

Aprendizaje no Supervisado



Delta Analytics construye capacidad técnica alrededor del mundo.



El contenido de este curso está siendo desarrollado activamente por Delta Analytics, una organización sin fines de lucro 501(c)3 del Área de la Bahía que apunta a capacitar a las comunidades para aprovechar sus datos.

Por favor comuníquese con cualquier pregunta o comentario a inquiry@deltanalytics.org.

Descubre más sobre nuestra misión [aquí](#).



Módulo 4.3:

Aprendizaje no Supervisado



Checklist del módulo:

- ❑ Descripción general del aprendizaje no supervisado
 - ❑ Intuición
 - ❑ Pros y contras
- ❑ Algoritmos de Clustering
 - ❑ Clustering K-means
 - ❑ Análisis de componentes principales (PCA)

En la primera mitad de este curso, 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 frente a no supervisado

Aprendizaje Supervisado

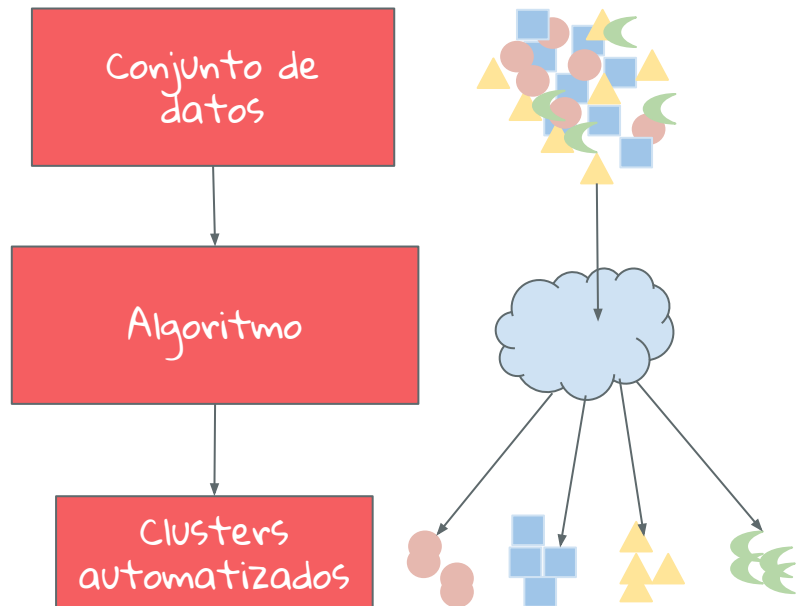
- Por cada x , hay un y
- El objetivo es predecir y usando x
- En la práctica, la mayoría de los métodos utilizados son supervisados.

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.

Aprendizaje no
Supervisado

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

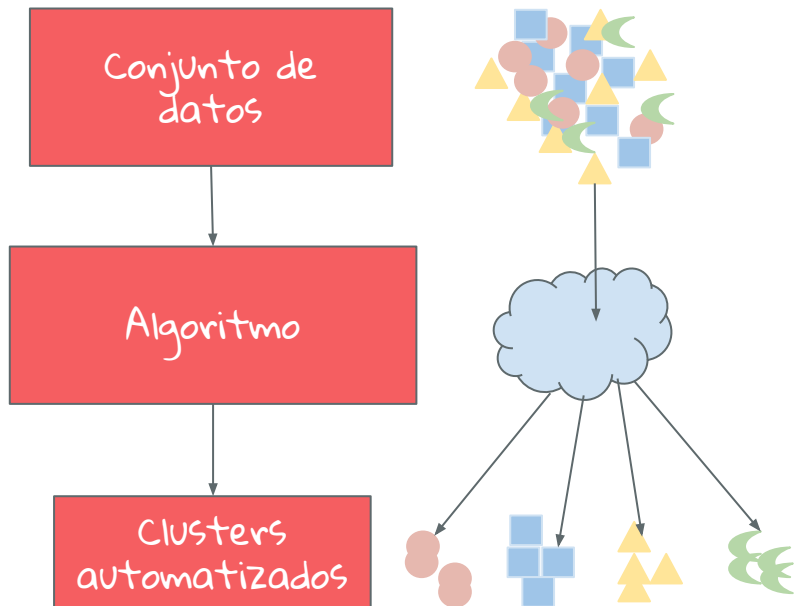


El aprendizaje no
supervisado implica
identificar patrones.

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

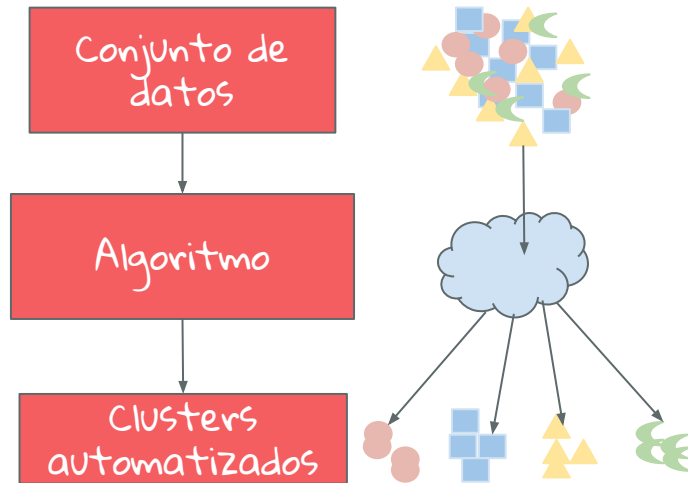
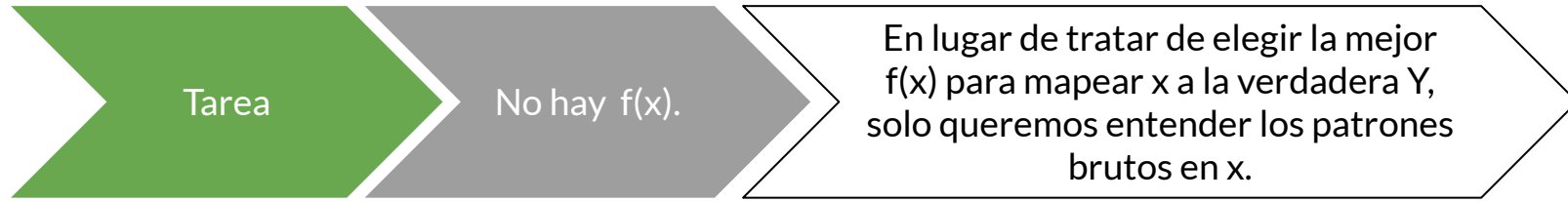
Aprendizaje no
Supervisado

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

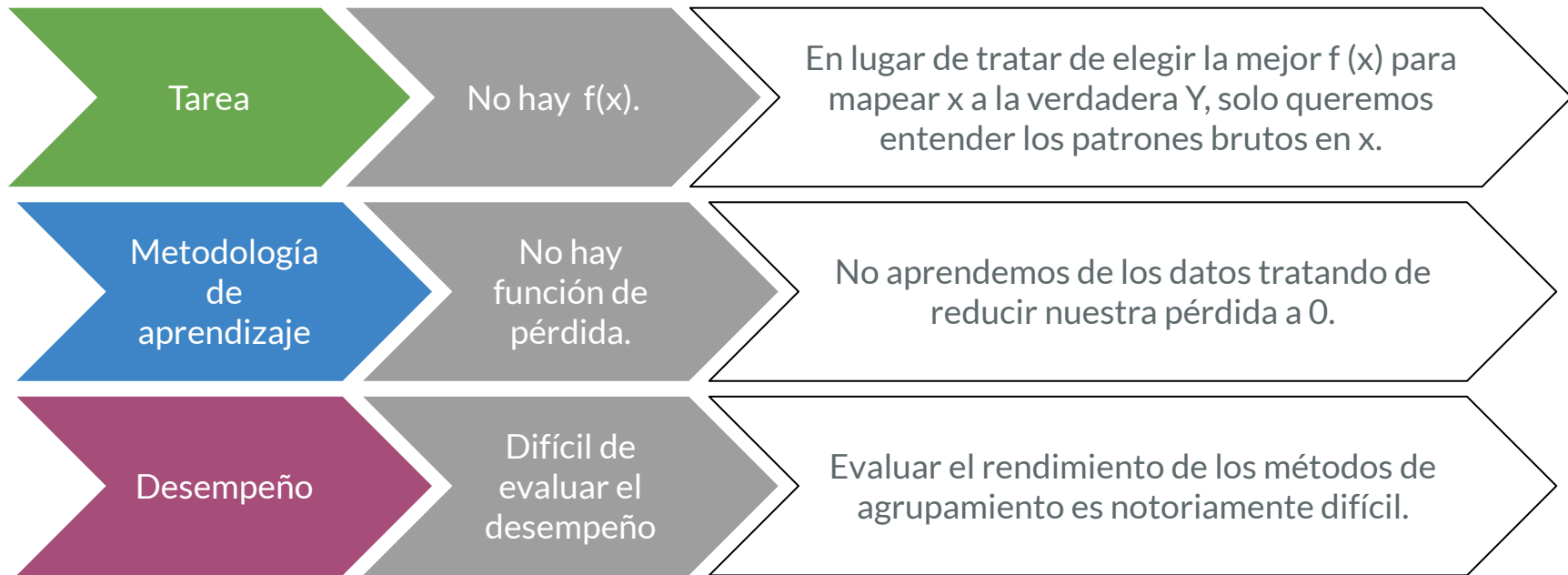


Resulta que lo que es extremadamente intuitivo para los bebés humanos es bastante difícil para que las computadoras lo hagan bien.

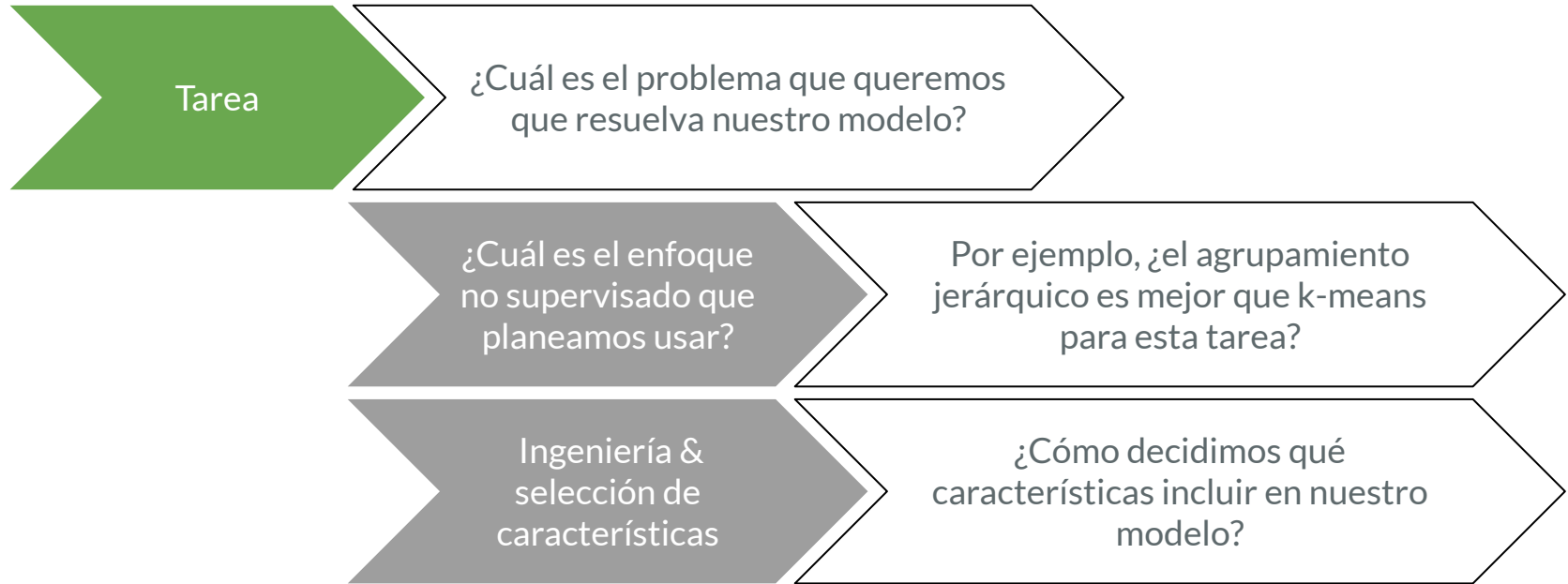
¿Cómo es este modelo diferente de lo que ya hemos visto?



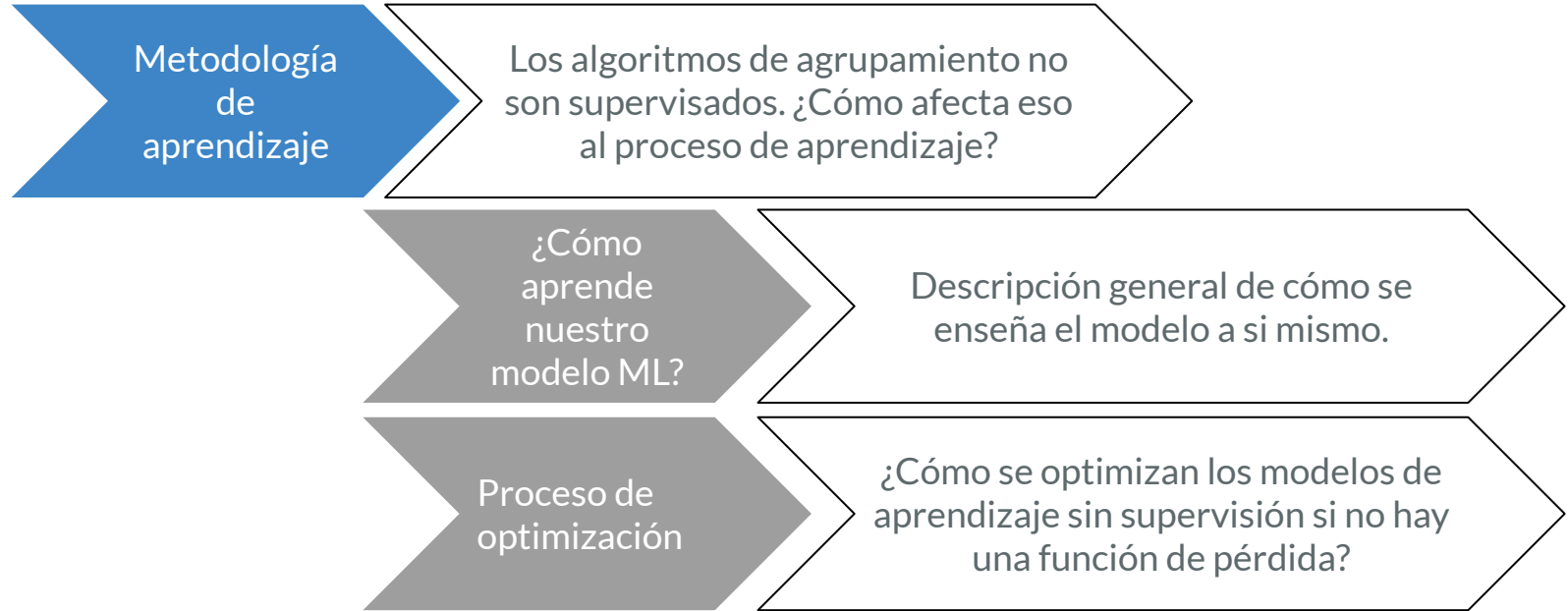
¿Cómo es este modelo diferente de lo que ya hemos visto?



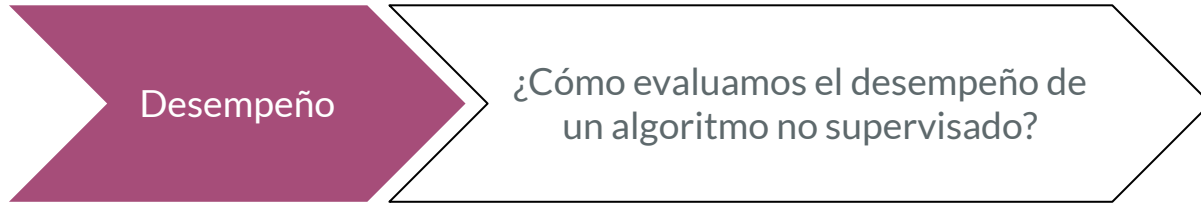
Los algoritmos no supervisados todavía tienen una tarea e involucran ingeniería y selección de características.



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



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



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.



Pros y contras de usar un enfoque no supervisado



¿Cuándo debería recurrir al aprendizaje no supervisado?

- Tienes datos dimensionales extremadamente altos (es decir, muchas características) que desea investigar
- Tienes una pregunta de investigación pero no tienes una característica de salida etiquetada
 - Esto es cierto para muchos conjuntos de datos
- Deseas detectar cualquier relación o patrón en tus datos.
 - Por ejemplo, datos de comportamiento del cliente
- No tienes tiempo para profundizar en la definición de un resultado
 - Utiliza 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.

Aprendizaje no
supervisado

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

Pregunta de
investigación

Limpieza
de datos

Análisis
exploratorio

Fase de
modelado

Desempeño

- Inferir propiedades complejas de los datos (como subgrupos)
- Descubrir métodos de visualización interesantes e informativos.
- A menudo utilizados en fases exploratorias del análisis de datos.

A menudo, puedes aprovechar 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.



El futuro del aprendizaje automático es el aprendizaje no supervisado.



El aprendizaje supervisado es la guinda del pastel

El aprendizaje no supervisado es el pastel en sí

Los seres humanos aprenden principalmente a través del aprendizaje no supervisado: absorbemos grandes cantidades de datos de nuestro entorno sin necesidad de una etiqueta.

Para alcanzar la verdadera inteligencia artificial (es decir, una máquina que piensa y aprende por sí misma), ML necesita mejorar en el aprendizaje no supervisado; debe aprender sin que tengamos que darle etiquetas o instrucciones explícitas.

Solo habremos rasguñado la superficie en esta clase.



Dicho esto, pasemos a nuestros primeros algoritmos no supervisados: **algoritmos de clustering**.



Algoritmos de Clustering

1. Clustering K-means
2. Análisis de Componentes Principales

Concepto central: identificar subgrupos similares dentro de los datos.



¿Dónde estamos?

Algoritmos
supervisados

Regresión Lineal

Árboles de
decisión

Algoritmos
combinados

Algoritmos no
supervisados

k-means

Análisis de
Componentes
Principales

Hemos terminado de discutir sobre algoritmos supervisados y ahora presentaremos y discutiremos algoritmos no supervisados. Esto ayudará a sentar las bases para nuestra discusión sobre el procesamiento del lenguaje natural.





Comencemos con clustering K-means, un algoritmo que divide los datos en k agrupaciones (clusters) en base a las características seleccionadas.



1. Clustering K-means



Clustering K-Means:

Pros

- Fácil de representar físicamente
- No asume ninguna distribución subyacente (por ejemplo, sin suposición de distribución normal como en la regresión lineal)
- Produce agrupaciones intuitivas
- Puede trabajar en muchas dimensiones.

Contras

- Lleva mucho tiempo encontrar la cantidad óptima de grupos
- Ingeniería de características que requiere mucho tiempo (las características deben ser numéricas y normalizadas)

Suposiciones

- Asume la existencia de agrupaciones subyacentes



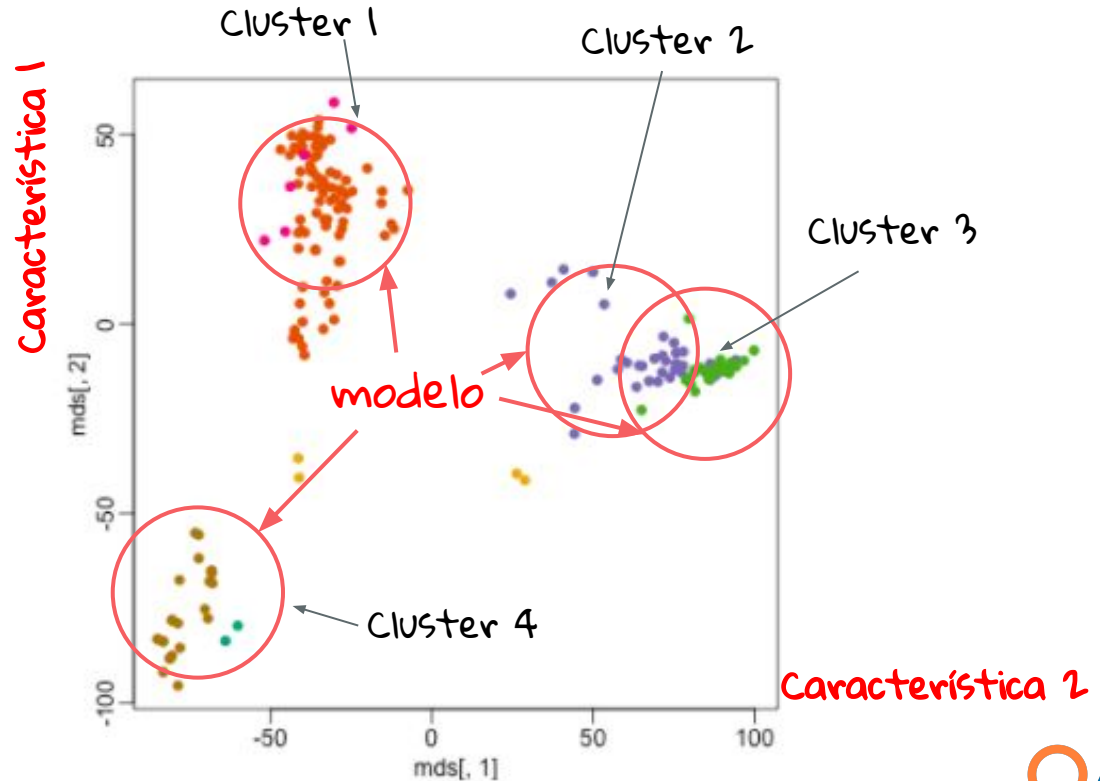
Tarea de Clustering

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 una verdadero Y.

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



Tarea de
Clustering

Definiendo
los grupos

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

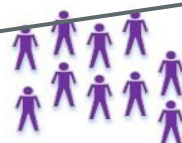


Conjunto de
datos de
características
del cliente



???

???



???

???





Tarea de
Clustering

Definiendo
los grupos

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

NOTA: Como propietario de la tienda, **crees** que algunos clientes son cazadores de ofertas, mientras que otros no tienen en cuenta los precios; algunos vienen todas las semanas mientras que otros llegan durante las vacaciones.

Pero en realidad no sabes definitivamente cuáles son cuáles.

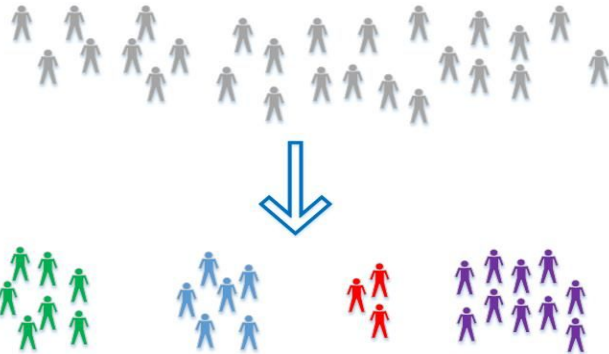
Podrías adoptar un enfoque supervisado para este problema e intentar **predecir** cuánto gastará un cliente o con qué frecuencia comprará un cliente. Pero aquí, solo estamos tratando de ver **qué nos dicen los datos**. Esto es lo que hace que la agrupación o clustering sea un problema no supervisado.



Tarea de Clustering

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 visitas	Cant_prom_gastada	Tipo_de trafico	%_de_visitas_durante_ofertas
	X1	X2	X3	X4
1237482	5	\$92	organic	20%
1213345	50	\$35	Email_sale	100%
2323764	20	\$200	Email_new_collection	10%
2326734	1	\$40	organic	100%

¿Qué características serán relevantes para nuestra tarea de agrupamiento?



Tarea de Clustering

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



id_cliente	número_de_visitas	Cant_prom_gastada	Tipo_de_tráfico	%_de_visitas_durante_ofertas
	X1	X2	X3	X4
1237482	5	\$92	organic	20%
1213345	50	\$35	Email_sale	100%
2323764	20	\$200	Email_new_collection	10%
2326734	1	\$40	organic	100%

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.



Tarea de
Clustering

Definiendo
Grupos

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



Conjunto de
datos de
características
del cliente



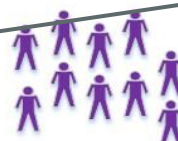
Compradores de
alto valor

Compradores de
valor medio



Compradores de bajo valor

Compradores casuales

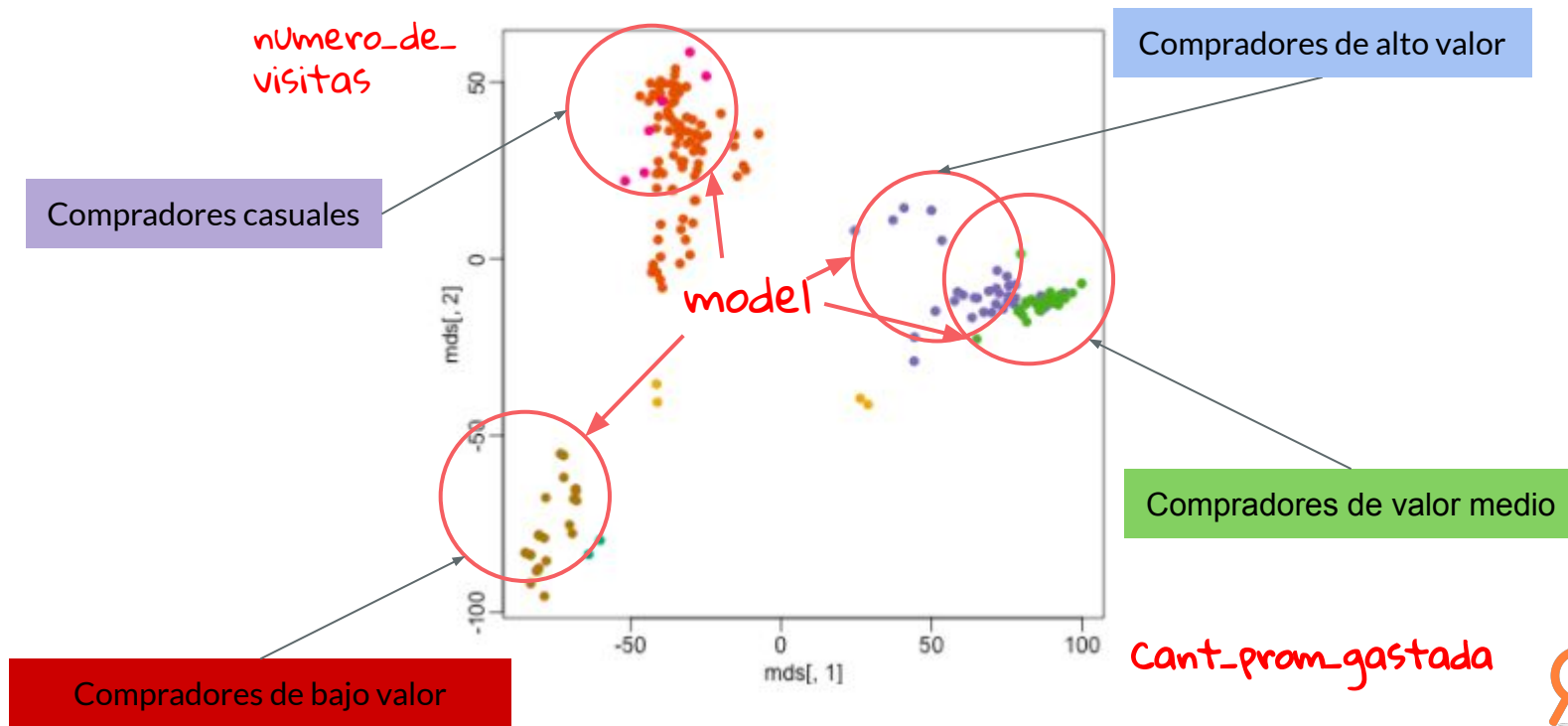


Tarea

Definiendo
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.



Tarea

Definiendo
Grupos

Usando dos características, podemos decir algo sobre nuestros clientes.

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

High value buyers

Middle value
buyers

Low Value Buyers

Casual Buyers

- Not price-sensitive
- Frequent buyers

- Price-sensitive
- Infrequent buyers

- Not price-sensitive
- Infrequent buyers

- Price-sensitive
- Frequent buyers





Tarea

Ingeniería de
características
& Selección

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

En este ejemplo simple, usamos 2 características para agrupar, y produjimos 4 grupos diferentes ($K = 4$).

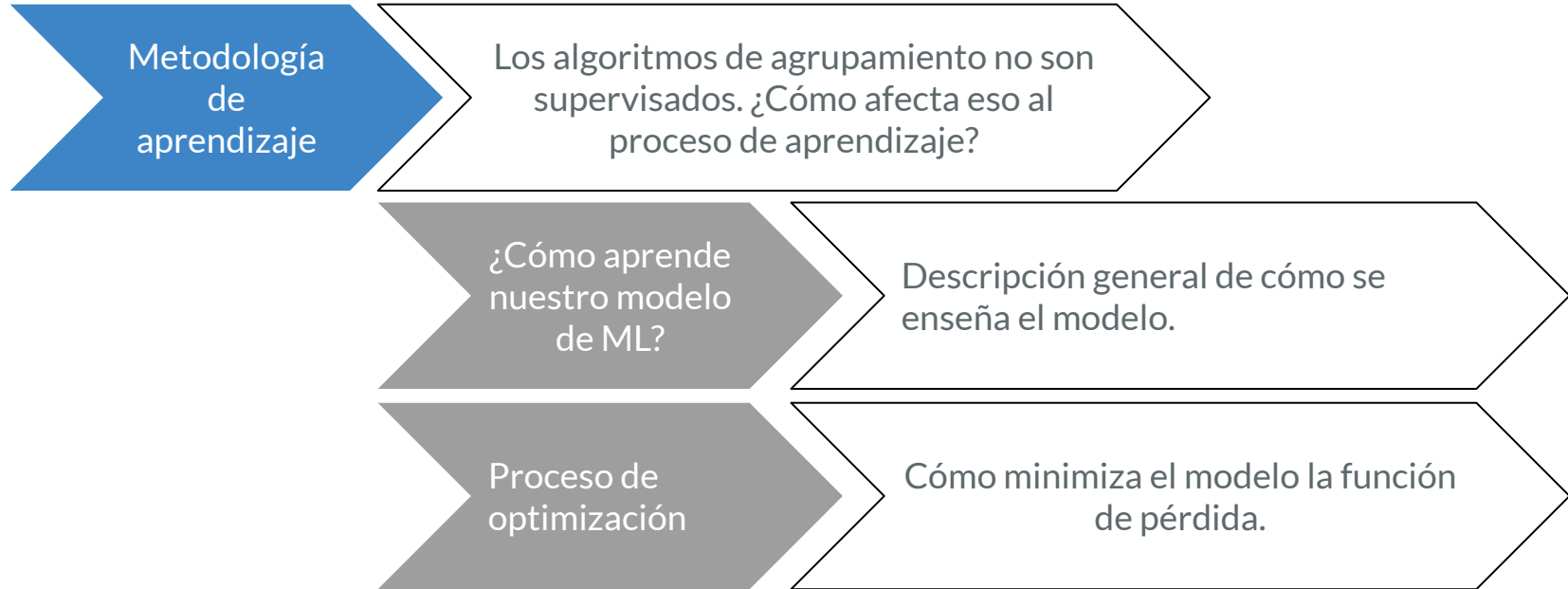
Pero podemos incluir tantas características como queramos y determinar cuántos clústeres producir. Volveremos a este punto más tarde, pero primero, veamos cómo aprende este algoritmo.



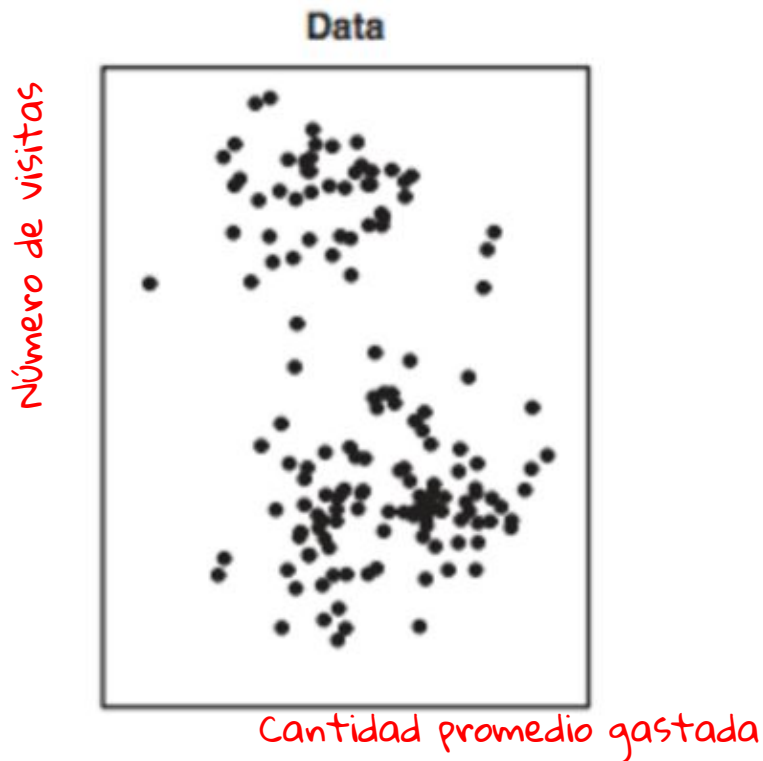
¿Cómo aprende nuestro algoritmo k-mean de los datos para agrupar observaciones similares?



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



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?

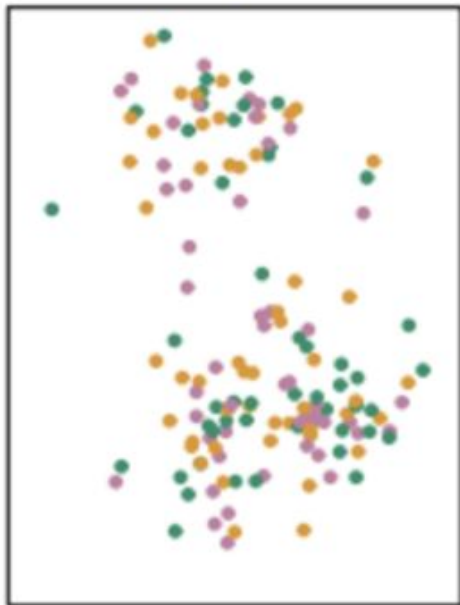
Por ahora, digamos $K = 3$.

Fuente: Intro to Statistical Learning with Applications in R.pdf

Paso 1: asigne aleatoriamente a cada cliente a un clúster

Número de visitas

Step 1



- Los puntos de datos se han asignado aleatoriamente en grupos de color rosa, amarillo, y verde.

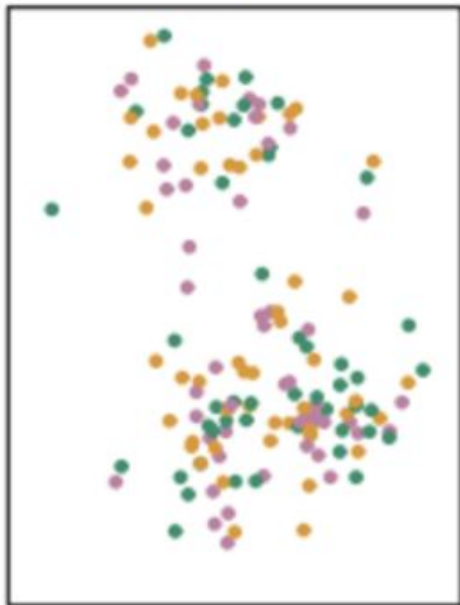
Hay 3 colores (grupos) aquí. ¿Por qué?

Cantidad promedio gastada

Paso 1: asigne aleatoriamente a cada cliente a un clúster

Número de visitas

Step 1

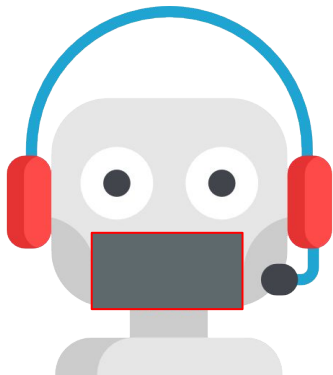


- Los puntos de datos se han asignado aleatoriamente en grupos de color rosa, amarillo, y verde.

Aquí hay 3 colores (grupos) porque decidimos $K = 3$.

Cantidad promedio gastada

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



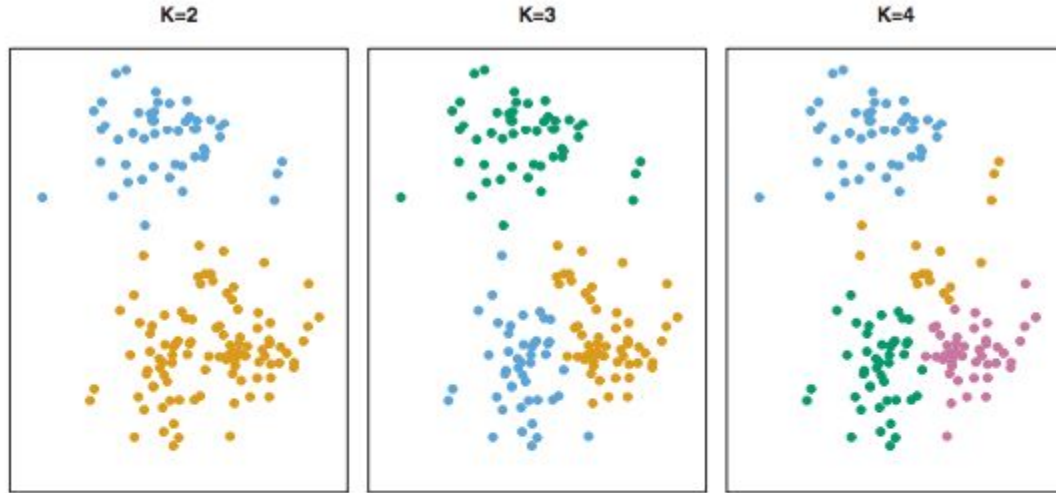
El número de grupos (k) es un ejemplo de hiperparámetro.

Los hiperparámetros son establecidos por el investigador (¡tú!), No determinado por el modelo.

Hiperparámetros

Configuraciones de nivel superior
de un modelo que se fijan antes
de comenzar el entrenamiento

Podemos configurar nuestro hiperparámetro K para que sea $2, 3, 4 \dots n$.



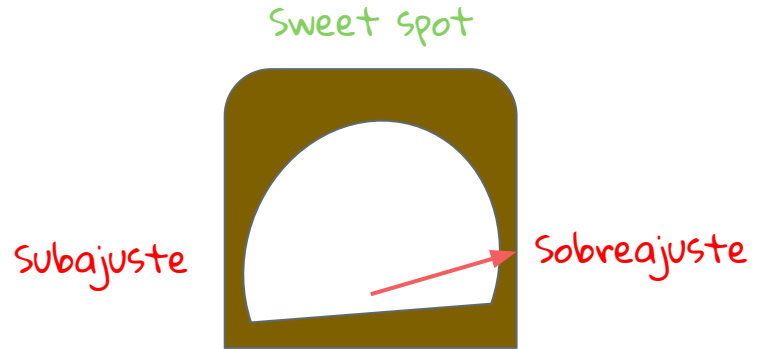
En nuestro ejemplo, especificamos 3 grupos, pero podemos decirle al modelo **cualquier "K"** que queramos.

¿Qué debería ser k ? Un número muy grande de grupos causará un **sobreajuste** (por ejemplo, si establece k igual al número de observaciones, ¡tendrá un grupo para cada observación!) Esto no es muy útil para dar sentido a los subconjuntos de nuestros datos.

Desempeño

¿Recuerdan el sobreajuste frente al subajuste? Si K es demasiado grande, podemos tener un sobreajuste; Demasiado pequeño y podemos estar sub ajustados.

Si estamos sub ajustados, es posible que nos falten subconjuntos naturales de clientes similares. Si estamos sobre ajustando, es posible que tengamos demasiados grupos, por lo que nuestro modelo no se generaliza bien para datos no vistos.

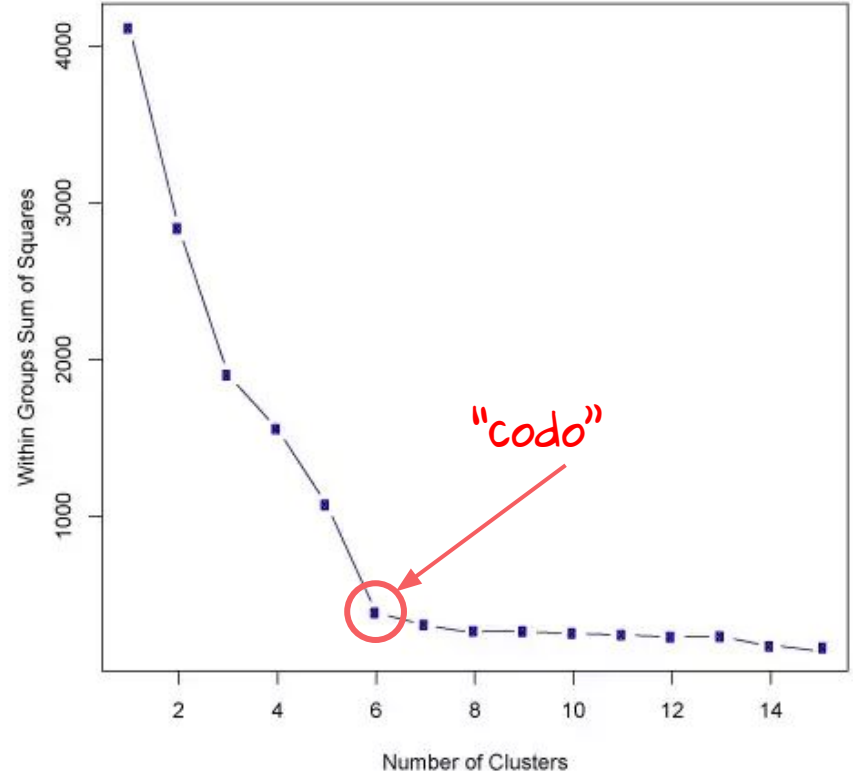


Un diagrama de codo nos ayuda a evitar el sub-ajuste al mostrar cómo el error del modelo disminuye con el número de clusters.

¿Cómo sabemos cuál debería ser el número óptimo de clústeres?

Un diagrama de codo visualiza cómo disminuye el error a medida que aumenta K.

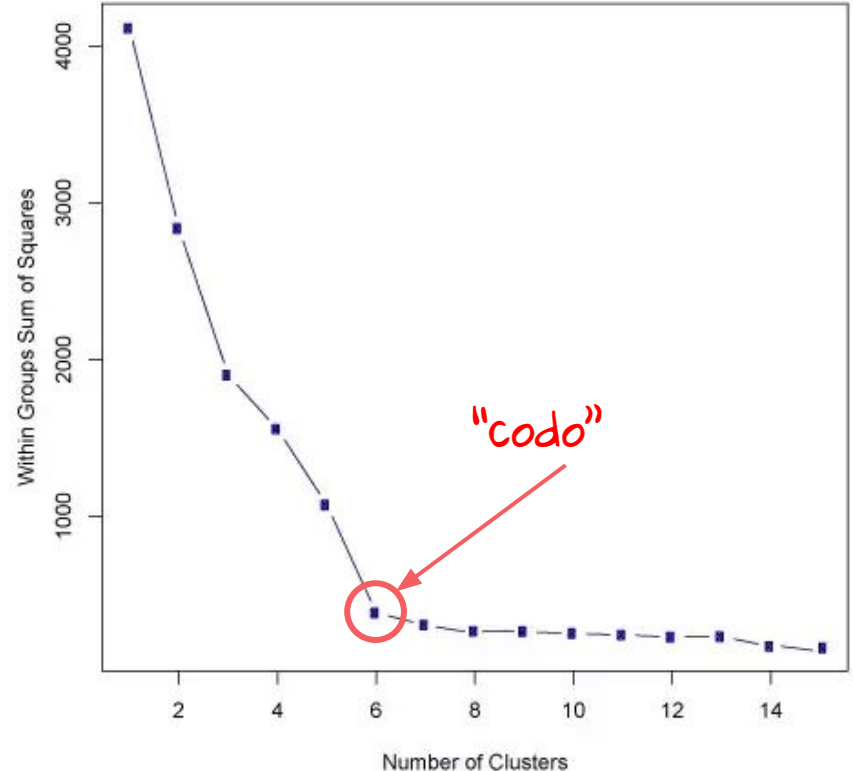
Este gráfico es útil porque visualiza la **compensación** entre sobreajuste y sub-ajuste. Necesitamos encontrar un equilibrio, llamado "codo".



Un diagrama de codo nos ayuda a evitar el sub-ajuste al mostrar cómo el error del modelo disminuye con el número de clusters.

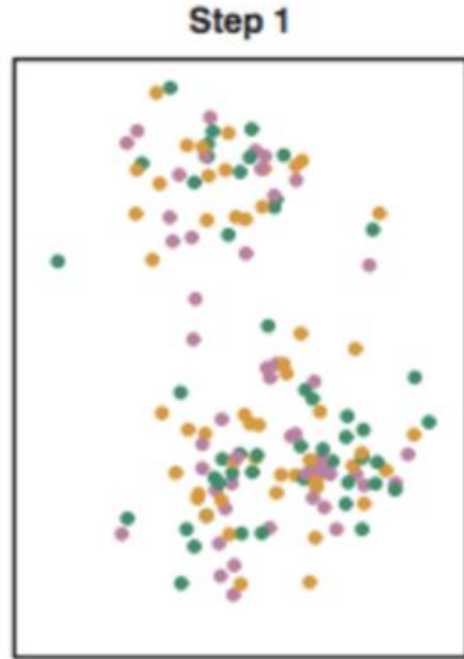
Si $K == n$ (número de observaciones), entonces la distancia es $= 0$ (cada observación es su propio cluster)! **Esto, como ya sabemos, es un caso de overfitting.**

En el gráfico de la derecha, debemos elegir $K=6$. Este parece ser el número de clústeres donde se han logrado las mayores ganancias en la reducción de errores.



Para empezar, establezcamos $k = 3$. Esto significa que queremos encontrar 3 grupos.

Número de clusters



Sospechamos que estos 3 grupos corresponderán a:

1. Clientes con poco presupuesto que compran con nosotros con frecuencia
2. Clientes de gama alta que compran con nosotros con frecuencia
3. Clientes que compran con nosotros con poca frecuencia

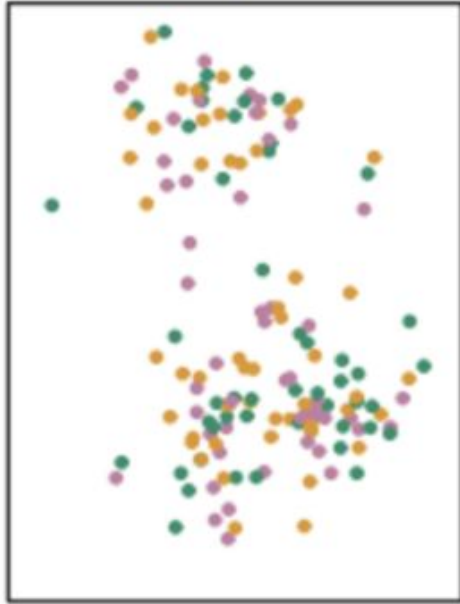
Sin embargo, es posible y de hecho es probable que haya naturalmente más de 3 subconjuntos de clientes.

Cantidad promedio gastada

¿Es útil nuestra asignación inicial aleatoria de observaciones a grupos?

Número de clusters

Step 1



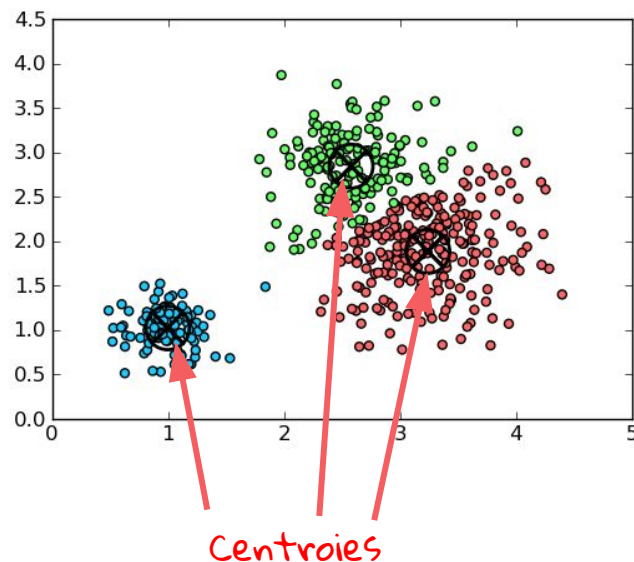
Recuerde que nuestro primer paso fue asignar aleatoriamente a cada cliente a un clúster.

Claramente, nuestra asignación aleatoria inicial a los grupos es pobre. ¿Qué tan pobre? ¿Y cómo lo mejoramos?

Cantidad promedio gastada

Para evaluar cuán pobres son nuestros grupos, podemos usar **la distancia entre una observación y el centroide de su cluster.**

Un centroide es el punto central de un cluster.



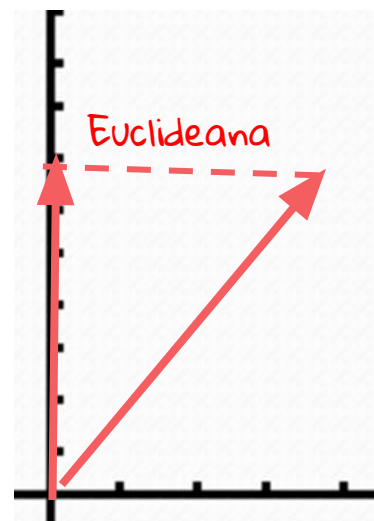
¡Revisión! ¿Cómo calculamos la distancia entre puntos?

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

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

- En este ejemplo, la diferencia entre el cliente c_6 (7,2) y el centro del clúster (4,7) sería $\text{sqrt}[(7-4)^2 + (2-7)^2] = 5.8$.

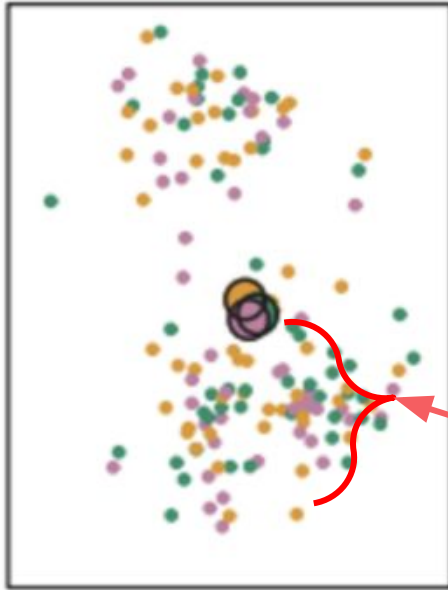
Nota importante: el clustering no toma características categóricas como entradas, solo continuas. La distancia entre puntos categóricos no sería significativa.



Calculamos un centroide para cada grupo. Nuestro objetivo es minimizar la distancia desde el centroide a cualquier observación.

Número de clusters

Iteration 1, Step 2a



- Los centroides se calculan como el centro de su grupo asignado aleatoriamente.
- Aquí, los centroides están muy juntos, y la distancia a los puntos externos es muy grande, ¡definitivamente podemos hacerlo mejor!

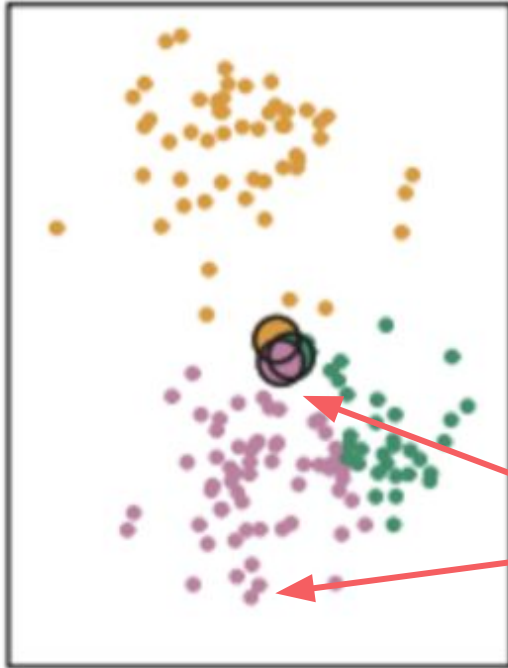
El objetivo es minimizar la distancia desde el centroide a cualquiera de las observaciones en su grupo

Cantidad promedio gastada

Paso 2b: reasigna cada observación al centroide más cercano

Iteration 1, Step 2b

Número de clusters



- Ahora reasignamos cada observación al cluster del centroide más cercano. ¡Ahora nuestros clusters comienzan a verse mejor!

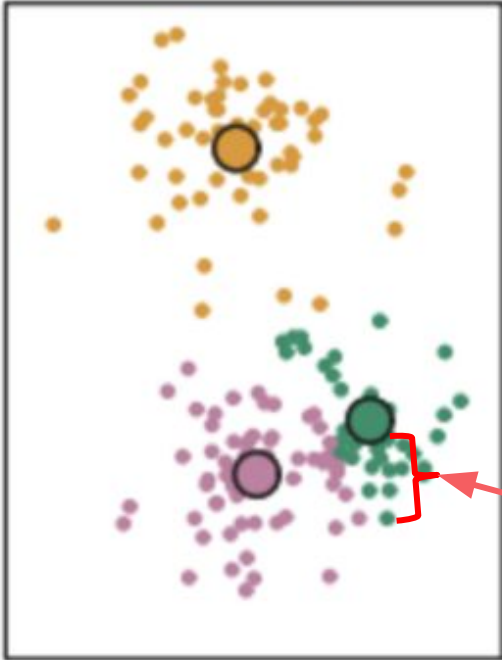
De los tres centroides, esta observación es la más cercana al rosa, por lo que lo asignamos al grupo rosado

Cantidad promedio gastada

Iteración 2 Paso 2a: ¡Repita los cálculos del centroide!

Iteration 2, Step 2a

Número de clusters



Cantidad promedio gastada

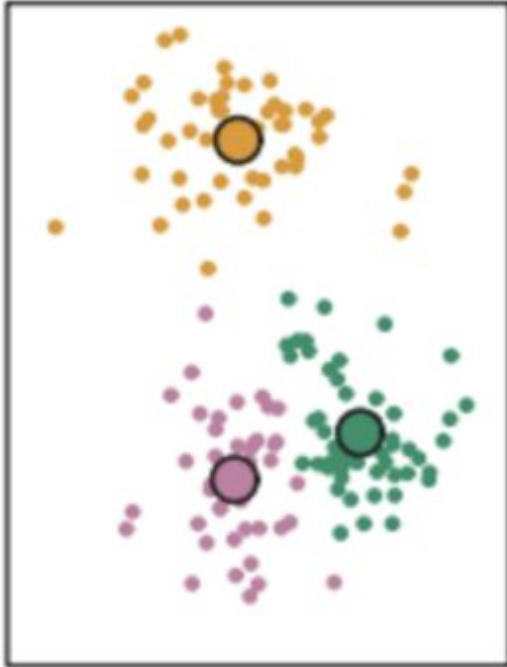
- ¡Aquí vamos de nuevo!
Nuestro centroide recalculado ahora está más separado, y podemos ver la minimización de la distancia entre observaciones y centroides.

Las distancias totales son mucho más pequeñas, inos estamos acercando!

Iteración 2 Paso 2b: reasignar
clusters según el centroide más
cercano

Final Results

Número de clusters



Cantidad promedio gastada

- Esta vez, menos observaciones cambiaron de cluster.
- Seguimos repitiendo el cálculo del centroide / reasignaciones de cluster hasta que ya nada se mueva - ¡el modelo está completo!

Ahora que conocemos la mecánica del algoritmo K-means, pensemos en un ejemplo que puede recrear con nuestros datos.

¿Cómo agrupamos los tipos de préstamos solicitados en el sitio web de Kiva?



Clustering Ejercicio 1

¿Cómo agruparíamos estos préstamos en grupos de características más similares?

Clustering task:

Loan 1:
\$25
1 borrower
Funded in 1 day

Loan 2:
\$1000
2 borrowers
Funded in 30 days

Loan 3:
\$50
3 borrowers
Funded in 30 days

Loan 4:
\$25
1 borrower
Funded in 1 day

Loan 5:
\$2000
5 borrower
Funded in 1 day

Loan 11:
\$3500
3 borrowers
Funded in 40 days

Loan 6:
\$1000
1 borrower
Funded in 1 day

Loan 7:
\$75
1 borrower
Funded in 60 days

Loan 8:
\$25
1 borrower
Funded in 1 day

Loan 9:
\$50
1 borrower
Funded in 1 day

Loan 10:
\$25
1 borrower
Funded in 1 day



Clustering Ejercicio 1

Podemos usar las características "tiempo para financiar" y "monto del préstamo"

Loan 9:
\$500
4 borrower
Funded in 2 days

Grandes
préstamos que
muchos
prestatarios
financiaron
rápidamente

Loan 5:
\$2000
5 borrower
Funded in 1 day

Loan 6:
\$1000
1 borrower
Funded in 1 day

Loan 11:
\$3500
3 borrowers
Funded in 40 days

Loan 2:
\$1000
2 borrowers
Funded in 30 days

Loan 3:
\$50
3 borrowers
Funded in 30 days

Loan 7:
\$75
1 borrower
Funded in 60 days

Loan 1:
\$25
1 borrower
Funded in 1 day

Loan 8:
\$25
1 borrower
Funded in 1 day

Loan 4:
\$30
2 borrower
Funded in 1 day

Loan 10:
\$25
2 borrower
Funded in 1 day

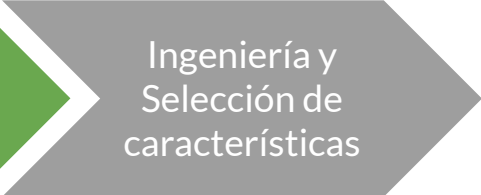
Mucho tiempo para financiar

Pequeños préstamos
que pocos
prestatarios
financiaron
rápidamente





Tarea de
Clustering



Ingeniería y
Selección de
características

¿Qué sucede si queremos usar más
de dos funciones para agrupar?

Queremos agrupar préstamos similares usando:

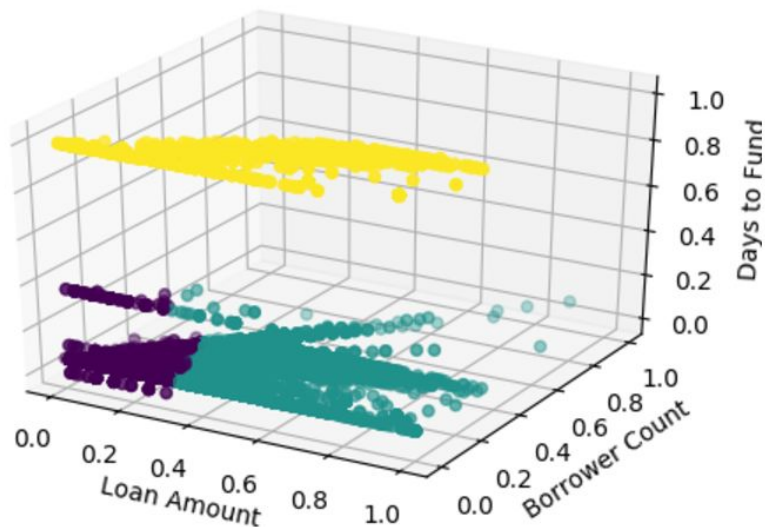
- Monto del préstamo: ¿cuánto \$ solicitó el prestatario?
- Recuento de prestatarios: ¿se trata de una solicitud de un solo prestatario o de un grupo de prestatarios?
- Tiempo para financiar: ¿cuánto tiempo tardó en financiarse la solicitud en el sitio?



Podemos extender nuestro agrupamiento a dimensiones más altas agregando más funciones.

Hasta ahora, nuestros clústeres se han visualizado en 2D para simplificar.

Pero esto puede extenderse a n dimensiones. El diagrama a la derecha usa 3 características para agrupar, esto da como resultado una visualización en 3D.



La verdadera utilidad del clustering proviene de dar sentido a los datos de alta dimensión.

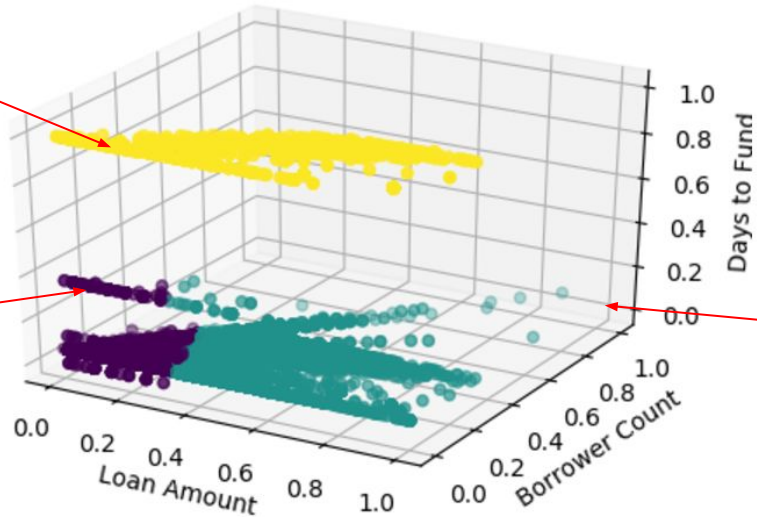
Tarea de
Clustering

Ingeniería y
Selección de
características

Ejemplo de agrupamiento utilizando 3 características: tiempo para financiar, el número de prestatarios y el monto del préstamo.

Préstamos
financiados a
última hora

Pequeños
préstamos con
pocos prestatarios
que se financian
rápidamente



Características de agrupamiento:

- Tiempo para financiar
- Número de prestatarios
- Monto del préstamo

Grandes
préstamos que
se financian
rápidamente

¡Fascinante! Si tú fueras el CEO de Kiva, ¿qué harías con esta información?

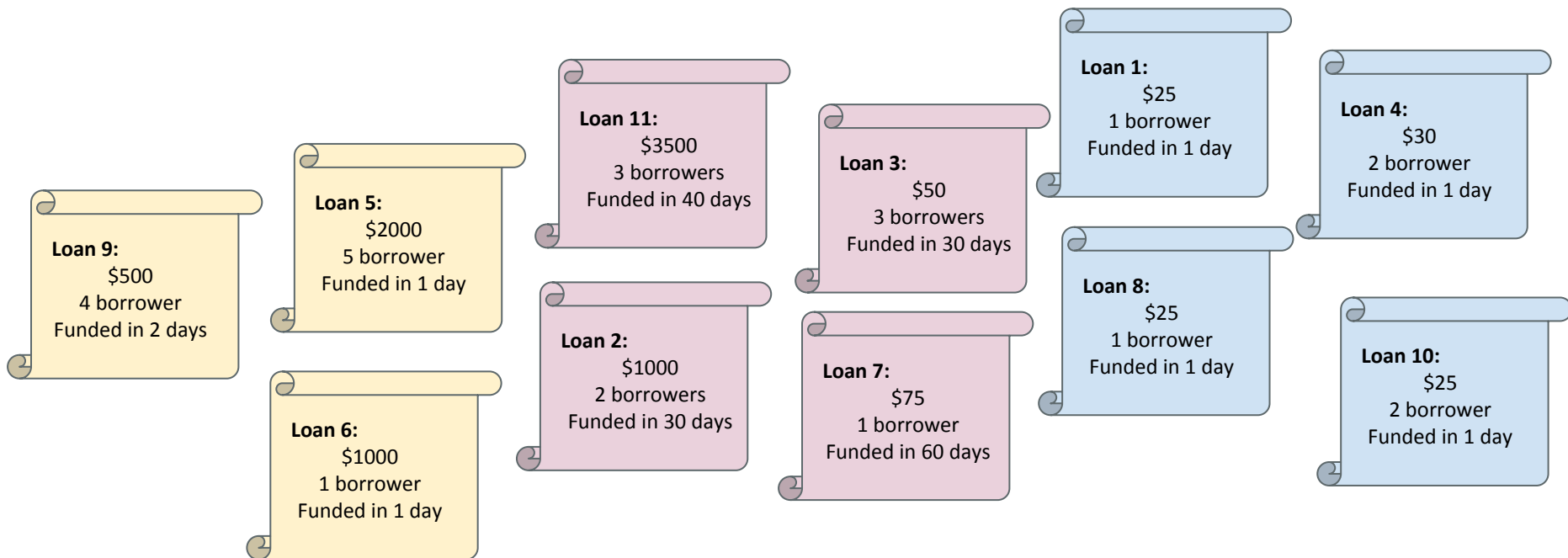


Podemos cambiar cuántas características usamos para agrupar nuestros datos, pero también podemos elegir la cantidad de clústeres (k) que queremos ver en nuestros datos.



Ejercicio 2

En este ejemplo, vemos tres clusters separados ...



Ejercicio 2

¿Qué pasaría si pensáramos que hay 4 clusters distintos?

Mucho tiempo para financiar

Loan 9:
\$500
4 borrower
Funded in 2 days

Loan 5:
\$2000
5 borrower
Funded in 1 day

Loan 11:
\$3500
3 borrowers
Funded in 40 days

Loan 3:
\$50
3 borrowers
Funded in 30 days

Loan 1:
\$25
1 borrower
Funded in 1 day

Loan 4:
\$30
2 borrower
Funded in 1 day

Loan 8:
\$25
1 borrower
Funded in 1 day

Loan 10:
\$25
2 borrower
Funded in 1 day

Grandes
préstamos que
muchos
prestatarios
financiaron
rápidamente

Loan 6:
\$1000
1 borrower
Funded in 1 day

Loan 2:
\$1000
2 borrowers
Funded in 30 days

Loan 7:
\$75
1 borrower
Funded in 60 days

Grandes
préstamos.
Mucho tiempo
para financiar

Pequeños
préstamos.
Mucho tiempo
para financiar

Pequeños préstamos
que pocos
prestatarios
financiaron
rápidamente



Clustering k-means es útil cuando queremos identificar grupos en nuestros datos basados en similitudes entre características específicas.

Pero, ¿qué pasa si tienes 250 características? Es difícil comprender los clústeres según estas características.

El Análisis de Componentes Principales puede ayudar. PCA es un algoritmo no supervisado que ayuda a disminuir la dimensión de las características.



2. Análisis de Componentes Principales



¿Qué es el Análisis de Componentes Principales?

El análisis de componentes principales, o PCA, es un **método de reducción de dimensionalidad** que a menudo se usa para reducir la dimensionalidad de grandes conjuntos de datos.

Esto lo hace transformando ortogonalmente las n coordenadas originales de un conjunto de datos en un nuevo conjunto de n coordenadas llamadas componentes principales.

Son los primeros componentes los que contienen la mayor parte de la información, por lo que un gran número de componentes se pueden desechar.

PCA es útil para la reducción de características.



¿Qué es el Análisis de Componentes Principales?

La reducción de la cantidad de variables de un conjunto de datos es naturalmente a expensas de la precisión, pero el truco en la reducción de la dimensionalidad es un intercambio entre la precisión y la simplicidad.

El beneficio de esto es que los conjuntos de datos más pequeños son más fáciles de explorar y visualizar y hacen que el análisis de datos sea mucho más fácil y rápido para los algoritmos de ML.

En resumen, la idea de PCA es simple: reducir el número de variables de un conjunto de datos, al tiempo de conservar la mayor cantidad de información posible.

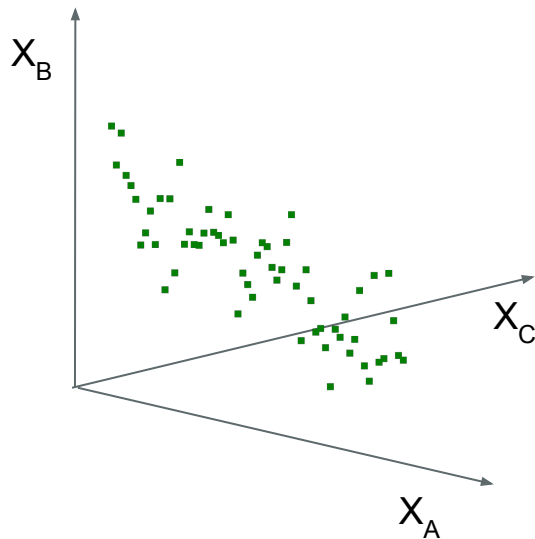


Asumir un set de datos simplificado con tres características: X_a , X_b y X_c .

1. Primero debemos **estandarizar** los datos. Llamemos a nuestras nuevas características estandarizadas X_A , X_B y X_C .

Para cada punto x por cada característica X , sustraer la media de X y dividir por la desviación estándar de X .

Por qué? Porque estaremos midiendo la variación. Si no estandarizamos los datos, podríamos convertir una característica desde *km* a *cm*, y provocar que la variación de esa característica aumente. La estandarización asegura que la variación de nuestros datos sea independiente de cualquier transformación que podamos usar.



Encontremos el componente principal #1

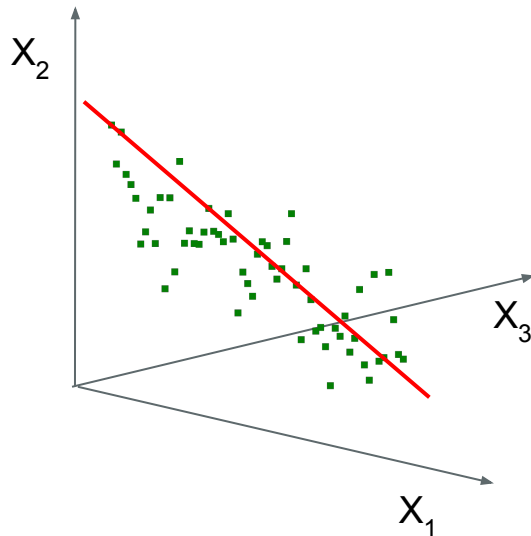
2. Para encontrar el primer componente principal, encuentra la **dirección de la varianza máxima**. ¡Este es el Componente Principal # 1 (PC1)!

Formalmente, PC1 es la combinación lineal de las características...

$$PC1 = c_1(X_A) + c_2(X_B) + c_3(X_C)$$

... que tiene la **mayor varianza**.

En nuestro ejemplo principal, observamos la dirección en la que los datos varían más.



Encontremos el componente principal #2

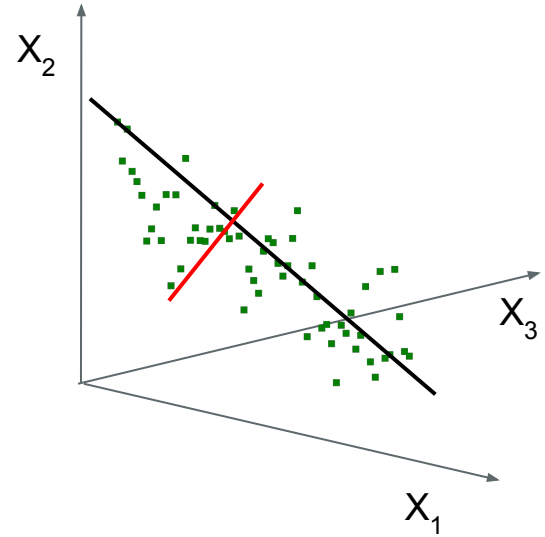
3. Para encontrar el segundo Componente Principal, debemos encontrar la **segunda más alta dirección de varianza**. Este es el Componente Principal #2 (PC2)!

Formalmente, PC2 es la combinación lineal de las características ...

$$PC2 = c_4(X_A) + c_5(X_B) + c_6(X_C)$$

... que tiene la segunda más alta varianza **Y no está correlacionada con PC1**.

En nuestro ejemplo simplificado, podemos mirar más o menos la dirección en que la varianza es la segunda más alta.S



Encontremos el componente principal #3

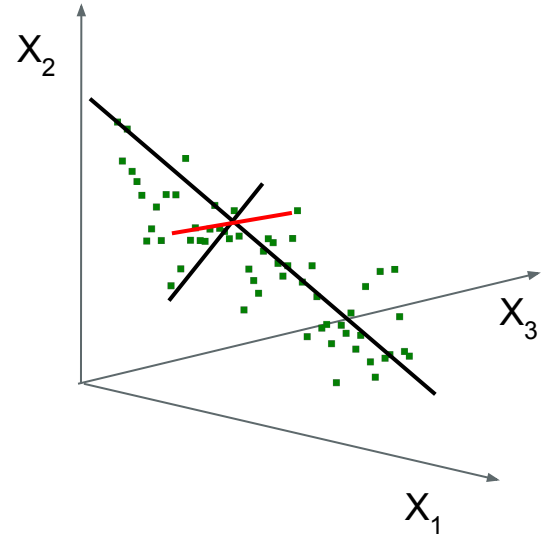
4. Para encontrar la tercera y final Componente Principal, debemos encontrar la **tercera más alta dirección de varianza**. Esta es la Componente Principal #3 (PC3)!

Formalmente, PC3 es la combinación lineal de las características...

$$PC3 = c_7(X_A) + c_8(X_B) + c_9(X_C)$$

... que tiene la tercera mayor variación **Y no está correlacionada con PC1 y PC2**.

En nuestro ejemplo simplificado, podemos observar aproximadamente la dirección en que la varianza es la tercera más alta.



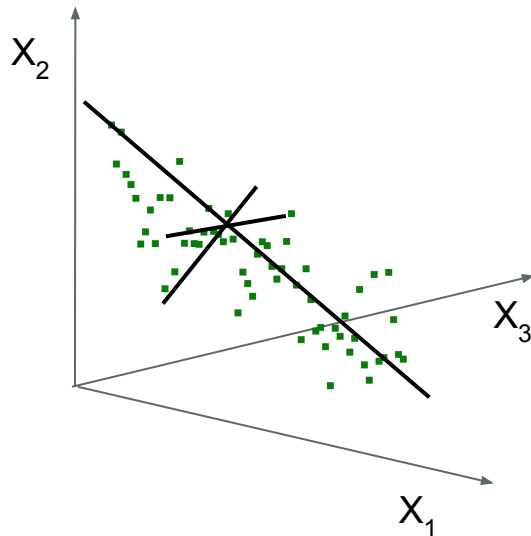
¿Cuántos componentes principales
podemos tener? ¿Cuántos queremos?

En este ejemplo, calculamos todos los componentes principales posibles (puede haber tantos componentes principales como características).

Sin embargo, como nuestro objetivo principal es la reducción de la dimensionalidad, mantendremos los “*Top*” componentes principales.

¿Cuántos debemos guardar?

Recordemos nuestra discusión sobre el diagrama del codo en la agrupación k-means. Usamos una lógica similar para ver cuántos componentes principales queremos.

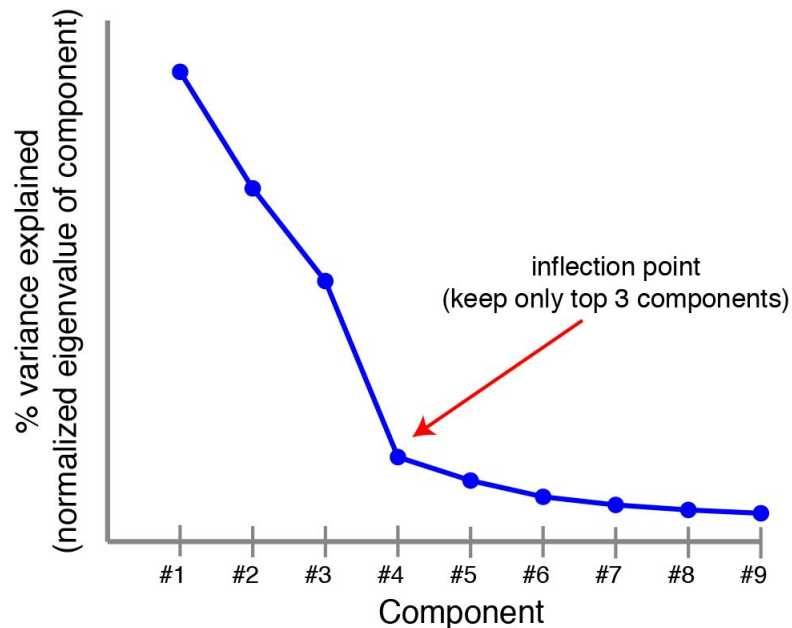


¿Cuántos componentes principales
podemos tener? ¿Cuántos queremos?

Si elegimos conservar solo la PC1, podremos capturar el 80% de la información de los datos originales. ¡Bastante bueno!

Si elegimos conservar PC1 y PC2, podremos capturar el 95% de la información de los datos originales. ¡Aun mejor!

Tener en cuenta que si optamos por conservar PC1, PC2 y PC3, capturaríamos el 100% de la información de los datos originales. Sin embargo, no elegiríamos hacer esto, ya que nuestra intención principal es la reducción de dimensiones

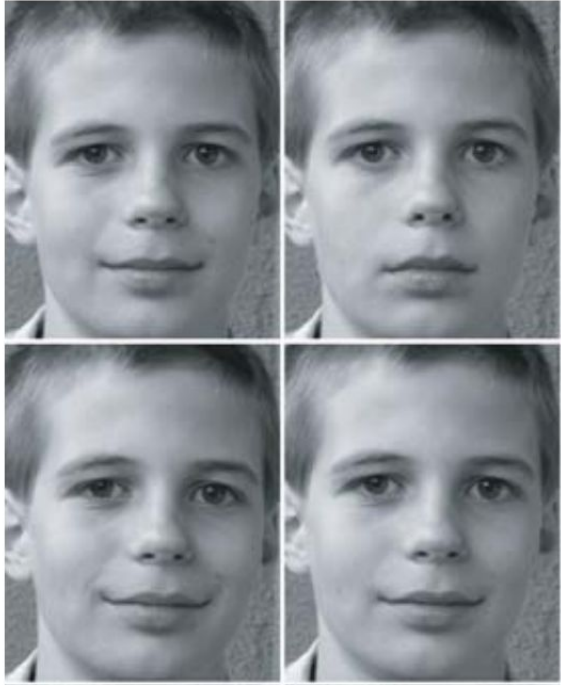


Por simplicidad, nuestro ejemplo tenía un conjunto de datos con solo 3 características. En la realidad, probablemente no aplicarías PCA a un conjunto de datos tan simple. La verdadera utilidad de PCA proviene de cuando la aplicamos a conjuntos de datos con cientos o miles de características.

Considere, por ejemplo, el **procesamiento de imágenes**, donde cada píxel de una imagen constituye una característica.



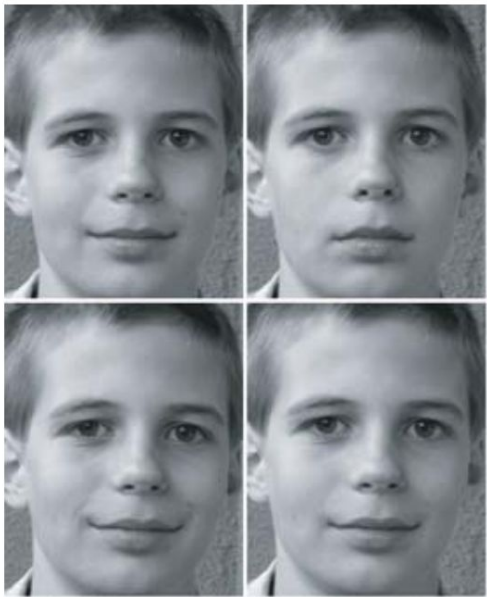
Para cada una de estas imágenes de 321×261 píxeles, cada píxel es una característica, lo que resulta en $321 * 261 = 83,781$ características.



Este es un conjunto de datos de muy alta dimensión, pero podemos usar PCA para simplificarlo.

Con PCA, podemos reducir un conjunto de datos

Compara las imágenes originales ...

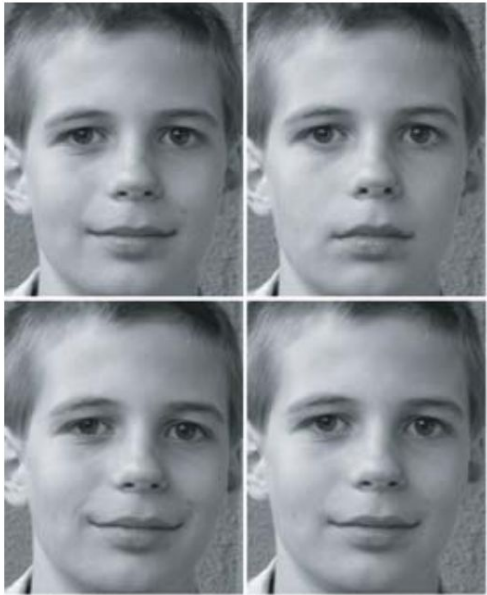


... con imágenes recreadas usando 4 componentes principales



PCA puede retener la mayor parte de la **información** del conjunto de datos original, al tiempo que condensa bastante los datos.

Fuente: <http://people.ciirc.cvut.cz/~hlavac/TeachPresEn/11ImageProc/15PCA.pdf>



Fin del módulo.



Recursos Avanzados



¿Quieres ir más allá? Aquí hay algunos recursos que recomendamos:

- Libros
 - Unsupervised learning, [Introduction to Statistical Analysis, Chapter 10.2](#)
 - Hierarchical clustering, [Introduction to Statistical Analysis, Chapter 10.3.2](#)
- Recursos en Línea
 - Clustering optimization using [Silhouette plots](#)
 - More applications of k-means: [Anomaly detection](#)



Felicidades! ¡Terminaste el módulo!

Obtén más información sobre el
machine learning de Delta para una
buena misión aquí.