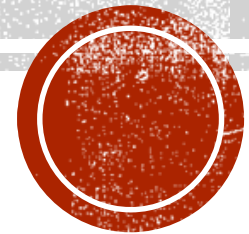
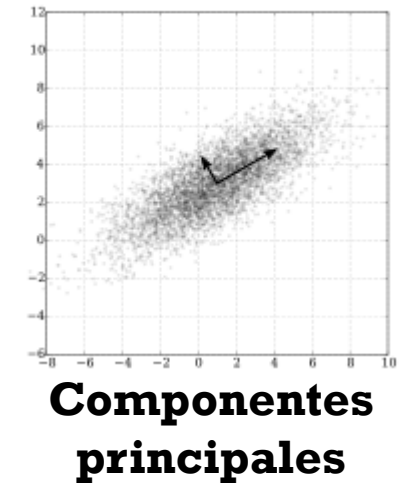
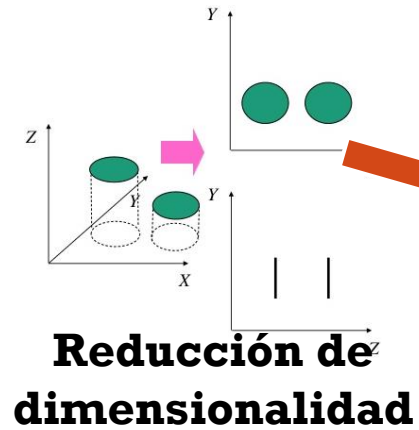
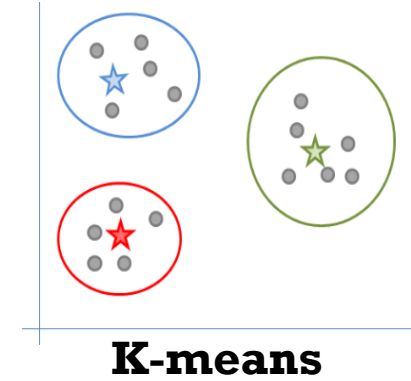
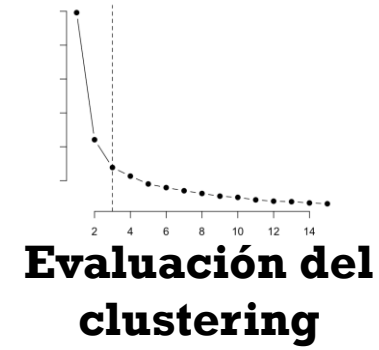
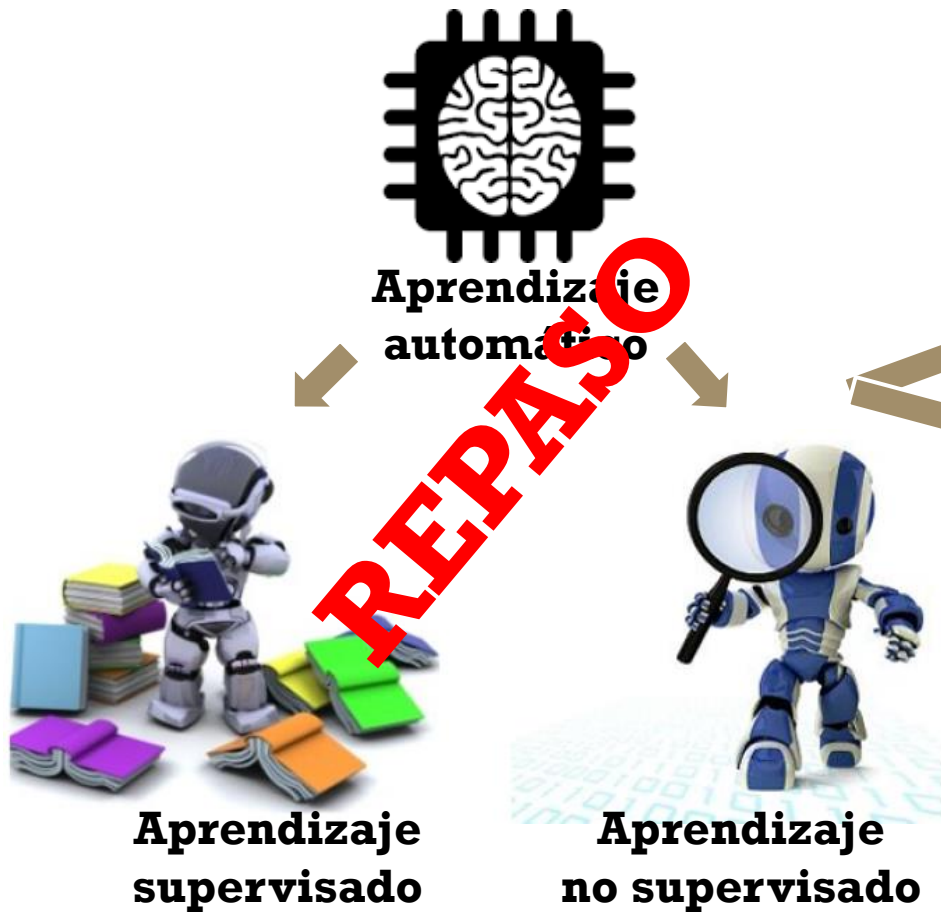


APRENDIZAJE NO SUPERVISADO



Javier Diaz Cely, PhD

AGENDA



APRENDIZAJE AUTOMÁTICO


Aprendizaje supervisado

- Aprender a partir de un “experto”
- Datos de entrenamiento **etiquetados** con una clase o valor:

$(x_1, x_2, \dots, x_n, y)$

Predictores, explicativos,
independientes

Dependiente, objetivo,
salida



- **Meta:** predecir una clase o valor

Aprendizaje no supervisado

- Sin conocimiento de una clase o valor objetivo
- Datos **no** están **etiquetados**

(x_1, x_2, \dots, x_n)

- **Meta:** descubrir factores no observados, estructura, o una representación mas simple de los datos



APRENDIZAJE AUTOMÁTICO

Aprendizaje supervisado

| Edad | Ingresos | Tiene carro? |
|------|-----------|--------------|
| 24 | 1'200.000 | NO |
| 23 | 4'500.000 | SI |
| 45 | 1'250.000 | SI |
| 32 | 1'100.000 | NO |

Factores/atributos/variables independientes,
predictores, explicativos

Dependiente, objetivo,
respuesta, salida

| | |
|----|-----------|
| 34 | 3'500.000 |
|----|-----------|

¿Cuál es el valor predicho
para una instancia dada?

Aprendizaje no supervisado

| Edad | Ingresos |
|------|-----------|
| 24 | 1'200.000 |
| 23 | 4'500.000 |
| 45 | 1'250.000 |
| 32 | 1'100.000 |

Factores/atributos/variables

Datos **no etiquetados**:
“¿Qué me puede decir
de mis datos?”

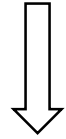
¿Se puede encontrar alguna
estructura en los datos?



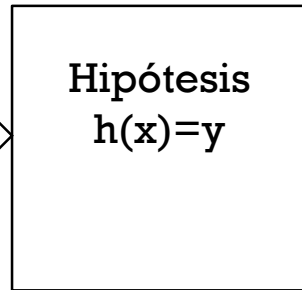
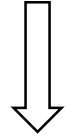
APRENDIZAJE AUTOMÁTICO

Aprendizaje supervisado

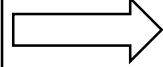
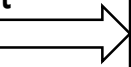
Set de entrenamiento(x_1, x_2, \dots, x_n, y)



Algoritmo de aprendizaje,
estimación de parámetros



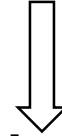
Set de text de test
(x_1', x_2', \dots, x_n')



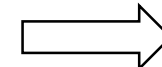
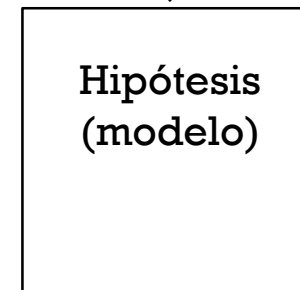
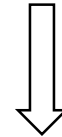
Resultado
(y')

Aprendizaje no supervisado

Set de entrenamiento(x_1, x_2, \dots, x_n)



Algoritmo de aprendizaje,
estimación de parámetros



Resultado
(estructura)



APRENDIZAJE NO SUPERVISADO

- No se interesa por la predicción sino por encontrar una estructura, un nuevo punto de vista, una simplificación o un resumen de los datos
- Usualmente se incluye en la fase exploratoria de datos
- Tipos de tareas:
 - Segmentación (clustering)
 - Cambio de representación (e.g. reducción de dimensiones, selección de factores)
 - Reglas de asociación
 - Detección de anomalías (i.e. excepciones)
- Difícil de validar los resultados, ya que no se cuenta con un “gold standard”



CLUSTERING



CLUSTERING

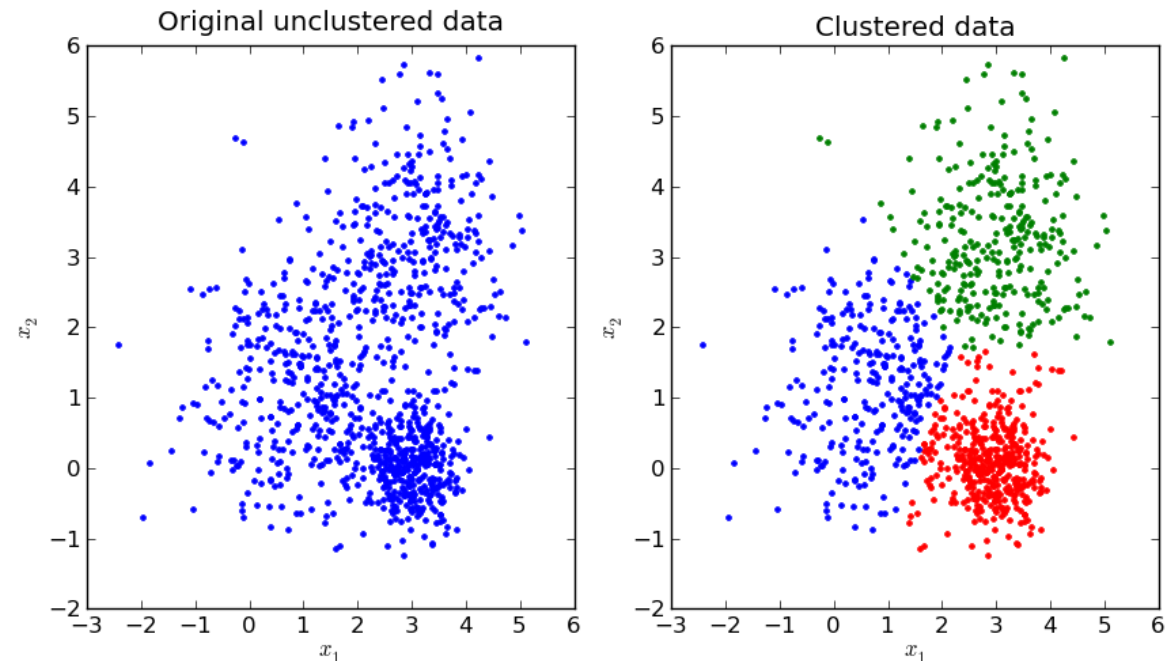
- No se tiene una variable objetivo
- Se busca agrupar los datos similares para encontrar patrones globales de los datos
- Agrupamiento por similitud, proximidad, densidad
- Particionar un conjunto heterogéneo en grupos, de forma que elementos en un grupo sean similares entre sí y tan diferentes como sea posible de elementos en otros grupos.

<http://pypr.sourceforge.net/kmeans.html>



CLUSTERING POR DISTANCIA

- Objetivo: descubrir **k** grupos o segmentos desconocidos que
 - Minimicen la distancia dentro de los grupos
 - Maximicen la distancia por fuera de los grupos
- Se basan en una noción de **distancia**
 - Definición de la medida a utilizar
 - Unidades de los atributos tienen gran influencia
 - **Normalizar**
 - **Estandarizar**



<http://pypr.sourceforge.net/kmeans.html>



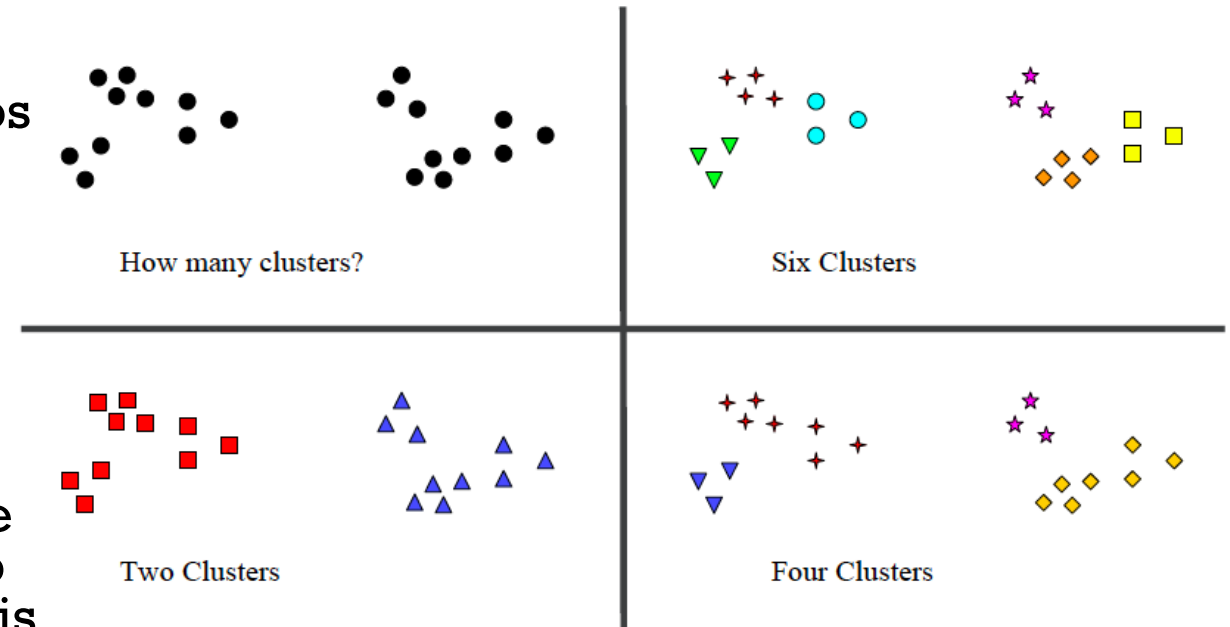


How many clusters?



CLUSTERING POR DISTANCIA

- Se pueden buscar segmentos de observaciones o de atributos (usando los mismos algoritmos)
- No existe un método universal absoluto para establecer **k**, solo heurísticos
- Requiere juicio humano, más difícil de automatizar
- La interpretación de los resultados no se debe de hacer de manera absoluta, sino como un punto de partida para el análisis
- Los datos puede que no contengan estructura, por lo que su segmentación no va a tener tanto sentido



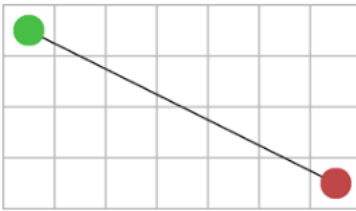
<http://governingstochastic.weebly.com/blog/category/clustering>



CLUSTERING – DISTANCIAS

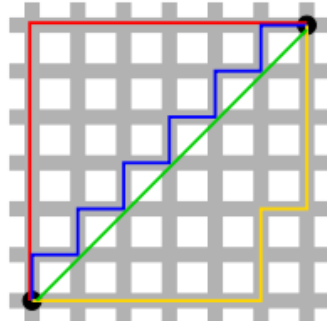
- Medidas de **similitud** o **distancia**:

- Euclidiana**: tamaño del segmento linear que une las dos instancias comparadas.

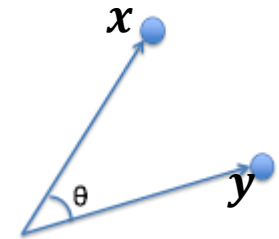


$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2}$$
$$= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}.$$

- Manhattan**: basada en una organización en bloques rectilíneos



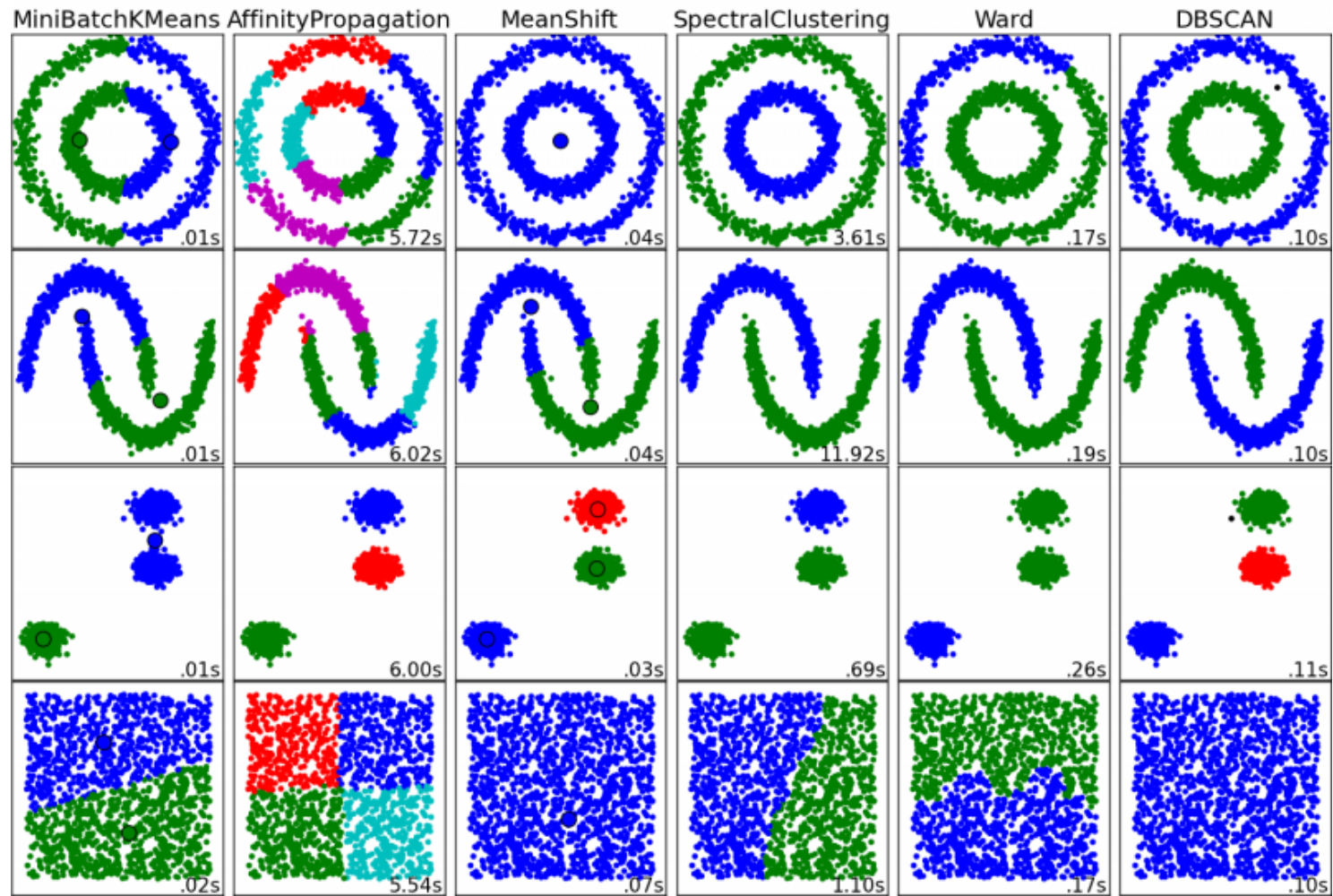
- Coseno**: coseno del ángulo entre las dos instancias comparadas → Alta dimensionalidad y **big data**



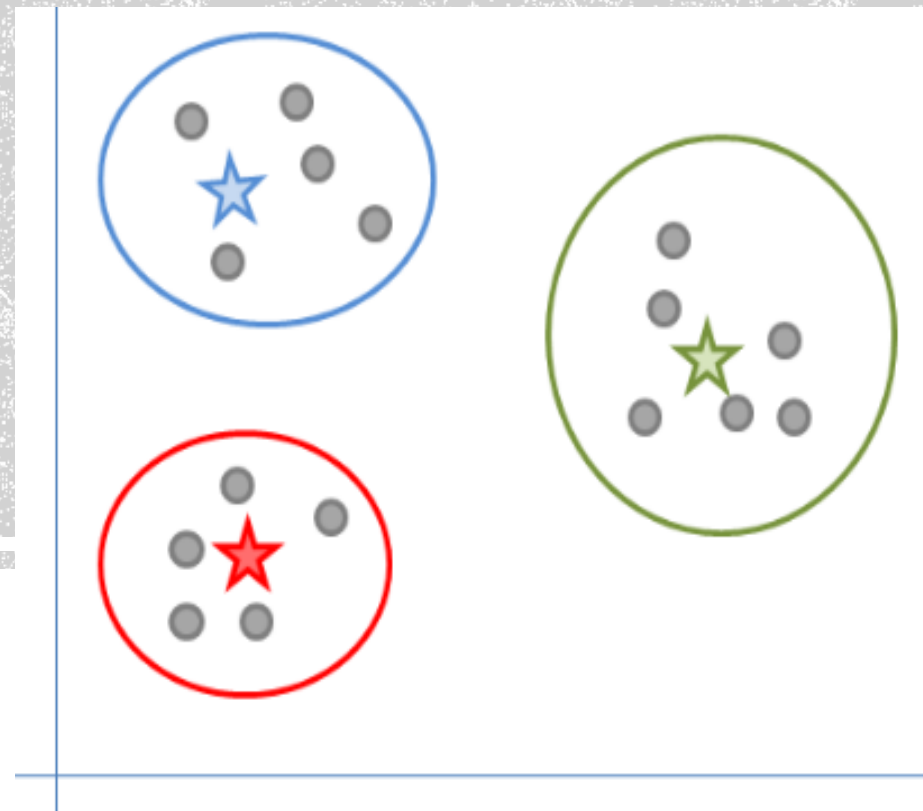
$$sim(\mathbf{x}, \mathbf{y}) = \cos(\theta_{\mathbf{x}, \mathbf{y}}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{\sum_i x_i * y_i}{\sqrt{(\sum_i x_i * x_i) * \sum_i y_i * y_i}}$$



CLUSTERING



K-MEANS



K-MEANS



(a)



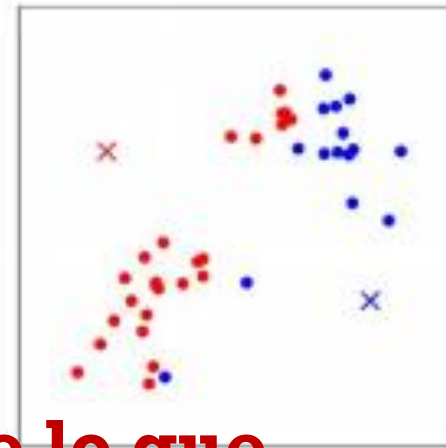
K-MEANS



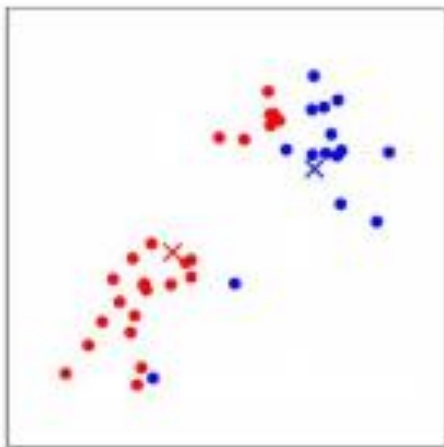
(a)



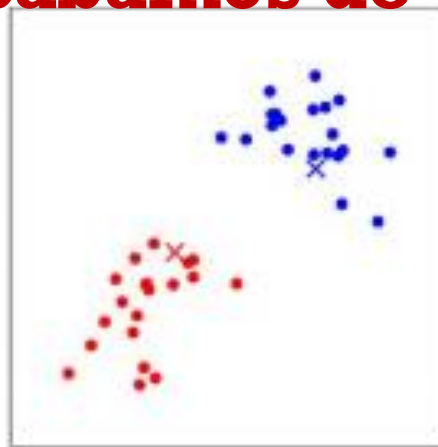
(b)



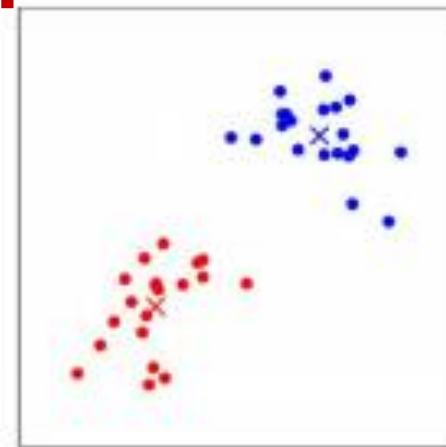
(c)



(d)



(e)



(f)

¿Cuál es el algoritmo de lo que acabamos de ver?



K-MEANS

- Algoritmo:
 1. Inicializar los K centroides
 2. Asignar cada instancia al cluster del centroide más cercano
 3. Re calcular los centroides de cada cluster (el baricentro/promedio)
 4. Repetir pasos 2 y 3 hasta convergencia (hasta que los centroides permanezcan estáticos)
- Cada observación se asigna a un solo cluster, de manera absoluta
- Los clusters no se sobrelapan
- Objetivo: minimizar la variación dentro de los clusters (Within Sum of Squares - WSS):

$$WSS = \sum_{i=1}^{\#instancias} distancia(\mathbf{x}_i - \text{centroide}(\mathbf{x}_i))^2$$



K-MEANS

- **Consideraciones:**

- ¿Cómo estimar el número de clusters (K)?

- Mardia (1979): $\sqrt{n/2}$
 - Método “del codo”
 - Método Silhouette
 - Medida de CH

- ¿Cómo inicializar los centroides de los clusters?

- Escoger centros completamente aleatorios
 - Escoger puntos existentes aleatoriamente
 - Escoger los centroides utilizando K-Means ++



K-MEANS

K-Means++

- La idea es inicializar los centros lo más lejanos los unos de los otros
- Algoritmo K-Means++:
 1. Se comienza con un conjunto M de centroides vacío
 2. Se escoge aleatoriamente una instancia que no sea ya un centroide y se agrega a M .
 3. Se calcula la distancia mínima de las instancias que quedan con los centroides en M
 4. Se escoge una nueva instancia como centroide de manera aleatoria asociando una probabilidad a cada instancia dada por su distancia mínima calculada anteriormente
 5. Repetir pasos 3 y 4 hasta haber seleccionado K centroides
 6. Continuar con el algoritmo K-Means clásico.



K-MEANS

- **Consideraciones:**

- ¿Qué distancia escoger?
 - Depende del problema
 - e.g. Euclidiana, Manhattan, correlación

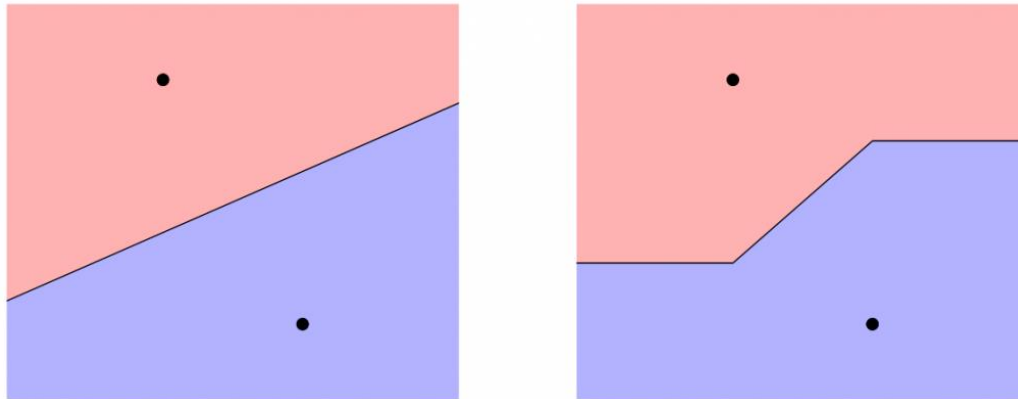
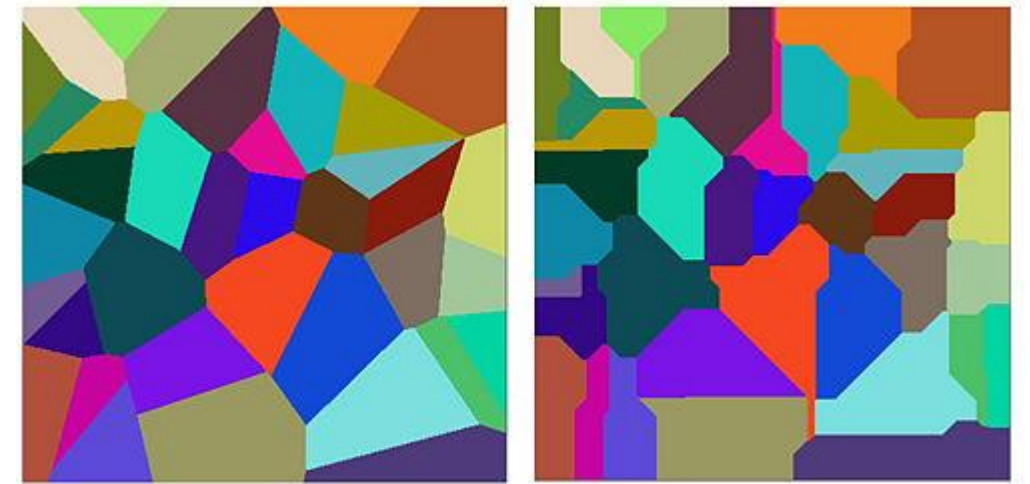


Diagrama de Voronoi: particionamiento espacial basado en la distancia con los centroides



Euclidean

Manhattan

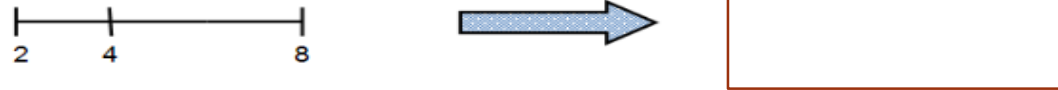
<http://math.stackexchange.com/>



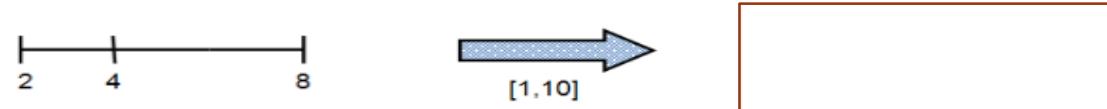
K-MEANS

Consideraciones:

- Normalización [0, 1]

$$Y = \frac{X - \text{mínimo_original}}{\text{máximo_original} - \text{mínimo_original}}$$


- Normalización [newmin, newmax] → Generalización, cambio de escala a otro intervalo cualquiera, no necesariamente [0, 1], ni [oldmin, oldmax]

$$Y = \text{min} + \frac{X - \text{mínimo_original}}{\text{máximo_original} - \text{mínimo_original}} (\text{max} - \text{min})$$


- Normalización z-score (estandarización)
 - Supuesto de distribución normal
 - Sea Z la representación estandarizada del dato
 - X la representación actual del dato
 - μ el valor promedio de los datos
 - σ la desviación estándar del campo

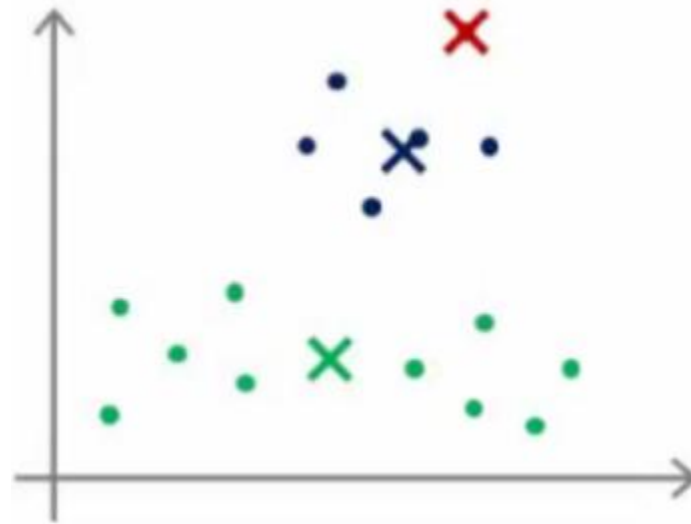
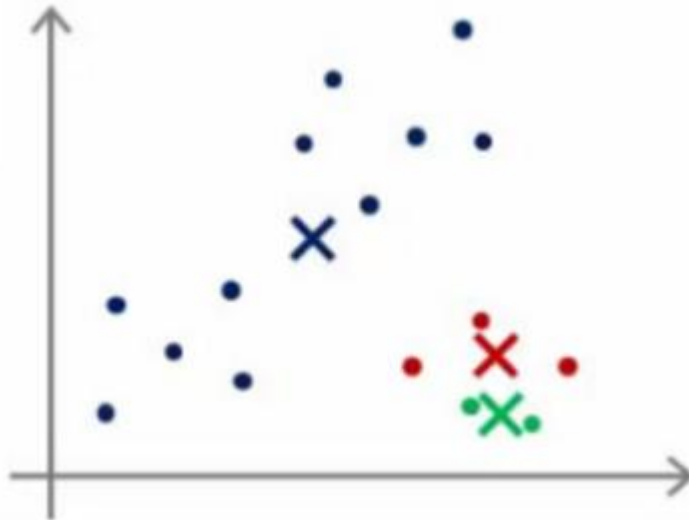
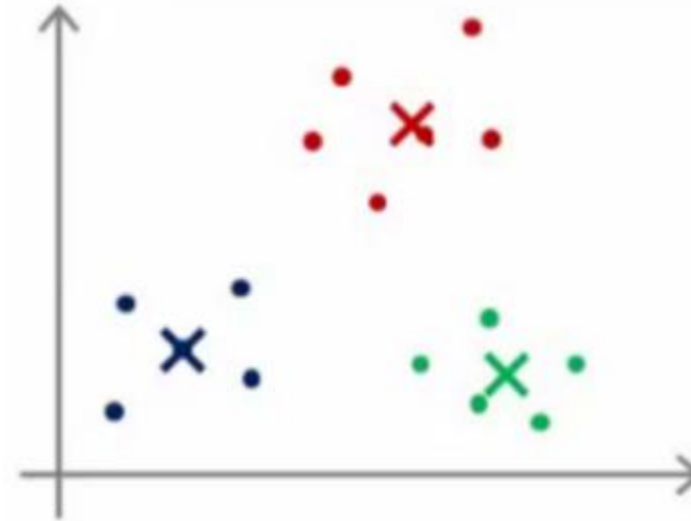
$$Z = \frac{X - \mu}{\sigma}$$



K-MEANS

- Consideraciones:

- ¿Cómo evitar los óptimos locales?
 - Ejecutar varias veces el algoritmo con diferentes inicializaciones, seleccionar el clustering con el mínimo WSS



K-MEANS

- **Consideraciones:**
 - Algoritmo de particionamiento
 - Muy fácil de implementar
 - Más rápido que clustering jerárquico
 - Sólo trabaja con atributos numéricos (noción de promedio)
 - Muy sensible a excepciones
 - Funciona muy bien con datos generados siguiendo un proceso Gaussiano
 - No funciona para identificar clusters con formas no convexas

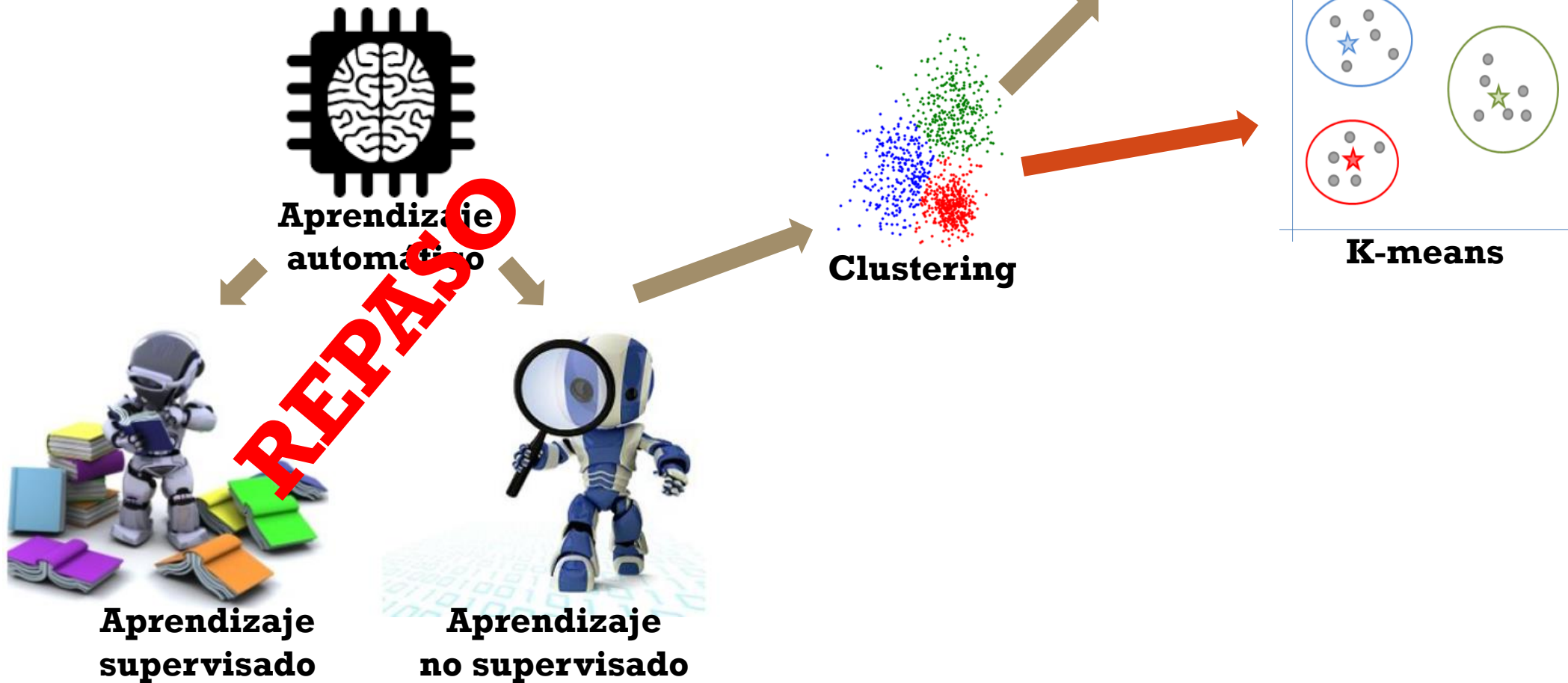


TALLER: K-MEANS — CLIENTES DE SUPER MERCADOS

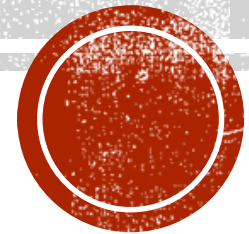
Desarrollar el taller de clustering de clientes de supermercado.



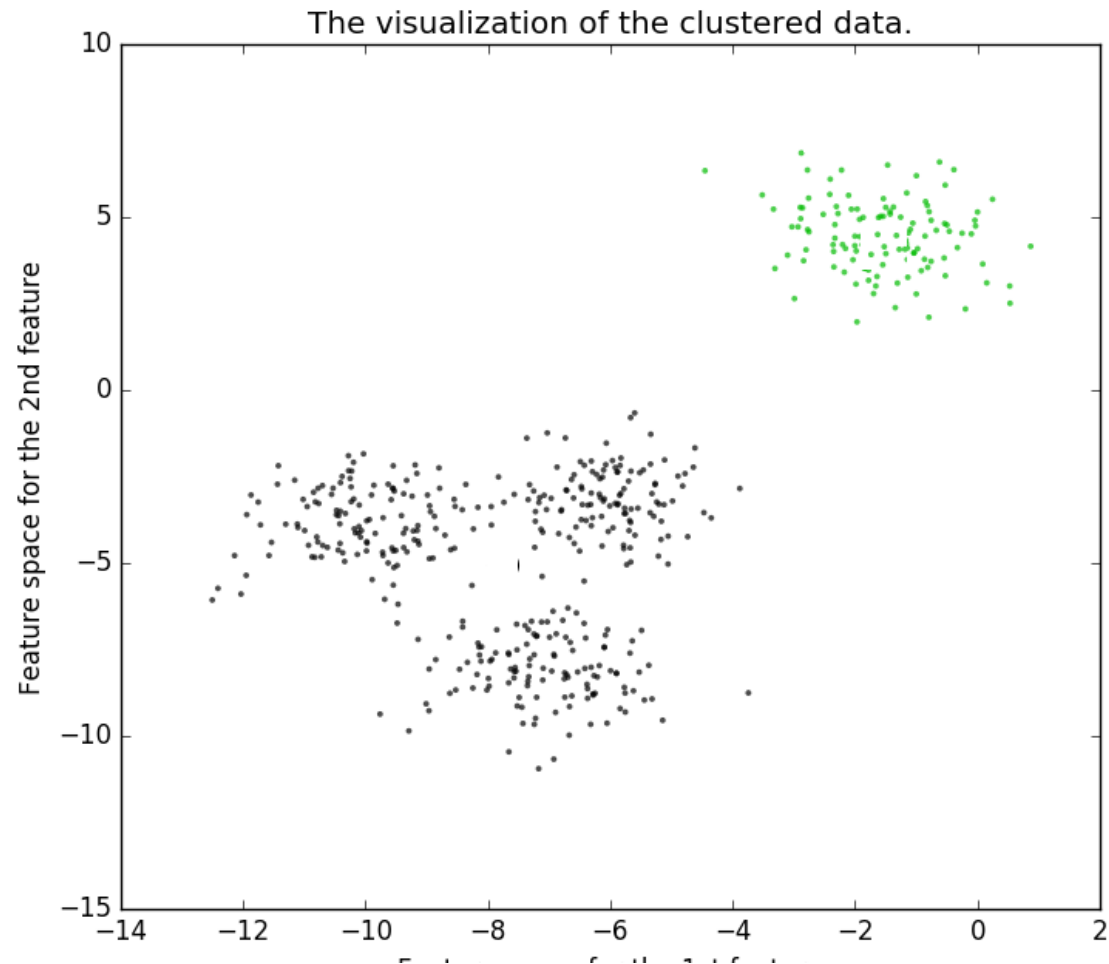
AGENDA



EVALUACIÓN DE CLUSTERING



ESCOGENCIA DEL K



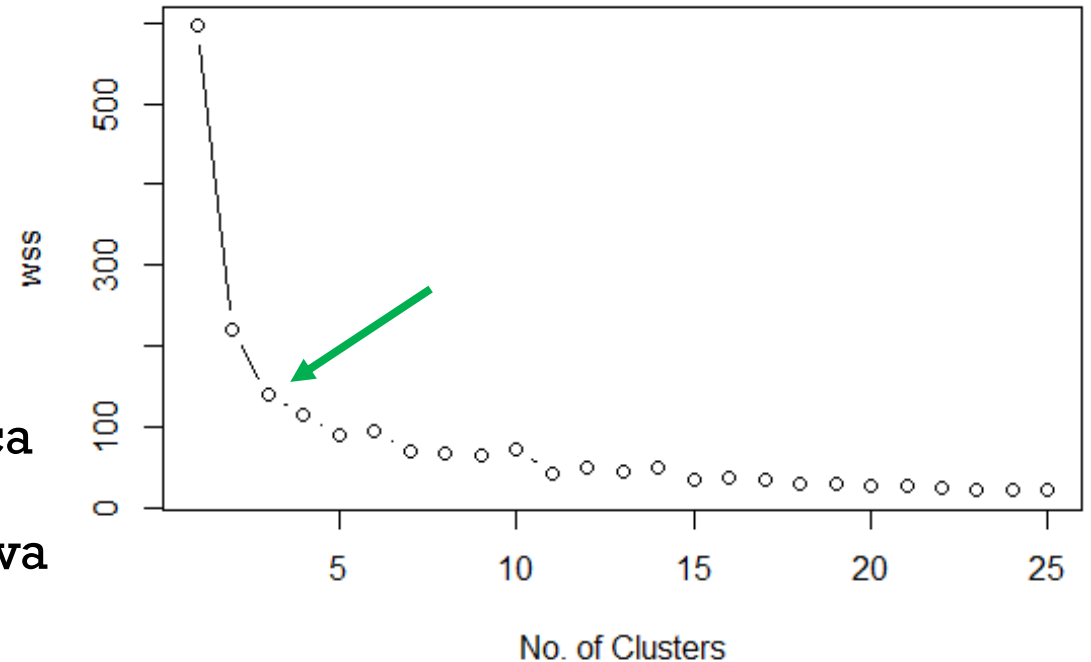
http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

¿Cuántos clusters ven uds aquí?



ESCOGENCIA DEL K — CODO

- Heurísticos:
 - No hay un método absoluto
 - Dependen del juicio del analista, se requiere conocimiento del negocio
- Método “del codo”:
 - Plotear WSS para cada valor de K
 - Escoger el último valor de K que implica una reducción “considerable” del WSS del clustering resultante, cuando la curva se vuelve aproximadamente lineal



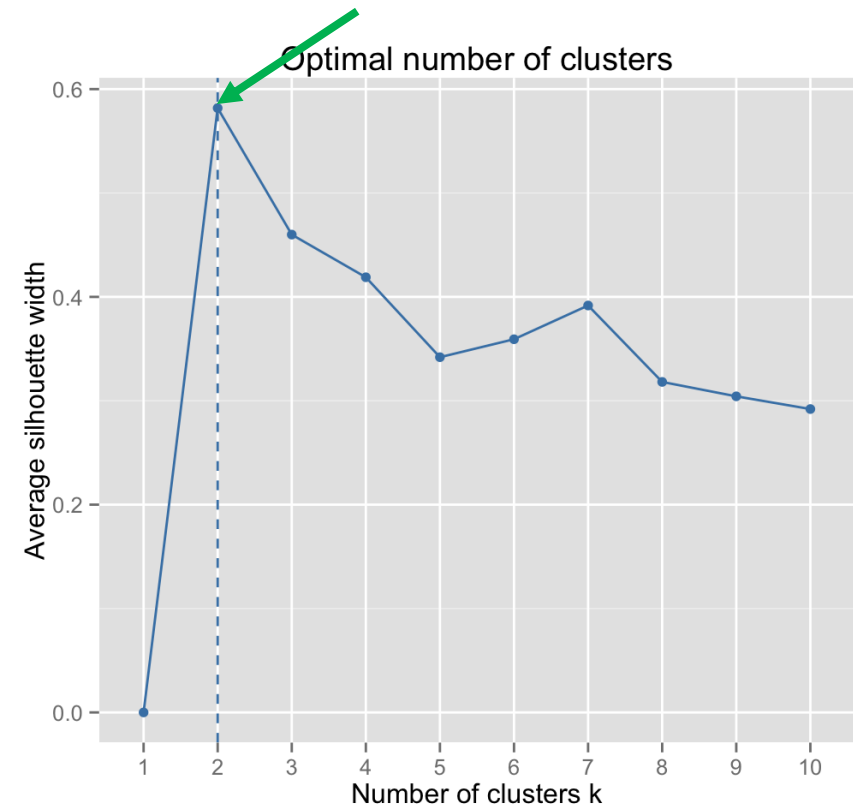
$$WSS = \sum_{i=1}^{\#instancias} distancia(x_i - centroide(x_i))^2$$



ESCOGENCIA DEL K — SILHOUETTE

■ Método Silhouette

- Analizar el ajuste de cada instancia al cluster al que fue asignado
- Qué tan cerca está cada observación de las demás de su propio cluster
 - 0,7-1,0: el cluster es fuertemente robusto
 - 0,5-0,7: el cluster es razonablemente robusto
 - 0,25-0,5: el cluster puede ser artificial y puede no denotar una noción de estructura necesariamente
 - Inferior a 0,25: el cluster debería descartarse, no indica estructura
- Se busca la maximización del valor Silhouette promedio de los clusters



ESCOGENCIA DEL K — SILUETA

■ Método Silueta (Silhouette)

■ Calcular el valor de silueta de cada punto:

- Cohesión del punto con su cluster C_i (promedio de distancias con puntos de su mismo cluster):

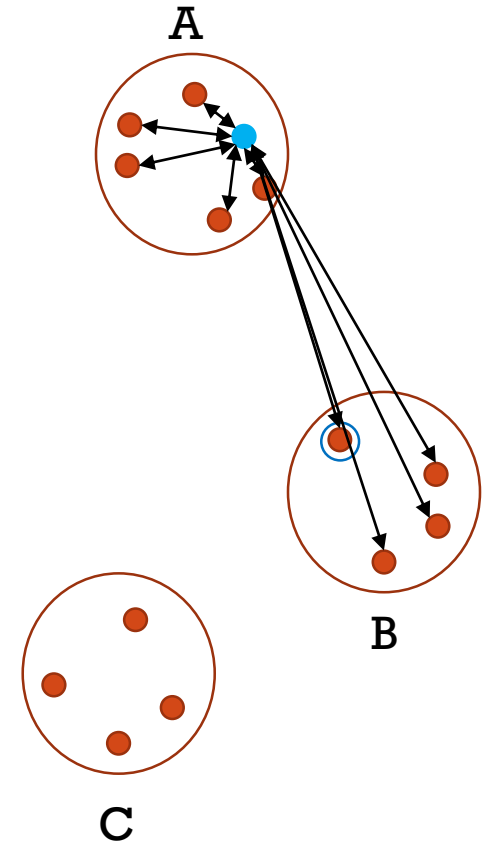
$$\text{cohesión}(p) = a(p) = \frac{\sum_{p' \in C_i, p' \neq p} \text{distancia}(p, p')}{|C_i| - 1}$$

- Separación de los puntos de otros clusters (distancia promedio con los puntos del cluster más cercano):

$$\text{separación}(p) = b(p) = \min_{C_j: 1 \leq j \leq k, j \neq i} \left(\frac{\sum_{p' \in C_j} \text{distancia}(p, p')}{|C_j|} \right)$$

- El valor de silueta del punto es entonces:

$$\text{silueta}(p) = s(p) = \frac{b(p) - a(p)}{\max(b(p), a(p))}$$



ESCOGENCIA DEL K — SILUETA

- Método Silueta (Silhouette)

- Calcular el valor de silueta de cada cluster (promedio de las siluetas de sus puntos).

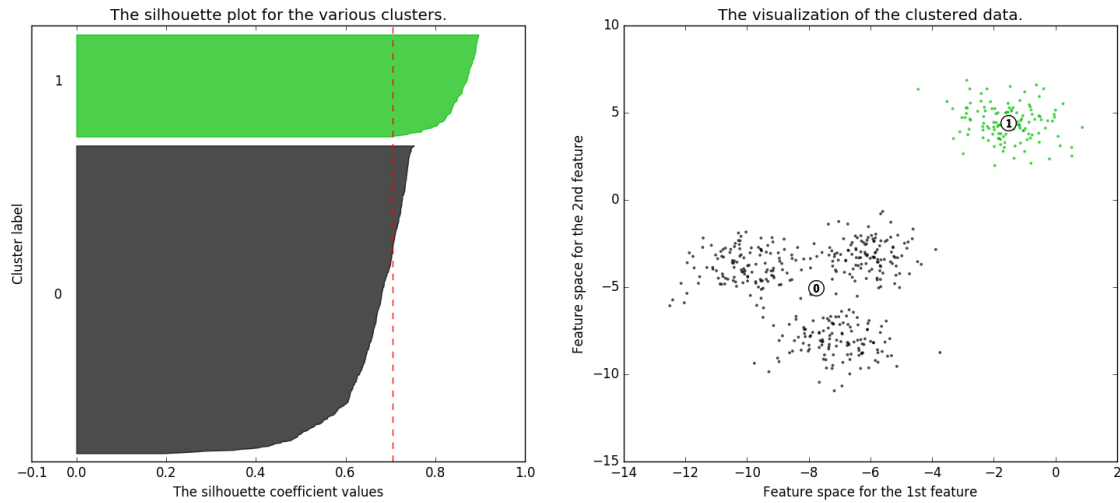
$$silueta(C_i) = \frac{1}{|C_i|} \sum_{p \in C_i} s(p)$$

- Analizar los puntos y clusters, buscando posibles problemas de asignación dados por el valor del K:
 - El rango de la silueta está entre -1 y 1
 - Una silueta de 0 implica que la asignación de un punto a su cluster es indiferente
 - Se espera que los puntos del mismo cluster estén más cercanos al punto en cuestión: para que la silueta sea positiva tenemos que $a(p) < b(p)$

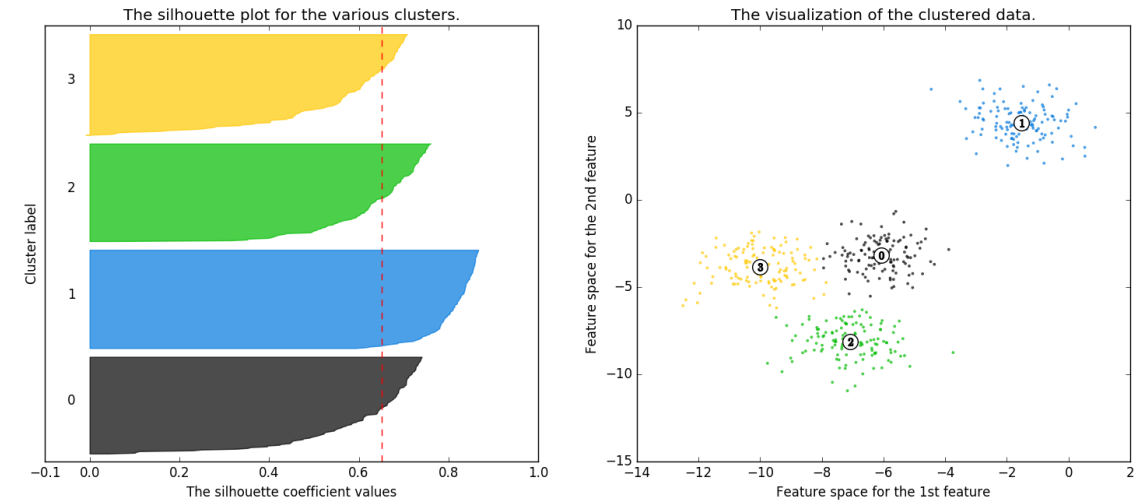


ESCOGENCIA DEL K — SILHOUETTE

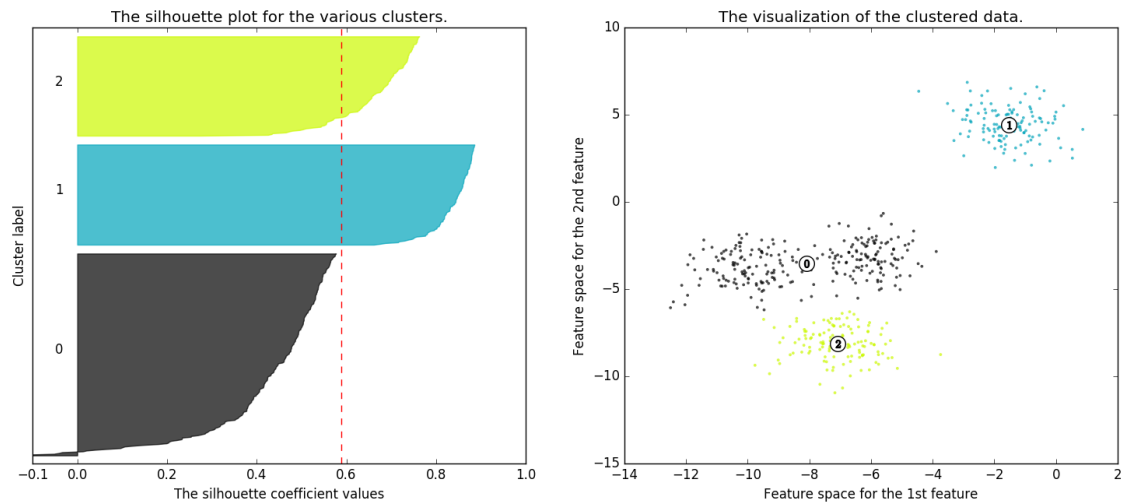
Silhouette analysis for KMeans clustering on sample data with $n_clusters = 2$



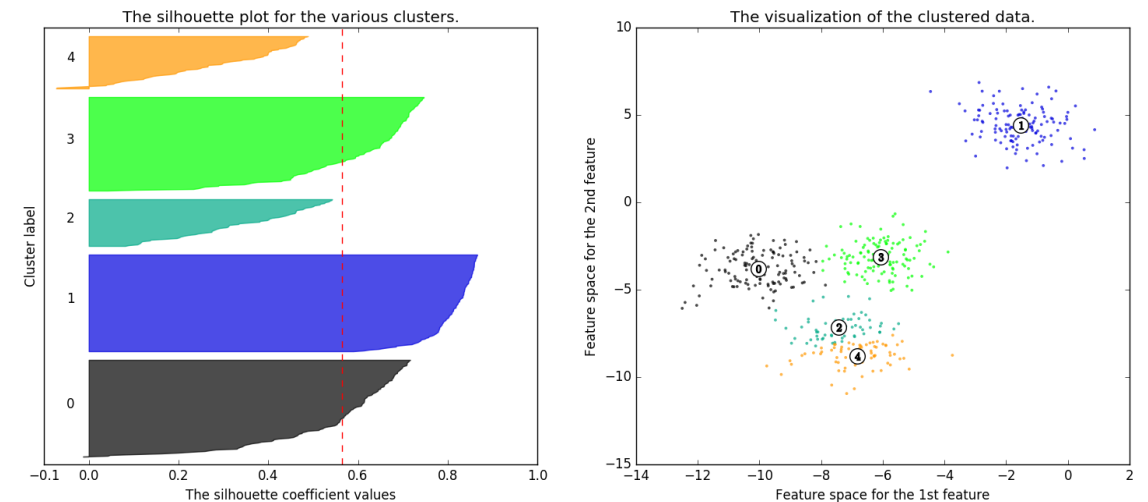
Silhouette analysis for KMeans clustering on sample data with $n_clusters = 4$



Silhouette analysis for KMeans clustering on sample data with $n_clusters = 3$

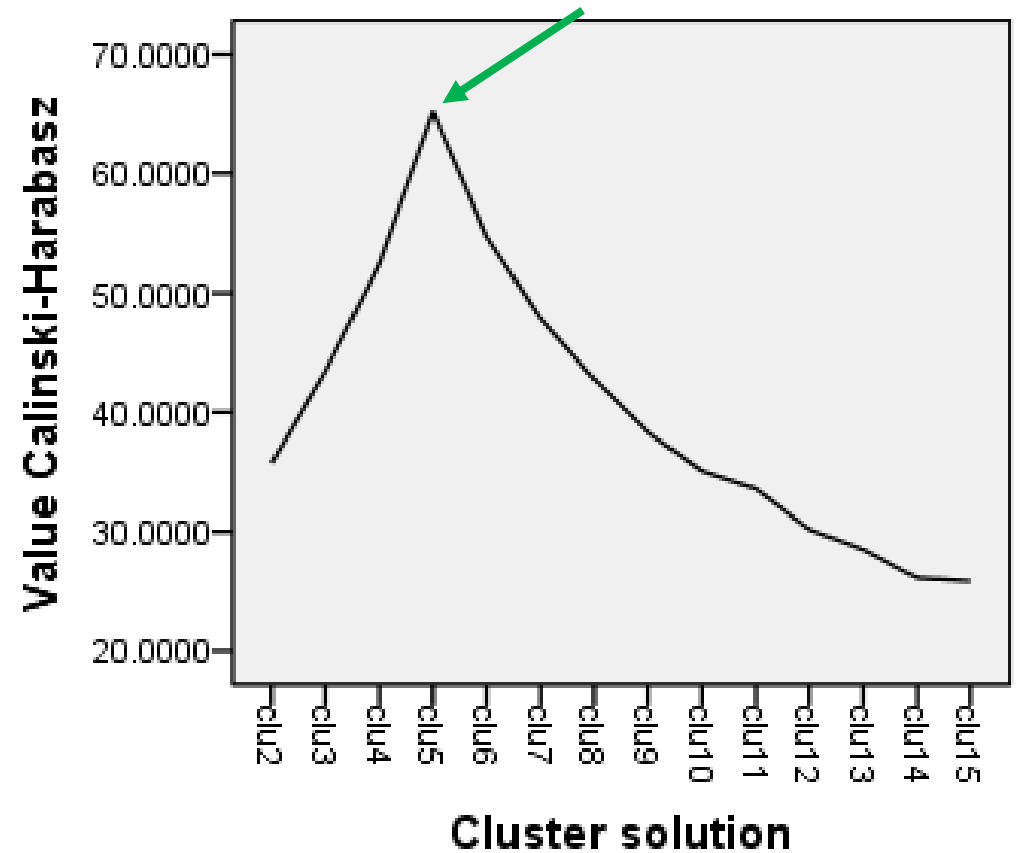


Silhouette analysis for KMeans clustering on sample data with $n_clusters = 5$



ESCOGENCIA DEL K — CALINSKI-HARABASZ

- Método de Calinski-Harabasz:
 - TSS = variación total (entre todos los datos y el centro global)
 - WSS = variación intra-cluster (entre los puntos de cada cluster y sus centroides)
 - BSS = variación inter-cluster (entre los centroides de los clusters y el centro global). $BSS = TSS - WSS$
 - CH = ratio entre la variación entre clusters (BSS) y el promedio de la variación interna de los clusters (WSS)
 - Se busca el K que maximice el valor de la medida CH

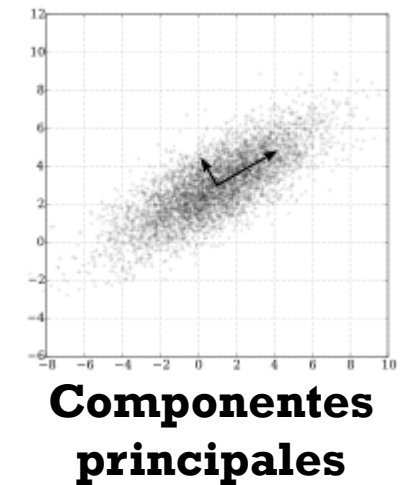
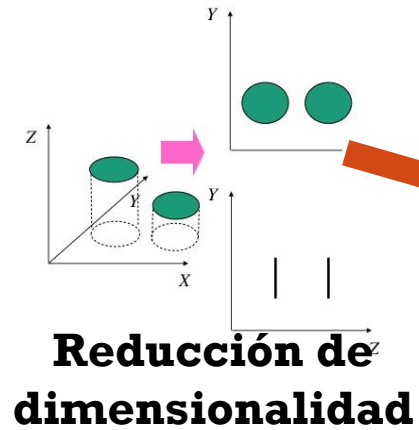
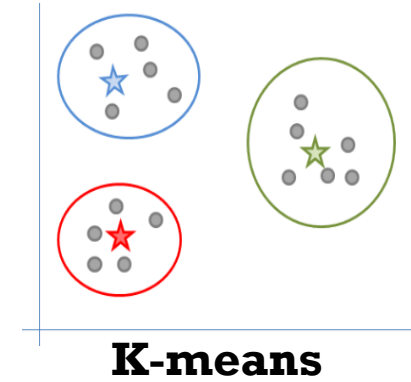
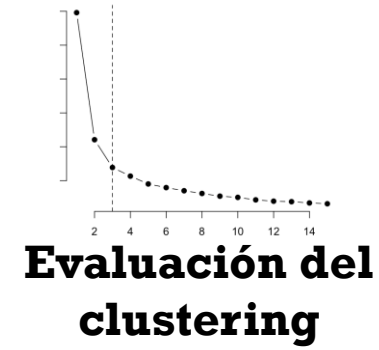
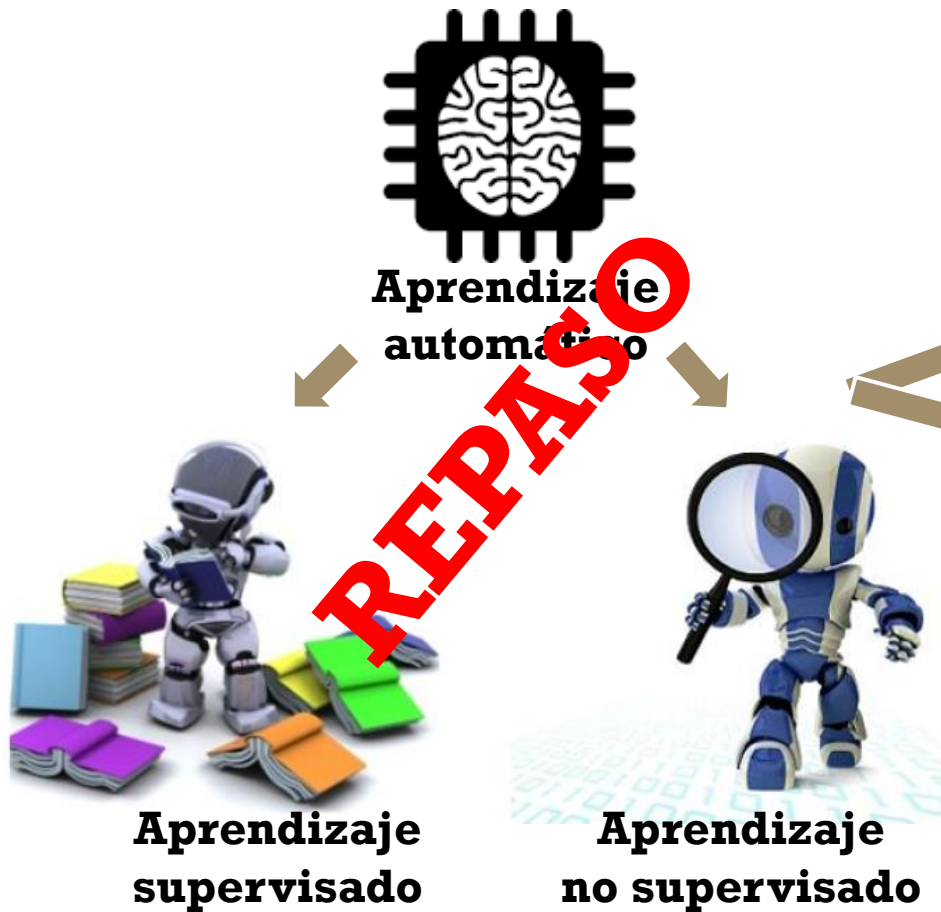


TALLER: EVALUACIÓN DE CLUSTERING

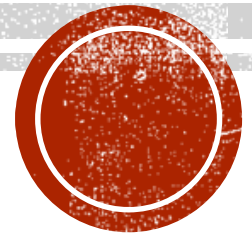
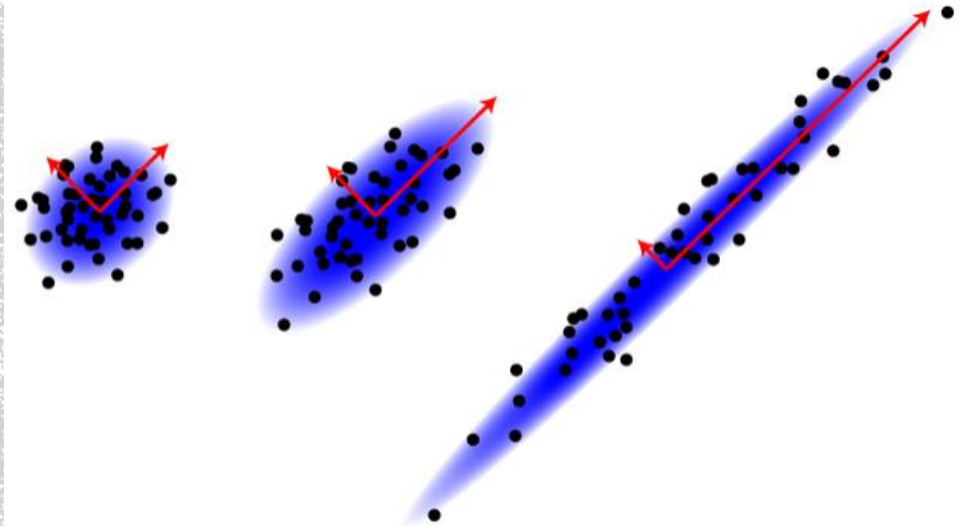
Continuar con el taller de clustering de clientes de supermercado con la parte dedicada a la evaluación del número de clusters.



AGENDA



COMPONENTES PRINCIPALES

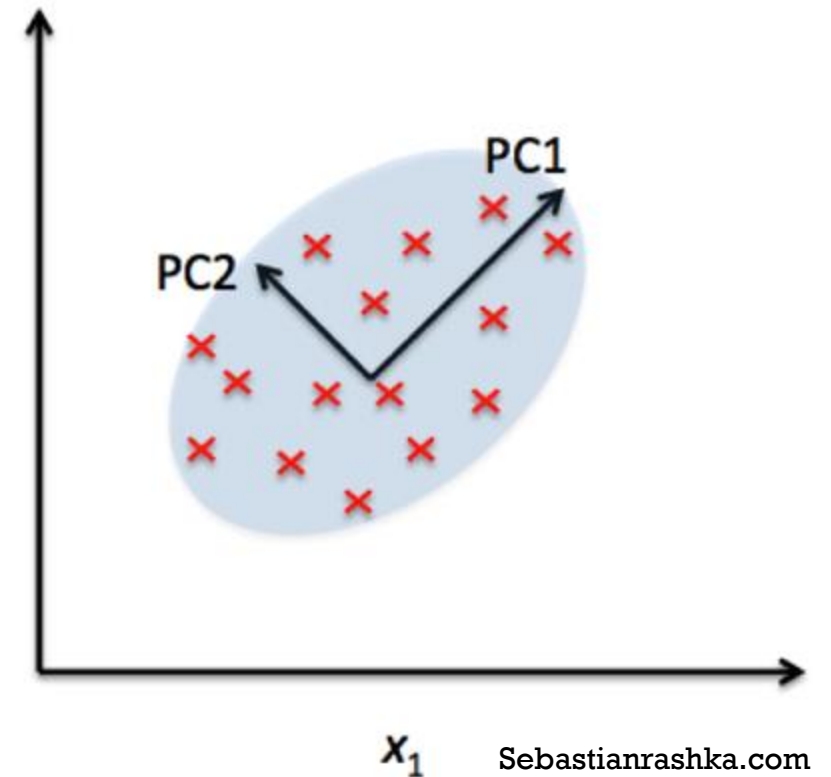


COMPONENTES PRINCIPALES

PCA: Principal Component Analysis

Objetivo: Simplificar el dataset, encontrando una representación de **baja dimensionalidad** que conserva la mayor parte de la información

- **Combinación lineal** de las dimensiones (atributos) originales del dataset que maximiza la x_2 varianza
- **Rotación** de los ejes originales
- Permite una **visualización** los datos en problemas de aprendizaje supervisado y no supervisado
- Se limitan las dimensiones que estén altamente **correlacionadas** entre ellas



Sebastianrashka.com



COMPONENTES PRINCIPALES

- Hay tantos componentes principales (PCs) como dimensiones, ortogonales entre ellos
- Cada PC es una combinación lineal normalizada de los atributos del dataset (X_1, X_2, \dots, X_N) :

$$PC_i = \Phi_{1i}X_1 + \Phi_{2i}X_2 + \dots + \Phi_{Ni}X_N, \quad \text{con} \quad \sum_{j=1}^N \Phi_{ji}^2 = 1$$

- Cada PC tiene asociada una carga o **loading** de cada una de las dimensiones originales (los Φ_{ji}). El vector de loadings de un PC indica su dirección
- A cada PC se le puede establecer la cantidad de información original especificada. Esta va decreciendo con cada PC considerado, por lo que los primeros **p** PCs van a representar mucha más información que las primeras **p** dimensiones originales
- Las instancias originales se proyectan en el espacio dado por los primeros **p** PCs



COMPONENTES PRINCIPALES

Consideraciones

- La varianza de cada uno de los atributos (dimensiones) originales depende de su escala, por lo que se debe **normalizar** los datos originales
- El número de dimensiones originales no puede ser superior al número de instancias del dataset
- Puede que la varianza este bien distribuida en los atributos originales, por lo que aplicar PCA no tendría efecto



TALLER: COMPONENTES PRINCIPALES

Descargar el taller de componente principales que analiza los niveles de criminalidad de diferentes estados de USA.



REFERENCIAS

- *Python Machine Learning*, Sebastian Raschka, Packt, 2015
- *Introduction to Statistical Learning with Applications in R (ISLR)*, G. James, D. Witten, T. Hastie & R. Tibshirani, 2014
- EMC2, “Data science and big data analytics”, 2015, John Wiley & Sons
- *Data Science for Business*, Foster Provost & Tom Fawcett, O'Reilly, 2013
- *Practical Data Science with R*, Nina Zumel & John Mount, 2014

