IIC 2433 Minería de Datos

https://github.com/marcelomendoza/IIC2433

- OUTLINE -

¿Qué vamos a ver?

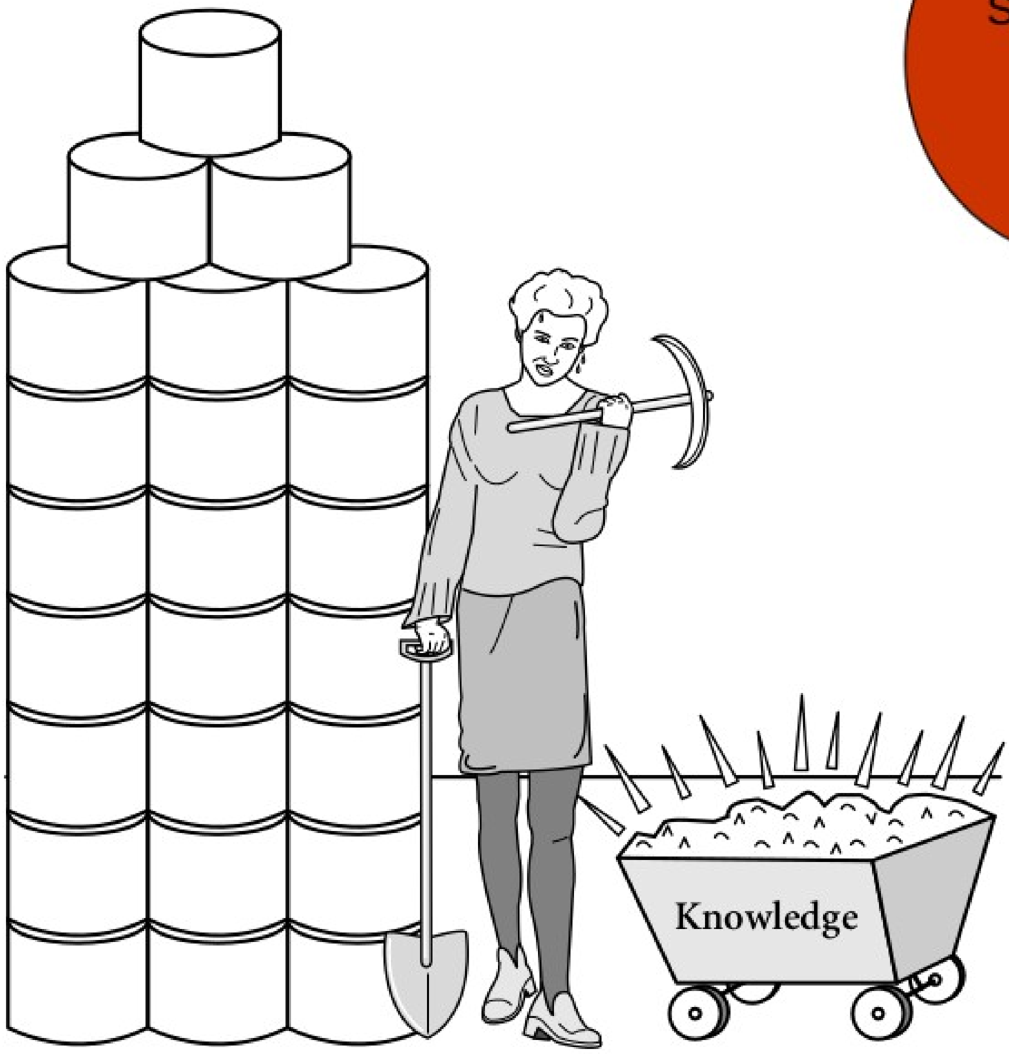
PCA, t-SNE

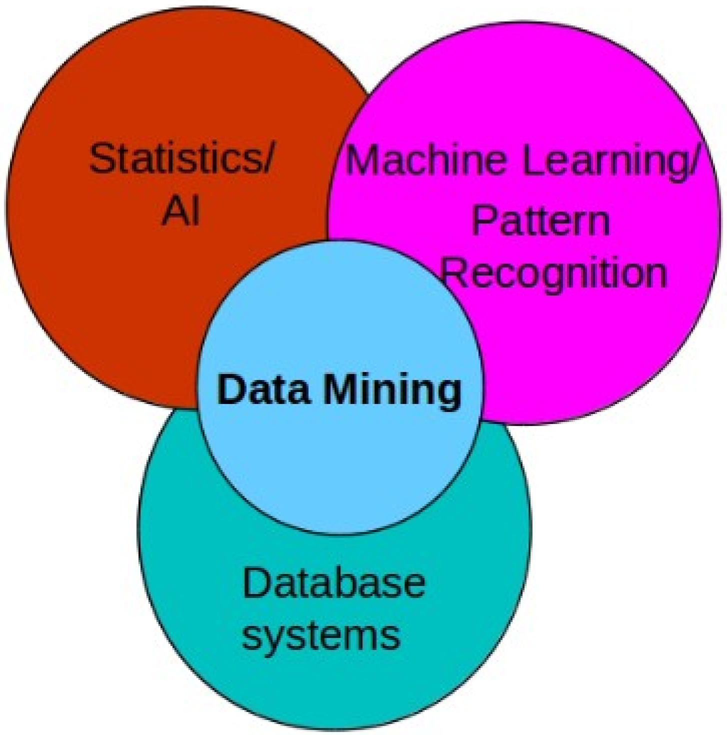
k-means, HAC, DBSCAN, Louvain

Perceptron, SVM, ensembles

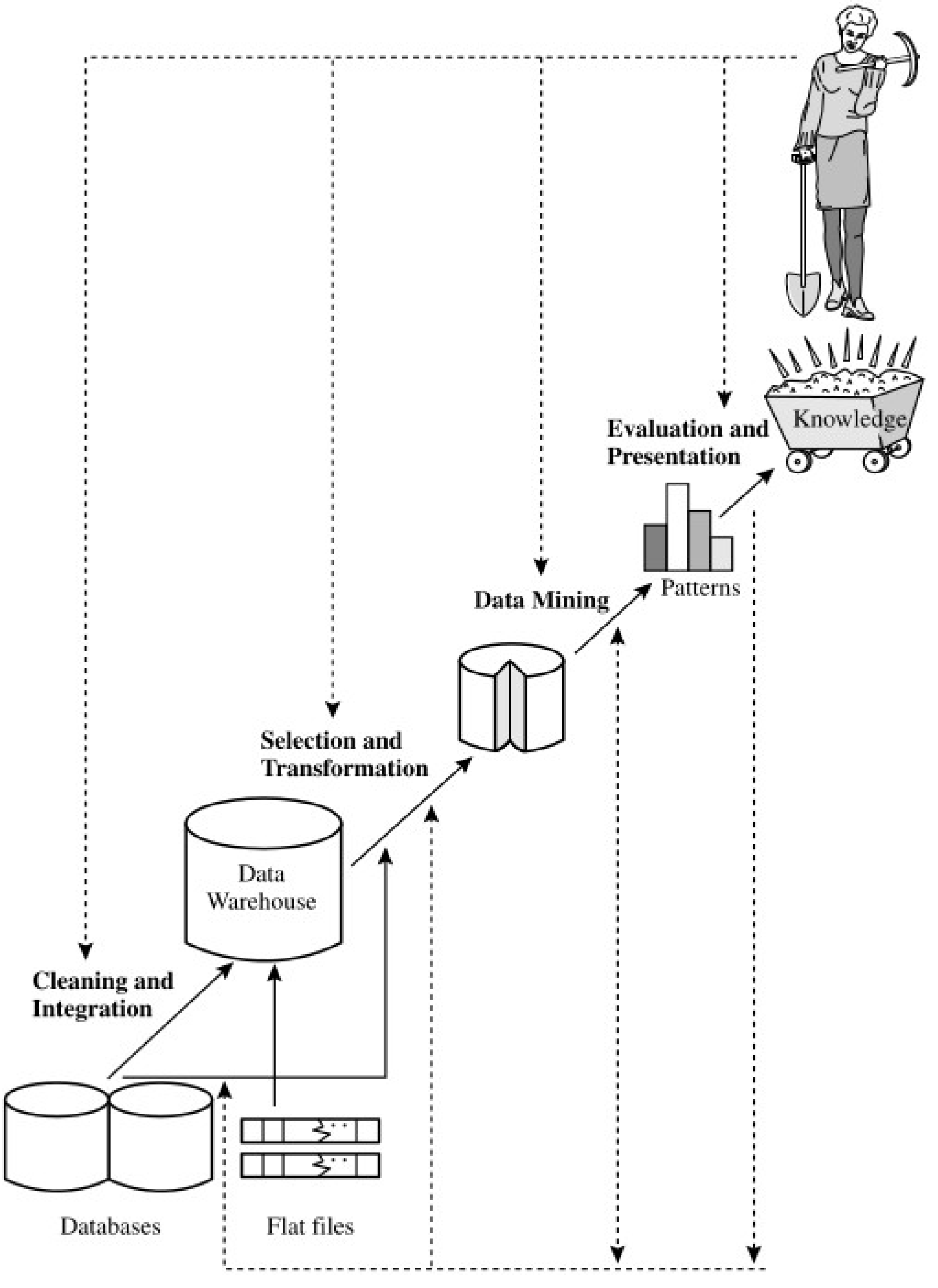
AE. VAE

Data Mining


Knowledge

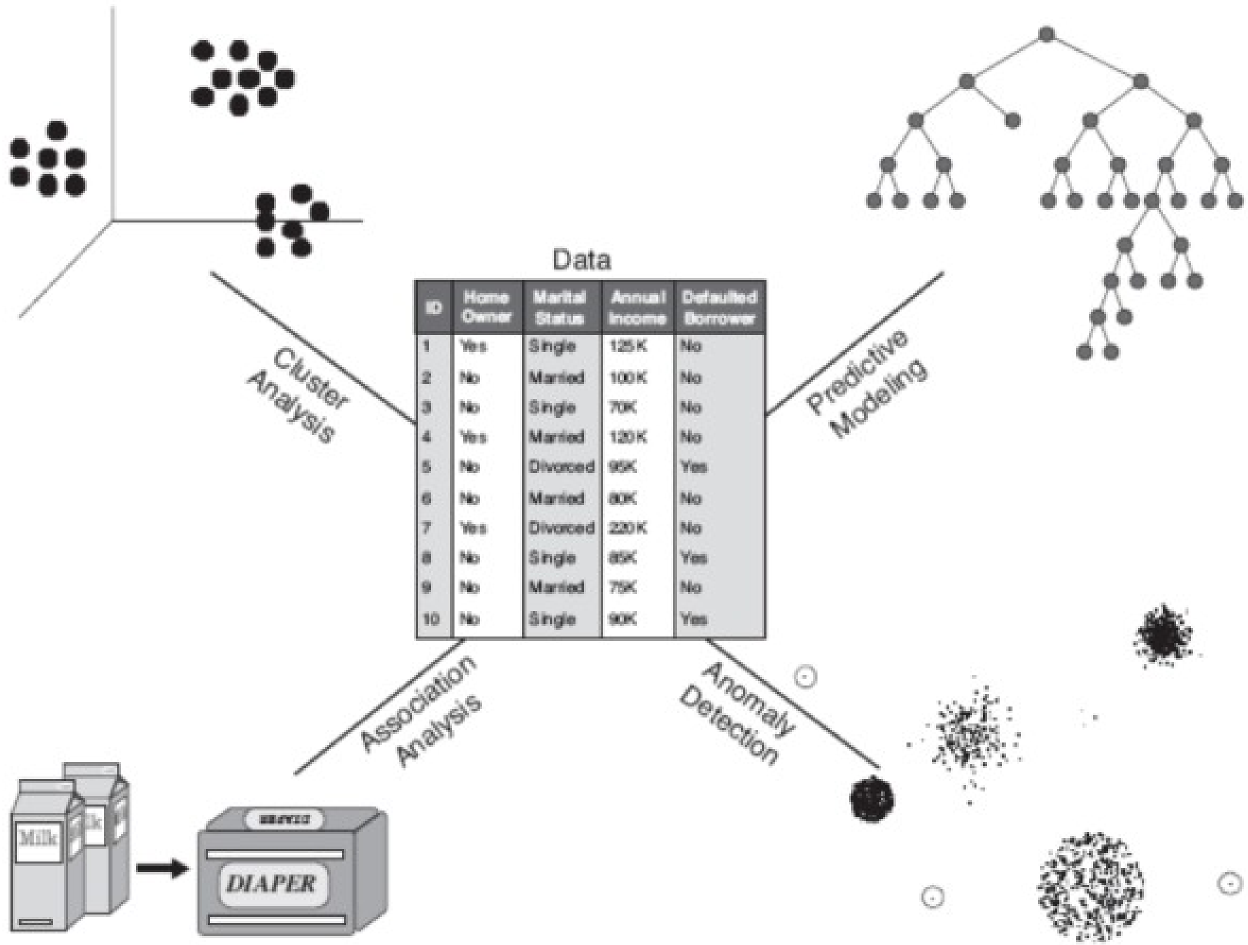

Statistics/
AI
Machine Learning/
Pattern
Recognition
Data Mining
Database
systems

Data Mining



Knowledge
Evaluation and
Presentation
Patterns
Data Mining
Selection and
Transformation
Data
Warehouse
Cleaning and
Integration
Databases
Flat files

Data Mining



ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Preprocesamiento de datos -

Fuentes de Datos

Tid	Refund	Marital Status	Taxable Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

(a) Record data.

TID	ITEMS
1	Bread, Soda, Milk
2	Beer, Bread
3	Beer, Soda, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Soda, Diaper, Milk

(b) Transaction data.

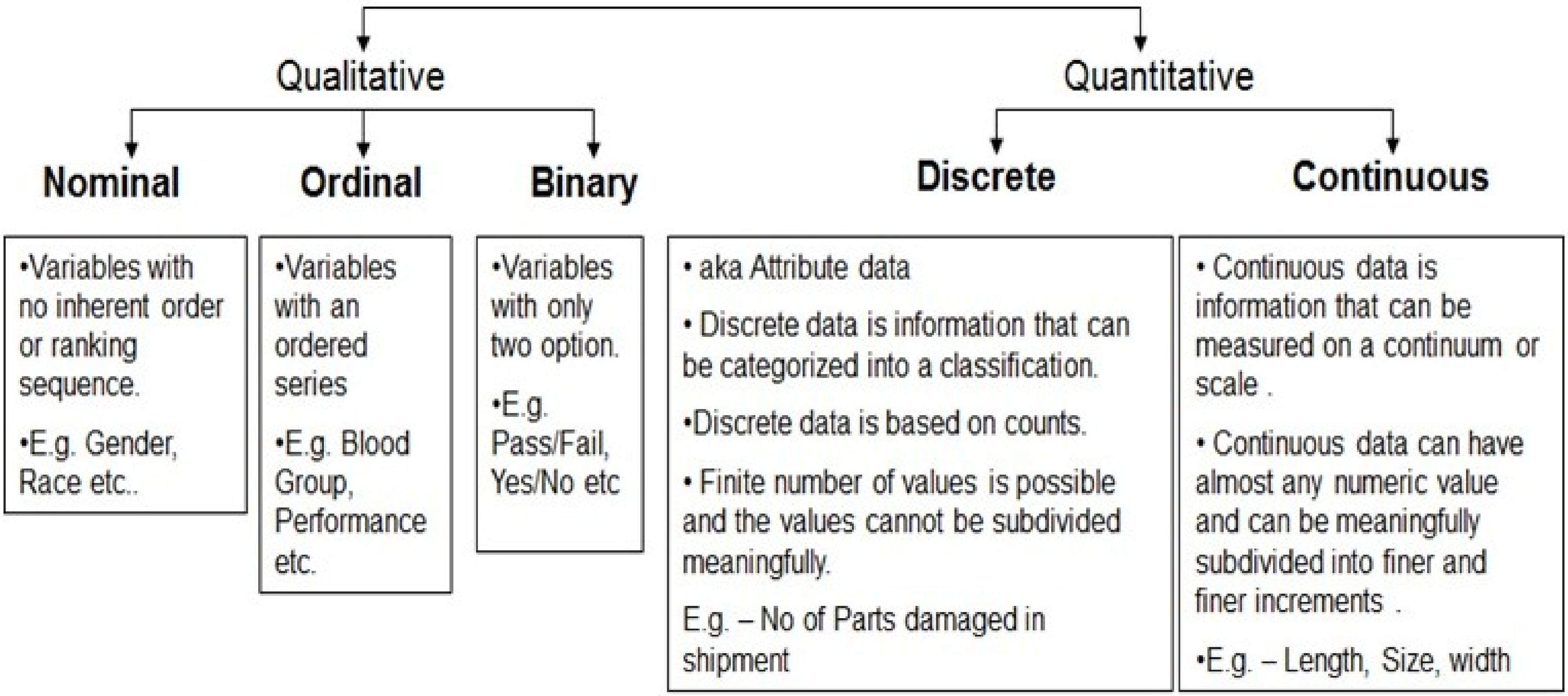
Projection of x Load	Projection of y Load	Distance	Load	Thickness
10.23	5.27	15.22	27	1.2
12.65	6.25	16.22	22	1.1
13.54	7.23	17.34	23	1.2
14.27	8.43	18.45	25	0.9

(c) Data matrix.

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

(d) Document-term matrix.

Tipos de características



Características cuantitativas

Normalización:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \longrightarrow [0, 1]$$

$$X_{m-norm} = \frac{X - \mu}{X_{max} - X_{min}} \longrightarrow \text{¿Intervalo?}$$

Características cualitativas

Codificación:

One-hot encoding

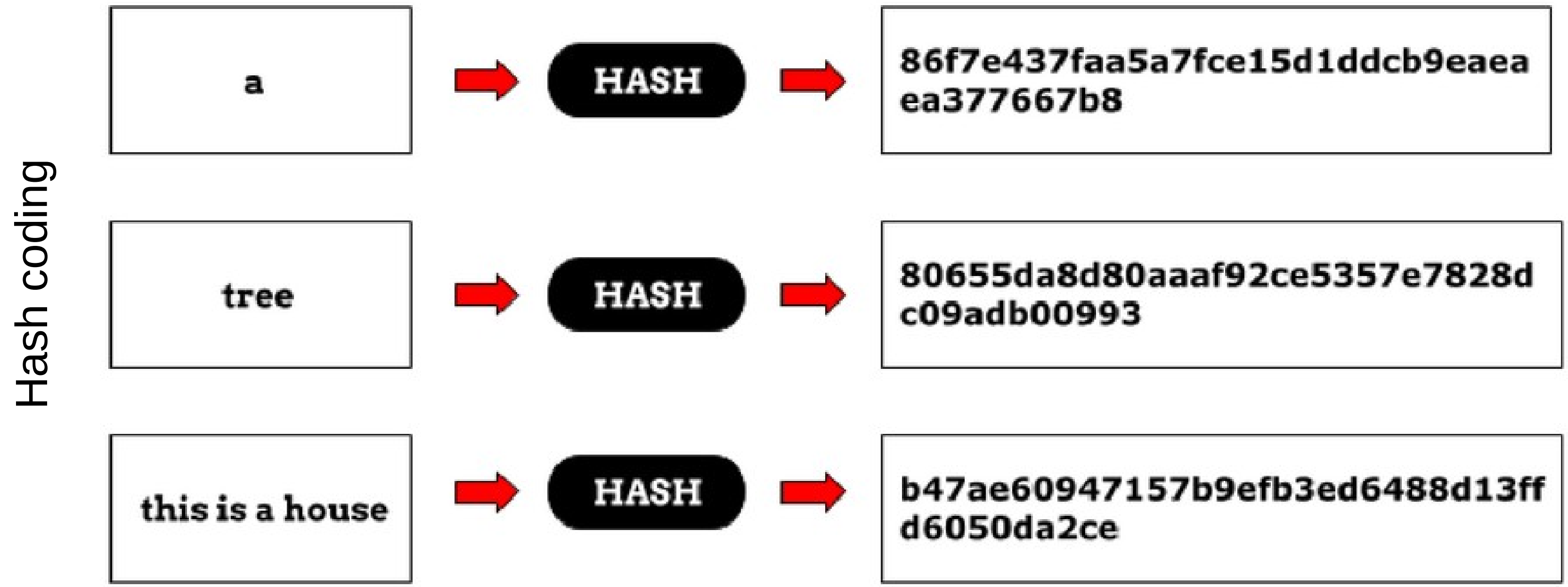
Index	Animal
0	Dog
1	Cat
2	Sheep
3	Horse
4	Lion

One-Hot code

Index	Dog	Cat	Sheep	Lion	Horse
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	0	1
4	0	0	0	1	0

Características cualitativas

Codificación:



Vectores y características

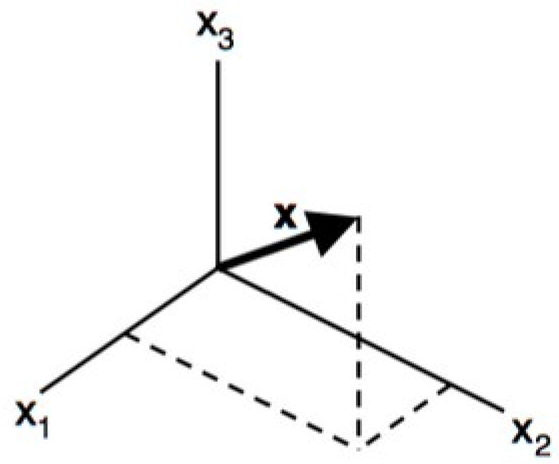
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \\ x_d \end{bmatrix}$$

Feature vector

Vectores y características

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \\ X_d \end{bmatrix}$$

Feature vector

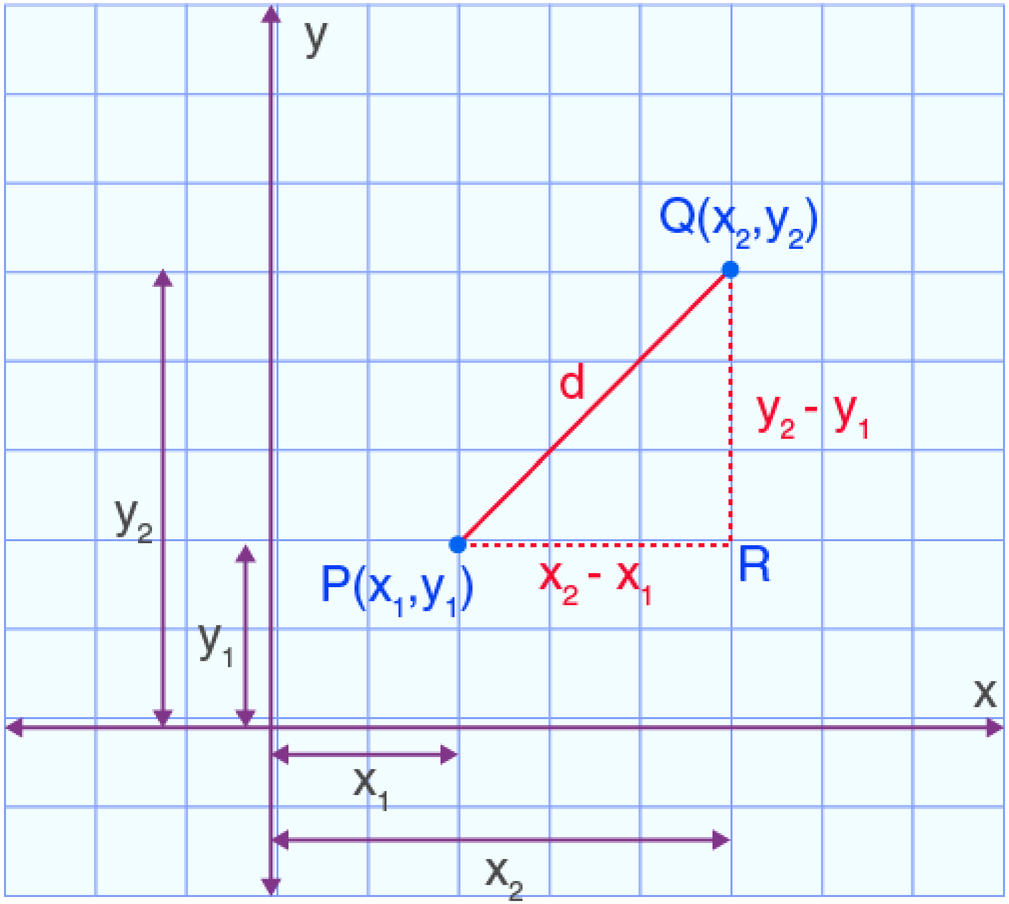


Feature space (3D)

- Distancia y proximidad -

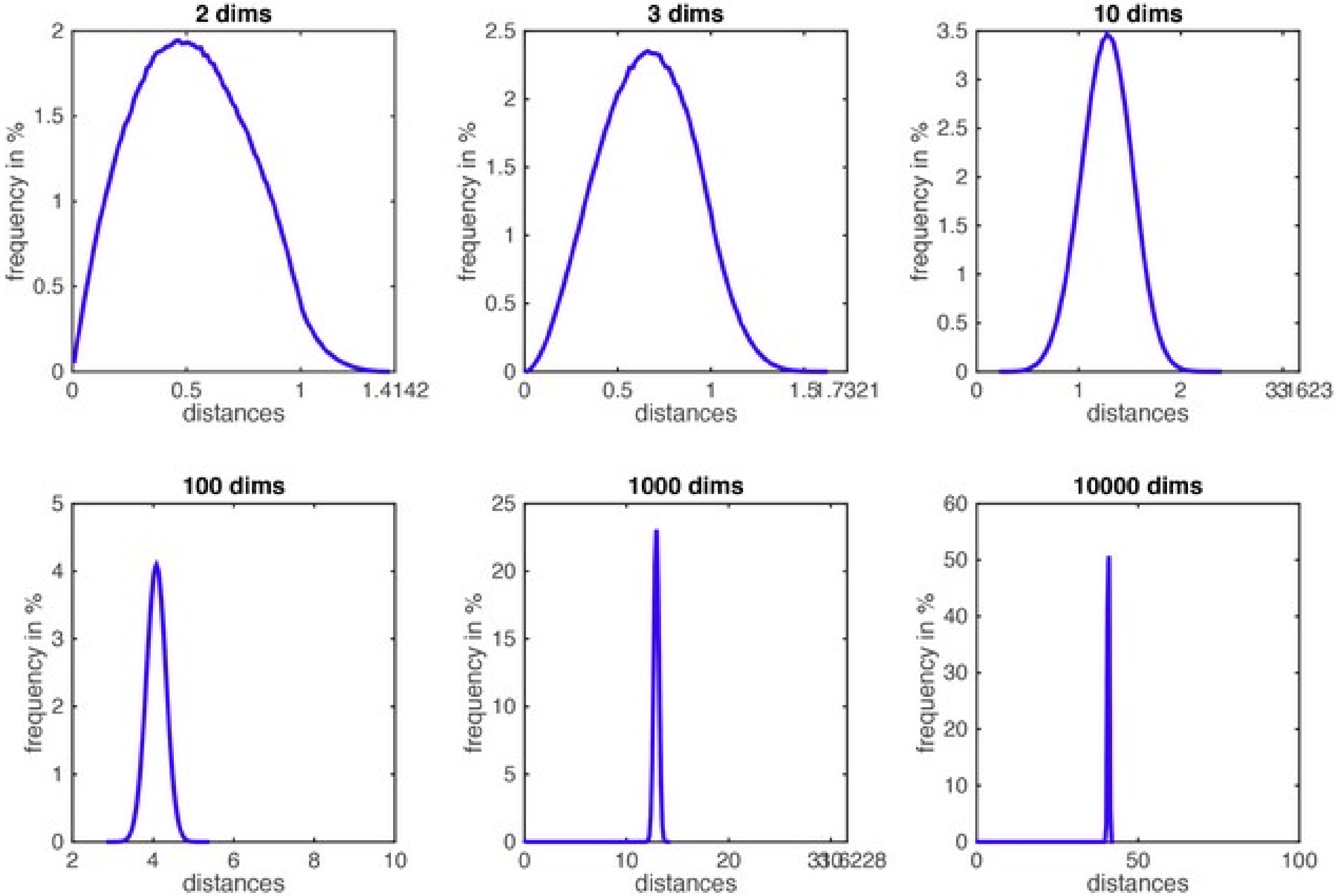
Distancia Euclideana

2D:



n-dimensional: $$d(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

Distancia Euclideana



$$\sqrt{2} = 1.41$$

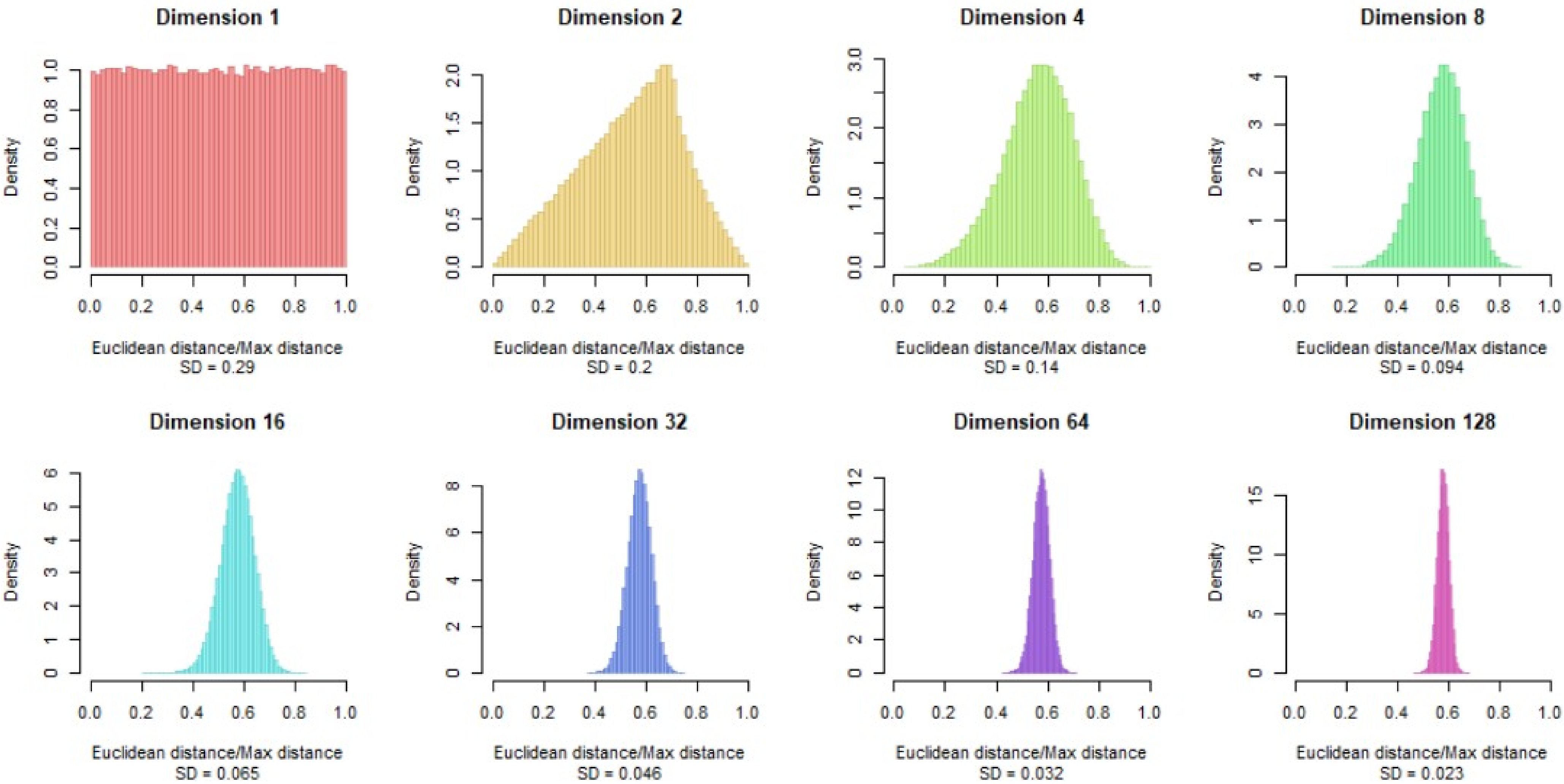
$$\sqrt{3} = 1.73$$

$$\sqrt{10} = 3.16$$

$$\sqrt{1000} = 31.6$$

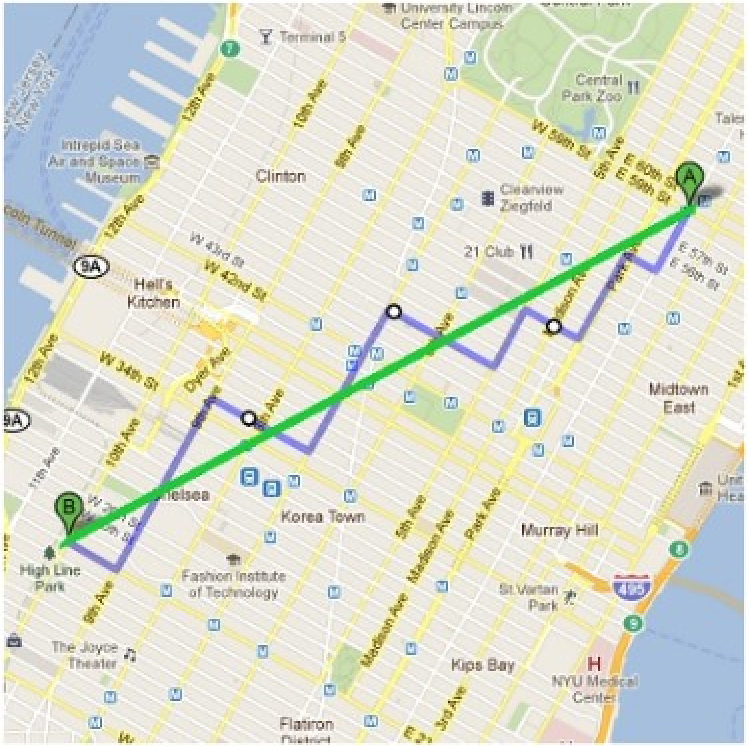
Maldición de la dimensionalidad

Distancia Euclideana



Maldición de la dimensionalidad (normalizado)

Distancias

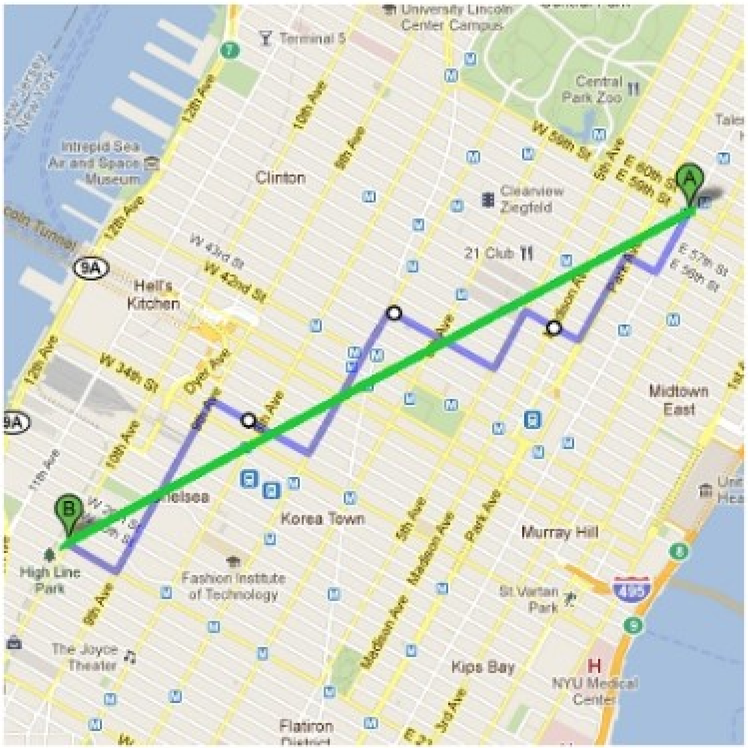


Distancia Manhattan

Distancias

Generalización (Minkowski):

$$Dist(\overline{X},\overline{Y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}$$



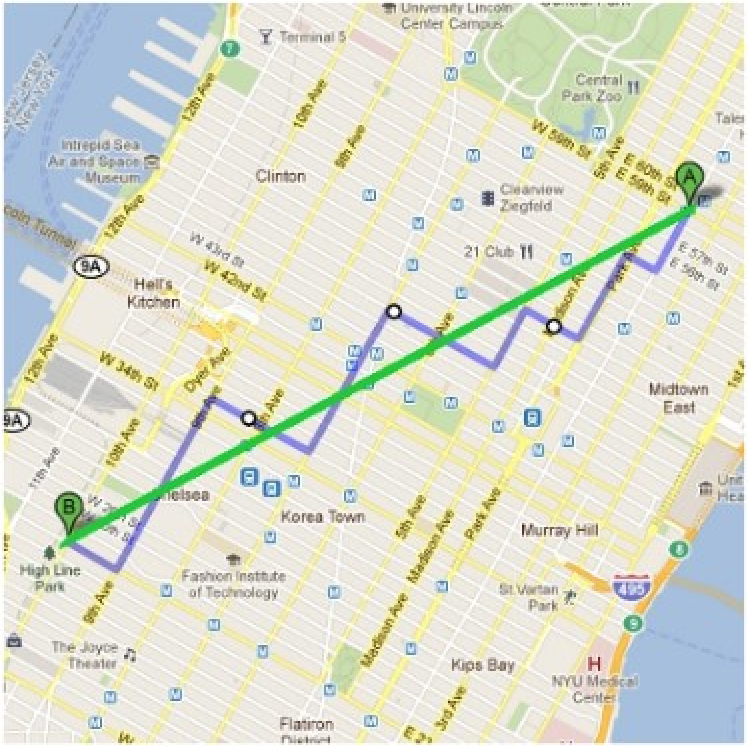
$\longleftarrow$ $Manhattan\ (p = 1)$

Distancia Manhattan

Distancias

Generalización (Minkowski):

$$Dist(\overline{X},\overline{Y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}$$



$Manhattan\ (p = 1)$

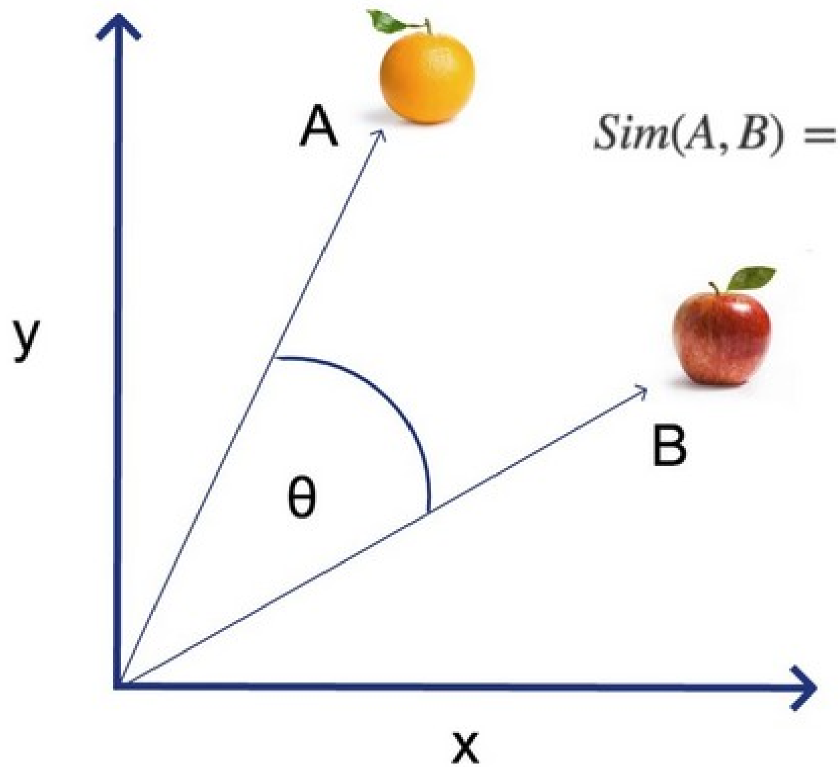
$Euclidean\ (p = 2)$

Distancia Manhattan

Proximidades

Proximidad de vectores de alta dimensionalidad:

Coseno:

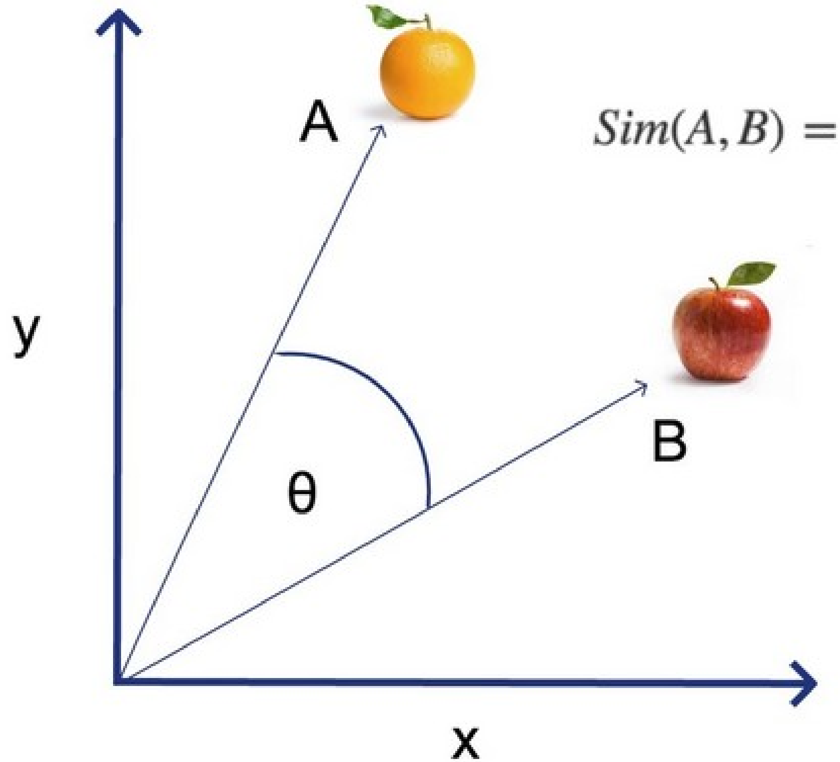


$$Sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

Proximidades

Proximidad de vectores de alta dimensionalidad:

Coseno:



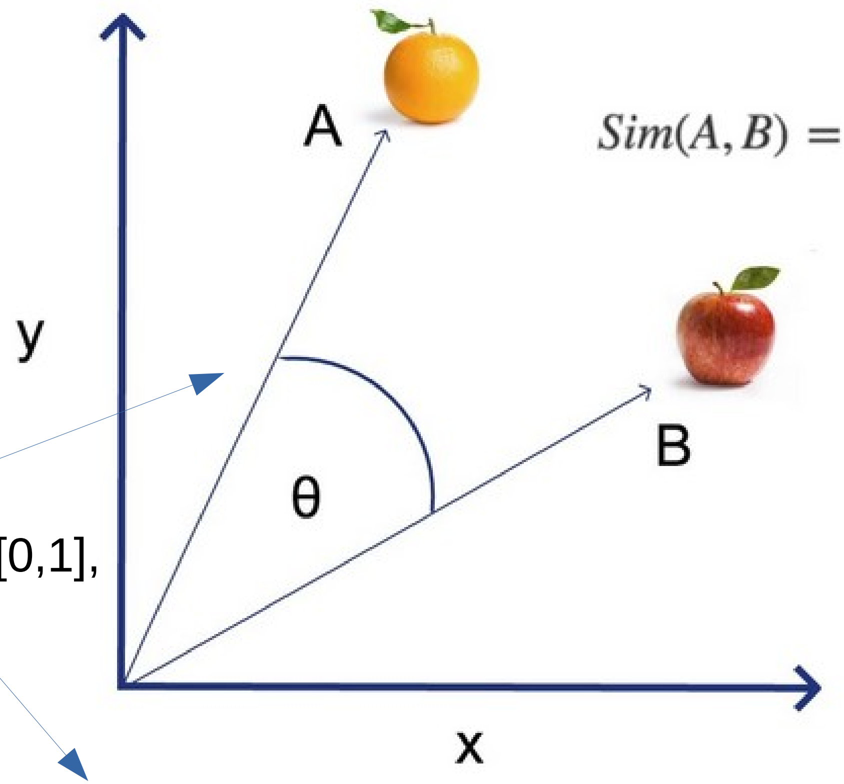
$$Sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

$$\cos(\overline{X}, \overline{Y}) = \frac{\sum_{i=1} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{d} x_i^2} \cdot \sqrt{\sum_{i=1}^{d} y_i^2}}$$

Proximidades

Proximidad de vectores de alta dimensionalidad:

Coseno:



$$Sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

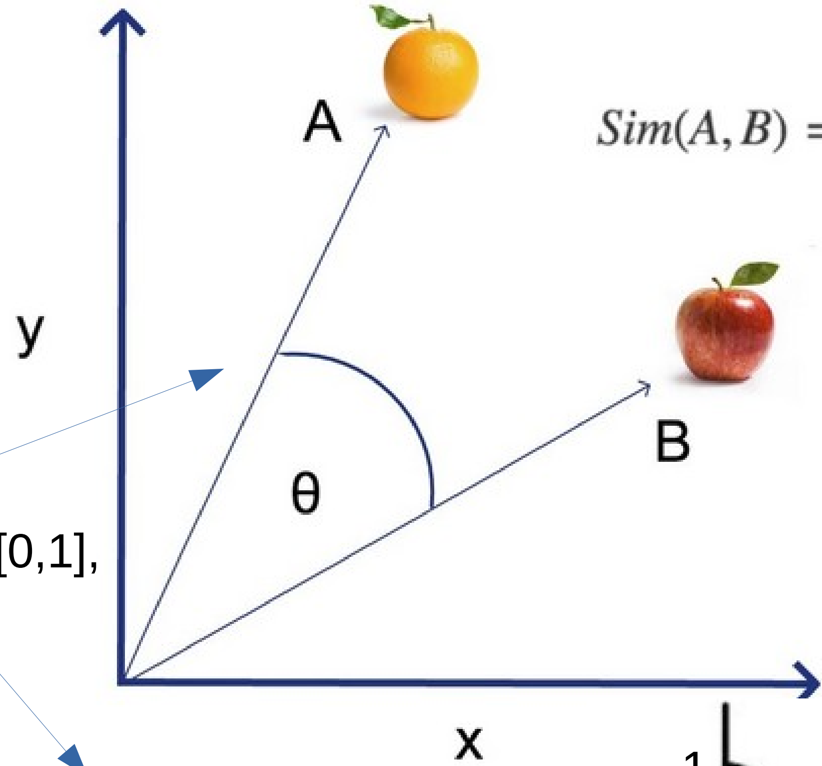
Si las características están en [0,1], los ángulos están en [0°, 90°]

$$\cos(\overline{X}, \overline{Y}) = \frac{\sum_{i=1}^{d} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{d} x_i^2} \cdot \sqrt{\sum_{i=1}^{d} y_i^2}}$$

Proximidades

Proximidad de vectores de alta dimensionalidad:

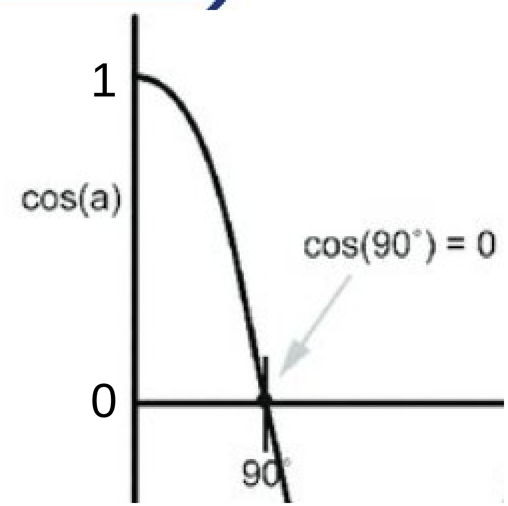
Coseno:



$$Sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

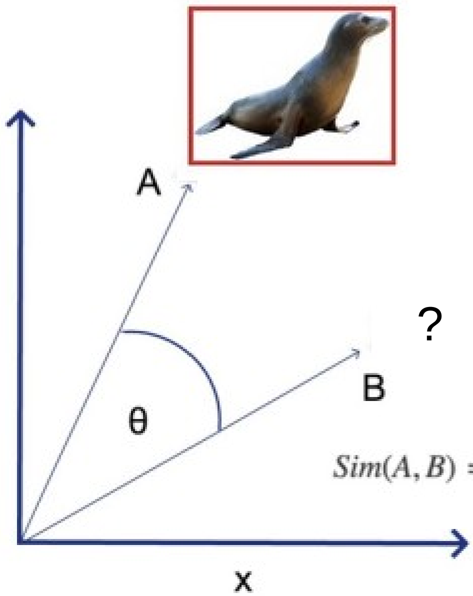
Si las características están en [0,1], los ángulos están en [0°, 90°]

$$\cos(\overline{X},\overline{Y}) = \frac{\sum_{i=1}^{d} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{d} x_i^2} \cdot \sqrt{\sum_{i=1}^{d} y_i^2}}$$



Proximidades

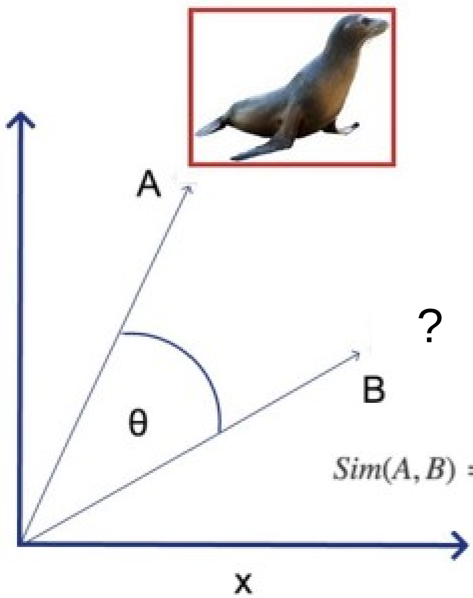
(Zoo dataset)



$$Sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

Proximidades

(Zoo dataset)



$$Sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

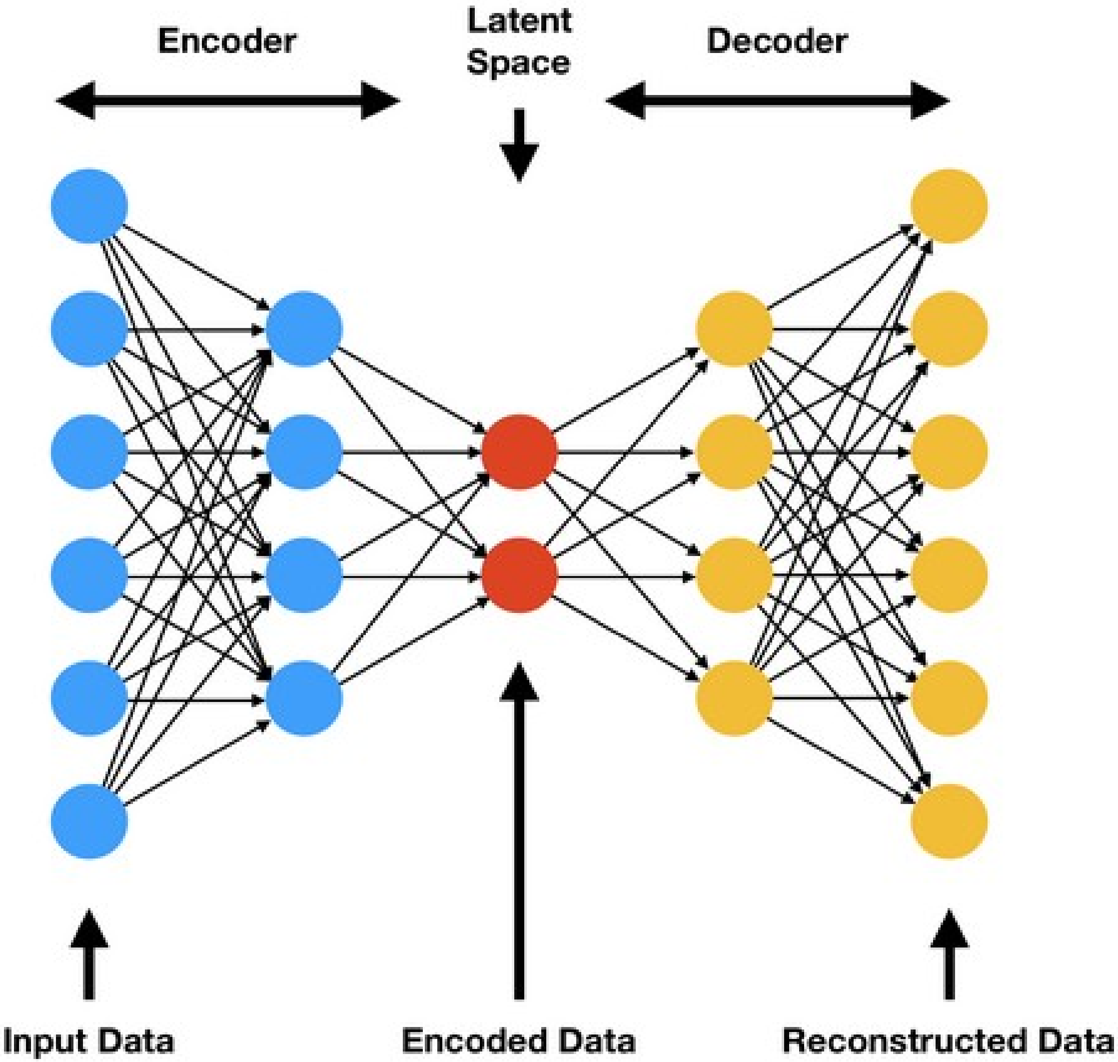
dolphin: 0.875 mink: 0.875 porpoise: 0.875 seal: 0.875 boar: 0.8125 cheetah: 0.8125 leopard: 0.8125 lion: 0.8125

- PCA -

Proyección

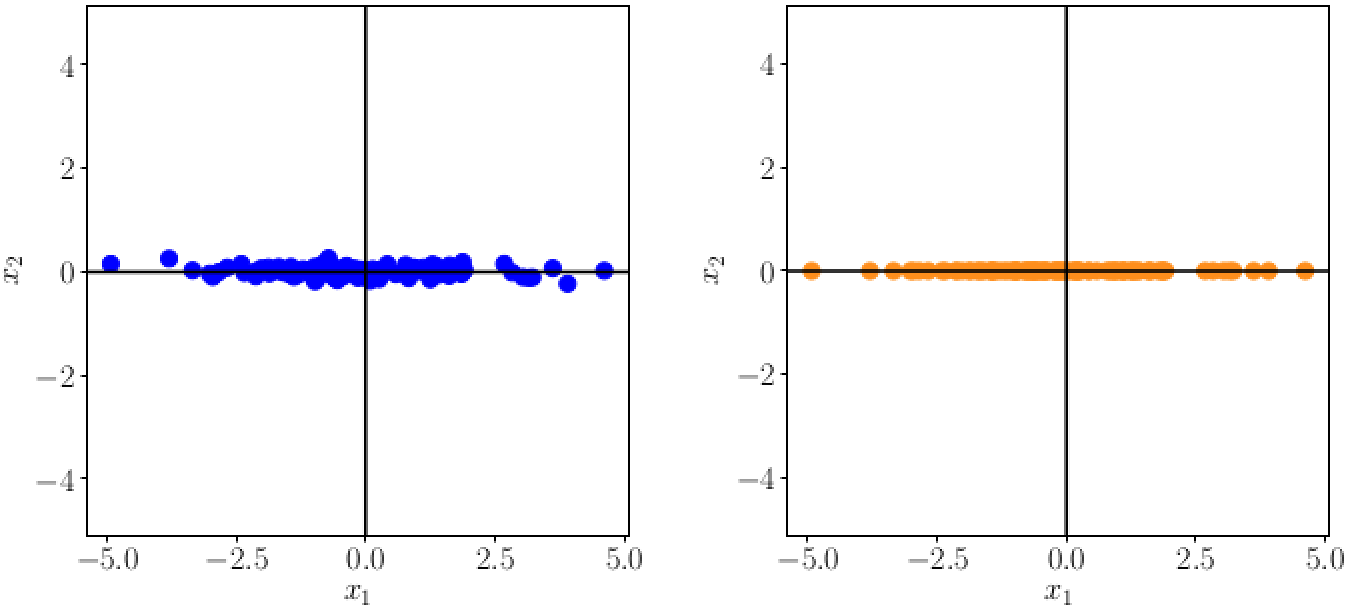
Codificación:

Compression coding



Ej.: PCA

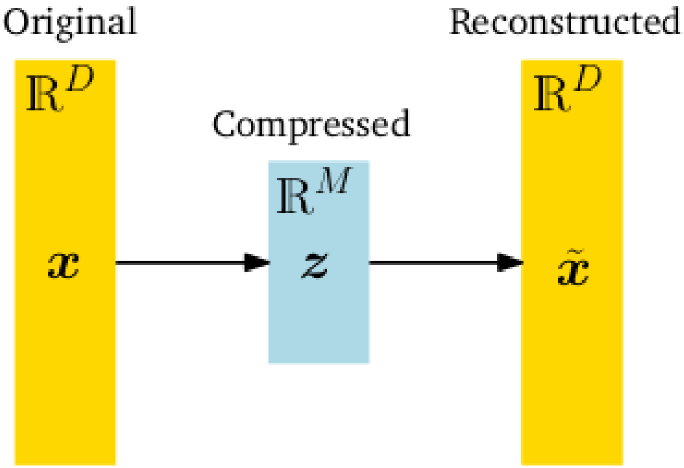
Análisis de Componentes Principales (PCA)



X1 retiene la mayor parte de la varianza por lo que remover x2 es neutro en términos de compresión.

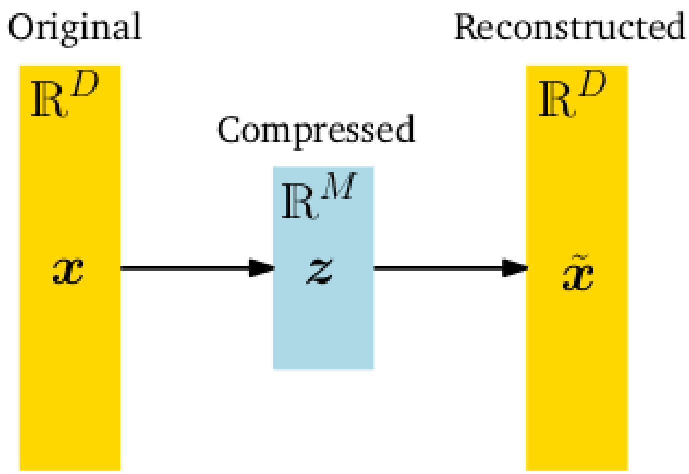
Análisis de Componentes Principales (PCA)

dataset $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, $\boldsymbol{x}_n \in \mathbb{R}^D$



Análisis de Componentes Principales (PCA)

dataset $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}, \boldsymbol{x}_n \in \mathbb{R}^D$

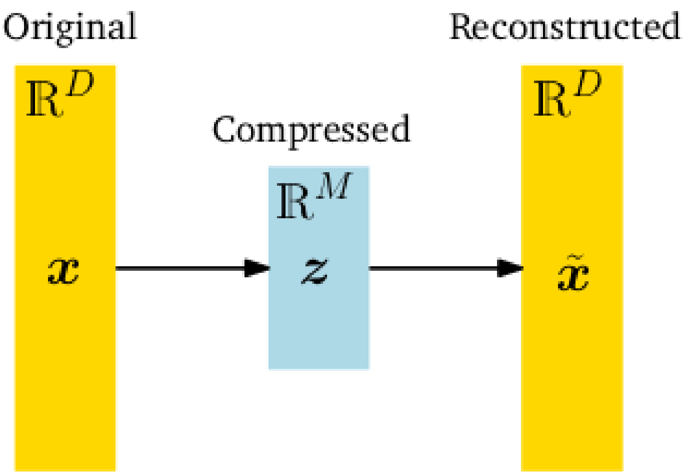


$\boldsymbol{z}_n = \boldsymbol{B}^\top \boldsymbol{x}_n \in \mathbb{R}^M$ → Baja dimensionalidad

→ Base de la descomposición $\boldsymbol{B} := [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_M] \in \mathbb{R}^{D \times M}$.

Análisis de Componentes Principales (PCA)

dataset $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, $\boldsymbol{x}_n \in \mathbb{R}^D$



$\boldsymbol{z}_n = \boldsymbol{B}^\top \boldsymbol{x}_n \in \mathbb{R}^M$ → Baja dimensionalidad

→ Base de la descomposición $\boldsymbol{B} := [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_M] \in \mathbb{R}^{D \times M}$.

→ $\boldsymbol{b}_i^\top \boldsymbol{b}_j = 0$ y $\boldsymbol{b}_i^\top \boldsymbol{b}_i = 1$

La proyección se calcula usando la SVD.

Análisis de Componentes Principales (PCA)

Proceso iterativo:

$$\hat{\boldsymbol{X}} := \boldsymbol{X} - \sum_{i=1}^{m-1} \boldsymbol{b}_i \boldsymbol{b}_i^\top \boldsymbol{X} = \boldsymbol{X} - \boldsymbol{B}_{m-1} \boldsymbol{X} \,,$$

con $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$

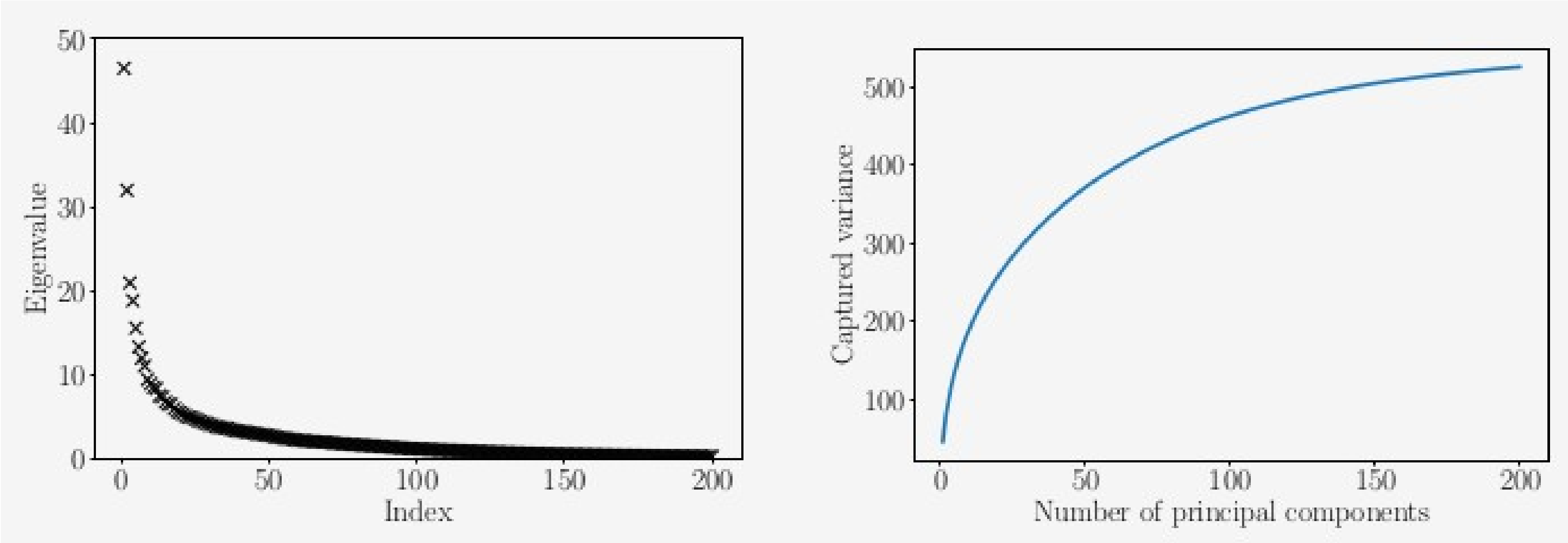
y $\boldsymbol{B}_{m-1} := \sum_{i=1}^{m-1} \boldsymbol{b}_i \boldsymbol{b}_i^\top$

Análisis de Componentes Principales (PCA)

Proceso iterativo:

$$\hat{\boldsymbol{X}} := \boldsymbol{X} - \sum_{i=1}^{m-1} \boldsymbol{b}_i \boldsymbol{b}_i^\top \boldsymbol{X} = \boldsymbol{X} - \boldsymbol{B}_{m-1}\boldsymbol{X}\,, \qquad \text{con} \quad \boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$$

$$\text{y} \quad \boldsymbol{B}_{m-1} := \sum_{i=1}^{m-1} \boldsymbol{b}_i \boldsymbol{b}_i^\top$$



Análisis de Componentes Principales (PCA)

- Aspectos prácticos:

    - Usa la full SVD (LAPACK) para datos densos.
    - Usa la SVD truncada (ARPACK) para datos dispersos.

- Implementaciones:

    - Python: sklearn

https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html