Aprendizaje No Supervisado

Tutoría

{desafío} latam_

¿Qué se entiende por Machine Learning? ¿Qué tipos de aprendizaje existen?

Autoaprendizaje

Recursos asincrónicos

- ¿Revisaste los recursos de la semana 2 (Guía y desafío)?
- ¿Tienes dudas sobre alguno de ellos?

{desafío} latam_

Ideas fuerza

Clustering *es una* ***técnica de análisis de datos*** *en el aprendizaje no supervisado que tiene como objetivo* ***agrupar objetos similares*** *en conjuntos llamados* ***clusters.***

Algunos algoritmos de clustering son K Means, Fuzzy C Means y Cluster Jerárquicos. Cada uno tiene sus ventajas y desventajas propias.

Existen diferentes ***métodos de validación*** *de algoritmos de clustering, a través de* ***métricas de calidad*** *y* ***criterios cualitativos*** *a utilizar.*

{desafío} latam_

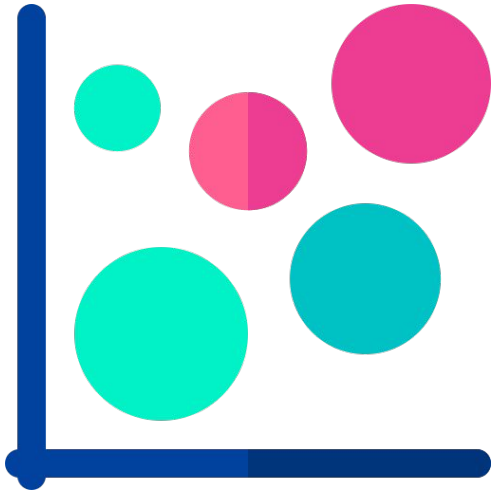
/* Clustering */

Clustering

¿Qué es?

El **clustering** es un conjunto de técnicas de aprendizaje no supervisado que buscan **agrupar objetos similares entre sí** a partir de los datos con los que se intenta describir los objetos a agrupar. Así, la definición de los grupos va a depender de qué variables se utilizan para generar los clusters.

Un **cluster** es una colección o conjunto de objetos que comparten características similares entre sí y difieren de otros objetos en el conjunto de datos. Estos son los resultantes de aplicar un algoritmo de clustering a un set de datos.



/* Métodos de evaluación de clusters */

{desafío}
latam_

Validación cuantitativa de clustering

Distancia Intra Cluster (SSE)

La distancia intracluster mide la coherencia de los objetos dentro de un mismo grupo o cluster. Cuanto menor sea la distancia intracluster, mayor será la cohesión dentro del cluster. Se busca que los objetos dentro del mismo cluster estén lo más cerca posible unos de otros.

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} (x - m_i)^2$$

m_i: corresponde al promedio del cluster i.
C_i: corresponde al conjunto de muestras del cluster i

Validación cuantitativa de clustering

Coeficiente de Silhouette

Mide la cohesión y separación de los clusters. Proporciona un valor entre -1 y 1, donde un valor cercano a 1 indica una buena separación entre los clusters y una cohesión interna alta.

Se puede calcular para puntos individuales, como también para clusters (promedios)

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

a: distancia promedio de i a los puntos de su propio cluster
b: mínimo distancia promedio de i a puntos de otro cluster

Validación cuantitativa de clustering

Índice de Davies-Bouldin

Mide la similitud media entre los clusters y la distancia entre los centroides de los clusters. Un valor más bajo indica una mejor separación y cohesión de los clusters.

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

$$DB = \frac{1}{k} \sum_{i-1}^{k} max_{i \neq j} R_{ij}$$

s_i: promedio de distancia de cada punto del cluster i al centroide del cluster
d_{ij}: distancia entre el centroide del cluster i y j

Validación de Expertos

La validación por expertos implica la participación de personas con conocimiento y experiencia en el dominio de los datos para evaluar y validar los resultados del clustering.

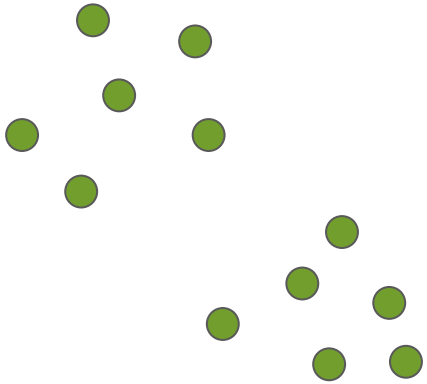
- Revisar y evaluar visualmente los clusters generados en función de su conocimiento y experiencia.
- Interpretar y asignar etiquetas o categorías a los clusters identificados.
- Evaluar la coherencia y relevancia de los grupos generados en relación con los objetivos del análisis.

/* Algoritmos de Clustering */

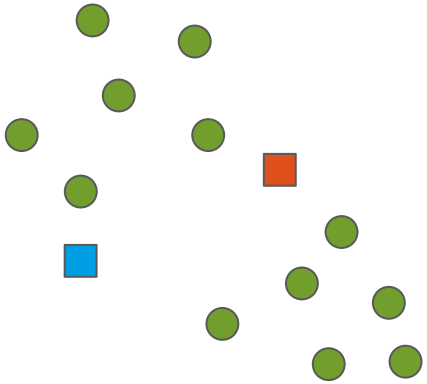
Algoritmos de clustering

KMeans

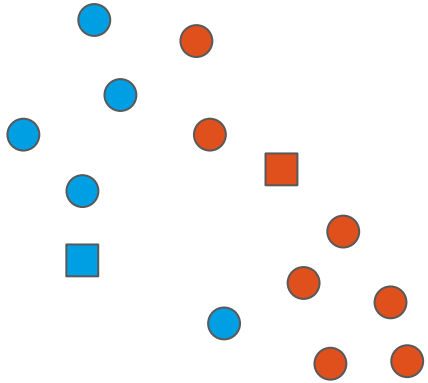
K = 2



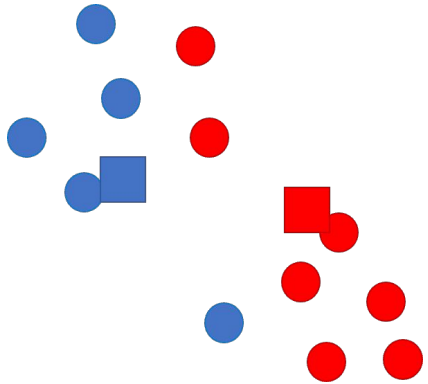
Centroides



Agrupamos



Recalculamos

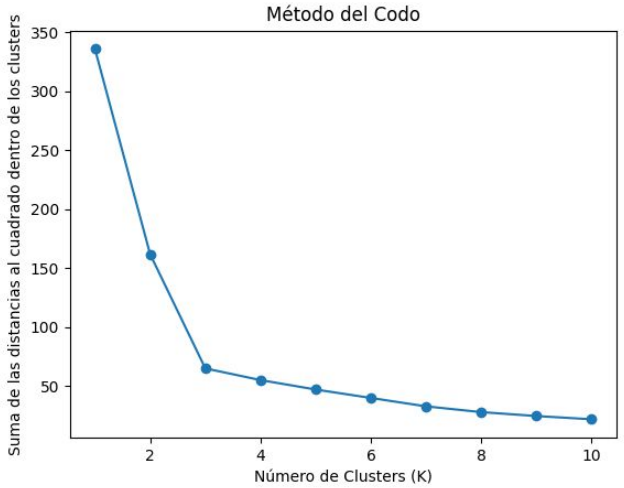


/* Método del codo*/

Método del codo

Se utiliza para determinar el número óptimo de clusters en el algoritmo de clustering. La idea principal es identificar el valor de K para el que se produce un cambio significativo en la variabilidad explicada por los clusters.

La idea es ejecutar el algoritmo para diferentes número de clusters, y graficar cómo varía la distancia intracluster. En la figura, se observa que para 3 clusters se produce un “codo”.



/* Fuzzy C-means */

Fuzzy C-means

Algoritmo

1. **Inicialización**: Se selecciona el número de clusters (K) y se inicializan aleatoriamente los centroides y los grados de pertenencia para cada punto de datos. (valores entre 0 y 1)
2. **Cálculo de los centroides:** Se calculan los centroides ponderando con los grados de pertenencia.
3. **Actualización de los grados de pertenencia:** Se actualizan los grados de pertenencia de cada punto de datos utilizando una función de pertenencia basada en la distancia a los centroides.
4. **Iteración**: se iteran los pasos anteriores, como con KMean
5. **Resultados**: Al finalizar las iteraciones, se obtienen los centroides finales y los grados de pertenencia de los puntos de datos.

Fuzzy C-means

Pertenencia

La fórmula de actualización de grados de pertenencia es

$$u_{ij} = \left(\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}} \right)^{-1}$$

donde:

- u_{ij} es el grado de pertenencia del punto i al cluster j
- d_{ij} es la distancia del punto i al cluster j
- m es el **parámetro de fuzziness.** Para él se tiene que:
    - Si m = 1 la pertenencia es binaria (KMeans)
    - Si m > 1, la pertenencia es difusa. Cuanto mayor sea el valor de m, más difusas serán las asignaciones.

¡Manos a la obra!
Clusters con Fuzzy C means

{desafío}
latam_

Ejercicio

Clusters con Fuzzy C Means

Veremos ahora cómo podemos utilizar Python para aplicar Fuzzy CMeans. Para ello, observa los pasos que te mostrará tu profesor en Jupyter Notebook. Puedes abrir tú también un archivo para reproducir los pasos.

Aprenderemos a:

1. Determinar el valor óptimo del parámetro m
2. Determinar los clusters usando Fuzzy CMeans
3. Validar el método con indicadores numéricos

/* Cluster Jerárquicos */

Cluster Jerárquicos

¿Qué son?

Corresponden a un método de agrupamiento que crea una jerarquía de clusters de manera recursiva.

Existen 2 grandes tipos y son:

1. **Aglomerativos**: Comienzan con cada punto de los datos como un cluster individual, y se van fusionando los cluster hasta tener un solo cluster con toda la data.

2. **Divisivo**: Comienzan con todos los puntos en un solo cluster, y van paso a paso dividiendo en cluster pequeños hasta que cada punto sea un cluster.

Cluster Jerárquicos

Criterios de enlace

1. **Simple Linkage:** La menor distancia entre 2 puntos en los grupos.

$$d_{single}(G, H) = min_{i \in G, j \in H}(d_{ij})$$

2. **Complete Linkage:** La mayor distancia entre 2 puntos en los grupos.

$$d_{complete}(G, H) = max_{i \in G, j \in H}(d_{ij})$$

3. **Average Linkage:** La distancia promedio entre todos los puntos del grupo opuesto.

$$d_{average}(G, H) = \frac{1}{n_G n_H} \sum_{i \in G, j \in H} d_{ij}$$

4. **Enlace de Ward (Ward's linkage)**: Minimiza la suma de las diferencias cuadradas dentro del cluster al fusionar dos clusters.

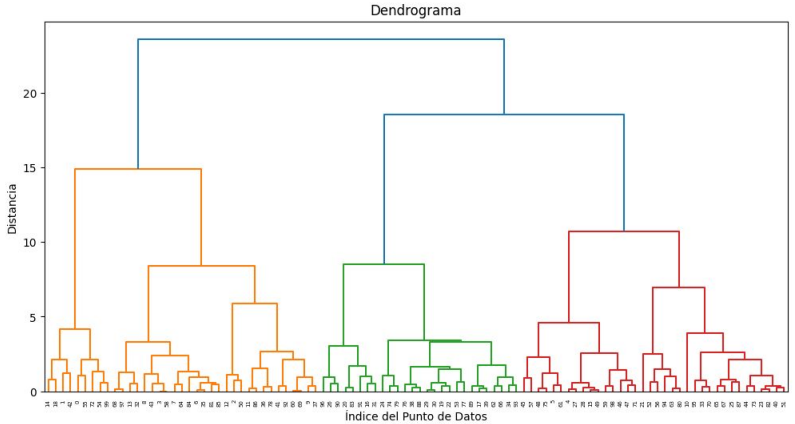
$$d_{Ward} = (G, H) = |G - H|^2$$

Cluster Jerárquicos

Dendogramas

Un dendrograma es una representación visual de la estructura jerárquica de los clusters.

Los puntos de datos se muestran en la parte inferior del dendrograma y se agrupan en clusters a medida que se mueven hacia arriba.



¡Manos a la obra!
Clusters jerárquicos

{desafío}
latam_

Ejercicio

Clusters jerárquicos

Veremos ahora cómo podemos utilizar Python para generar clusters jerárquicos. Para ello, observa los pasos que te mostrará tu profesor en Jupyter Notebook. Puedes abrir tú también un archivo para reproducir los pasos.

Aprenderemos a:

1. Generar clusters jerárquicos aplicando diferentes métodos de enlace
2. Crear dendogramas
3. Identificar el mejor método en cada caso

{desafío} latam_

Desafío - Segmentación de clientes

{desafío}
latam_

Desafío

"Segmentación de clientes"

- ¿Leíste el desafío de esta semana? ¿Comprendes bien lo que se solicita en cada caso?
- ¿Hay contenidos que necesitas repasar antes de comenzar este desafío?
- ¿Necesitas algún ejemplo o indicación para alguna pregunta o requerimiento específico?

{desafío}
latam_

{desafío}
latam_

Academia de
talentos digitales