IMT2200

INTRODUCCIÓN A CIENCIA DE DATOS

2022-2

CLUSTERING

CLASE 23

REPASO: APRENDIZAJE SUPERVISADO

- El objetivo es realizar predicciones precisas para *nuevos datos* con características similares a los datos usados para construir el modelo → generalización
- **Entrenamiento y testeo:** el conjunto de datos de ejemplo $\mathcal{D}$ se divide aleatoriamente en dos
  - Datos de entrenamiento ($\mathcal{D}_{\text{train}}$) → para entrenar el modelo
  - Datos de prueba ($\mathcal{D}_{\text{test}}$) → para evaluar qué tan bien funciona el modelo frente a datos que no conoce. No se usa para el entrenamiento.
  - Etiquetas (y) → valores o categorías asignadas a los datos de ejemplo

$$\mathcal{D} \rightarrow \mathcal{D}_{\text{train}} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \quad 70 - 80\%$$

ALEATORIO

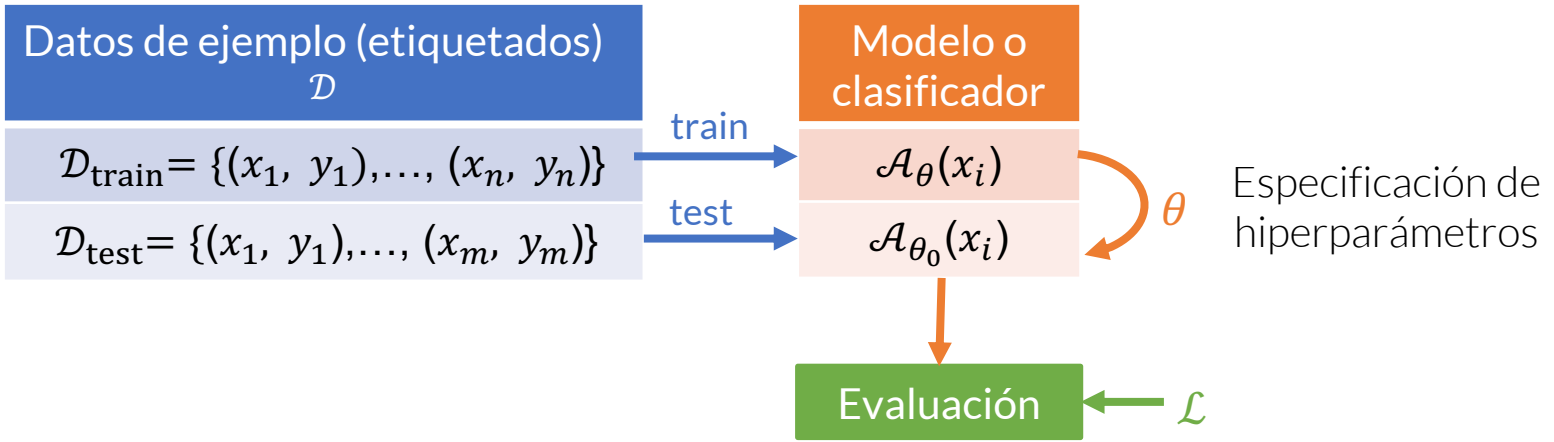
$$\mathcal{D} \rightarrow \mathcal{D}_{\text{test}} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\} \quad 20 - 30\%$$

- Función de pérdida ($\mathcal{L}$) → función que mide la diferencia o pérdida entre la predicción y el valor real de una etiqueta o valor.

REPASO: APRENDIZAJE SUPERVISADO

- El objetivo es realizar predicciones precisas para *nuevos datos* con características similares a los datos usados para construir el modelo → generalización

- Entrenamiento y testeo:
  - Hiperparámetros (θ) → parámetros libres del modelo que no son determinados por el algoritmo, sino entregados como input



09:56

REPASO: APRENDIZAJE SUPERVISADO

Entrenamiento y testeo: el conjunto de datos de ejemplo $\mathcal{D}$ se divide aleatoriamente en dos

- Datos de entrenamiento ($\mathcal{D}_{\text{train}}$) ➔ para entrenar el modelo
- Datos de prueba ($\mathcal{D}_{\text{test}}$) ➔ para evaluar qué tan bien funciona el modelo frente a datos que no conoce. No se usa para el entrenamiento.
- Etiquetas (y) ➔ valores o categorías asignadas a los datos de ejemplo

$$\mathcal{D} \rightarrow \mathcal{D}_{\text{train}} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \quad 70 - 80\%$$

ALEATORIO

$$\mathcal{D} \rightarrow \mathcal{D}_{\text{test}} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\} \quad 20 - 30\%$$

```
#seleccionamos las variables predictoras y la variable dependiente
X = df[['Age','EstimatedSalary']]
y = df['Purchased']

#datos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 5)
```

09:56

REPASO: APRENDIZAJE SUPERVISADO

Datos de ejemplo (etiquetados) $\mathcal{D}$

```
#Creamos los datos de entrenamiento y prueba
from sklearn.model_selection import train_test_split

#datos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3)
```

Modelo o clasificador

$\mathcal{A}_\theta(x_i)$

$\mathcal{A}_{\theta_0}(x_i)$

```
#definimos el modelo y especificamos sus hiperparámetros
classifier = KNeighborsClassifier(n_neighbors = 15)
```

```
#ajustamos a los datos de entrenamiento
classifier.fit(X_train, y_train)
```

train

Evaluación

```
#predecimos las categorías para el dataset de prueba
y_pred_test = classifier.predict(X_test)
```

$\mathcal{L}$

```
from sklearn import metrics

#matriz de confusión
cm=metrics.confusion_matrix(y_test,y_pred_test) #true,pred
```

09:56

COMENTARIOS TAREA 4

- Regresión multilineal con regularización: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots + \beta_J X_J$

$$L_{LASSO}(\beta) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \beta^{\mathrm{T}} Xi|^2 + \alpha \sum_{m=1}^{M} |\beta_m|$$
$$L_{Ridge}(\beta) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \beta^{\mathrm{T}} Xi|^2 + \alpha \sum_{m=1}^{M} {\beta_m}^2$$

Al combinar variables con distintos rangos de valores, se penalizará más fuertemente aquellas con mayor peso en la regresión (i.e. mayor β_j)

➔ Es necesario **normalizar los datos** antes de entrenar el model (Ej: `StandardScaler`)

- Uds. deben determinar qué variables considerar en el modelo final.
- **Analizar y justificar** todas las decisiones y respuestas. Se evaluará la calidad de los argumentos.

09:56

COMENTARIOS I2

- Fecha: martes 15-11-2022, 18:30-20:30
- Sala: B22
- Contenidos: clases 13 – 22.

- Preguntas de desarrollo/alternativas + ejercicios de programación.
- Pueden consultar apuntes, material de clases, etc.

- Recomendaciones:
  - Estudiar clases y notebooks
  - Comprender conceptos
  - Asegurarse de comprender qué hace cada código/método utilizado

- Cargar su computador
- Traer enchufes/ alargadores

APRENDIZAJE NO SUPERVISADO

Algoritmos de aprendizaje de máquina donde **no existe un output conocido**, sino que se pide al algoritmo extraer conocimiento a partir de la data de entrada.

En general, hay **dos grandes clases** de algoritmos de aprendizaje no supervisado:

1. Clustering ➔ algoritmos para particionar la data en grupos disímiles pero con ítems similares.

2. Transformaciones ➔ algoritmos que crean una nueva representación de los datos que puede ser más fácil de entender para humano o máquinas en comparación a la representación original.

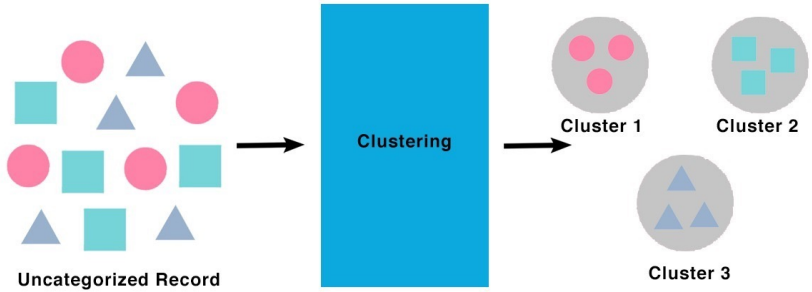
- Reducción de dimensionalidad.
- Extracción de tópicos

Desafío: evaluar si el algoritmo aprendió lo que se buscaba. Muchas veces se requiere inspeccionar o evaluar manualmente los resultados.

- Aprendizaje no supervisado se utiliza muchas veces en una etapa de exploración, o como parte del preprocesamiento para algoritmos supervisados.

CLUSTERING

- Algoritmos para particionar la data en grupos disímiles pero con ítems similares.
- Cada observación del dataset es asignada a sólo una categoría según el valor de las variables consideradas para la clasificación.
- Los clusters se definen de manera que cada uno agrupe observaciones similares entre sí, pero difieran entre grupos.
- El análisis se simplifica enfocándose sólo en los perfiles de cada cluster, que permiten interpretar la estructura de data multi-variable.

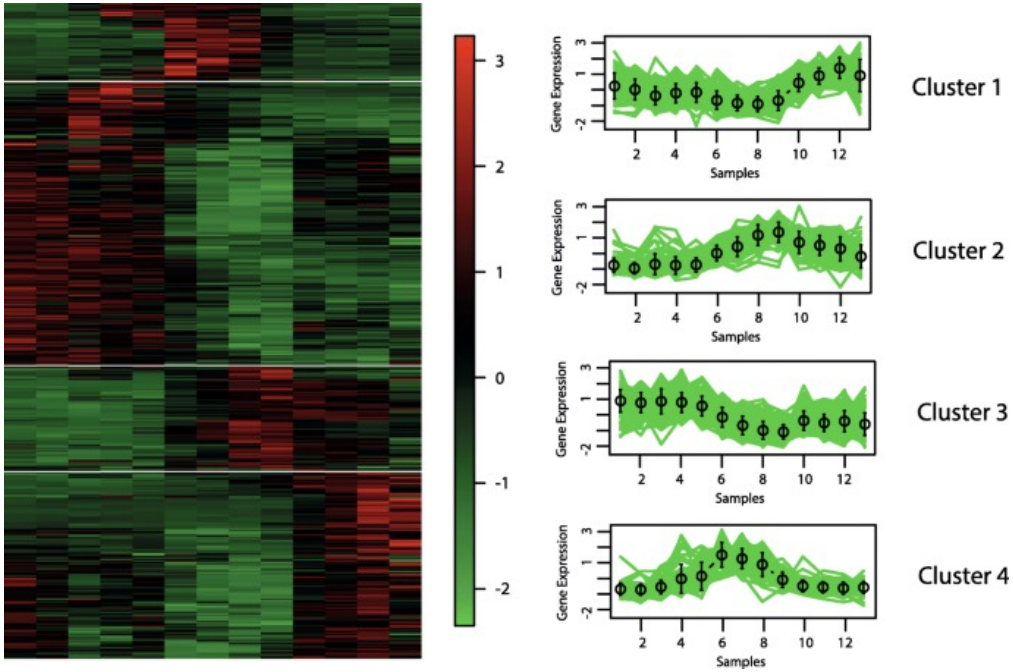


CLUSTERING: EJEMPLOS

A multi-objective gene clustering algorithm guided by apriori biological knowledge with intensification and diversification strategies

Jorge Parraga-Alava, Marcio Dorn & Mario Inostroza-Ponta

BioData Mining **11**, Article number: 16 (2018) | Cite this article

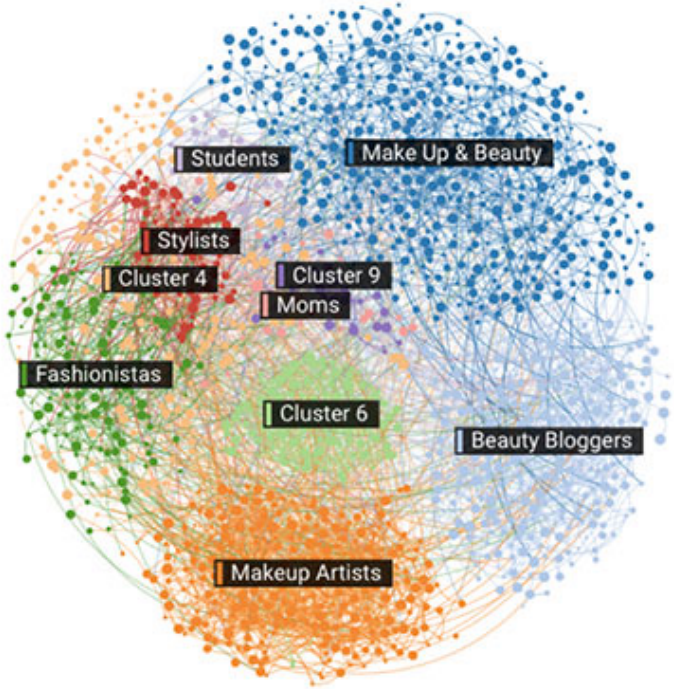


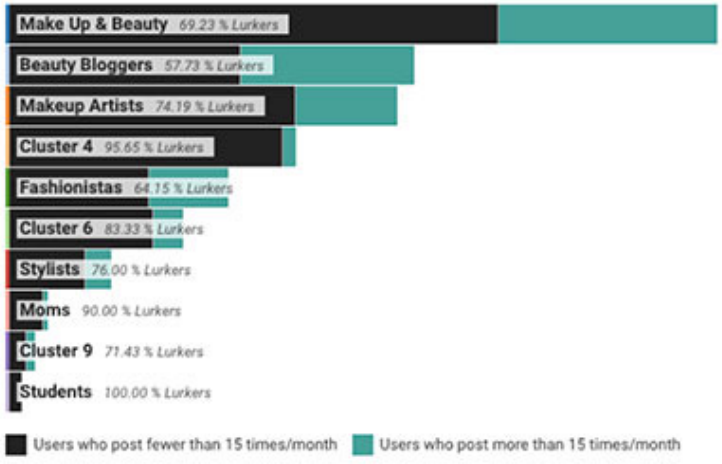
09:56

CLUSTERING: EJEMPLOS

Affinio Deep Learning Social Network Analysis Clustering

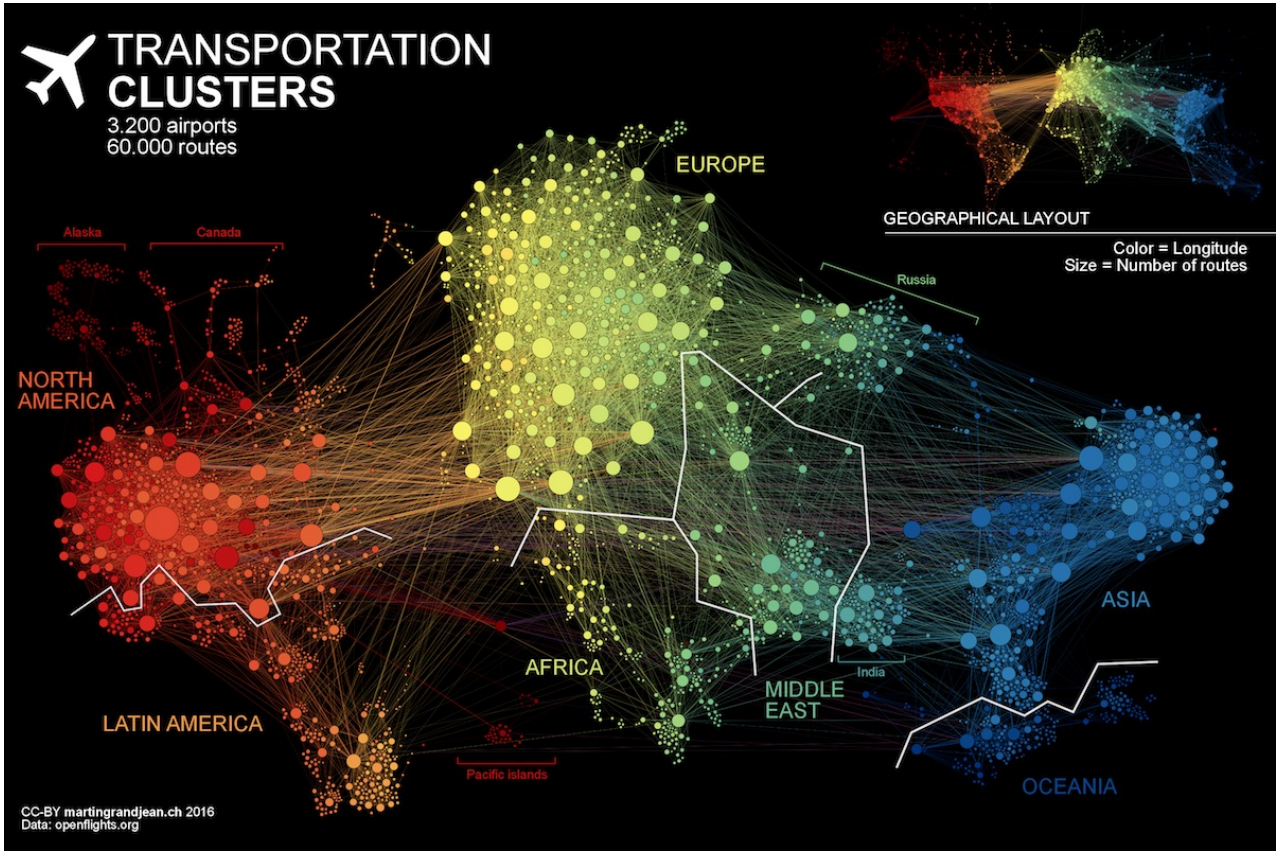
by Mariya Yao | Mar 5, 2018





09:56

CLUSTERING: EJEMPLOS

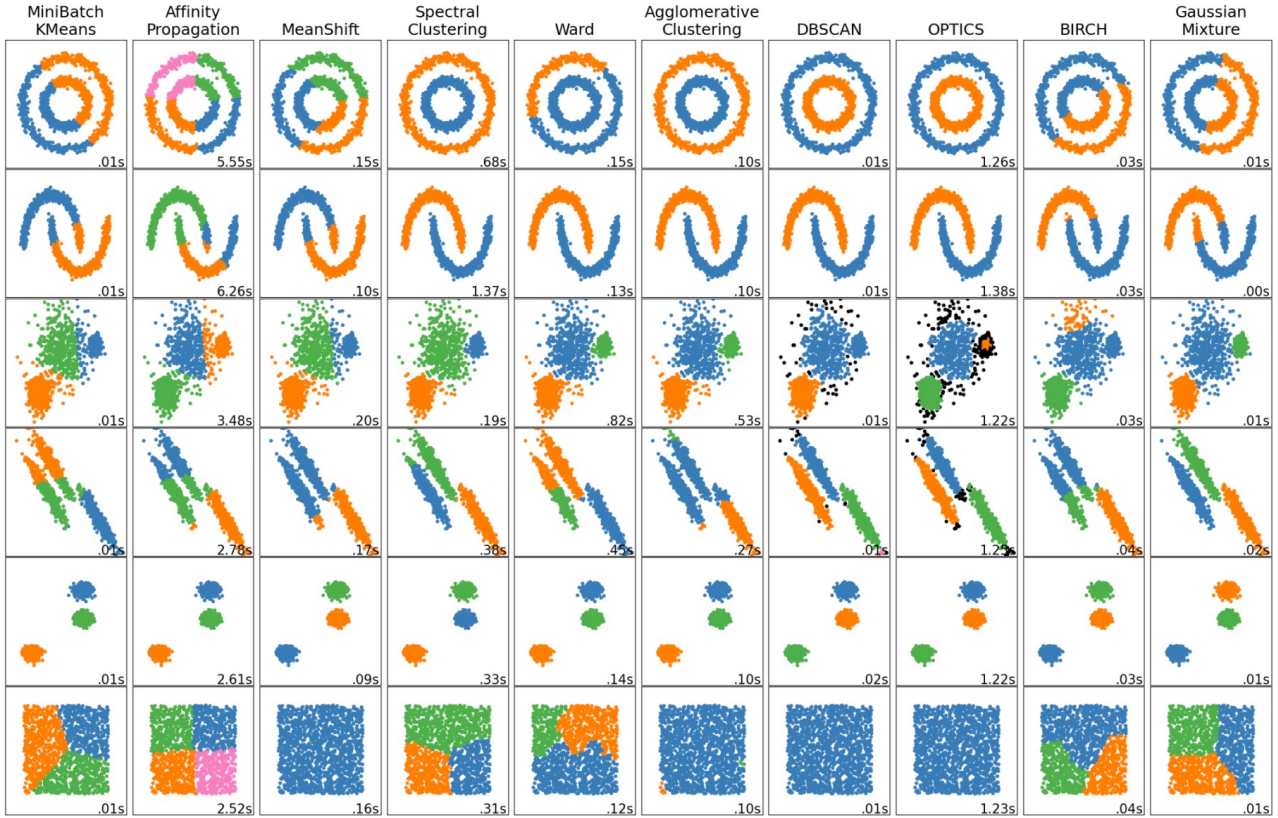


MÉTODOS DE CLUSTERING

Existen **múltiples tipos de clustering** que difieren en:

- Cómo se define la similaridad entre observaciones y disimilaridad entre grupos

- Operacionalización del algoritmo de clasificación



A comparison of the clustering algorithms in scikit-learn

09:56

MÉTODOS DE CLUSTERING

Algunos de los métodos más usados son:

- **Métodos de partición :** requieren que el usuario especifique de antemano el número de clusters.

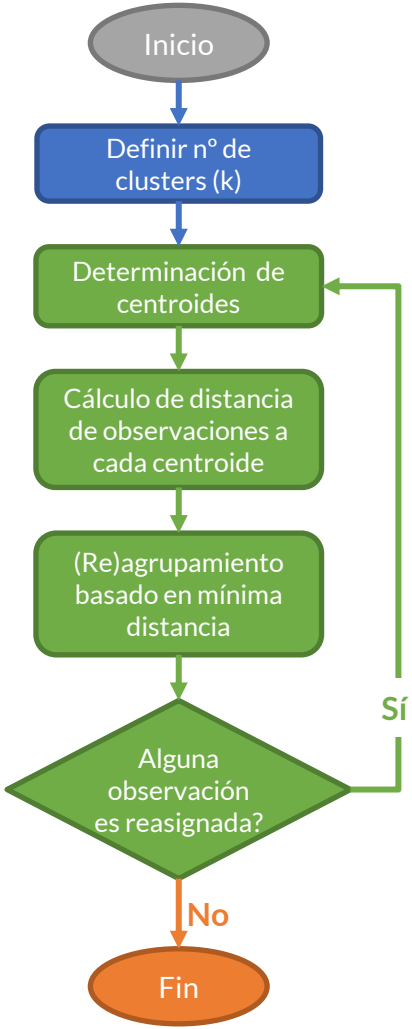
- **Métodos jerárquicos :** se busca crear una jerarquía para la fusión de observaciones fusionadas en clusters. No requiere que se pre-especifique el número de clusters.
  - Aglomerativos: el agrupamiento se inicia con todas las observaciones separadas, cada una formando un cluster individual. Los clusters se van combinado a medida que la estructura crece hasta converger en uno solo.
  - Divisivo: Se inicia con todas las observaciones contenidas en un mismo cluster y se suceden divisiones hasta que cada observación forma un cluster individual.

- **Métodos basados en densidad:** forma de identificar clusters siguiendo el modo intuitivo en el que lo hace el cerebro humano, identificando regiones con alta densidad de observaciones separadas por regiones de baja densidad.

CLUSTERING k-MEANS

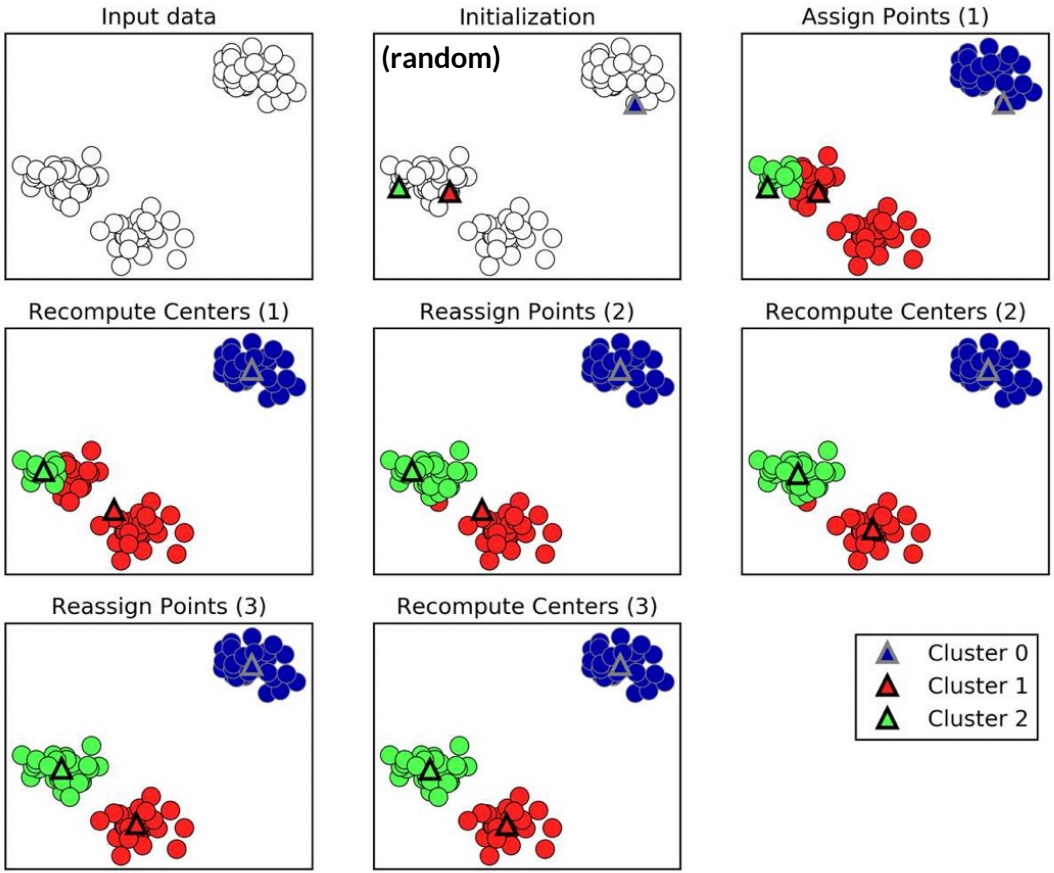
- **Método de partición:** cada observación as asignada a uno de ***k* clusters (predefinido)**, de manera que la observación que está más cerca del centro de su cluster, que de cualquier otro.

- El problema se resuelve mediante un **proceso iterativo** de asignación y actualización:
  - Primero, se genera un centro aleatorio para cada cluster.
  - Se calcula la media multivariada (centroide) de cada cluster.
  - Se calcula la distancia entre los centros y observaciones para asignar cada dato al cluster más cercano.
  - Se recalcula el centro de los clusters
  - El proceso de continúa hasta que no son necesarias nuevas reasignaciones.



09:56

CLUSTERING k-MEANS



El proceso continúa hasta que no son necesarias nuevas reasignaciones.

09:56

CLUSTERING k-MEANS

¿Cuándo termina el proceso?

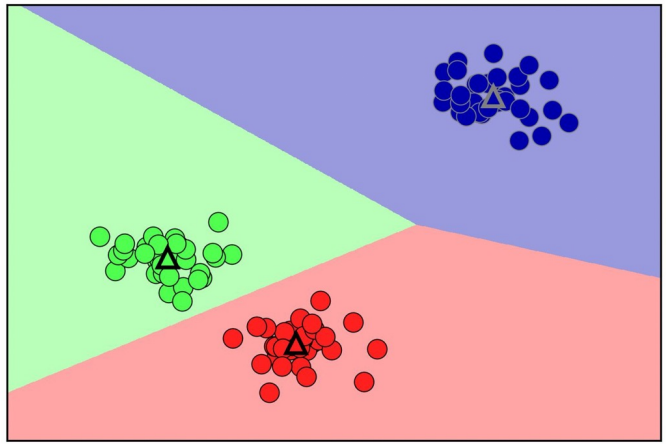
- El objetivo es lograr una agrupación que **minimice la disimilaridad dentro de cada cluster**.
- **Disimilaridad** ➔ suma de distancias euclideanas o errores cuadráticos (SSE) entre cada observación y el centroide de su cluster. También se denomina **inercia** de clusters.

$$SSE = \sum_{i=1}^{n} \sum_{j=i}^{k} w^{i,j} ||\boldsymbol{x}^i - \boldsymbol{\mu}^j||^2$$

Δ μ^j: centroide del cluster j

o x^i: observación i

$$w^{i,j} = \begin{cases} 1 \text{ si la observación } x^i \text{ está en el cluster } j \\ 0 \text{ en otro caso} \end{cases}$$



➔ El algoritmo continúa hasta que ya no es posible reducir SSE.

09:56

CLUSTERING k-MEANS: PREPROCESAMIENTO

- La función objetivo o disimilaridad es sensible a la escala de las variables:
  - Variables de un dataset pueden tener unidades no-comparables (Ej: dimensiones, m, kg, $)
  - Variables con las mismas unidades pueden tener muy distintas escalas de valores y varianzas

- Si una variable tiene una escala mucho mayor que el resto, determinará en gran medida el valor de distancia/similitud obtenido al comparar las observaciones, dirigiendo así la agrupación final.

- Para evitar que las variables con mucha varianza dominen la función objetivo, es común **estandarizar las variables** antes de clusterizar:

  - **Estandarización z** ➔ restar la media y dividir por la desviación estándar

$$z = \frac{x - \mu}{\sigma}$$

CLUSTERING k-MEANS: IMPLEMENTACIÓN

sklearn.cluster.KMeans

class sklearn.cluster.KMeans(*n_clusters=8*, ***, *init='k-means++'*, *n_init=10*, *max_iter=300*, *tol=0.0001*, *verbose=0*, *random_state=None*, *copy_x=True*, *algorithm='auto'*) [source]

Attributes:

cluster_centers_ : *ndarray of shape (n_clusters, n_features)*
Coordinates of cluster centers. If the algorithm stops before fully converging (see `tol` and `max_iter`), these will not be consistent with `labels_`.

labels_ : *ndarray of shape (n_samples,)*
Labels of each point

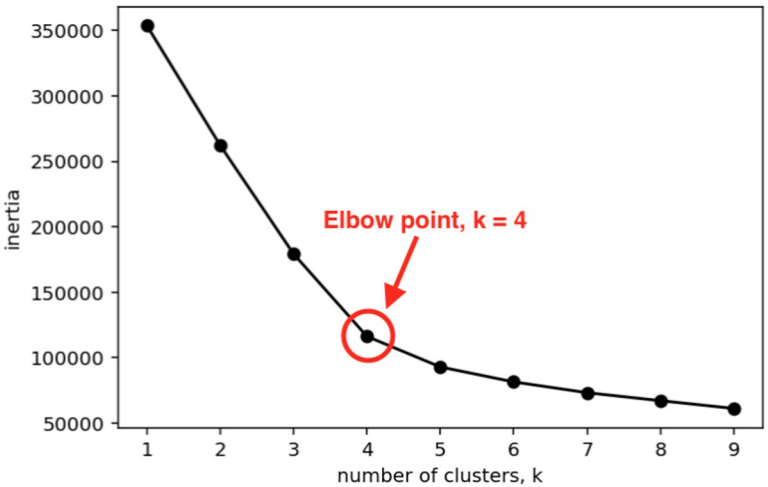
inertia_ : *float*
Sum of squared distances of samples to their closest cluster center, weighted by the sample weights if provided.

CLUSTERING k-MEANS: SELECCIÓN DE *k*

- El algoritmo requiere definir a priori el **número de clusters (k):** ¿cómo lo elegimos?

- Típicamente, se consideran varios valores para k , y se comparan los clusters resultantes en términos de la función objetivo (SSE).

- **Método del codo:**
  - Se grafica la función objetivo vs. valores crecientes de k
  - Mientras no se alcance el número óptimo de clusters, la función objetivo sigue mejorando sustancialmente,
  - Cuando se excede k óptimo, la curva se aplana.



09:56

CLUSTERING k-MEANS: LIMITACIONES

- El algoritmo requiere definir a priori el **número de clusters (k)** ➔ no siempre lo sabremos.

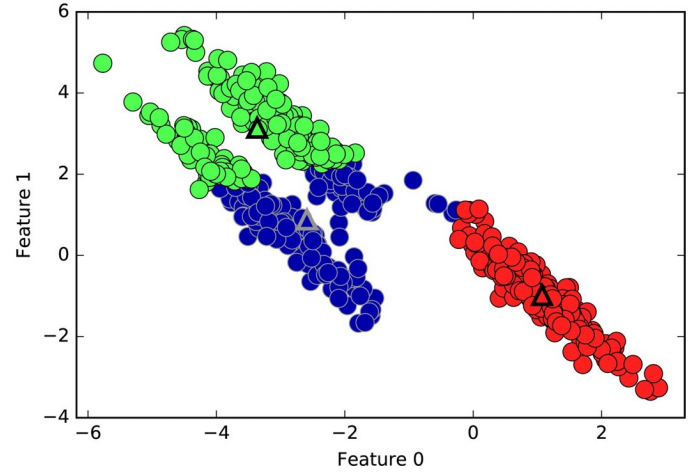
- Los clusters se inicializan aleatoriamente ➔ puede que el algoritmo no converja, o que el resultado sea inestable.

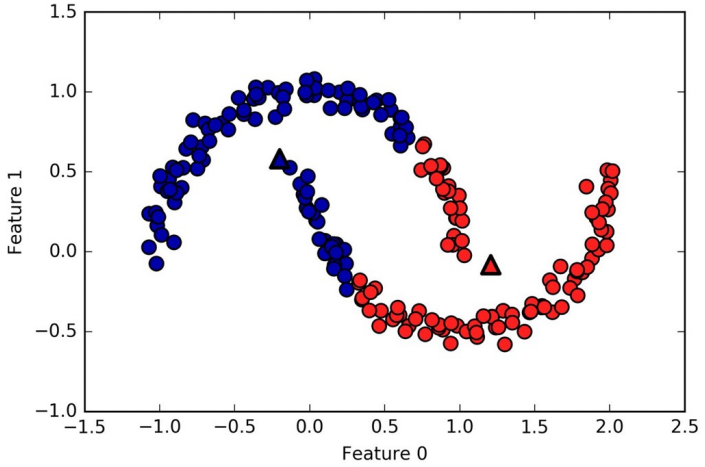
  ➔ `sklearn` realiza 10 iteraciones por defecto

- El algoritmo asume que cada punto pertenece al cluster más cercano, y que todas las direcciones son igualmente importantes

  ➔ Permite identificar clusters esféricos, pero **no funciona bien para datos con geometrías complejas**.

- En sensible a outliers.
- Puede ser lento para grandes cantidades de observaciones, ya que en cada paso hay que calcular todos los pares de distancias.



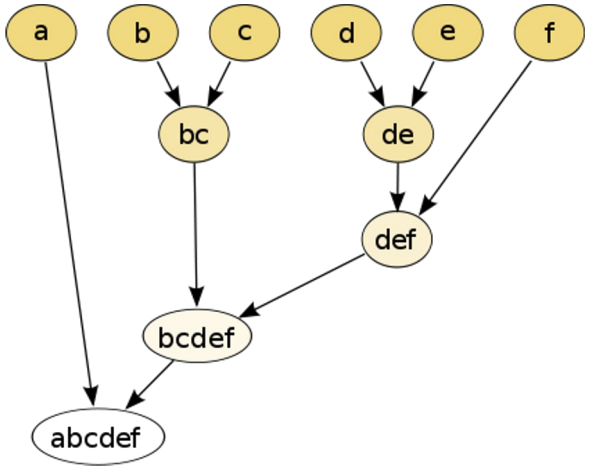


09:56

CLUSTERING AGLOMERATIVO (JERÁRQUICO)

Familia de algoritmos en que se construyen clusters anidados a partir de **fusiones sucesivas** de clusters.

- Primero, todos los puntos son considerados como clusters individuales
- Se calculan las distancias entre todos los pares de centroides (matriz de distancias) de clusters y **se unen los dos más cercanos.**
- Se recalcula el centro del cluster fusionado.
- El proceso de unión de clusters continua hasta llegar al número deseado de clusters.

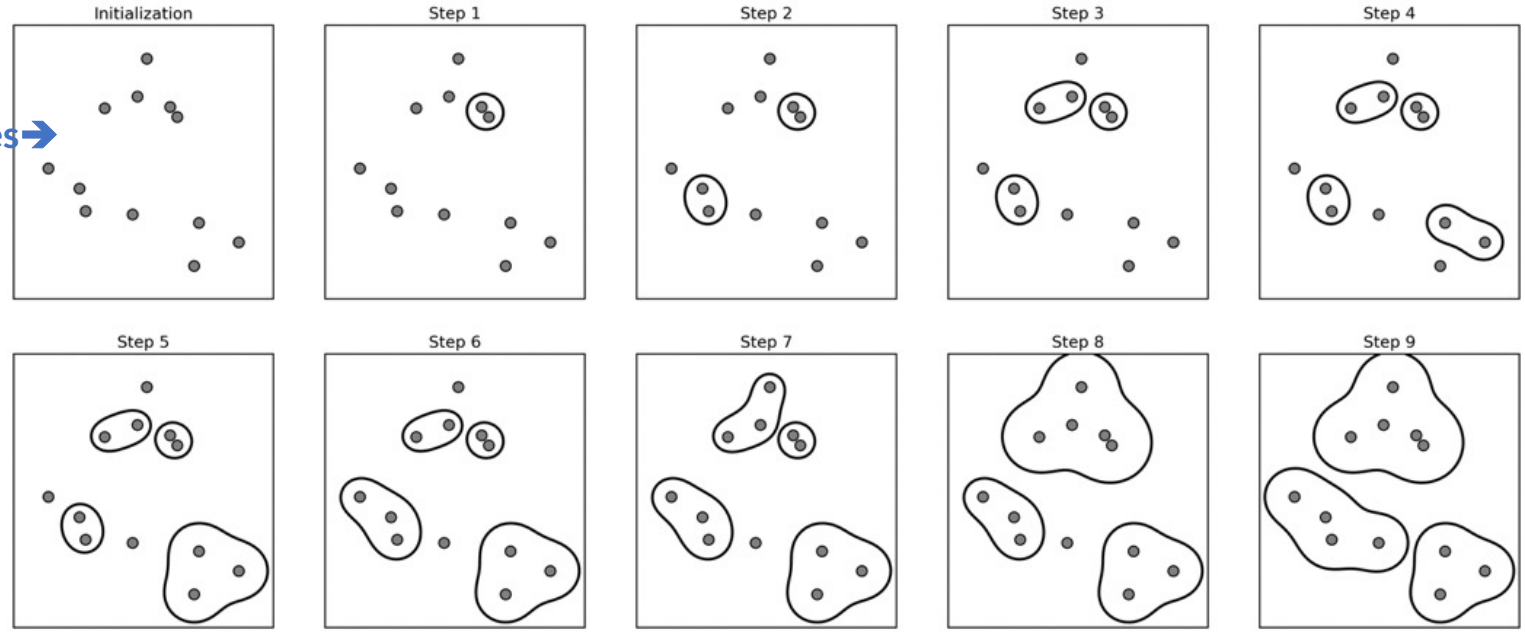


09:56

CLUSTERING AGLOMERATIVO

Clusters individuales

Observaciones ➔

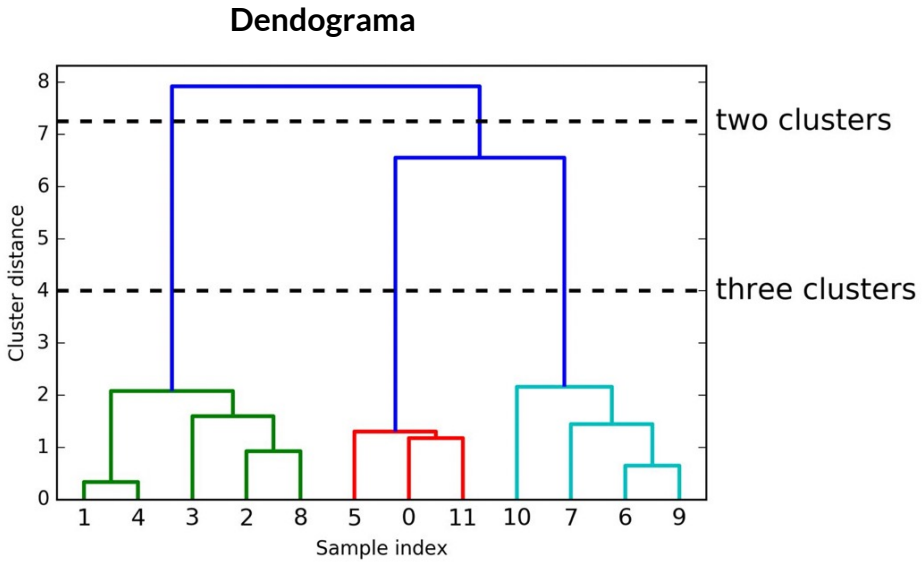


Se calculan las distancias entre todos los pares de centroides de clusters y se fusionan los más cercanos.

09:56

CLUSTERING AGLOMERATIVO (JERÁRQUICO)

- Para datasets multidimensionales, el proceso de fusión de clusters puede visualizarse mediante un dendograma:



- Cada observación (leaf) es inicialmente un cluster.
- Los nodos muestra fusiones de clusters.
- La distancia vertical (branch) representa la distancia entre los clusters que se fusionan.