Módulo 5: Aprendizaje no supervisado

Clustering 1ra parte

Diplomatura Cs. de Datos - FaCENA-UNNE

Docentes: Magdalena Lucini, Luis Duarte, Griselda Bóbeda

Clustering (Agrupamiento)

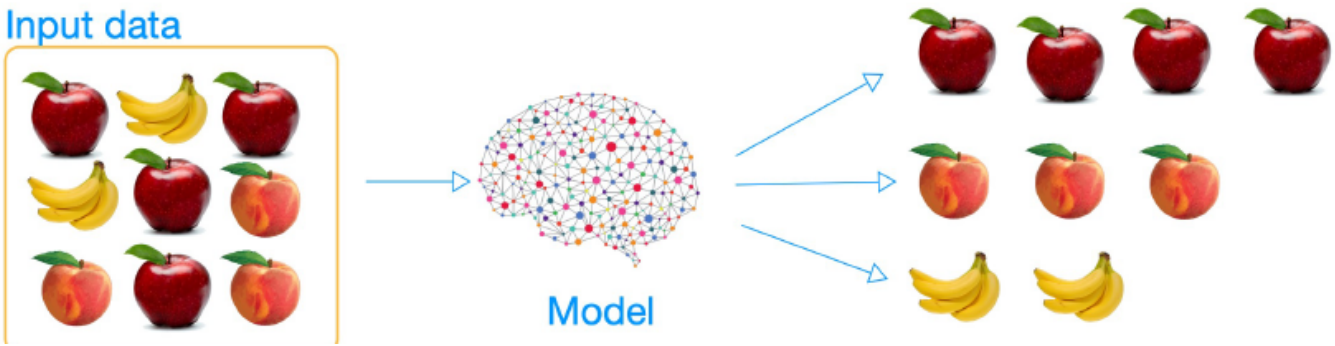
Objetivo: Encontrar la agrupación "natural " de los datos, de manera no supervisada. Agrupar objetos semejantes (basados en sus características o features). Quizás encontrar estructuras ocultas.

individuos	Variable X_1	$\ldots$	Variable X_j	$\ldots$	Variable X_p
x_1			$\vdots$		
$\vdots$			$\vdots$		
x_i	$\ldots$	$\ldots$	$x_{i,j}$		
$\vdots$					
x_n					

- n : número de individuos/objetos/elementos/observaciones $\mathrm{x}_1, \ldots, \mathrm{x}_n$
- p: número de variables $\mathrm{X}_1, \ldots, \mathrm{X}_p$
- $x_{i,j}$: respuesta de un individuo/objeto/elemento i a la variable j

¿Cómo funciona?

- Entrada: n objetos (observaciones) en un espacio p-dimensional
- Salida: conglomerados (clusters) de objetos semejantes (cercanos según alguna distancia o criterio de similitud)



¿Cómo se agrupa?

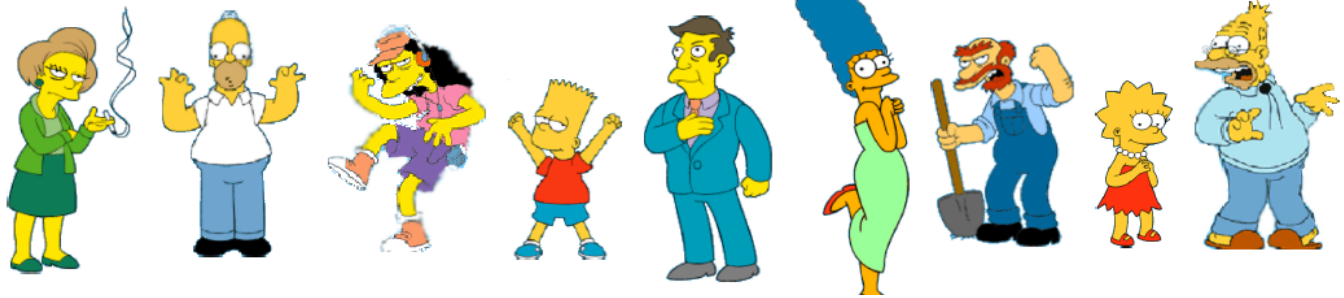


Figure: Simpson vs no Simpson

Figure: Mujeres vs Hombres

¿Cómo se agrupa?

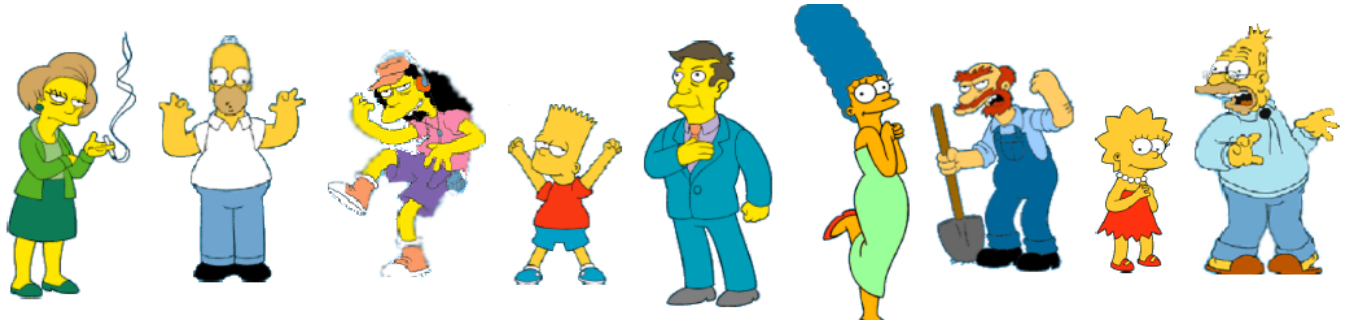


Figure: Mujeres vs Hombres

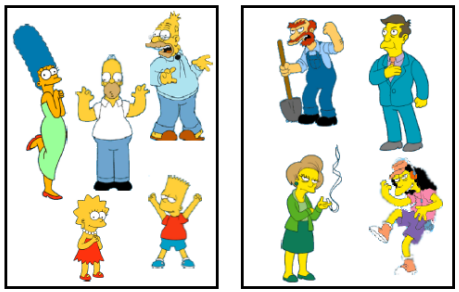
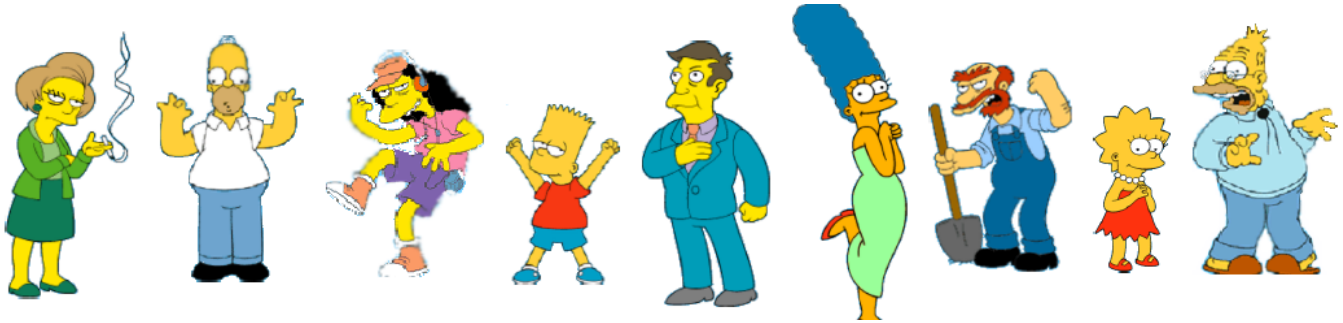


Figure: Simpson vs no Simpson

¿Cómo se agrupa?



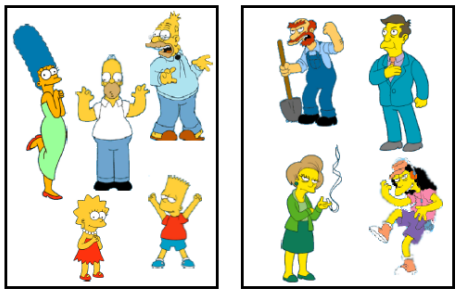


Figure: Simpson vs no Simpson

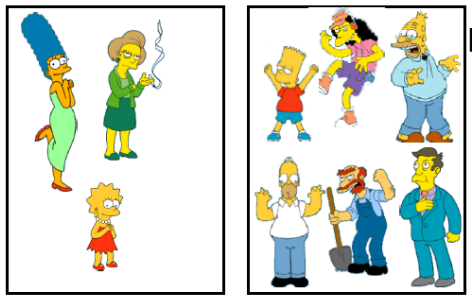


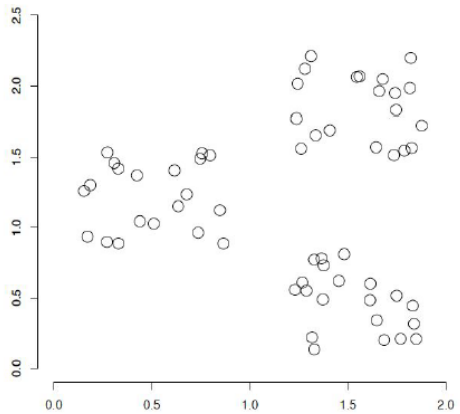
Figure: Mujeres vs Hombres

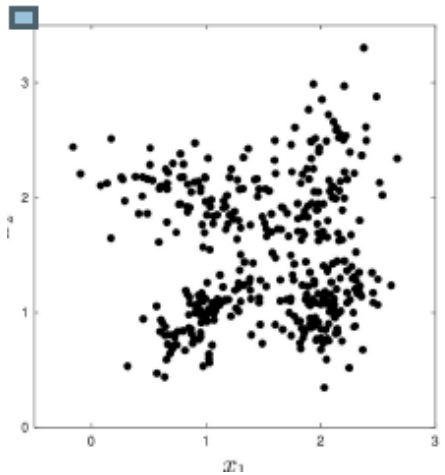
Clustering (Agrupamiento)

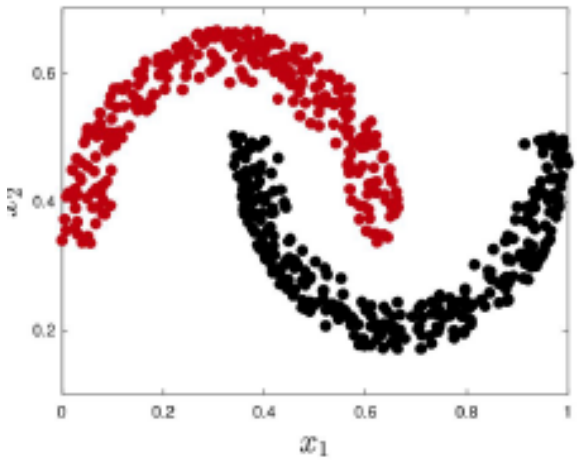
Agrupamiento

- Consiste en ordenar objetos en grupos de forma que el grado de asociación/similitud entre miembros del mismo cluster (clase) sea más fuerte que el grado de asociación/similitud entre miembros de diferentes clusters.
- Permite descubrir asociaciones y estructuras en los datos que no son evidentes a priori pero que pueden ser útiles una vez que se han encontrado.

Buscamos algoritmos que permitan encontrar clusters en situaciones como:

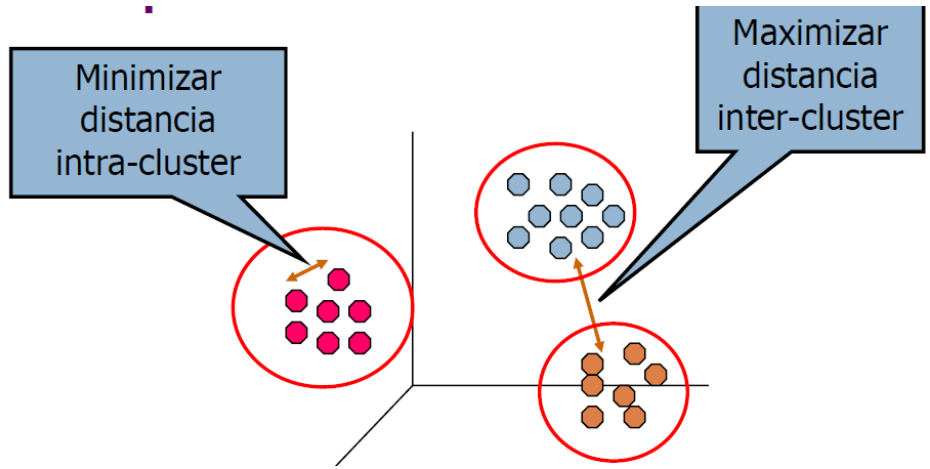






Agrupamiento - Objetivos

- Buscar patrones en un conjunto de datos agrupando los individuos en conglomerados (clusters).
- Agrupamiento óptimo:
  - elementos en cada grupo similares entre sí
  - grupos no similares entre sí
- "Similaridad"
  - Alguna medida de distancia,
  - Comparación entre la variabilidad dentro del cluster y entre los clusters
- Econtrar agrupamiento "natural" de los datos.



Agrupamiento

- Otros nombres: Reconocimiento de patrones (métodos de clasificación no supervisada), taxonomía numérica.
- Aplicaciones: Medicina, psiquiatría, geología, segmentación de imágenes, meteorología, estudios de mercado, etc.

Consideraciones:

- Naturaleza datos: ¿continuos, categóricos, ordinales, mixtos?
- Dimensionalidad del espacio: ¿es necesario reducirla antes de hacer un agrupamiento?
- ¿Hay que usar todas las variables?, ¿se debe normalizar? EDA fundamental!
- Tipo de espacio:¿qué distancia (o similitud) se debe usar?
  - Euclideo: ¿se puede usar la distancia euclidea o conviene usar otra?
  - No Euclideo: similaridades?
- ¿Cuántos clusters?: exploratorio
- ¿Clusters tienen alguna distribución?
- Métodos de validación: ¿Cuán bueno es el agrupamiento obtenido?
- Considerar distintos métodos e interacción con experto de dominio.

Definición de un algoritmo:

- Medidas de similitud/disimilitud: Se expresan en términos de distancias: Si $d(\mathrm{x}_i, \mathrm{x}_j) > d(\mathrm{x}_i, \mathrm{x}_k) \Rightarrow$ el i-esimo dato es "más parecido" al dato k-ésimo que al dato j-ésimo.
  - Distancia intra cluster
  - Distancia inter cluster
- Estrategia de particionado

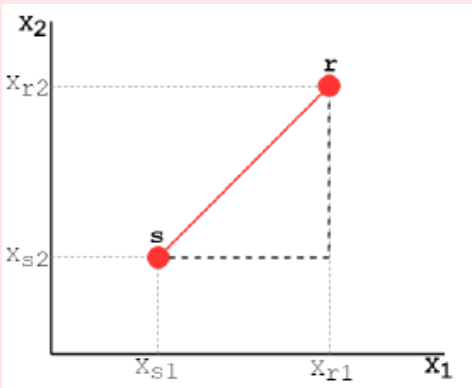
La definición de similitud/distancia depende de la naturaleza de los datos: entre documentos: semántica, entre clientes: hábitos de consumo, entre imágenes: objetos físicos distintos.

Distancias (medidas de similitud/disimilitud) variables continuas

Si $\mathrm{x}_r = (x_{r1}, \ldots, x_{rp})$ y $\mathrm{x}_s = (x_{s1}, \ldots, x_{sp})$ son dos observaciones, entonces:

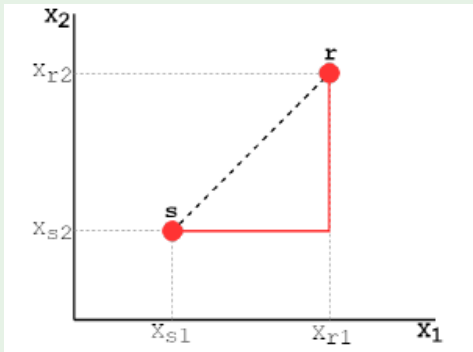
Distancia Euclidea

$$d_{rs} = \sqrt{\sum_{k=1}^{p} (x_{rk} - x_{sk})^2}$$



Distancia de Manhatan

$$d_{rs} = \sum_{k=1}^{p} |x_{rk} - x_{sk}|$$



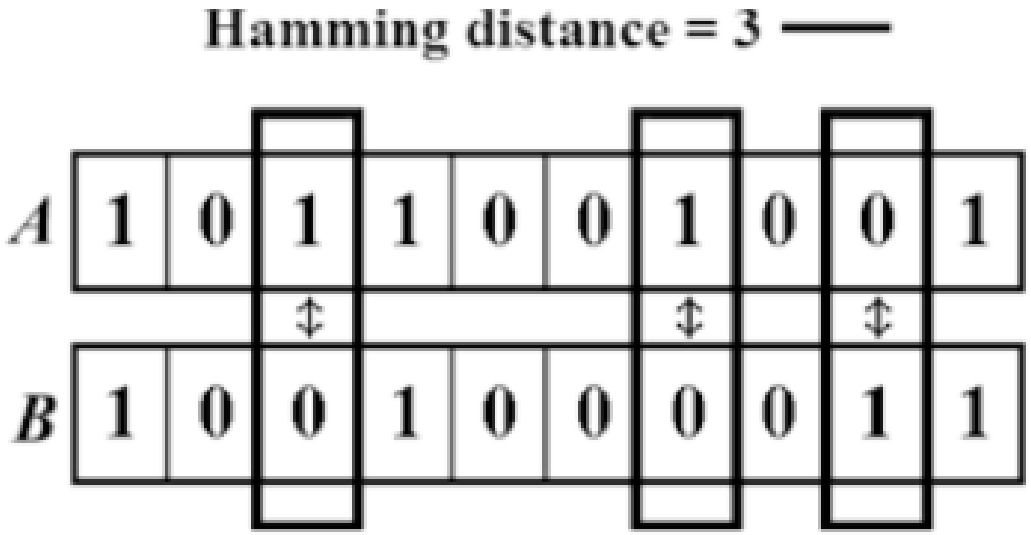
Distancia de Minkowsky

$$d_{rs}^{k} = \left[\sum_{j=1}^{p} |x_{rj} - x_{sj}|^{k} \right]^{1/k}$$

- k=1, distancia Manhattan (City Block)
- k = 2, distancia euclídea
- Si $k \to \infty$, distancia de Tchebychef

Algunas otras

- **Datos discretos**: Distancia de Hamming
  Si x_r y x_s son dos vectores de la misma dimensión con valores en $\Omega = \{0, 1, 2, \ldots, k-1\}$, entonces la distancia Hamming entre ellos se define como el número de entradas diferentes que tienen los dos vectores.



- **Datos categóricos**: Distancia de edición o distancia de Levenshtein
  Número de operaciones necesarias para transformar una cadena en otra. d("night", "noche")=3, d("efecto", "defecto")=1
- **Variables mixtas**: distancia de Gower

Tipos de agrupamiento

Partición de datos (Ej Kmeans, GMM)

Se dividen los individuos en g grupos (g prefijado) internamente homogéneos:

- Cada individuo pertenece solamente a un grupo
- Todos los individuos quedan clasificados

Métodos basados en densidad (Ej DBSCAN, HDBSCAN)

- identifica distintos grupos en los datos basándose en la idea que un cluster es una región de alta densidad de puntos
- regiones de alta densidad de puntos están separadas por regiones de baja densidad de puntos
- Útil para descubrir agrupaciones de formas variadas y detectar outliers.

Agrupamiento

Métodos Jerárquicos

- Crea un árbol de clusters
- Se estructuran los individuos(variables) en forma jerárquica por su similitud.
- Los datos se ordenan en niveles, de manera que los niveles superiores contienen a los inferiores.
- Aglomerativo:Se comienza con n clusters (1 por individuo) y se termina con un cluster que contiene los n individuos (o viceversa, algoritmo divisivo).
- Definen la estructura de asociación en cadena que puede existir entre los elementos.

Método de Partición: K-means

Número k de clusters que se desea obtener se define previamente.

1. Inicialización:Seleccionar k observaciones como centro de los grupos iniciales (semillas).
2. Asignación: Se calculan distancias euclídeas de cada observación a los centroides de los k clusters. Se asigna cada elemento al centroide más próximo. Recalcular centroides.
3. Actualización:Una vez asignados todas las observaciones a los clusters, se recalculan los centroides (media de cada grupo).
4. Iteración: Se repiten los pasos 2(asignación) y 3(actualización) hasta que se cumpla un criterio de convergencia, o hasta satisfacer un número máximo de iteraciones.

K-means

Selección de semillas (baricentros) iniciales:

- Seleccionar k observaciones aleatoriamente que estén mutuamente separadas por una distancia $> r$
- Seleccionar las primeras k observaciones del conjunto de datos que estén mutuamente apartadas por una distancia $> r$
- Seleccionar k observaciones del conjunto de datos que estén mutuamente más apartadas
- Seleccionar los k centroides de la solución de k clusters de un método jerárquico.

Resultados sensibles a la elección de las semillas iniciales y a la presencia de outliers.

Algoritmo K-means

Criterio de convergencia - Suma de cuadrados dentro

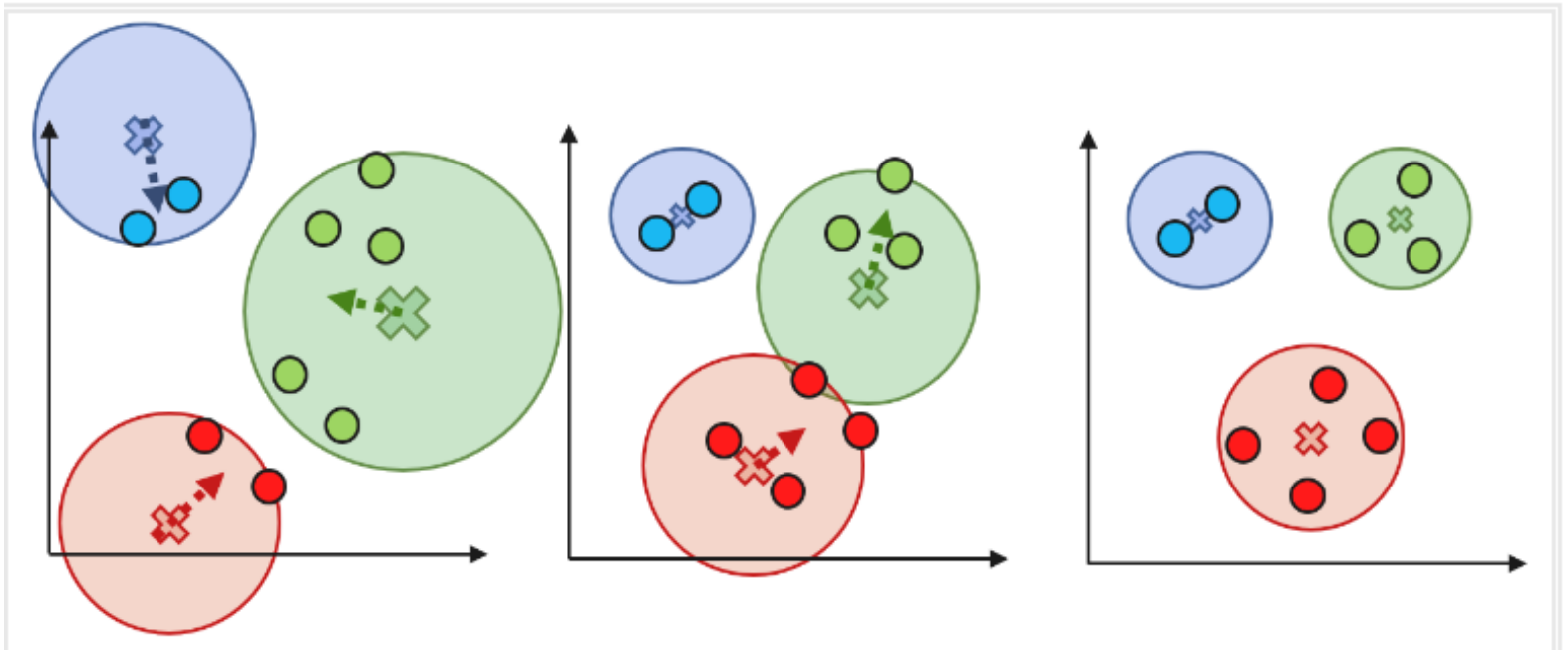
Minimizar las distancias (euclideas) al cuadrado entre los centros de los grupos y los puntos que pertenecen a ese grupo, esto es, encontrar los centroides que hagan mínima:

$$W = \sum_{g=1}^{k} \sum_{i=1}^{n_g} (x_{ig} - \bar{x}_g)(x_{ig} - \bar{x}_g)^t$$

n_g número de observaciones en el grupo g, $\bar{x}_g$= media de ese grupo

- Minimizando distancias de todas las variables en los grupos $\Rightarrow$ grupos más homogéneos.
- No es invariante ante cambios de escalas $\Rightarrow$ conviene estandarizar variables si están en distintas unidades

K-means

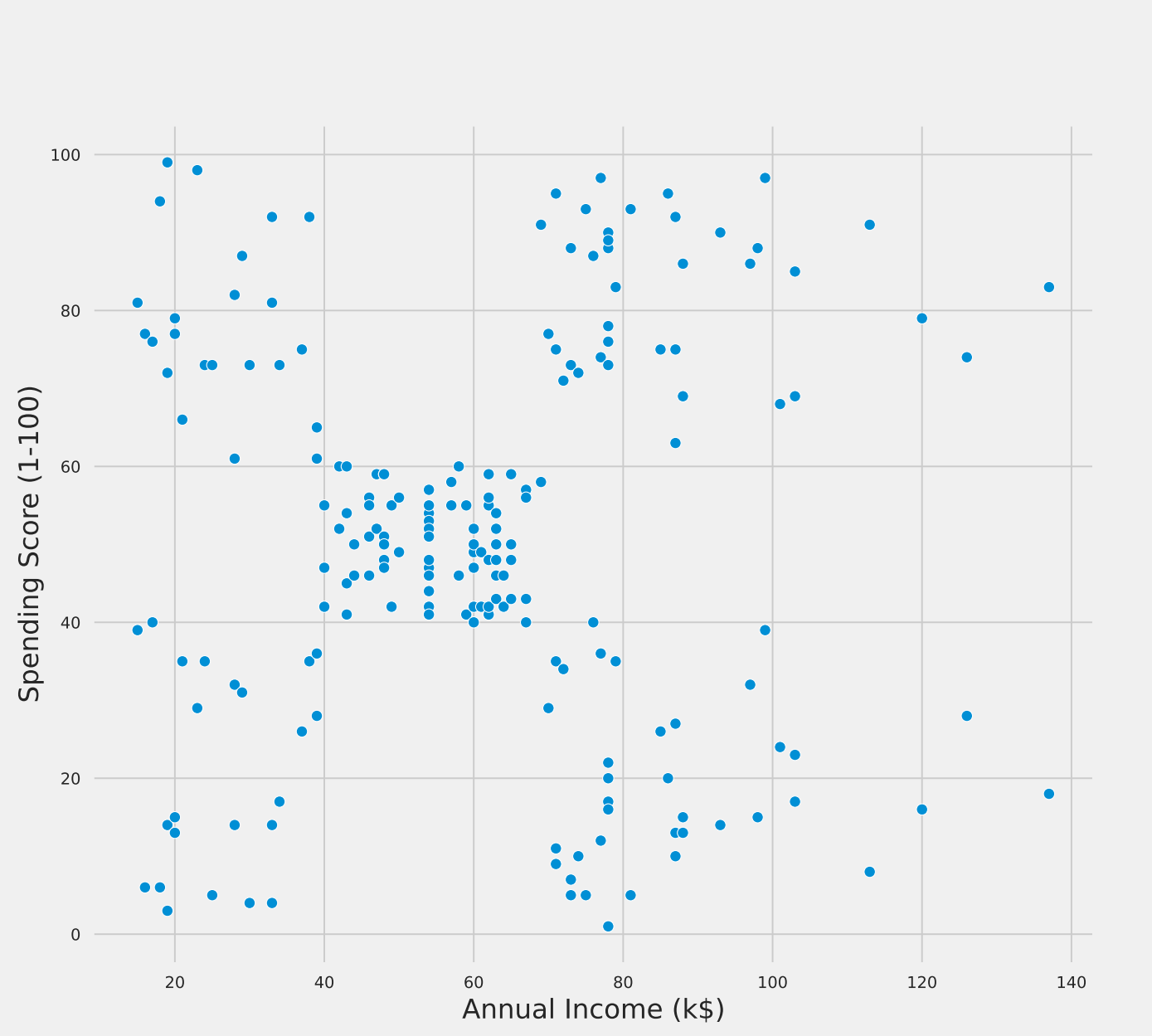


Ejemplo: Datos consumo

Ilustraremos con la base de datos "MallCustomers.csv" descargada de `https://www.kaggle.com/datasets/`donde se registraron datos de ingresos y consumos anuales de un grupo de personas basados en género y edad.

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

Datos consumo



Algoritmo K-means

Este algoritmo busca la partición óptima con la restricción de que en cada iteración sólo se permite mover un elemento de un grupo a otro.

1. Fijado el número de grupos y seleccionados los centros,
2. comprobar si moviendo algún elemento se reduce W
3. Si la respuesta es positiva $\Rightarrow$ mover elemento, recalcular medias (centroides) de los dos grupos afectados por el cambio y volver al paso anterior.
4. Si no es posible reducir W o se excedió un número prefijado de iteraciones $\Rightarrow$ terminar.

Observaciones

- Resultado afectado por semillas iniciales y asignación inicial de elementos a grupos
- Se sugiere implementarlo varias veces desde distintas semillas iniciales

Algoritmo K-means - ¿Como determinar número de clusters?

1. No existe un criterio óptimo
2. Puede seleccionarse k = número de clusters final de algún método jerárquico
3. Criterio empírico: Calcular la diferencia entre la SCD con g y $g+1$ grupos, analizando la reducción de variabilidad relativa luego de un agrupamiento adicional:

$$F = \frac{\mathrm{W(g) - W(g+1)}}{\mathrm{W(g+1)/(n-g-1)}}$$

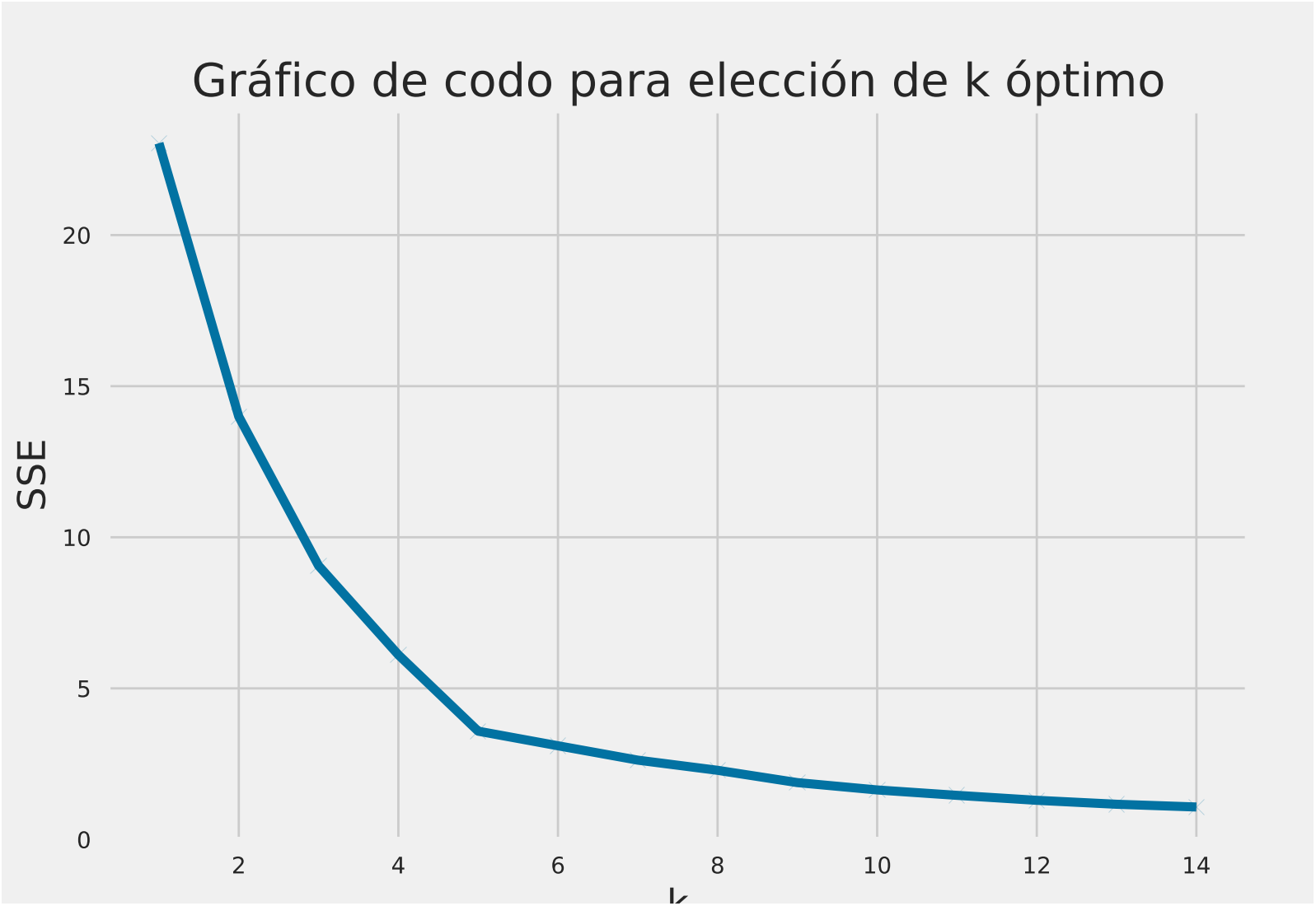
   Se compara con una distribución F con $p, p(n-g-1)$ grados de libertad. Si el cociente $F > 10$, se sugiere usar $g+1$ grupos (Hartigan (1975))
4. Gráfico de inercias en función de g (cantidad de grupos),
5. Coeficiente de silhoutte (siluetas)

Gráfico de inercias

$$\text{Inercia} = \sum_{i=1}^{g} \sum_{x \in C_i} d^2(x, \bar{x}_i)$$

- La inercia es la suma de distancias cuadradas dentro de cada cluster en la partición final
- Es una medida de cuan coherentes son los grupos
- Si se grafica la inercia en función de g, el número de grupos, se considera que el número de grupos más apropiado ocurre cuando se desacelera la reducción de la inercia.

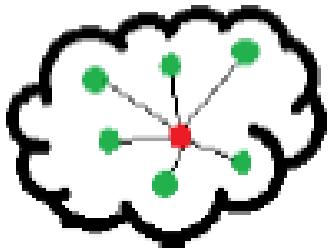
Gráfico de inercias



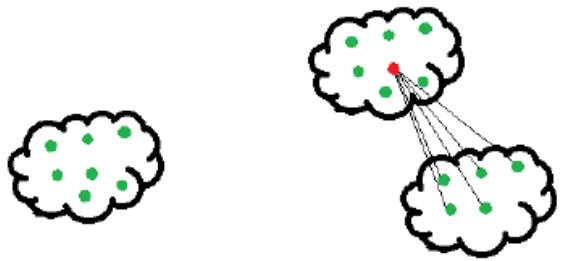
Análisis de siluetas

Para cada observación $\mathrm{x}_i \in C_k$ se calculan los indices :

- cohesión, similaridad promedio $a(i) = \frac{1}{|C_k|-1} \sum_{j \in C_k, i \neq j} d(i,j)$
  Distancia promedio de x_i a todos los puntos en el mismo cluster.



- separación, disimilaridad mínima promedio
  $b(i) = \min_{k \neq l} \frac{1}{|C_l|} \sum_{j \in C_l} d(i,j)$
  distancia promedio de x_i a todos los demás puntos en el cluster más cercano.



Análisis de siluetas

Se definen

$$s(i) = \begin{cases} 1 - a(i)/b(i) & si\ a(i) < b(i) \\ 0 & si\ a(i) = b(i) \\ b(i)/a(i) - 1 & si\ a(i) > b(i) \end{cases}$$

y

$$SC = \frac{1}{n_g} \sum_{i=1}^{n_g} s(i)$$

Métrica de uso interno (elección de medida de distancia, algoritmo de agrupamiento o de número de grupos) que indica cuán similar es un elemento a su propio grupo (cohesión) en comparación con otros grupos (separación)

Análisis de siluetas

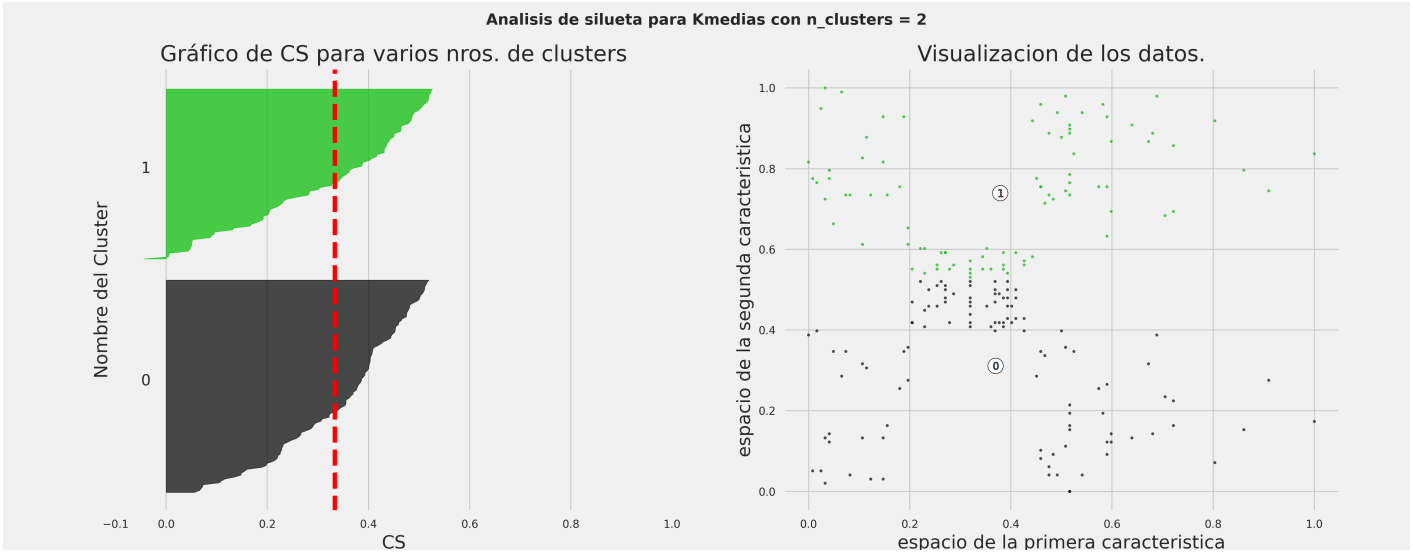
Observaciones

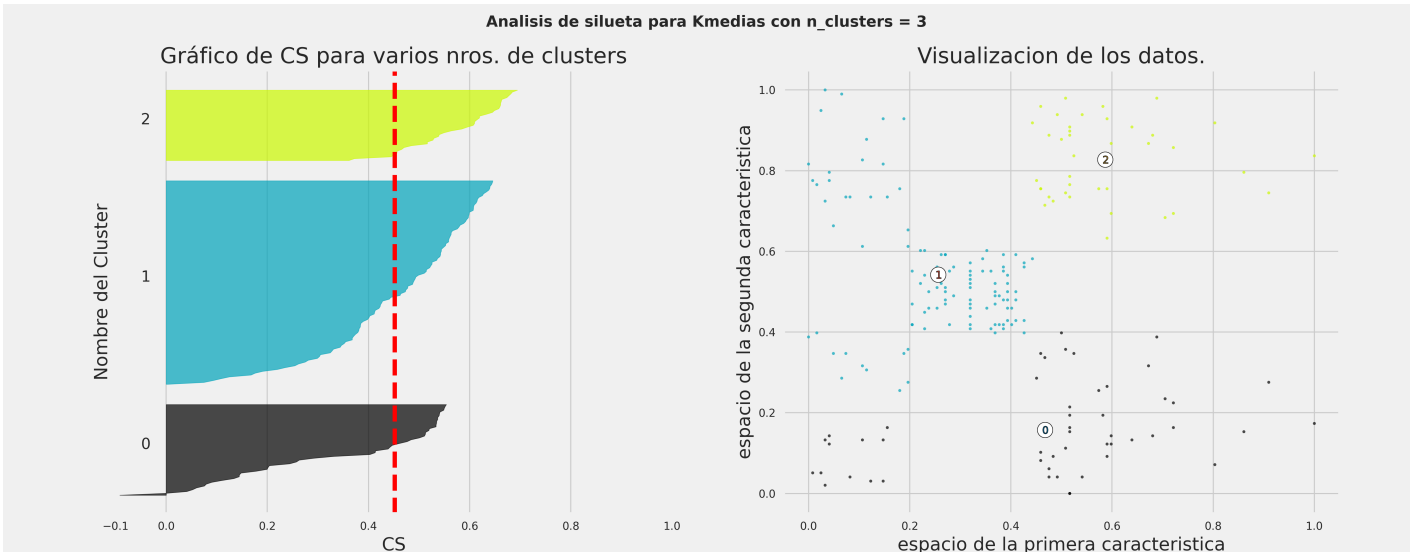
- $-1 < s(i) < 1$
- Coeficientes $s(i)$:
  - cercanos a 1 $\Rightarrow$ la muestra está lejos de los clusters vecinos
  - iguales o muy cercanos a 0 $\Rightarrow$ la muestra está muy cerca del borde de decisión entre clusters.
  - $< 0 \Rightarrow$ puntos asignados al cluster equivocado

Rousseew(1987) propuso la siguiente interpretación del coeficiente SC:

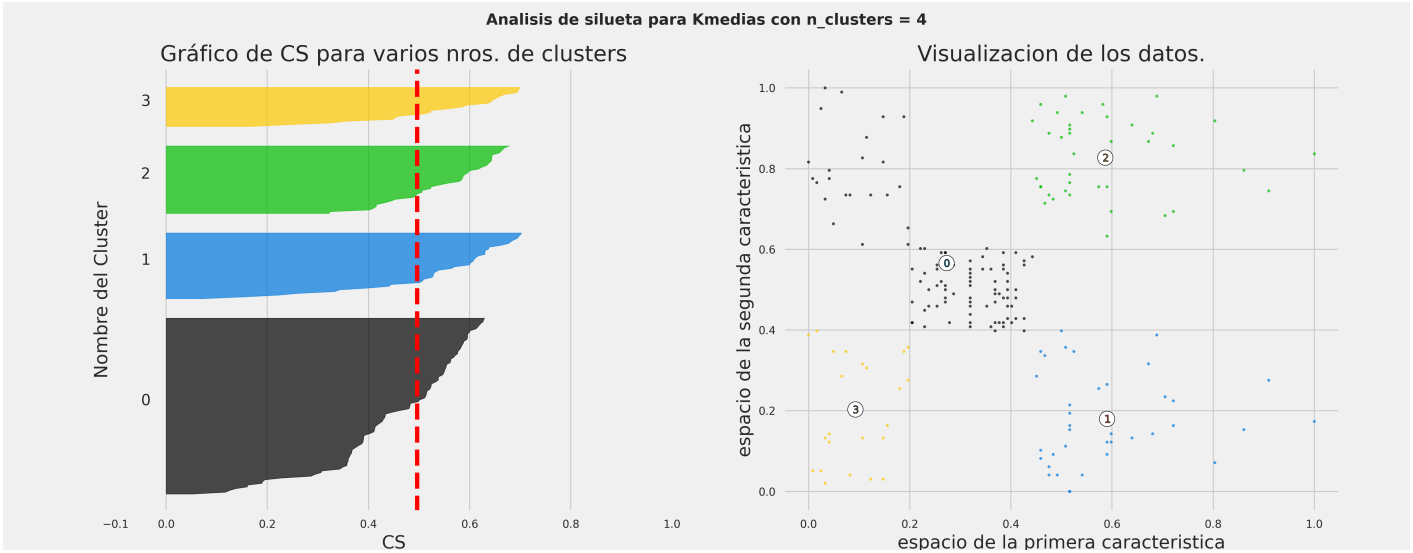
- $0.71 - 1$: estructura fuerte.
- $0.51 - 0.7$: se encontró una estructura razonable.
- $0.26 - 0.5$: estructura débil y podría ser artificial.
- menor a 0.25: sin estructura sustancial.

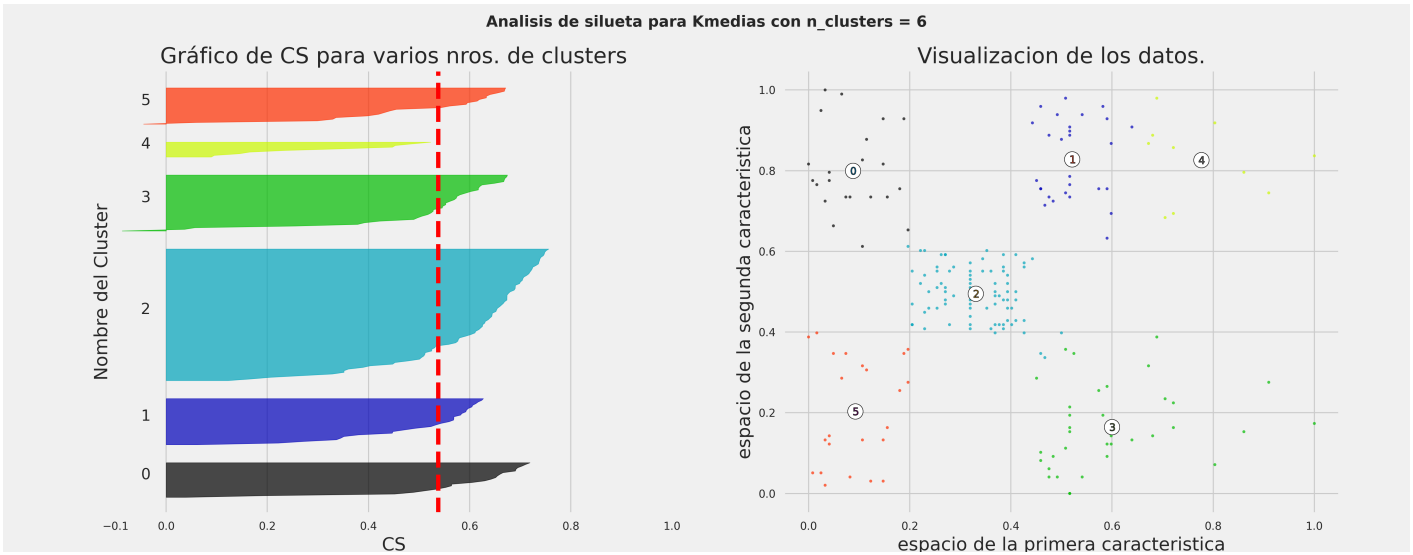
Análisis de siluetas



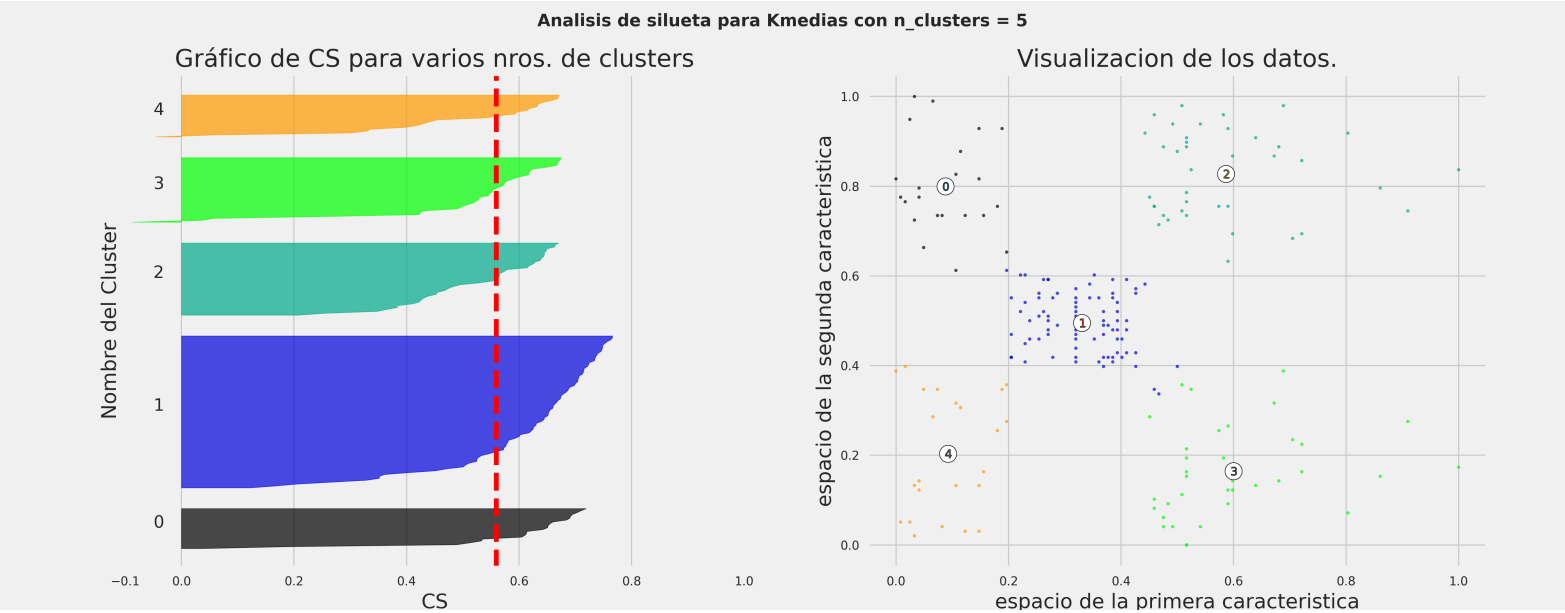


Análisis de siluetas

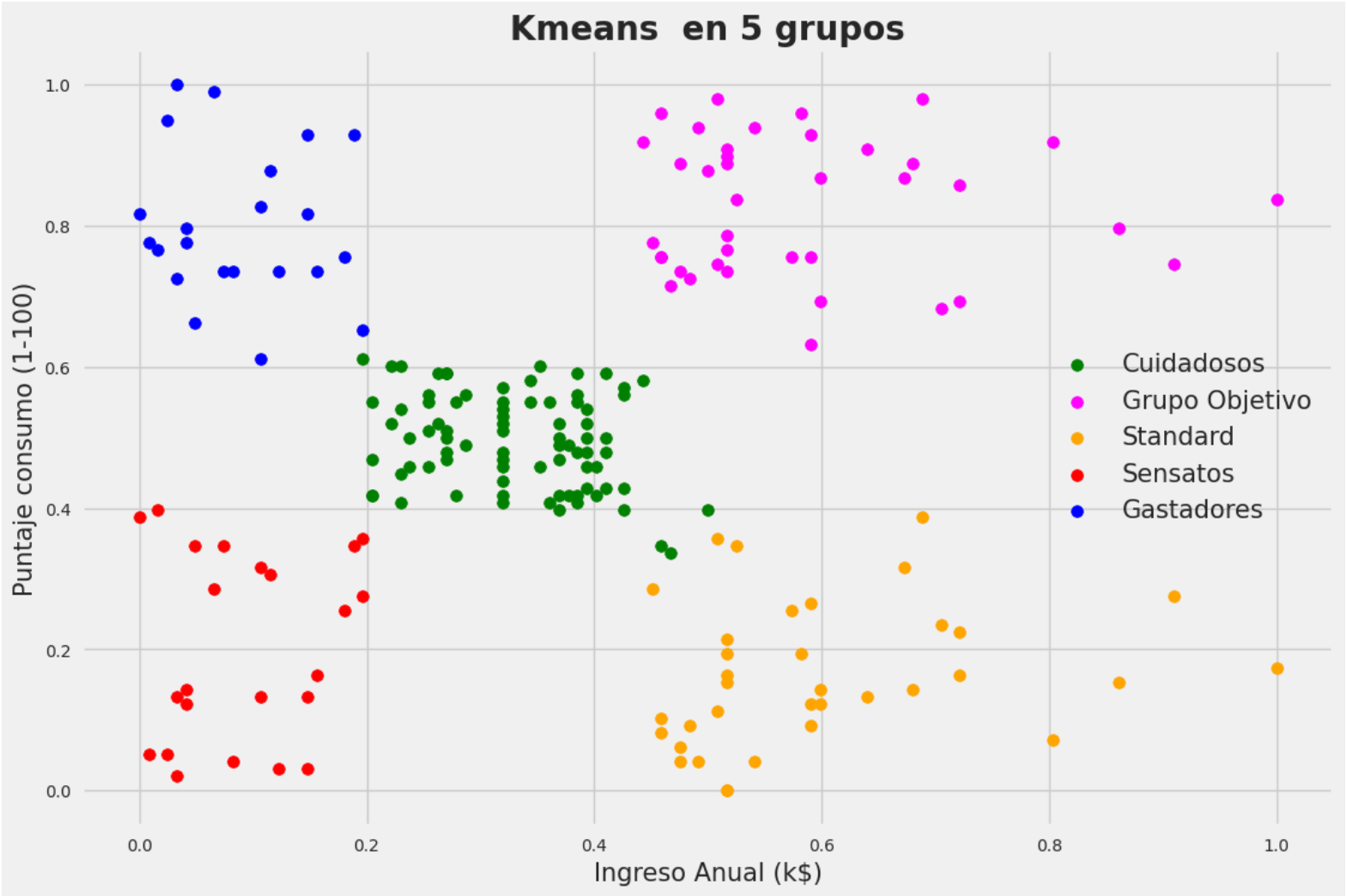




Análisis de siluetas



Ejemplo , k = 5


Kmeans en 5 grupos
Puntaje consumo (1-100)
Ingreso Anual (k$)
0.0
0.2
0.4
0.6
0.8
1.0
Cuidadosos
Grupo Objetivo
Standard
Sensatos
Gastadores

Aplicaciones K-means

- Análisis de mercado: agrupar clientes en función de su comportamiento de compra para adaptar estrategias de marketing
- Procesamiento de imágenes: segmentación (agrupa colores, texturas, etc similares)
- Agrupamiento de documentos: los organiza según contenidos (palabras claves, etc)

Variantes de K-means

- K-means++ : sólo difiere en la elección y actualización de centroides.
- Variables categóricas: Kmodas
- Variables mixtas: K-medoids, K-prototypes

Kmeans en sklearn

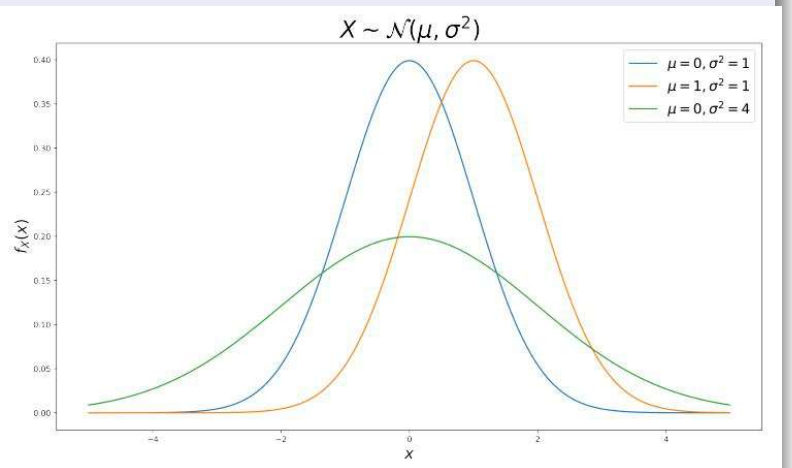
class sklearn.cluster.KMeans(n_clusters=8, *, init='k-means++', n_init='auto', max_iter=300, tol=0.0001, verbose=0, random_state=None, copy_x=True, algorithm='lloyd')[source]

Mezcla de Gaussianas (GMM: Gaussian Mixed Models)

Univariada

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

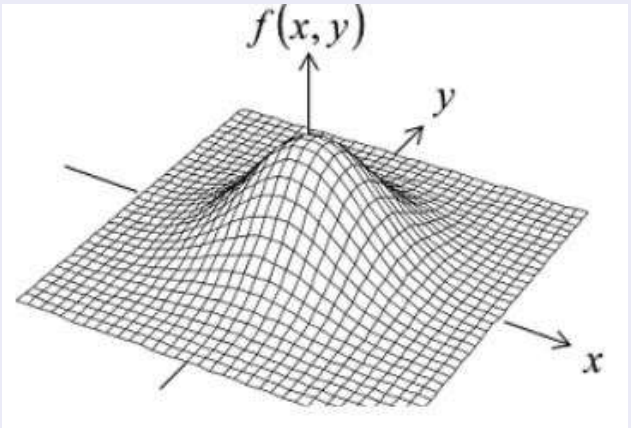
$x \in \mathbb{R}$



Multivariada

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{k/2}\det(\Sigma)^{1/2}} \times \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^t \Sigma^{-1}(\mathbf{x}-\mu)\right)$$

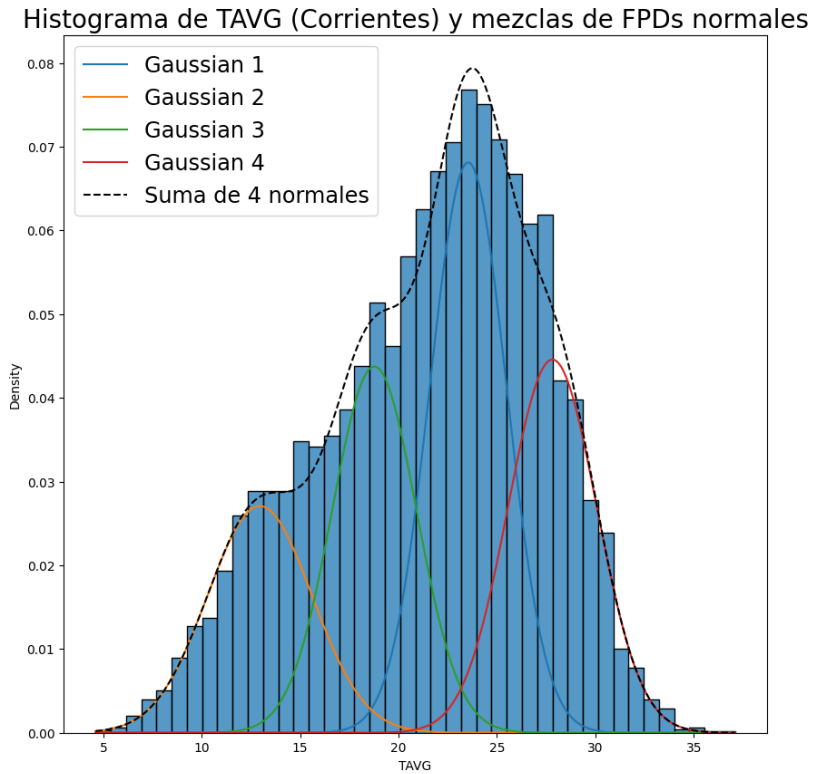
$x \in \mathbb{R}^k$



Suma de Gaussianas

Si $X_1, X_2, \ldots, X_n$ son variables aleatorias gaussianas, entonces

$Y = X_1 + X_2 + \ldots + X_n$ también es gaussiana! (la suma de gaussianas, es gaussiana)



Observaciones

- El agrupamiento por K-means coloca una hiper-esfera en el centro de cada cluster, con radios definido por el punto más distante al centroide del cluster (figura a).
- Si los objetos tiene otra distribución, este tipo de agrupamiento no es bueno. (figura b)

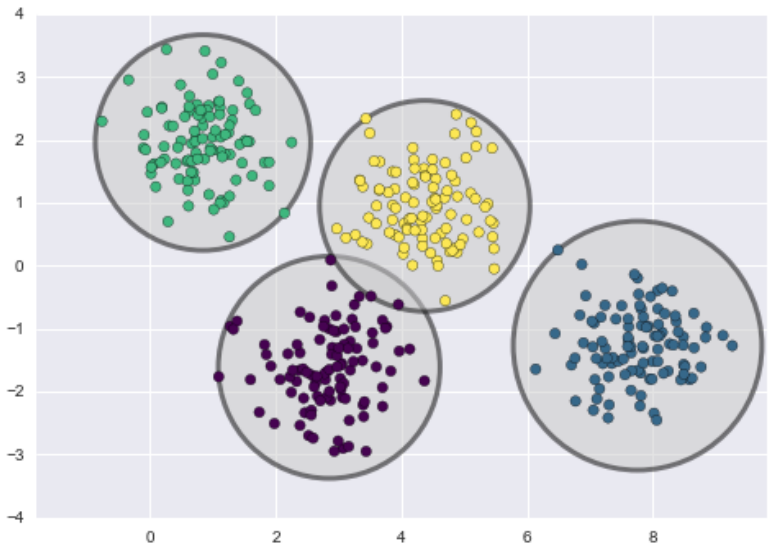


Figure: a

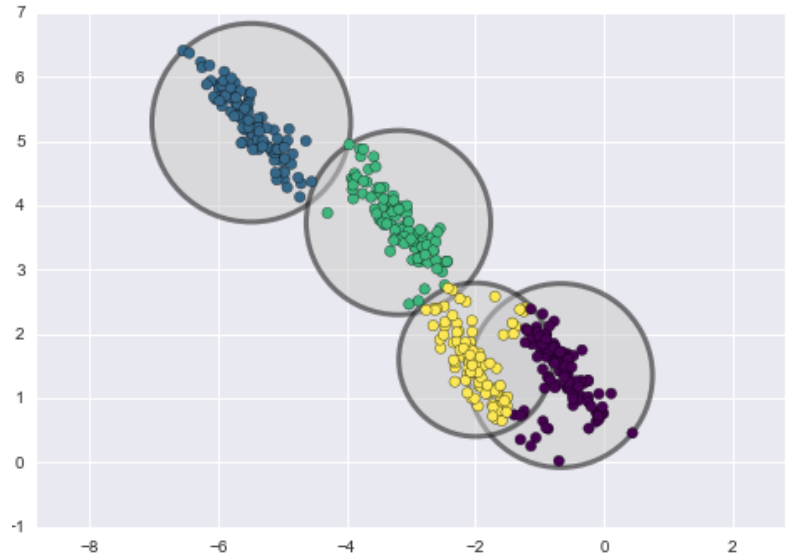


Figure: b

Agrupamiento GMM

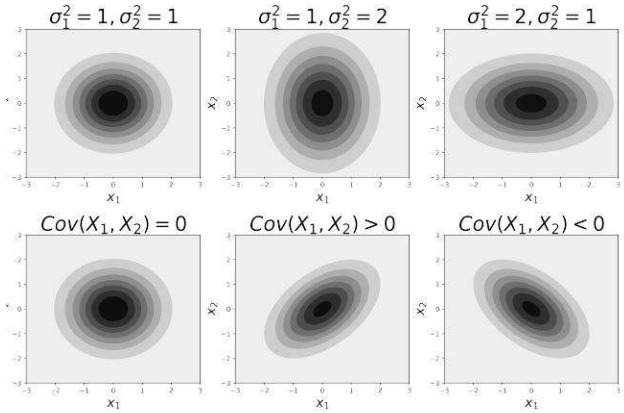


Figure: Proyeccion matriz covarianza gaussiana bivariada

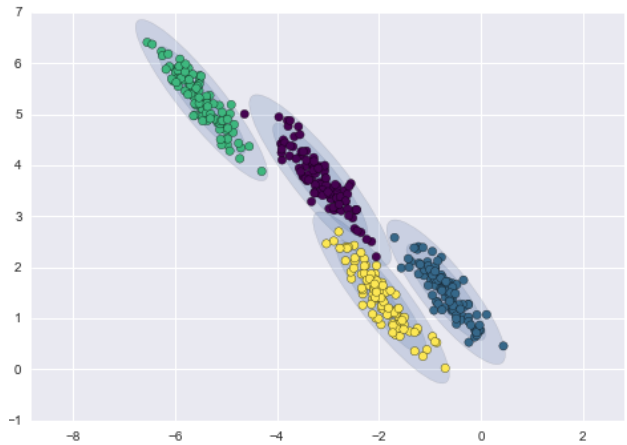


Figure: Agrupamiento con GMM

GMM

- Los modelos de mezcla gaussiana suponen que los datos fueron generados por una mezcla de varias distribuciones gaussianas, cada una de ellas representando un cluster. Esta suposición permite a los GMM modelar clusters de diferentes formas y tamaños, incluidos los clusters elípticos.
- En lugar de conformar clusters al centroide más cercano, se ajustan k distribuciones gaussianas a los datos (objetos)
- Se deben estimar los parámetros (media y varianza) para cada cluster, y el peso que se le dará a cada cluster. Para la estimación se usa el algoritmo EM (expectation -maximization) o máxima verosimilitud.
- Una vez estimados estos parámetros, para cada uno de los datos se calcula la probabilidad que pertenezca a cada uno de los datos y se multiplica cada distribución por un pesoπ_i (teniendo en cuenta la cantidad de objetos (datos) que pertenecen a cada cluster)

Algoritmo GMM

1. Decidir número de clusters (conocimiento de dominio, algún otro método de agrupamiento, criterio BIC, etc)
2. Dar valores iniciales a las medias μ_i, matrices de covarianza Σ_i y coeficientes de peso π_i
3. Usar algoritmo EM para:
    1. Expectation step (E): calcular la probabilidad que cada observación pertenezca a cada cluster. Luego evaluar la función de verosimilitud usando los parámetros actuales.
    2. Maximization step (M): actualizar los parámetros anteriores (medias, covarianzas, pesos) para maximizar la función de verosimilitud encontrada en el paso E.
    3. repetir hasta satisfacer algún criterio de convergencia.

Algoritmo GMM

Se asume que cada observación x_i proviene de una distribución de pobabilidad mezcla de K componentes gaussianas

Paso 1

Inicializar aleatoriamente

- medias μ_i
- covarianzas Σ_i
- pesos (coeficientes de mezclado) π_i Al inicio es igual para todos los clusters ($\pi_i = 1/k$ si se eligen k clusters).

Algoritmo GMM

Paso 2: E-step

Para cada observación x_i se calcula la probabilidad que pertenezca al cluster c y se calculan las responsabilidades

$$r_{ic} = \frac{\pi_c f(\mathrm{x}_i/\mu_c, \Sigma_c)}{\sum_{j=1}^{k} \pi_j f(\mathrm{x}_i/\mu_j, \Sigma_j)}$$

La responsabilidad r_{ic} mide cuán responsable es la c-ésima distribución gaussiana en la generación de la i-ésima observación.

El resultado de este paso es un conjunto de responsabilidades para cada punto y cada una de las k distribuciones.

Algoritmo GMM

Paso 3: M-step

Se usan las responsabilidades calculadas en el paso anterior para actualizar los estimadores de los parámetros del modelo

- $\pi_c = \frac{\sum_{i=1}^{n} r_{ic}}{n}$
- $\mu_c = \frac{\sum_{i=1}^{n} r_{ic} x_i}{\sum_{i=1}^{n} r_{ic}}$
- $\Sigma_c = \frac{\sum_{i=1}^{n} r_{ic}(x_i - \mu_c)^2}{\sum_{i=1}^{n} r_{ic}}$

Estos parámetros actualizados se usan nuevamente en el paso E para calcular nuevas responsabilidades. Y así se sigue iterando hasta satisfacer algún criterio de convergencia.

GMM agrupa los datos basándose en las probabilidades más altas.

Criterios de validación

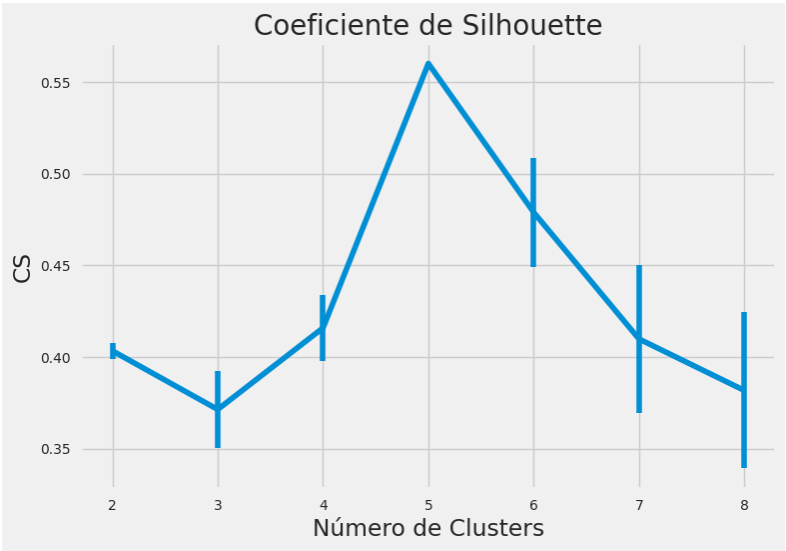
Pueden usarse los criterios ya vistos para validar número de clusters elegidos (siluetas, gráfico de codo). También puede usarse el criterio BIC (Bayesian Information Criterium)

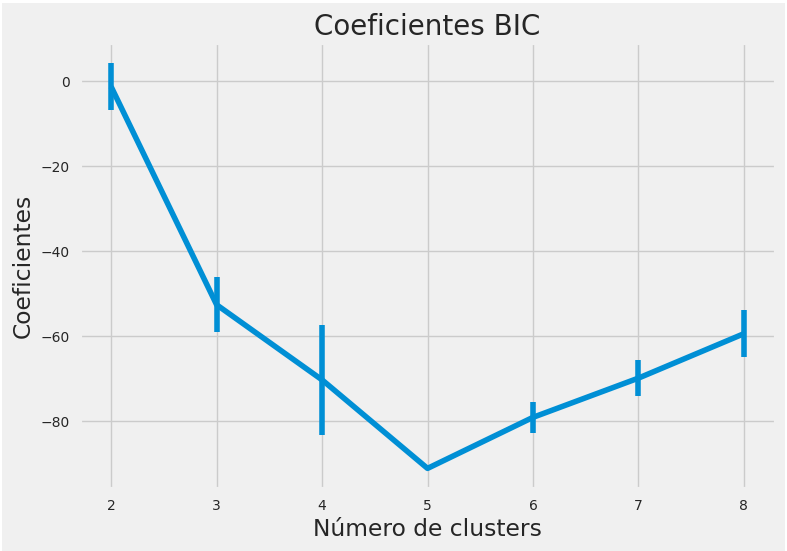
BIC (Bayesian Information Criterium

- Busca el modelo más estable entre un conjunto de modelos
- Mide la capacidad explicativa de un modelo y lo penaliza en función de su complejidad (número de parámetros del modelo)
- $BIC = -2 \cdot \ln(L) + \ln(n) \cdot k$
  $L =$ verosimilitd del conjunto de datos al modelo, n número de observaciones, k número de clusters.

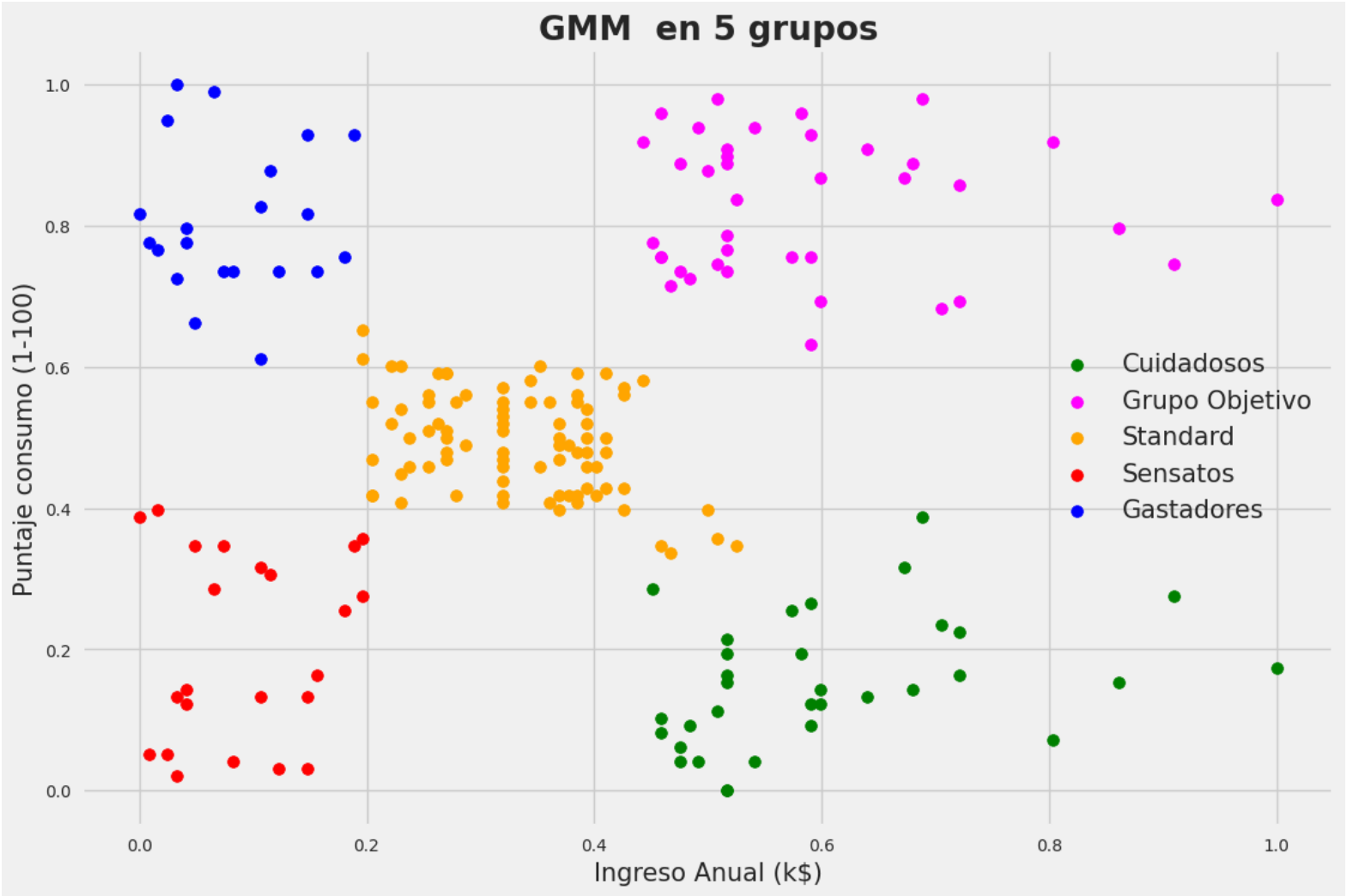
Se elige el modelo que tenga BIC más pequeño

Validación agrupamiento GMM





Ejemplo GMM , k = 5



Comentarios

- Al igual que Kmeans, GMM es sensible a parámetros iniciales
- Si se propone un número elevado de clusters, los GMM pueden sobreajustar los datos y capturar ruido
- Aplicaciones: deteccion de anomalías (identificación de valores atípicos en finanzas, tráfico de red), segmentación de imágenes, etc