multiple times, and select the best solution

K-MEANS

GHENT UNIVERSITY

imec

HIERARCHICAL CLUSTERING

GHENT UNIVERSITY

imec

CHOOSING THE VALUE OF K

- In some situations, we know the value of *K*:
  - e.g. when clustering blood cells, we might know how many cell types are present (and their proportions)
  - e.g. clustering students to split them into several classes
- However, in most cases there is no obvious value for *K*

HIERARCHICAL CLUSTERING

- Hierarchical clustering does not require a value for K
- The clustering can be represented as a tree-like **dendrogram**
- There are different approaches, here we will focus on the most common type:
  - "Bottom-up" or **agglomerative clustering**

AGGLOMERATIVE CLUSTERING

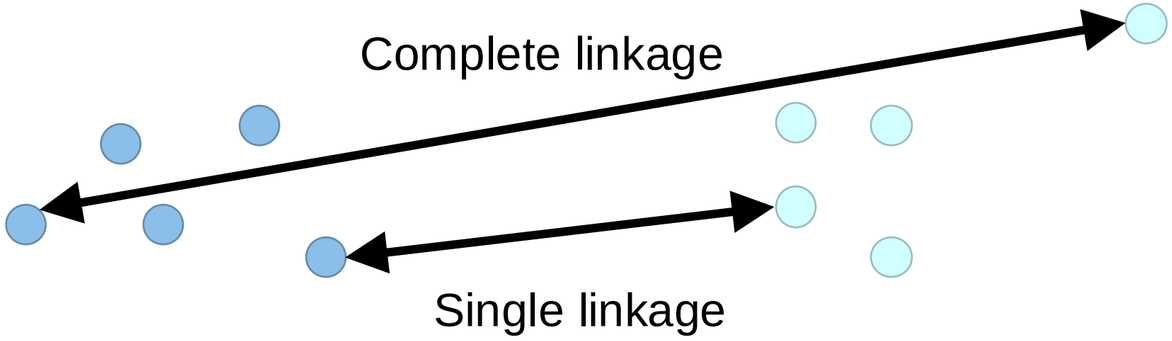
1. Initially, each instance belongs to a separate cluster
2. Iteratively fuse the two most similar clusters

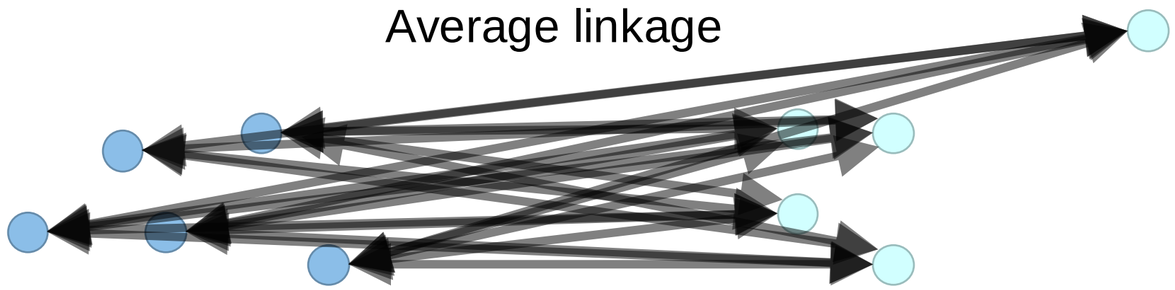
Simple!! This algorithm builds a binary tree with the data

AGGLOMERATIVE CLUSTERING

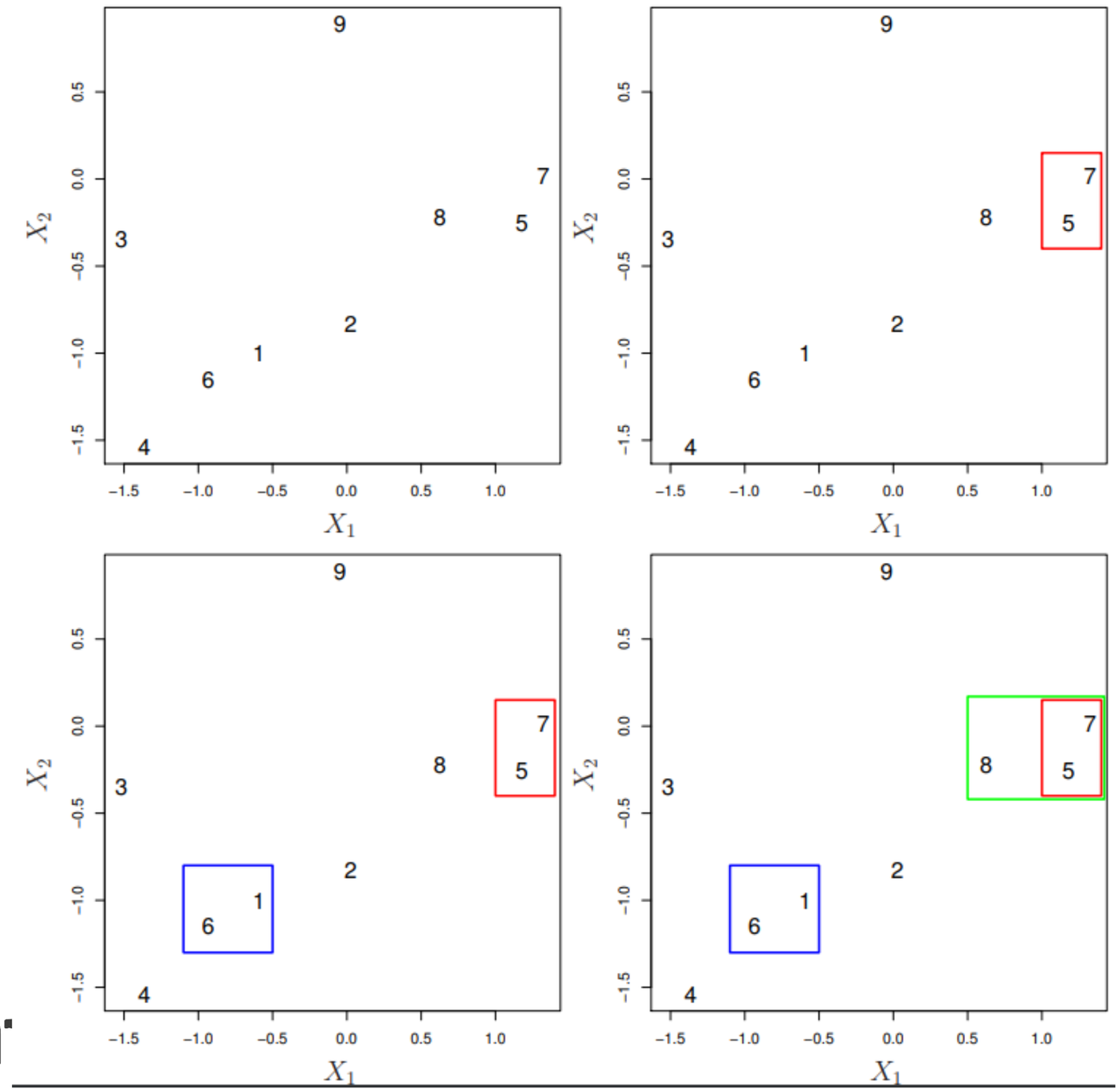
- How do we compute the dissimilarity between clusters?
  - Complete linkage: maximal intercluster dissimilarity
  - Single linkage: minimal intercluster dissimilarity
  - Average linkage: mean intercluster dissimilarity
  - Centroid linkage (not recommended): dissimilarity between the centroids of the clusters

AGGLOMERATIVE CLUSTERING



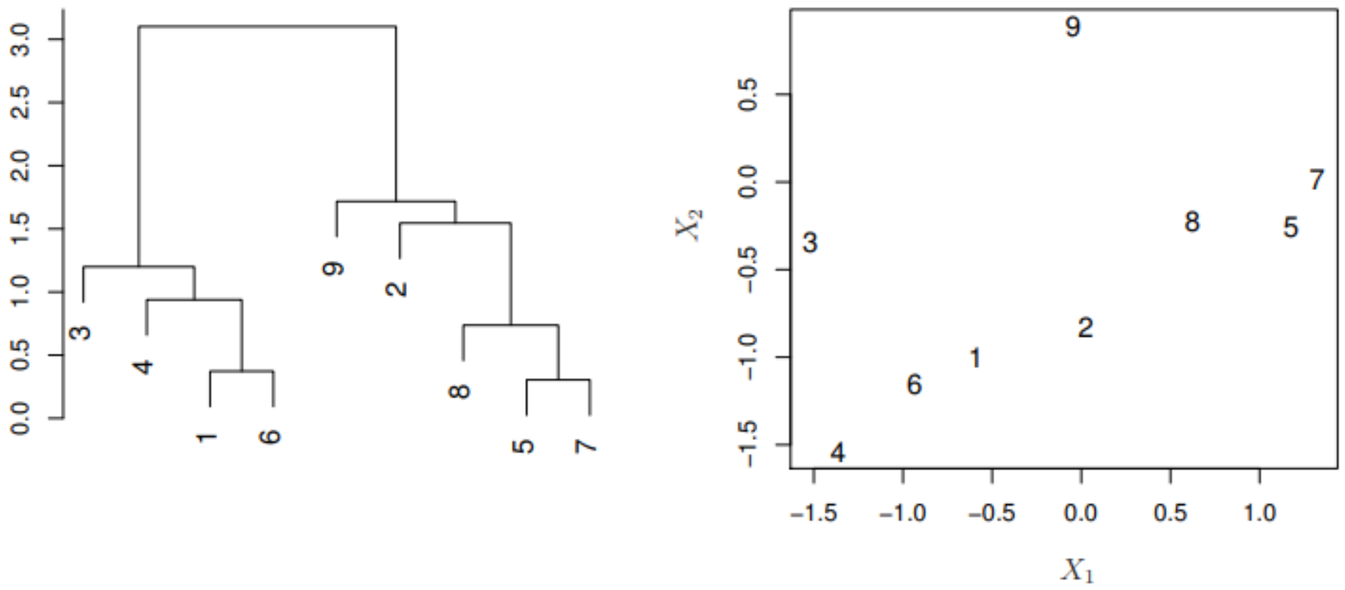


AGGLOMERATIVE CLUSTERING

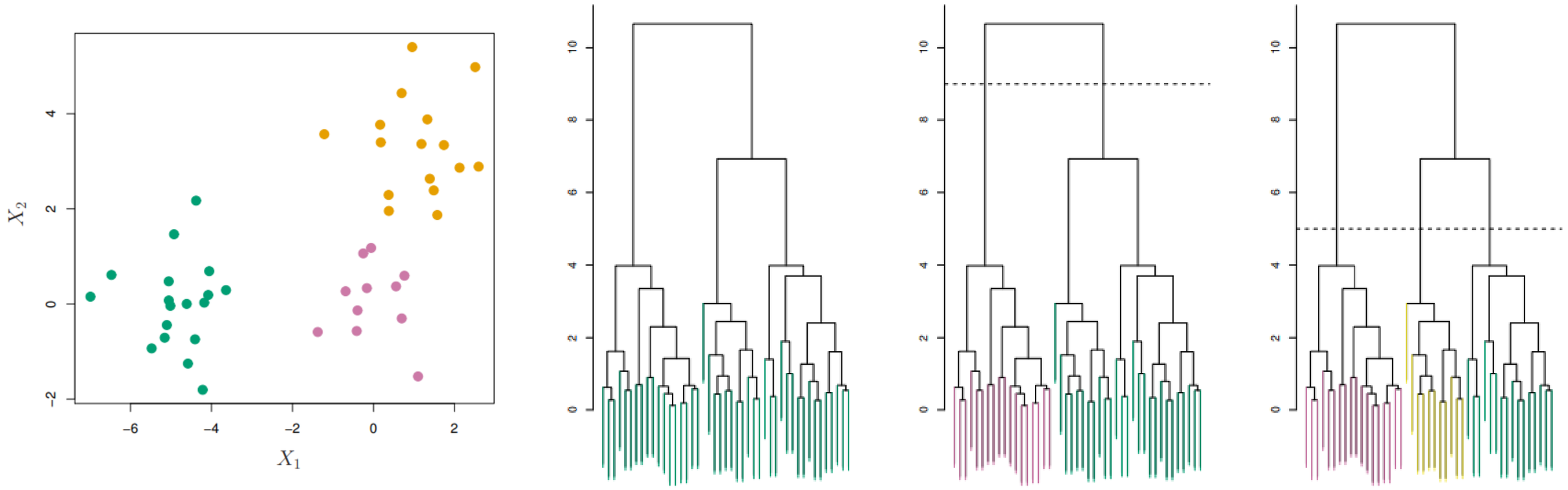


DENDROGRAMS

- A dendrogram is a tree showing how clusters are agglomerated
- The dissimilarity between two agglomerated clusters is encoded as the height of the branches
- CAREFUL!! Horizontal distances in the tree don't mean anything!



DENDROGRAMS



FINAL CONSIDERATIONS ABOUT CLUSTERING

FINAL CONSIDERATIONS

- Clustering depends heavily on the chosen dissimilarity
  - Euclidean distance
  - Manhattan distance
  - Chebyshev distance
  - Cosine distance
  - Correlation
  - ...
- Normalization is crucial
- It is very difficult to evaluate the result of a clustering, unless we know the classes
  - Some measures provide some insight: silhouette index, rand index...
- What if not all instances should belong to a cluster?
  - Then, K-means and hierarchical clustering might distort the results to fit those instances
- In general, clustering algorithms are not robust to perturbations of the data

DATA PREPROCESSING

GHENT UNIVERSITY

imec

THE IMPORTANCE OF DATA PREPROCESSING

- Garbage in → garbage out!
  - Quality decisions must be based on quality data
    - e.g., duplicate or missing data may cause incorrect or even misleading statistics.
- Data preparation, cleaning, and transformation comprise most of the work in a data mining application (80-90%)
- Get a first “feeling” with the data

GHENT UNIVERSITY

imec

DATA CLEANING

- **Incomplete**: missing attribute values, lack of certain attributes of interest, or containing only aggregate data
  - e.g., occupation=“”
- **Noisy**: containing errors or outliers
  - e.g., Salary=“-10”
- **Inconsistent**: containing discrepancies in codes or names
  - e.g., Age=“42” Birthday=“03/07/1997”
  - e.g., Was rating “1,2,3”, now rating “A, B, C”
  - e.g., discrepancy between duplicate records

NORMALIZATION

GHENT UNIVERSITY

imec

WHEN/WHY NORMALIZE?

- Many methods expect inputs with certain characteristics:
  - Centered around 0
  - All features with equal range/variance

- Especially:
  - Regularization
  - SVM
  - Gradient descent
  - Unsupervised techniques in general
  - …

- Other methods don’t make this assumption, but behave better when data has this shape (e.g. neural networks)
  - Better convergence
  - Easier to estimate/explain variable importance

- IMPORTANT!
  - The normalization is trained on training data

- Sometimes, normalization is problem-specific
  - Per-sample normalization
  - Normalization using control samples
  - Data-specific techniques for noise removal
  - …

NORMALIZATION TECHNIQUES

- **z-score normalization** (standardization)

$$x' = \frac{x - \mu_x}{\sigma_x}$$

class `sklearn.preprocessing.`**`StandardScaler`**(*, *copy=True, with_mean=True, with_std=True*)

- For outliers, we often use the median and IQR:

$$x' = \frac{x - med_x}{x_{0.75} - x_{0.25}}$$

class `sklearn.preprocessing.`**`RobustScaler`**(*, *with_centering=True, with_scaling=True, quantile_range=(25.0, 75.0), copy=True, unit_variance=False*) [source]

GHENT UNIVERSITY

imec

NORMALIZATION TECHNIQUES

- **Min-max normalization**:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}(x'_{max} - x'_{min}) + x'_{min}$$

  - Typical cases:

$$x'_{min} = 0, x'_{max} = 1 \Rightarrow x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

$$x'_{min} = -1, x'_{max} = 1 \Rightarrow x' = 2\frac{x - x_{min}}{x_{max} - x_{min}} - 1$$

```
class sklearn.preprocessing.MinMaxScaler(feature_range=(0, 1), *, copy=True, clip=False)
```

- Often, we use quantiles instead of the minimum and maximum for robustness against outliers:

$$x' = \frac{x - x_{0.99}}{x_{0.99} - x_{0.01}}$$

GHENT UNIVERSITY

imec

SCALING THE OUTPUTS

- In regression with MSE loss, highly skewed distributions of the output variable cause models to favor cases with high absolute values
  - It is better to ensure that outputs are centered around zero and symmetric (e.g. by log-scaling)

- In multi-output regression, different variables should have the same scale

- The model is trained on scaled outputs ➔ Un-scale the predictions!

DATA/DIMENSIONALITY REDUCTION

GHENT UNIVERSITY

imec

HIGH-DIMENSIONAL VERSUS LARGE DATA

- High dimensional datasets (# variables is high)
  - More and more high-dimensional data sets are emerging
  - Relevant features are often not known in advance
  - To avoid losing potentially interesting information:
    - Add as many features as possible
    - Let models decide

- Large datasets (# instances is high)

DATA REDUCTION METHODS

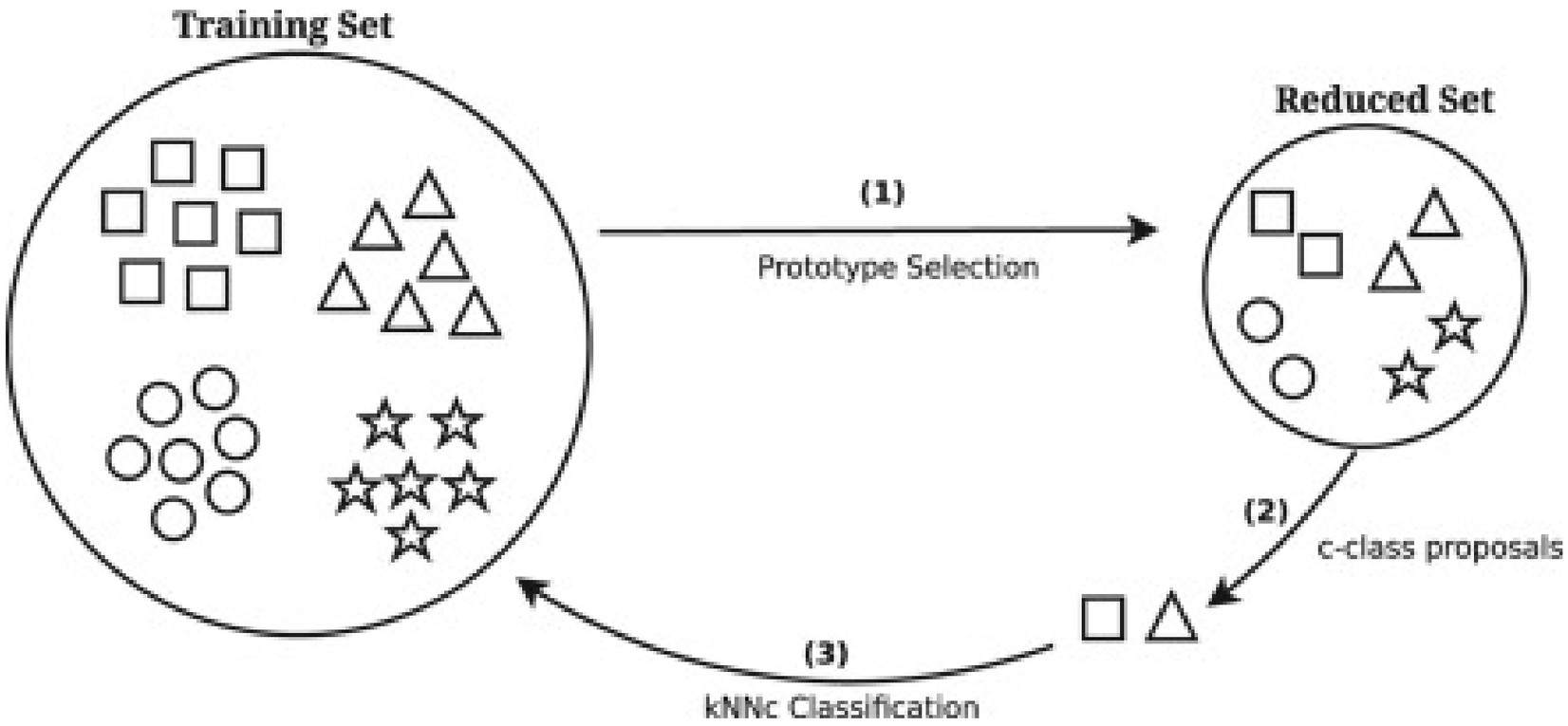
- Data is too big to work with
- Data reduction
  - Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Methods
  - Sampling
  - Prototype selection

SAMPLING METHODS

- Choose a **representative** subset of the data
  - Simple random sampling may have poor performance in the presence of skew.

- Develop adaptive sampling methods
  - Stratified sampling:
    - Approximate the percentage of each class (or subpopulation of interest) in the overall database
    - Used in conjunction with skewed data
  - Clustering:
    - Find clusters of similar data
    - Take a random sample or a centroid from every cluster

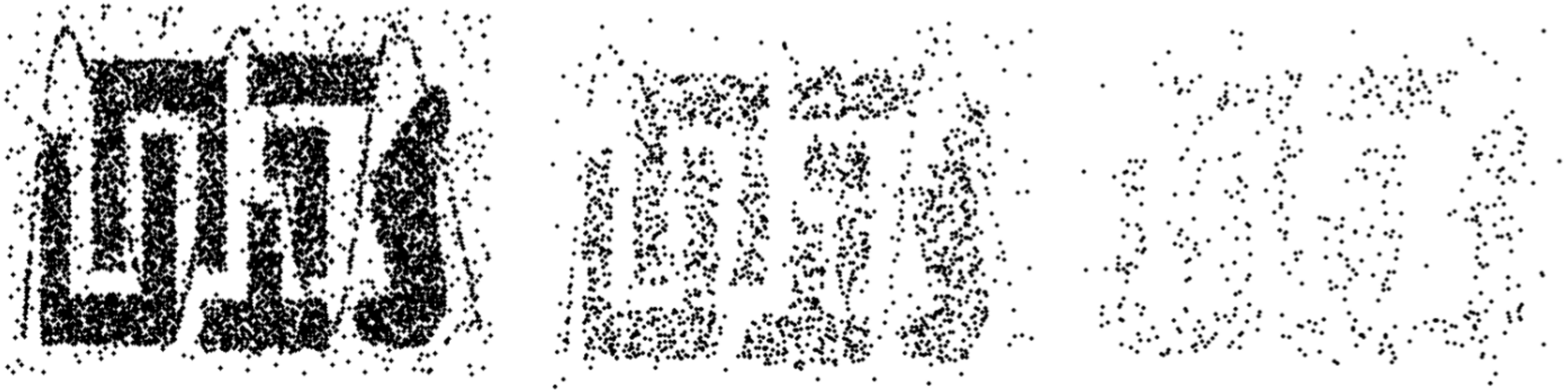
EXAMPLE: PROTOTYPE SELECTION



GHENT UNIVERSITY

imec

EFFECT OF SAMPLE SIZE WHEN REDUCING



8000 points

2000 Points

500 Points

GHENT UNIVERSITY

imec

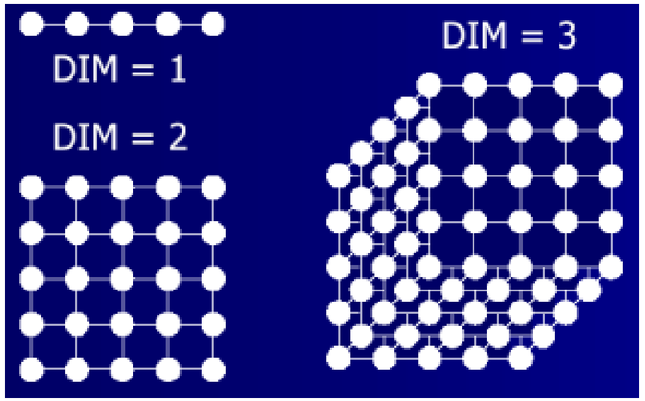
CURSE OF DIMENSIONALITY

[Bellman, 1961]

> "Sample size needed to estimate a function of several variables to a given degree of accuracy grows exponentially with the number of variables"

Example:

- sample a unit interval with no more than 0.01 distance between points:
  100 samples
- equivalent sampling of a 10-dimensional unit hypercube:
  100^{10} samples



“SMALL N LARGE P PROBLEMS”

- Problems with more features than instances ($N \ll p$)
- Also sometimes referred to as “wide data”

- Typically we want $N \gg p$ to perform accurate parameter estimation

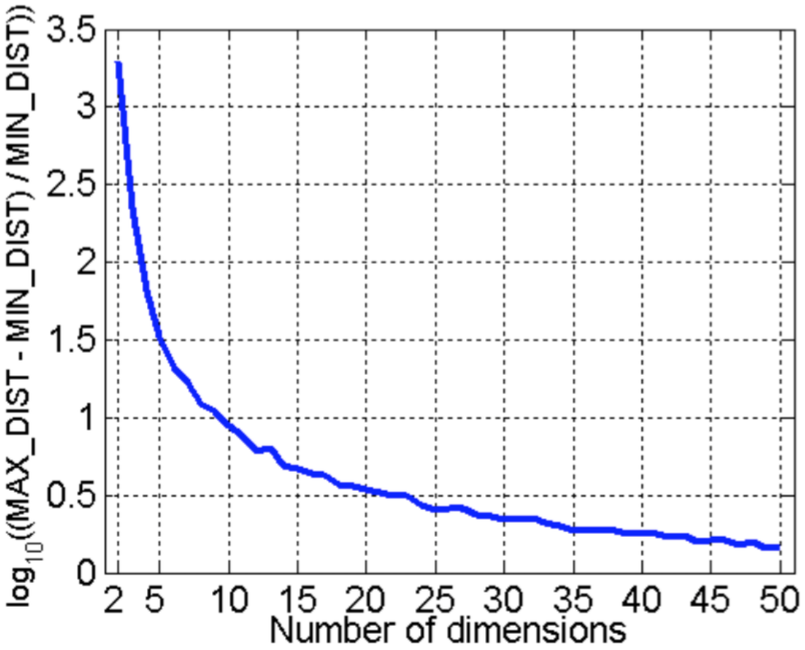
- Dimensionality reduction is key to avoid overfitting

SIMILARITY FADES AWAY IN HIGH DIMENSIONS

- When dimensionality increases, data becomes increasingly sparse in the space that it occupies
- Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful
- Things look more similar on average when more features are used to describe them

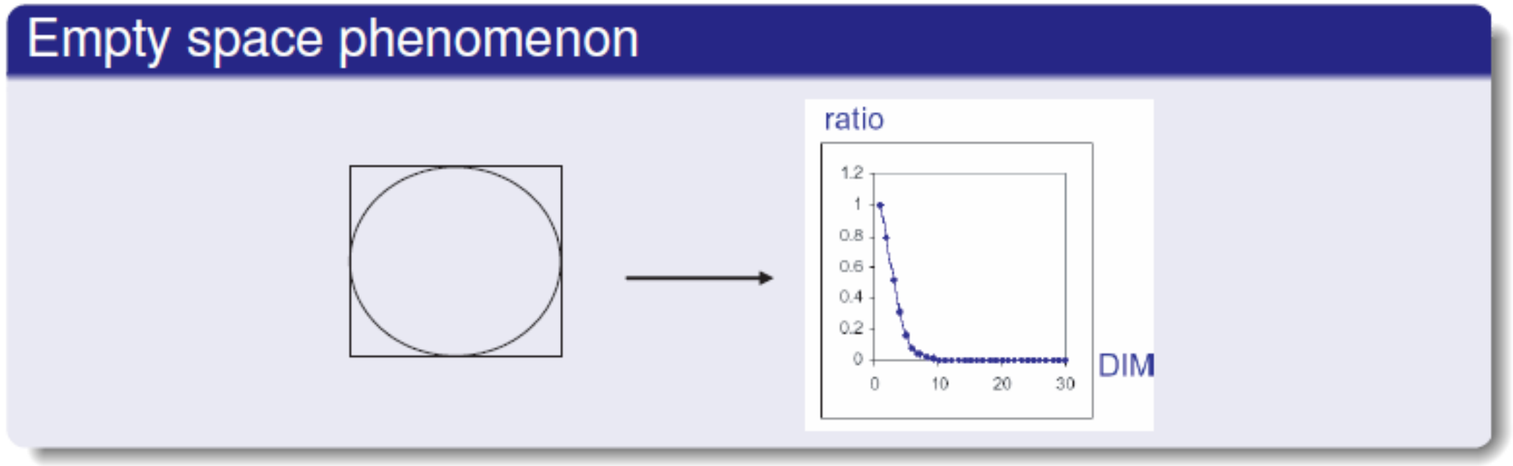
EXAMPLE: CURSE OF DIMENSIONALITY

- Randomly generate 500 points
- Compute difference between max and min distance between any pair of points



THE EMPTY SPACE PHENOMENON

- Empty space phenomenon: “High dimensional spaces are inherently sparse”
  - Counter-intuitively, corners (tails) are much more important than centers



DIMENSIONALITY REDUCTION

- Not all measured variables (features) are important for understanding the underlying phenomena

- Many methods are not designed to deal with HD datasets
  - Eliminate **irrelevant** features
  - Eliminate **redundant** features

- Motivation for many domains where the sample per feature ratio is low

ADVANTAGES OF DIMENSIONALITY REDUCTION

- Improves model performance
  - Classification: accuracy, AUC…
  - Clustering: sum of squares, silhouette index…
  - Regression: MSE…
- Faster and more cost-effective models
- Improves generalization (avoids overfitting)
- Gain deeper insight into the processes that generated the data

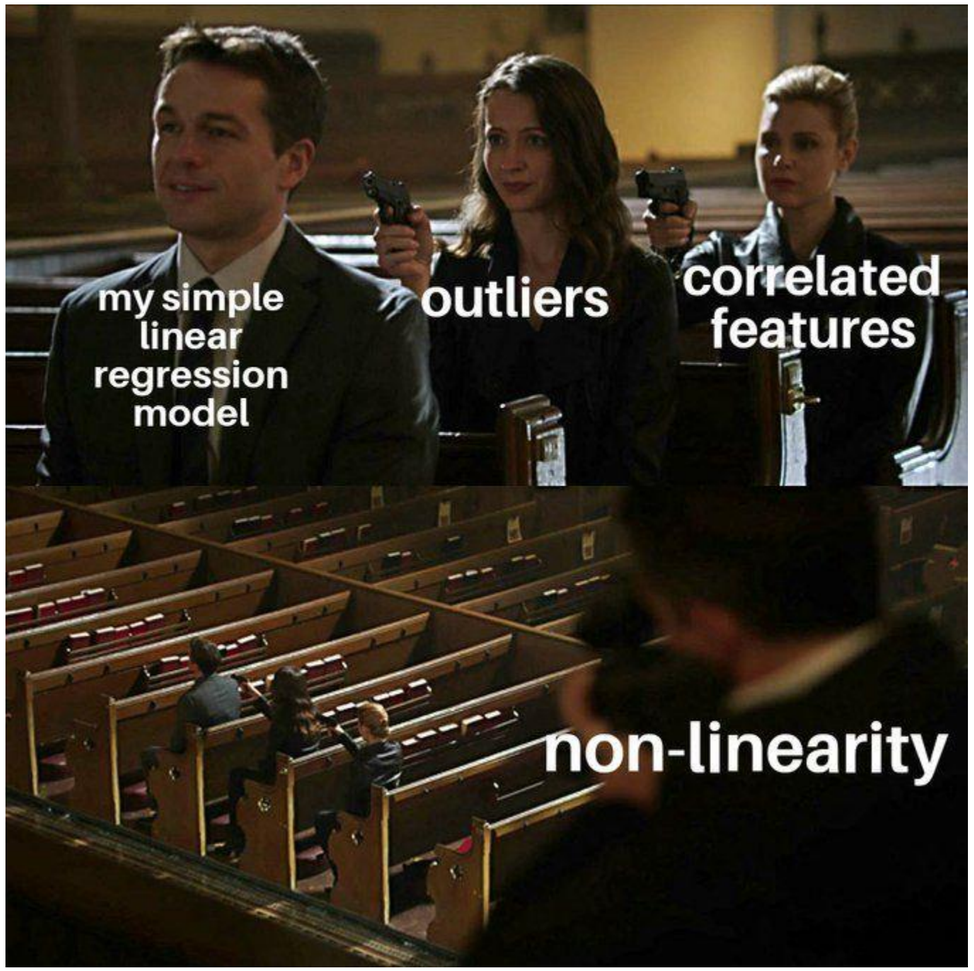
TYPES OF DR TECHNIQUES

- **Feature selection:**
  - **Filter**:
    - Compute a score for each feature
    - Discard features with low scores
  - **Wrapper**:
    - For different subsets of features, train a model
    - Keep the features whose model give the best results
  - **Embedded**:
    - Train a model with some degree of interpretability, that allows us to see which features are relevant (e.g. decision tree)

- **Feature extraction:** features are transformed into a new space
  - **Linear**
  - **Non-linear**

TYPES OF FEATURE EXTRACTION TECHNIQUES

- Linear vs Non-linear
- Supervised vs Non-supervised



OVERVIEW OF DR TECHNIQUES

Dimensionality reduction

Feature selection

	Supervised	Unsupervised
Filter	IG, Correlation, Mutual Information…	Laplacian score,…
Wrapper	Forward, Backward,…	Category utility, EM,…
Embedded	Decision tree, SVM weight,…	Q-α, biclustering,…

Feature transformation/projection/extraction

	Supervised	Unsupervised
Linear	LDA	PCA, LSA,…
Non-linear	Kernel DA, Local FDA,…	MDS, LLE, Isomap, tSNE, UMAP…

GHENT UNIVERSITY

imec

FORWARD / BACKWARD METHODS

- Greedy methods
- Suppose the original set of features is P
  - Recursive Feature Elimination (RFE)
    1. Selected features: $S = P$
    2. While the performance is improved:
       1. For each feature $i \in S$
          1. Train a model with features $S \setminus \{\mathrm{i}\}$
          2. Compute the performance m_i
       2. Remove the feature i that maximizes m_i: $S = S \setminus \{\mathrm{i}\}$

  - Sequential Feature Selector
    1. Selected features: $\emptyset$
    2. While the performance is improved:
       1. For each feature $i \in P \setminus S$
          1. Train a model with features $S \cup \{\mathrm{i}\}$
          2. Compute the performance m_i
       2. Select the feature i that maximizes m_i: $S = S \cup \{\mathrm{i}\}$

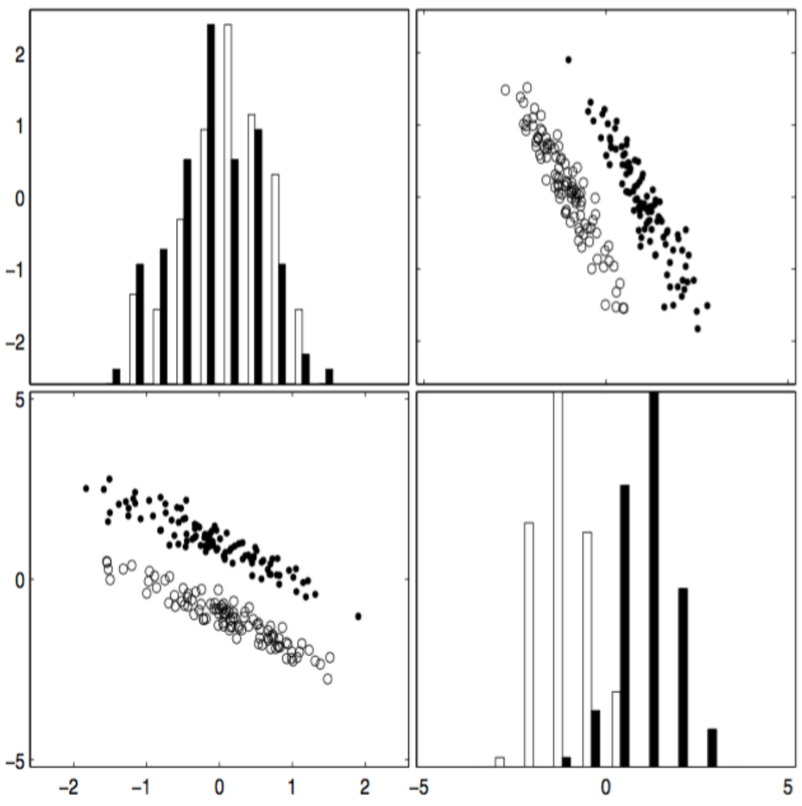
- Typically, the performance m_i is measured with a validation set!

GHENT UNIVERSITY

imec

SURPRISING FACTS ON VARIABLE COMBINATIONS

A variable useless by itself can be useful together with others

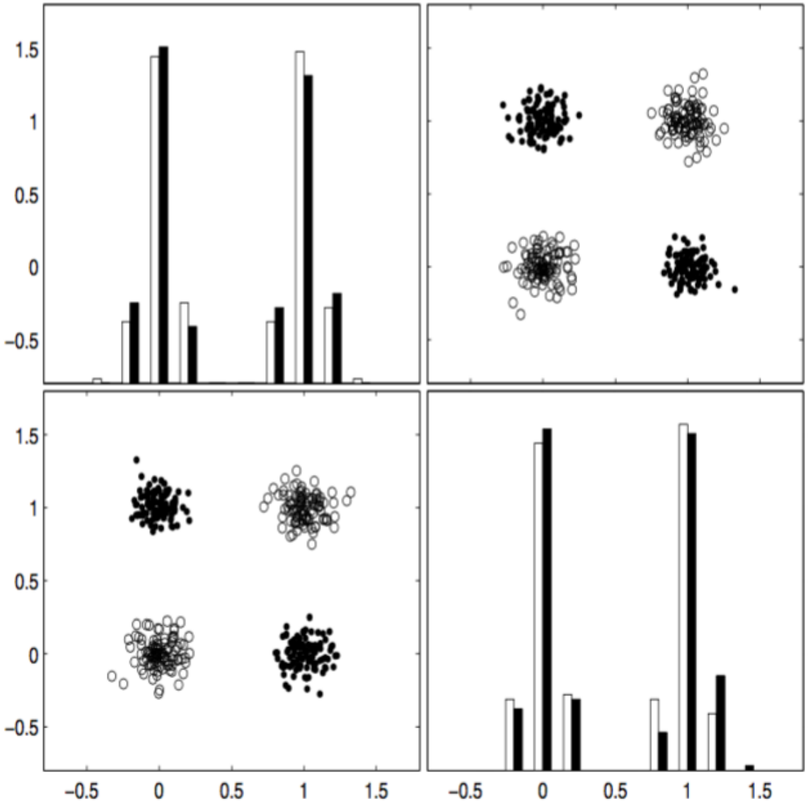


One variable has completely overlapping class conditional densities. Still, using it jointly with the other variable improves class separability compared to using the other variable alone.

GHENT UNIVERSITY

SURPRISING FACTS ON VARIABLE COMBINATIONS

Two variables useless on their own can be useful when combined

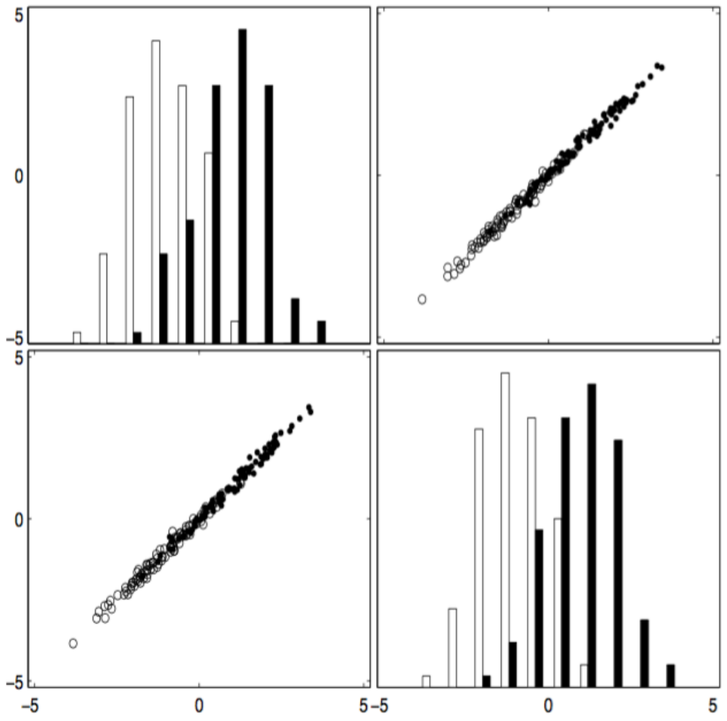


XOR-like or chessboard- like problems. The classes consist of disjoint clumps such that in projection on the axes the class conditional densities overlap perfectly. Therefore, individual variables have no separation power. Still, taken together, the variables provide good class separability .

GHENT UNIVERSITY

SURPRISING FACTS ON VARIABLE COMBINATIONS

Perfectly correlated variables are truly redundant in the sense that no additional information is gained by adding them.

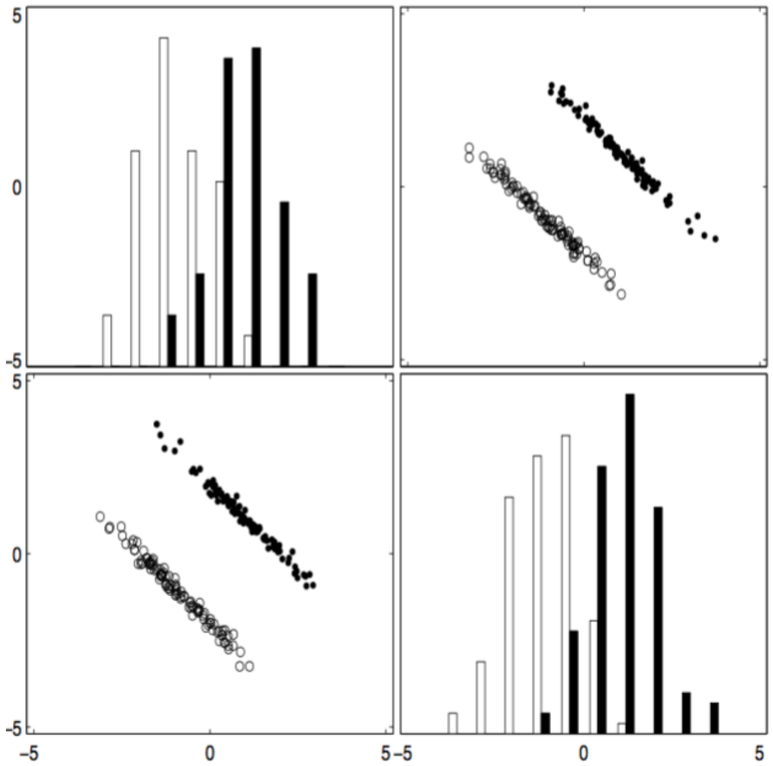


- The class conditional distributions have a high covariance in the direction of the line of the two class centers. There is no significant gain in separation by using two variables instead of just one.

GHENT UNIVERSITY

SURPRISING FACTS ON VARIABLE COMBINATIONS

Very high variable correlation (or anti-correlation) does not mean absence of variable complementarity.

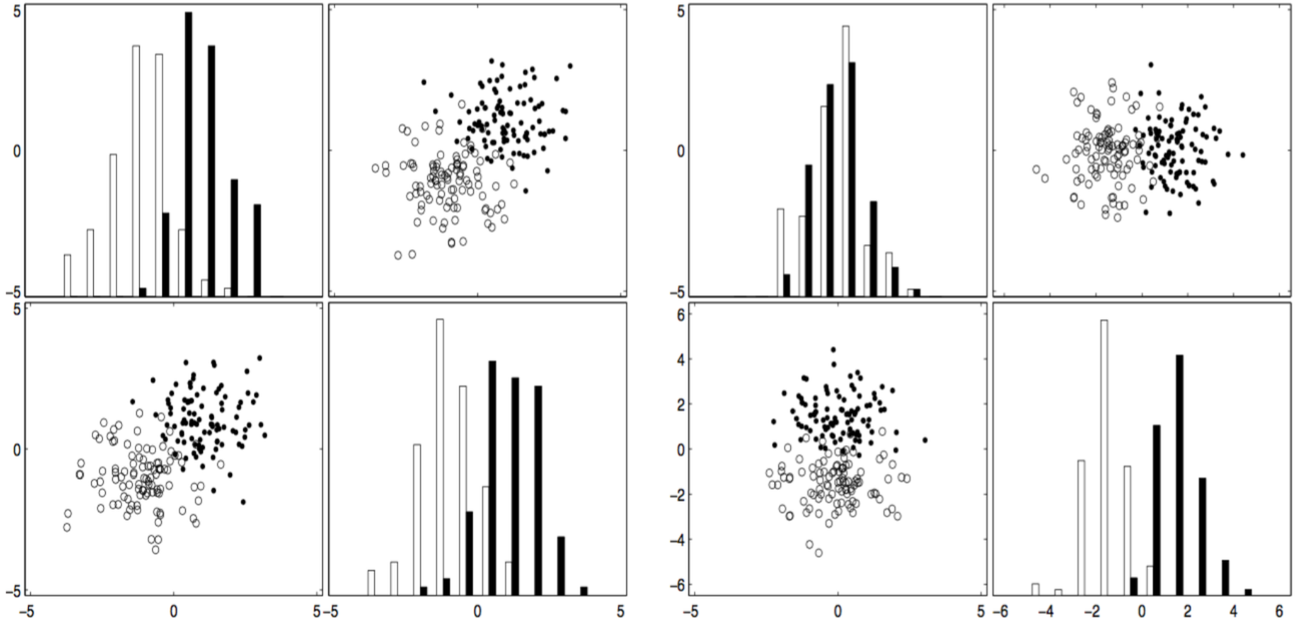


- The class conditional distributions have a high covariance in the direction perpendicular to the line of the two class centers. An important separation gain is obtained by using two variables instead of one.

GHENT UNIVERSITY

SURPRISING FACTS ON VARIABLE COMBINATIONS

Noise reduction and consequently better class separation may be obtained by adding variables that are presumably redundant.



(Left) A two class problem with independently and identically distributed (i.i.d.) variables. Each class has a Gaussian distribution with no covariance. (Right) The same example after a 45 degree rotation showing that a combination of the two variables yields a separation improvement. I.i.d. variables are not truly redundant.

GHENT UNIVERSITY

PRINCIPAL COMPONENT ANALYSIS

PRINCIPAL COMPONENTS

- Principal components of a dataset $\mathbf{X} \in \mathbb{R}^{N \times p}$
  - Directions along which the data are highly variable
  - Lines and subspaces as close as possible to the data cloud
  - A sequence of best linear approximations to that data, of all ranks $q \leq p$

PRINCIPAL COMPONENTS

- Linear model:

$$f(\lambda) = \mu + \mathbf{V}_q \lambda$$

- μ : location vector in $\mathbb{R}^p$
- $\mathbf{V}_q \in \mathbb{R}^{p \times q}$: matrix with q orthogonal unit vectors as columns
- $\lambda \in \mathbb{R}^q$: vector of parameters

- Goal: minimize the reconstruction error:

$$\min_{\mu, \lambda_i, \mathbf{V}_q} \sum_{i=1}^{N} \left\| x_i - \mu - \mathbf{V}_q \lambda_i \right\|^2$$

PRINCIPAL COMPONENTS

- Optimizing μ and λ_i is easy:

$$\hat{\mu} = \bar{x}$$
$$\widehat{\lambda_i} = \mathbf{V}_q^T (x_i - \bar{x})$$

- Now we only need the orthogonal matrix:

$$\min_{\mathbf{V}_q} \sum_{i=1}^{N} \left\| (x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x}) \right\|^2$$

- Assuming $\bar{x} = 0$ for simplicity, this can be solved using the singular value decomposition of matrix $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

- $\mathbf{U}$: $N \times p$ orthogonal matrix of left singular vectors
- $\mathbf{D}$: $p \times p$ diagonal matrix of singular values
- $\mathbf{V}$: $p \times p$ orthogonal matrix of right singular vectors

- In our context:
  - $\mathbf{V}_q$ corresponds to the first q columns of $\mathbf{V}$, for each $q \leq p$
  - The columns of $\mathbf{UD}$ are the principal components of $\mathbf{X}$
  - The variance of the first principal component $\mathbf{z}_j = \mathbf{X}v_1 = \mathbf{u}_1 d_1$ is $\frac{d_1}{\mathrm{N}}$, therefore:
    - $\mathbf{X}v_1$ has the highest variance among all linear combinations of the features
    - $\mathbf{X}v_2$ has the highest variance among all linear combinations of the features, satisfying v_2 being orthogonal to v_1
    - ...

GHENT UNIVERSITY

imec

PRINCIPAL COMPONENTS

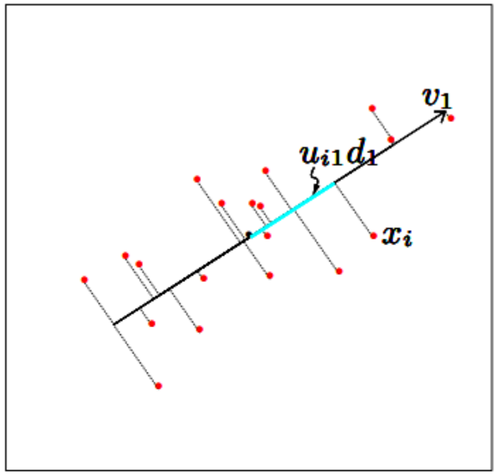


FIGURE 14.20. *The first linear principal component of a set of data. The line minimizes the total squared distance from each point to its orthogonal projection onto the line.*

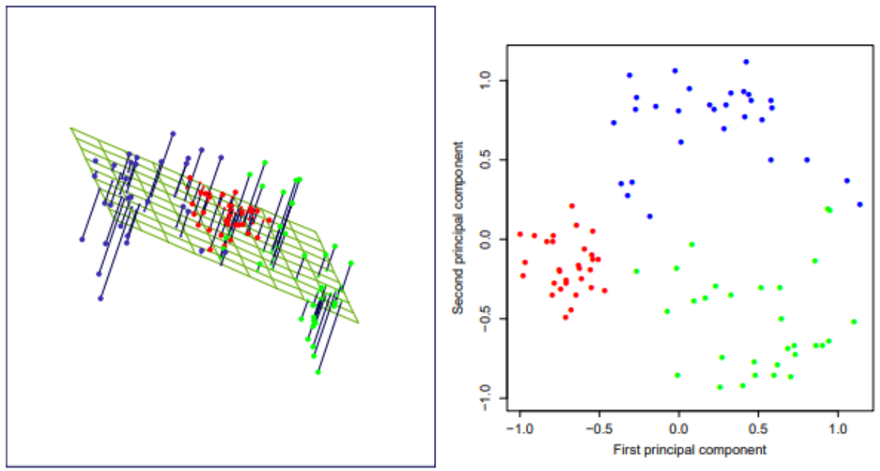


FIGURE 14.21. *The best rank-two linear approximation to the half-sphere data. The right panel shows the projected points with coordinates given by* $\mathbf{U}_2\mathbf{D}_2$, *the first two principal components of the data.*

PRINCIPAL COMPO

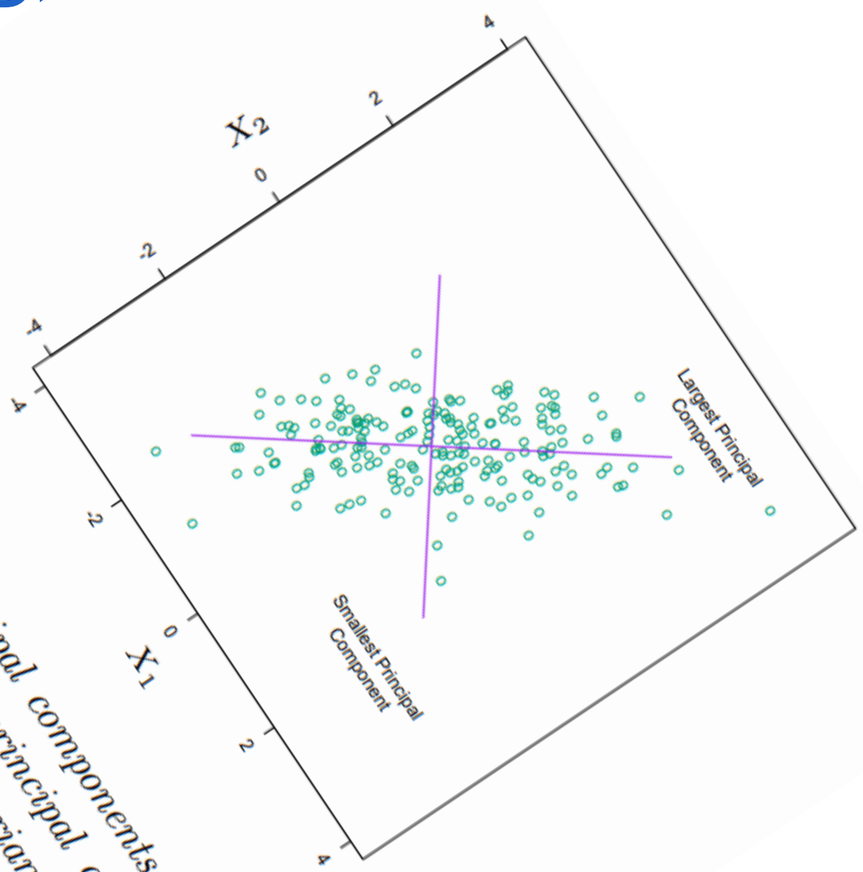
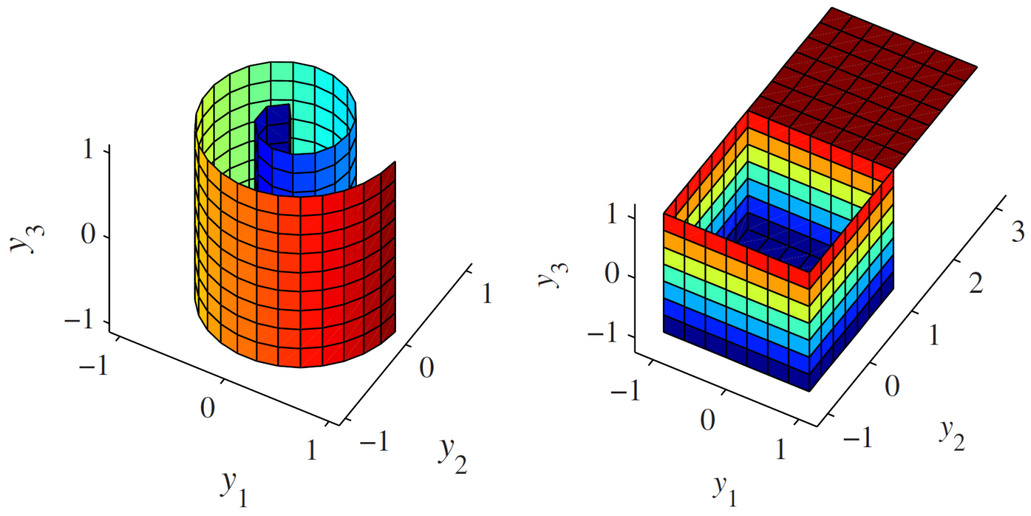


FIGURE 3.9. Principal components of some input data points. The largest principal component is the direction that maximizes the variance of the projected data, and the smallest principal component minimizes that variance. Ridge regression projects $\mathbf{y}$ onto these components, and then shrinks the coefficients of the low-variance components more than the high-variance components.

MANIFOLD LEARNING

MANIFOLDS

- A space of a certain dimensionality, embedded into a higher dimensionality space
- Locally Euclidean: objects look nearly “flat” on small scales.
- Manifold learning / Non-linear dimensionality reduction:
  - Unraveling manifolds



LOCALLY LINEAR EMBEDDING (LLE)

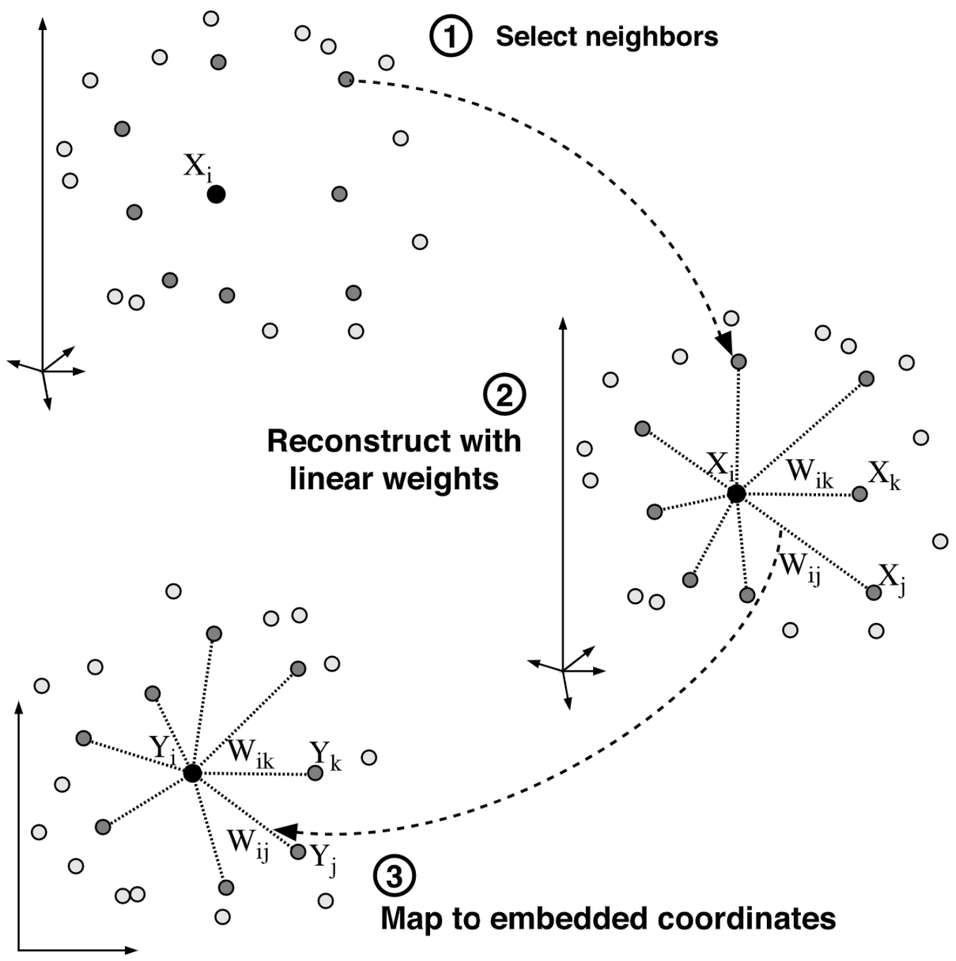
- For each instance X_i, find weights W_{ij} to reconstruct X_i from its nearest neighbors:

$$\epsilon(W) = \sum_i \left| X_i - \sum_j W_{ij} X_j \right|^2$$

- This transformation is rotation/translation invariant
- Find d-dimensional coordinates Y_i that minimize:

$$\Phi(Y) = \sum_i \left| Y_i - \sum_j W_{ij} Y_j \right|^2$$

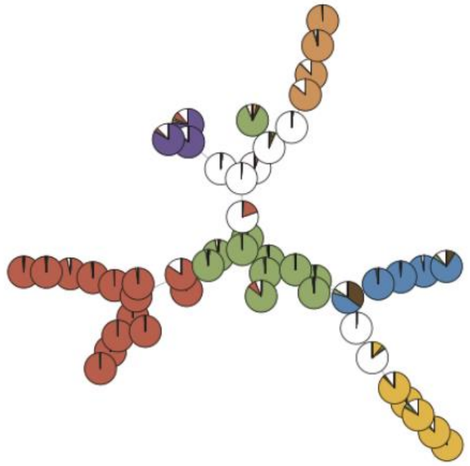
- Solved as a sparse $N \times N$ eigenvalue problem



VISUALIZING HIGH DIMENSIONAL DATA

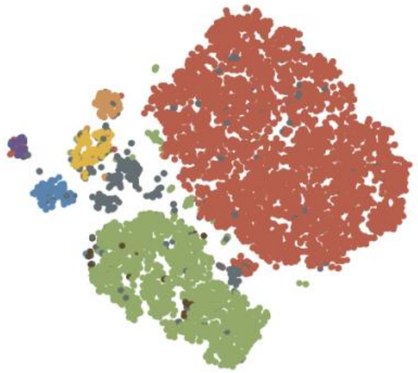
TWO MAIN ALTERNATIVES

- **Clustering**:
  - Visualize only representatives for each cluster



Examples: SOM+MST

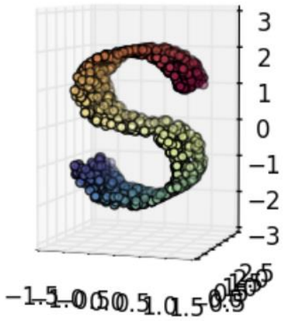
- **Dimensionality reduction**:
  - Find a lower-dimensional projection that maintains the main structure of the data



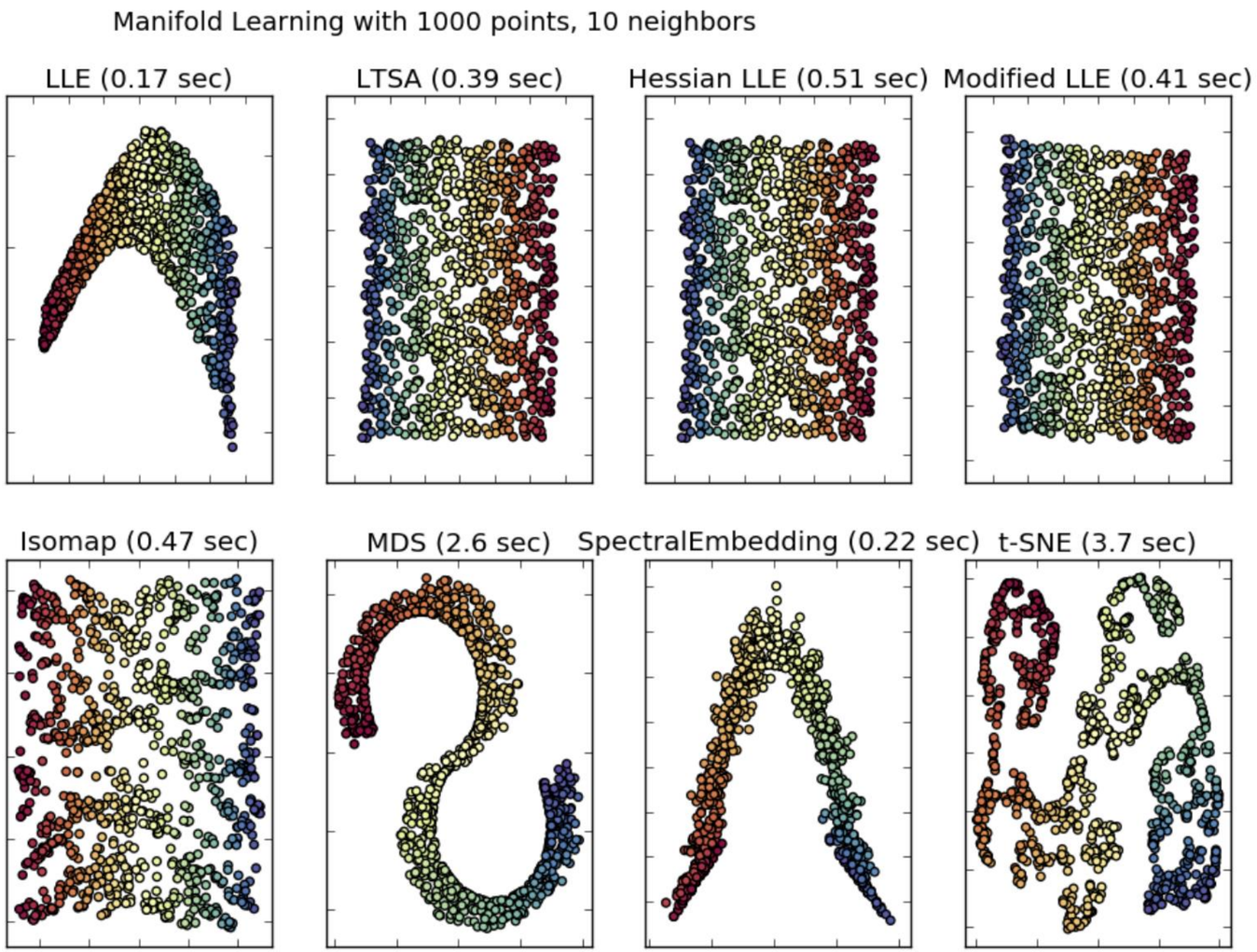
Examples: PCA, MDS, tSNE

DIMENSIONALITY REDUCTION

- PCA:
  - Linear
  - Sensitive to outliers



- Non-linear methods:



GHENT UNIVERSITY

imec

UMAP

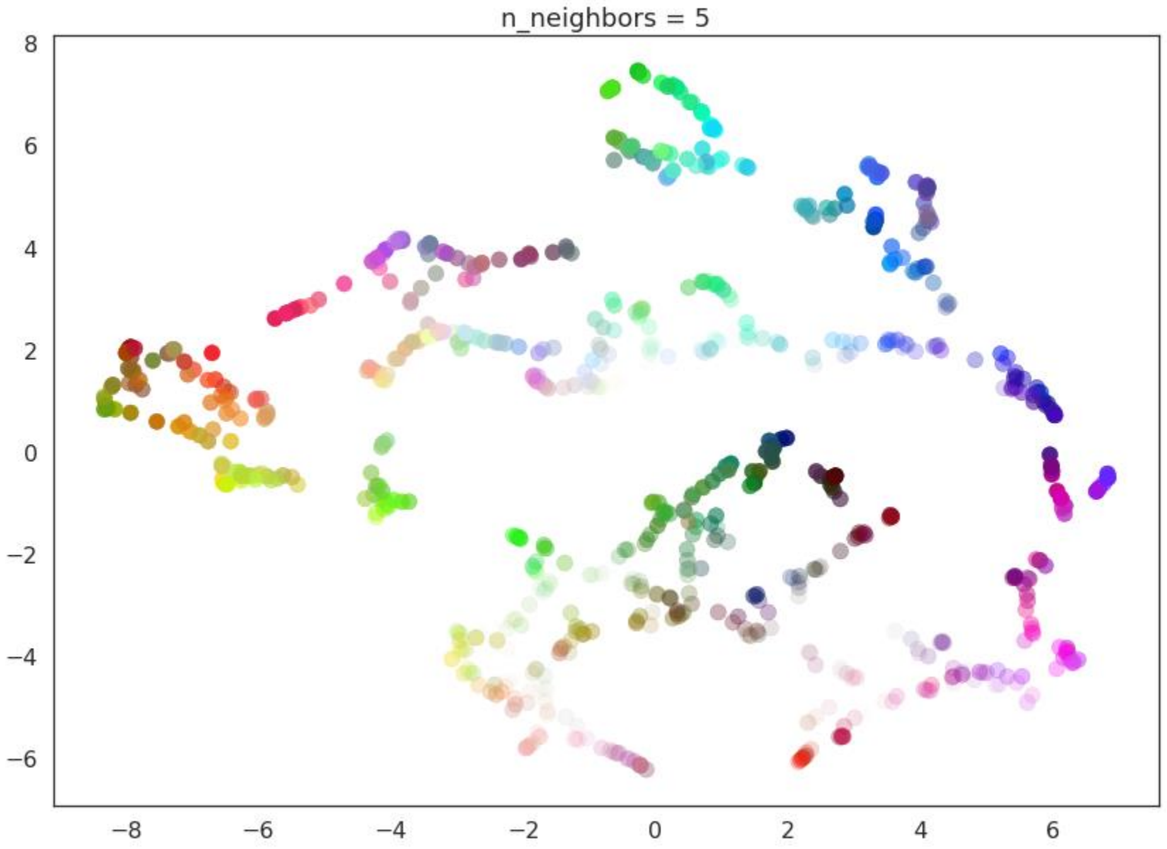
- Constructs a high dimensional graph representation of the data
- Optimizes a low-dimensional graph to be as structurally similar as possible

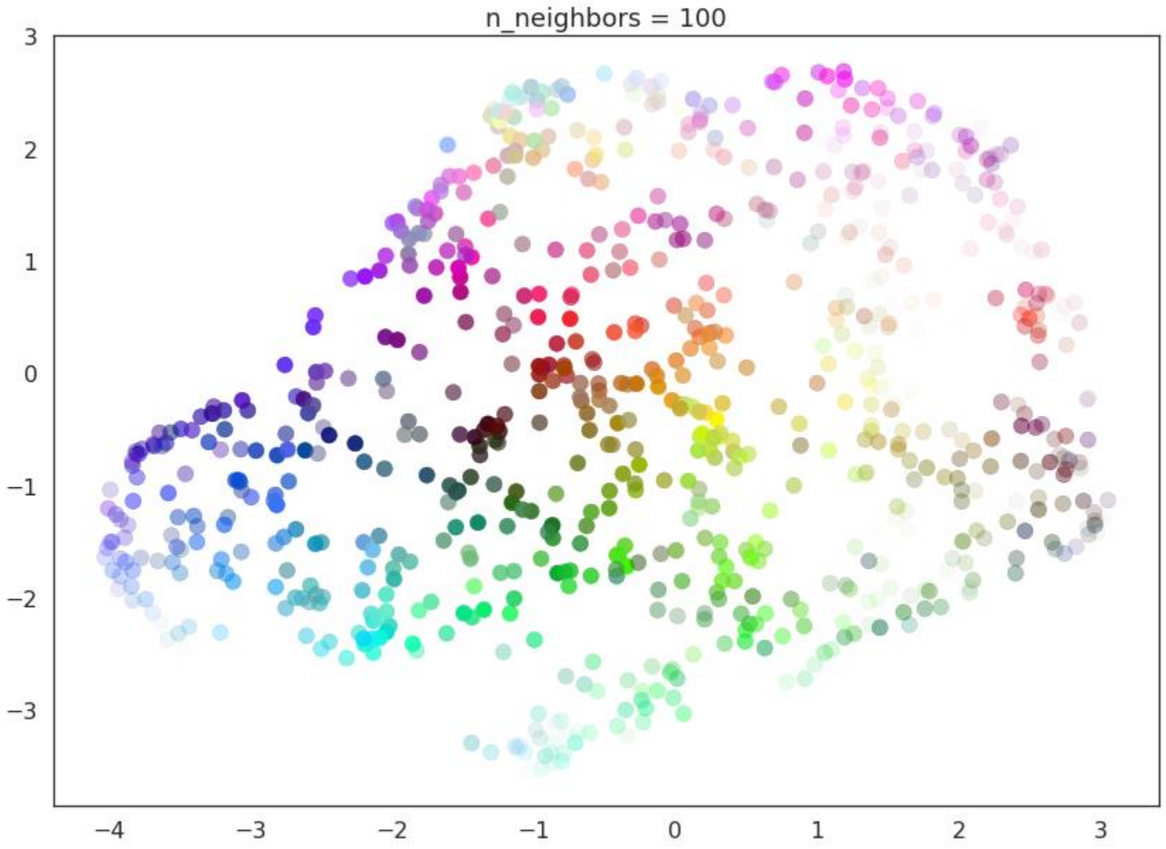
- Hyperparameters:
  - Number of neighbors when constructing the graph
  - Minimum distance between points in the low-dimensional representation
  - Number of dimensions of the output
    - For visualization: 2 or 3
    - For clustering or other algorithms: higher
  - Distance metric

- Python package: https://umap-learn.readthedocs.io/en/latest/
- Nice interface to try it out: https://pair-code.github.io/understanding-umap/

UMAP

– Number of neighbors



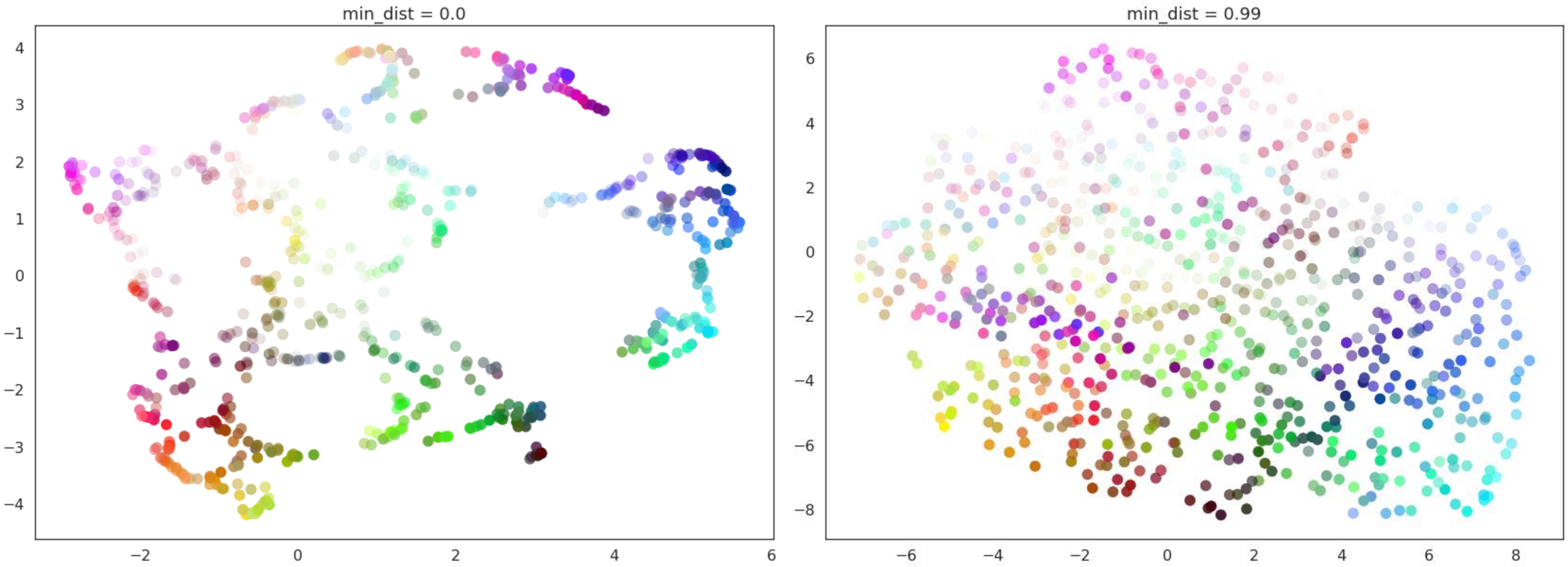


GHENT UNIVERSITY

imec

UMAP

– Minimum distance

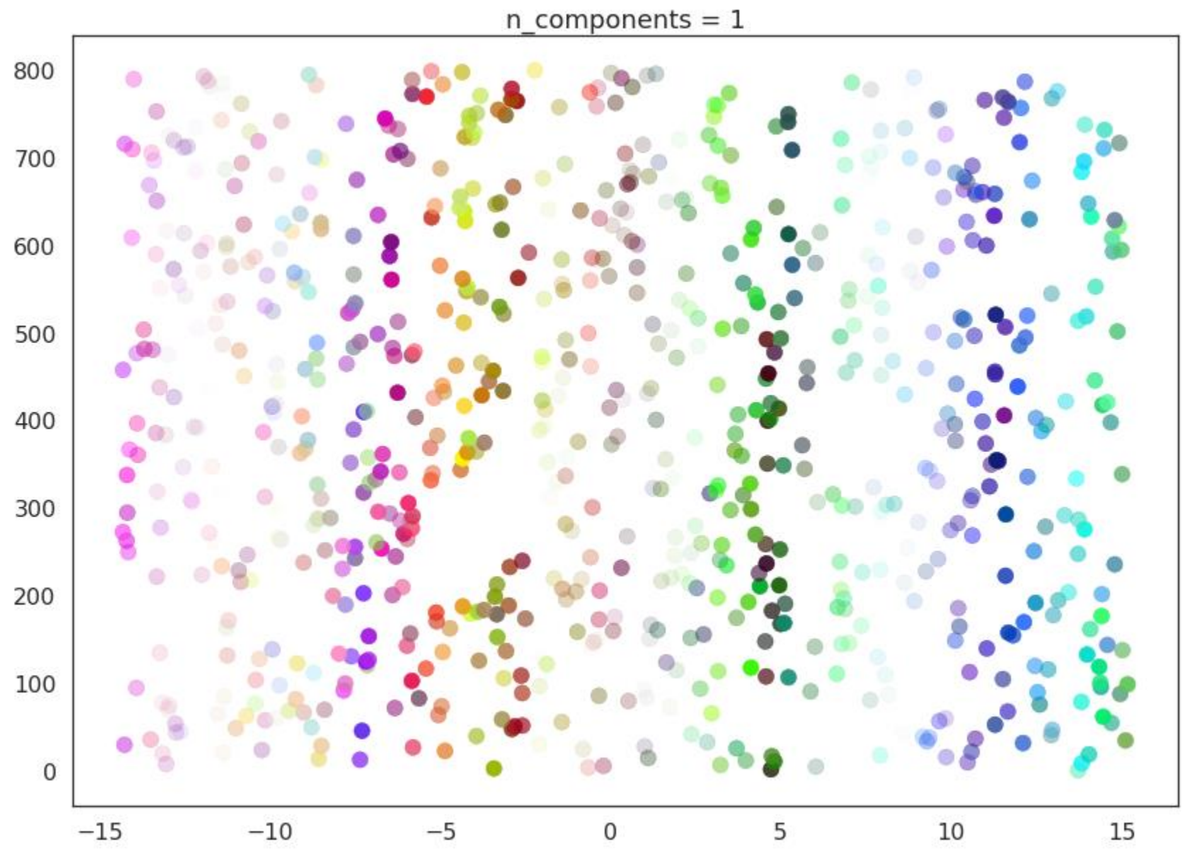


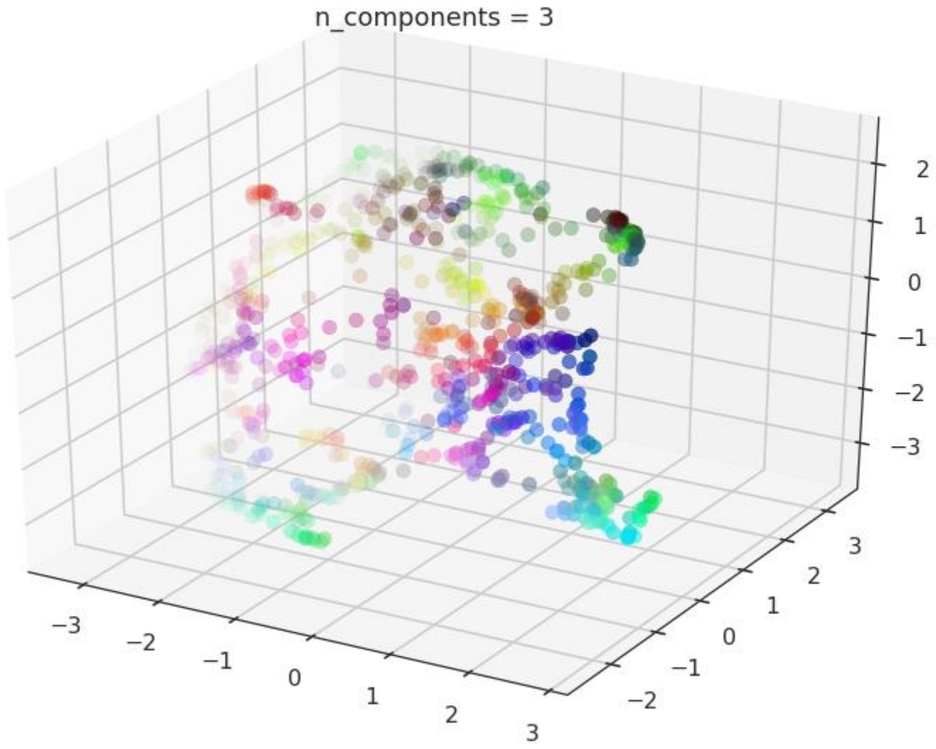
GHENT UNIVERSITY

imec

UMAP

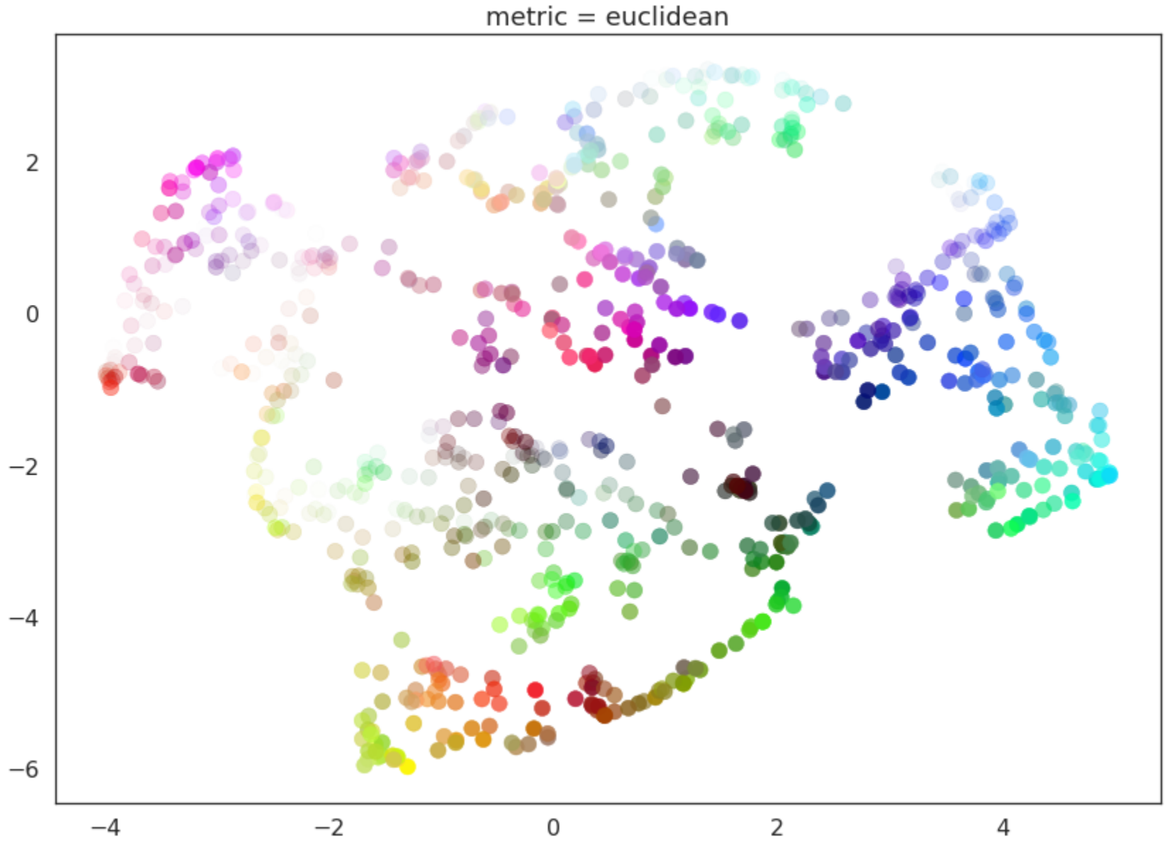
– Number of dimensions

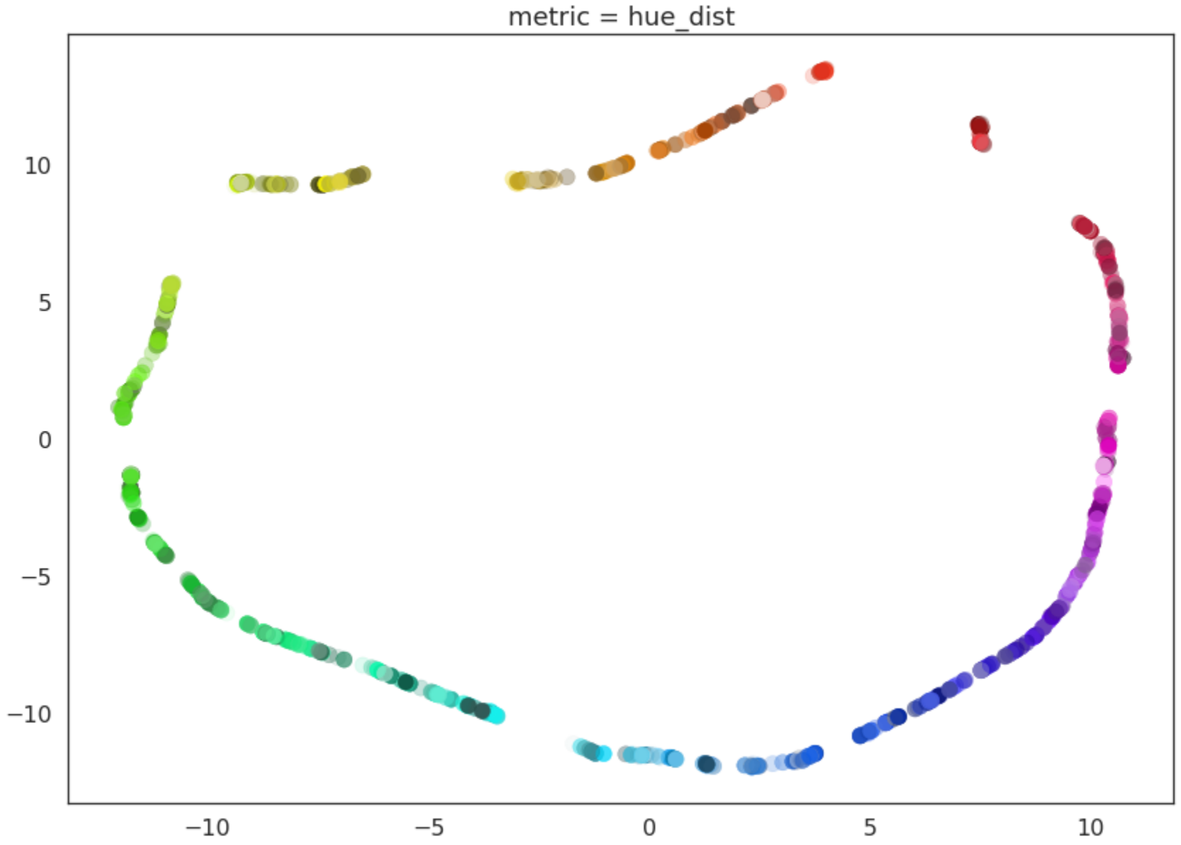




UMAP

– Distance metric





FACULTY OF ENGINEERING
AND ARCHITECTURE

Daniel Peralta

Doctor Assistant – Postdoctoral Researcher

DEPARTMENT OF INFORMATION
TECHNOLOGY - IDLAB

E daniel.peralta@ugent.be

Universiteit Gent
@ugent
@ugent
Ghent University

www.ugent.be

GHENT
UNIVERSITY

imec