APRENDIZAJE DE MÁQUINA

3/14/23

APRENDIZAJE NO SUPERVISADO



Programa

Aprendizaje no supervisado

Algoritmos de Clustering

Clustering K-means

Método del Codo

- > Descripción general del aprendizaje no supervisado
 - Descripción
- Algoritmos de Clustering
 - Clustering K-means
 - DBSCAN
 - Mean Shift



En la sesión pasada, aprendimos mucho sobre algoritmos de aprendizaje supervisado.

Ahora, pasamos a algoritmos de **aprendizaje no supervisados**:

¿qué son, cuándo son útiles y qué papel juegan en el futuro del aprendizaje automático?



Resumen: aprendizaje supervisado - no supervisado

Aprendizaje Supervisado

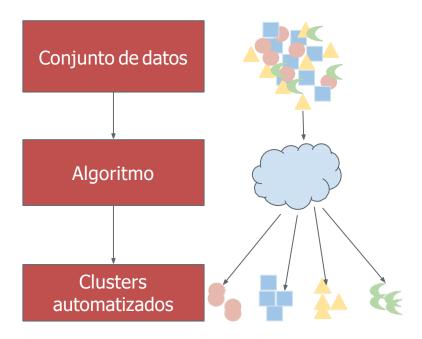
- Por cada x, hay un y
- El objetivo es predecir y usando
 x
- En la práctica, la mayoría de los métodos utilizados son supervisados.

Aprendizaje no Supervisado

- Por cada x, no hay y
- El objetivo no es la predicción, sino investigar x.
- Los métodos no supervisados leen primero los datos y luego sugieren qué esquema(s) de clasificación podrían aplicarse.



Muchos algoritmos de aprendizaje no supervisado implican la identificación de patrones.

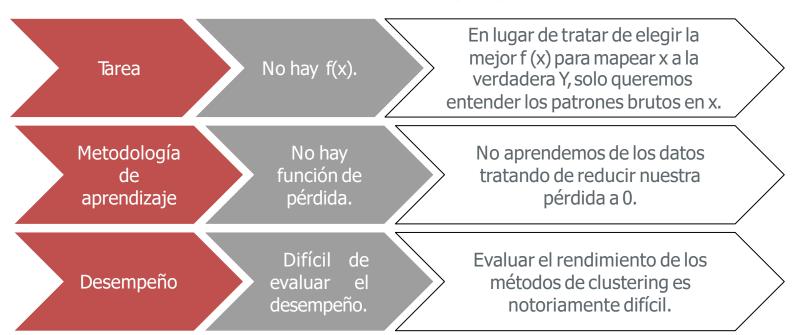


El aprendizaje no supervisado implica **identificar patrones.**

Esta es quizás la tarea humana más básica - incluso los bebés pueden hacerlo



¿Cómo es este modelo diferente de lo que ya hemos visto?





Los algoritmos no supervisados todavía tienen una tarea e involucran ingeniería y selección de características.

Tarea

¿Cuál es el problema que queremos que resuelva nuestro modelo?

¿Cuál es el enfoque no supervisado que planeamos usar?

Ingeniería & selección de características incluir en nuestro modelo?



Los algoritmos no supervisados siguen siendo algoritmos de machine learning, lo que significa que aprenden activamente de los datos.

Metodología de aprendizaje Los algoritmos de clustering no son supervisados. ¿Cómo afecta eso al proceso de aprendizaje?

¿Cómo aprende nuestro modelo ML?

Descripción general de cómo se enseña el modelo a si mismo.

Proceso de optimización

¿Cómo se optimizan los modelos de aprendizaje sin supervisión si no hay una función de pérdida?



Desafortunadamente, los algoritmos no supervisados son muy difíciles de evaluar en su desempeño.

Desempeño

¿Cómo evaluamos el desempeño de un algoritmo no supervisado?

Con el aprendizaje supervisado teníamos un objetivo claro: predecir nuestra variable de salida con alta precisión. *Evaluar el desempeño de los métodos de agrupación (clustering) es difícil. A menudo, confiamos en evaluaciones holísticas de nuestro algoritmo de agrupamiento.*

A diferencia del aprendizaje supervisado, no podemos verificar el rendimiento de nuestro modelo comparando la función de pérdida. La calidad del clustering es *subjetiva*, y *depende en gran medida de las suposiciones* hechas desde el principio.



¿Cuándo deberíamos recurrir al aprendizaje no supervisado?

- Tenemos datos dimensionales extremadamente altos (es decir, muchas características) que deseamos investigar.
- Tenemos una pregunta de investigación pero no tenemos una característica de salida etiquetada.
 - Esto es cierto para muchos conjuntos de datos.
- Deseamos detectar cualquier relación o patrón en nuestros datos.
 - Por ejemplo, datos de comportamiento del cliente.
- No tenemos tiempo para profundizar en la definición de un resultado.
 - Utilizamos el aprendizaje no supervisado como primer paso exploratorio.

A medida que crece la cantidad de datos en el mundo, recurriremos cada vez más a métodos de aprendizaje sin supervisión.



El aprendizaje no supervisado es una herramienta importante, a menudo utilizada como parte de nuestro análisis exploratorio de los datos.

- Inferir propiedades complejas de los datos (como subgrupos)
- Descubrir métodos de visualización interesantes e informativos.

A menudo, podemos aprovecftar un algoritmo no supervisado para ayudar a la selección de características durante el análisis exploratorio, antes de usar un algoritmo supervisado durante la fase de modelado.



Algoritmos de Clustering

- 1. Clustering K-means
- 2. Density-Based Spatial Clustering of Applications with Noise (DBSCAN)
- 3. Mean-Shift Clustering
- 4. Expectation—Maximization (EM) Clustering using Gaussian Mixture Models (GMM)
- 5. Agglomerative Hierarchical Clustering

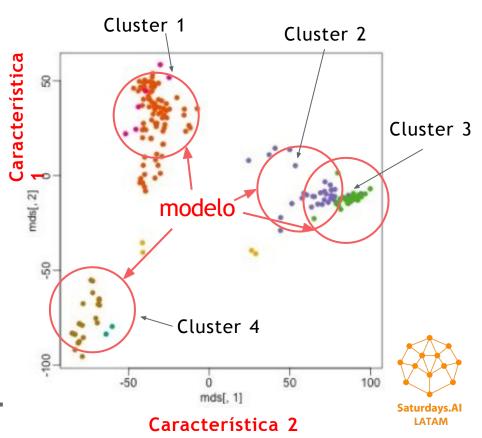
Concepto central: identificar subgrupos similares dentro de los datos.

Clustering es un poderoso algoritmo no supervisado que detecta patrones naturales en los datos.

El agrupamiento divide los datos para descubrir cómo las observaciones son similares en varias características diferentes.

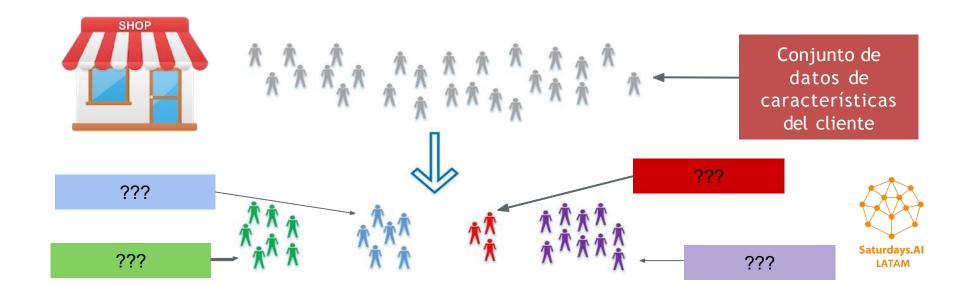
No estamos prediciendo un verdadero Y.

Los grupos son el modelo. Decidimos el número de grupos, representados como K.



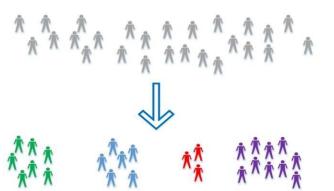
Definiendo los grupos

Imagina que eres el dueño de una tienda de ropa online. Deseas segmentar a tus clientes por sus ftábitos de compra.



Dado que tenemos un sitio web para nuestra tienda, tenemos datos valiosos sobre cómo se comportan nuestros consumidores en línea.

Conjunto de datos de características del cliente



id_cliente	Número_de_ Cant_prom_ visitas gastada		_	Tipo_de_ trafico	%_de_visitas_ durante_ofertas
	X1	X2	Х3	X4	
1237482 1213345 2323764 2326734	5 50 20 1	\$92 \$35 \$200 \$40		organic Email_sale Email_new_collection organic	20% 100% on 10% 100%

¿Qué características serán relevantes para nuestra tarea de agrupamiento?

> aturdays.A LATAM

Seleccionamos características que determinarán cómo se forman los clusters en nuestro algoritmo

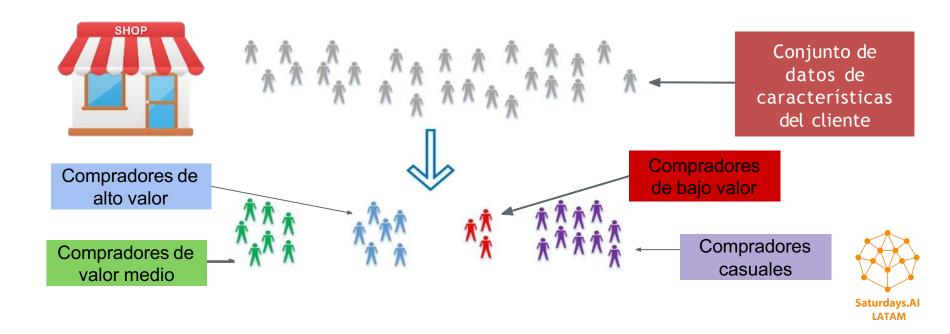


id_cliente	Número_d visitas	le_ Cant_pr gastada	_		%_de_visitas_ durante_ofertas
	X1	X2	Х3	X4	
1237482	5	\$92		organic	20%
1213345	50	\$35		Email_sale	100%
2323764	20	\$35 \$200		Email_new_collection	n 10%
2326734	1	\$40		organic	100
		7.0		_	0/0

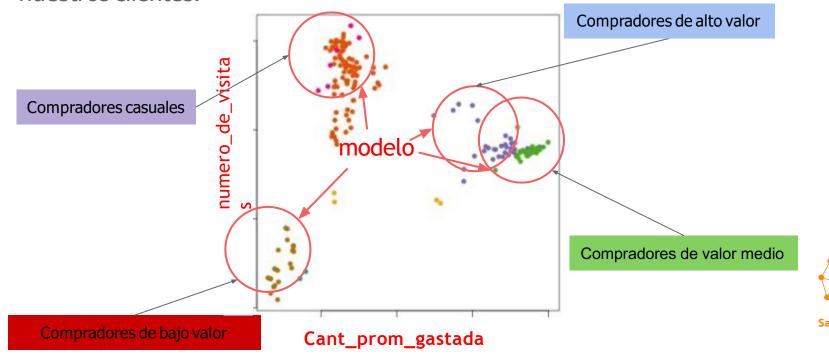
Dado que queremos segmentar a los clientes según sus hábitos de compra, probablemente queramos formar grupos utilizando las características "número de visitas" y "cantidad promedio gastada". Comencemos con estos.



Usando el número de visitas y la cantidad gastada, podemos ver los siguientes grupos:



Podemos demostrar el agrupamiento usando dos características en un espacio bidimensional: utilizamos cant_prom_gastada y numero_de_visitas para agrupar a nuestros clientes.



Una vez que agrupamos usando las funciones que seleccionamos, podemos decir algo sobre el valor de nuestros clientes.

Compradores de alto valor

- No sensibles a los precios
- Compradores frecuentes

Compradores de valor medio

- Sensibles a los precios
- Compradores poco frecuentes

Compradores de bajo valor

- No sensibles a los precios
- Compradores poco frecuentes

Compradores casuales

- Sensibles a los precios
- Compradores frequentes





La selección de características es tan importante en algoritmos no supervisados como supervisados.

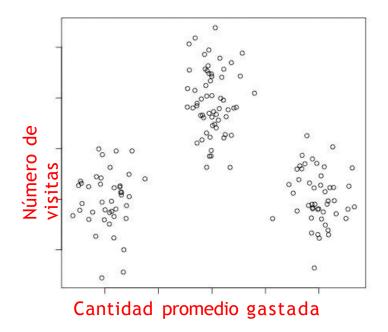
En este ejemplo simple, se usaron 2 características para agrupar, y se produjeron 4 grupos diferentes (K = 4).

Pero podemos incluir tantas características como queramos así como determinar cuántos clústeres producir.



Paso 1: elegir K

Comenzamos con un diagrama de dispersión simple del número de visitas contra la cantidad promedio gastada.



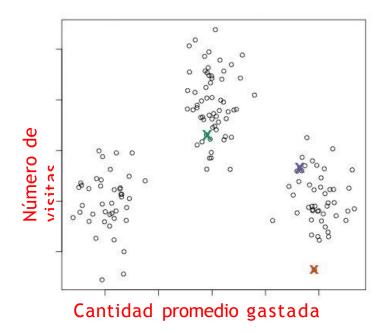
¿Cómo tomamos este diagrama de dispersión y segmentamos las observaciones en K grupos distintos?

Por ahora, digamos K = 3.



Paso 2: asignar los centroides

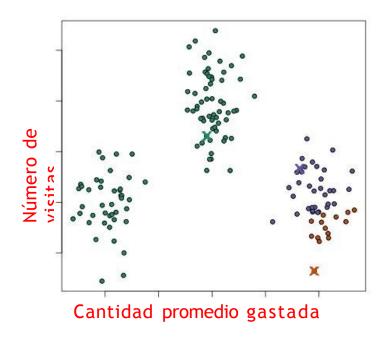
El segundo paso es asignar los K puntos aleatorios como centroides. Estos K puntos pueden ser puntos del conjunto de datos o del exterior.





Paso 3: asignar puntos a centroides.

En el tercer paso, se asigna cada punto al centroide más cercano, y con esto se forman k grupos o clusters.



La "distancia" aquí es la distancia Euclidiana (o distancia espacial), donde la distancia entre dos vectores u y v con n elementos es:

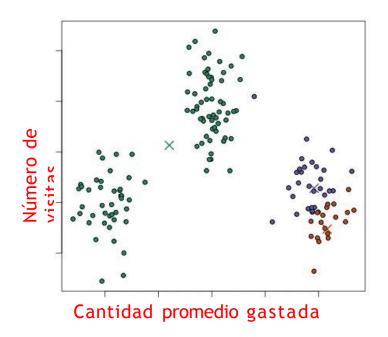
$$d = \sqrt{\sum_{n} (u_i - v_i)^2}$$

Nota importante: el clustering no toma características categóricas como entradas, solo continuas. La distancia entre puntos categóricos no sería significativa.



Paso 4: calcular nuevos centroides.

Recalculamos los nuevos centroides como los puntos centrales de cada cluster.

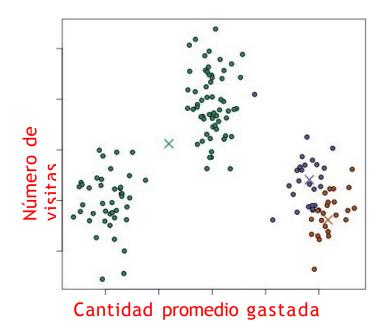


Nuestros centroides recalculados ahora están más separados.



Paso 5: iterar.

Seguimos repitiendo el cálculo del centroide / reasignaciones de cluster hasta que ya nada se mueva.

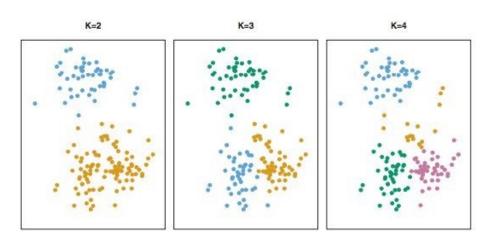


Vemos que la distancia entre observaciones y centroides es cada vez menor.



¿Por qué nuestros puntos de datos fueron asignados aleatoriamente a tres grupos? Establecimos K, o número de grupos, igual a 3.

El número de grupos (k) es un ejemplo de hiperparámetro.



¿Cuánto debería ser k?

Un número muy grande de grupos causará un sobreajuste y un número muy pequeño causará sub-ajuste



Método del codo

El método del codo nos ayuda a evitar el sobre o sub-ajuste al mostrar cómo el error del modelo disminuye con el número de clusters.

¿Cuánto debería ser k?

Un gráfico de codo nos permite visualizar cómo disminuye el error a medida que aumenta K.

Este gráfico es útil porque muestra la **compensación** entre sobreajuste y sub-ajuste. Necesitamos encontrar un equilibrio, llamado "codo".

