

Análisis de conglomerados: Métodos I

Diplomatura en Ciencia de Datos,
Aprendizaje Automático y sus Aplicaciones
FaMAF-UNC
2021

Mapa de ruta

- 1. Atributos continuos y discretos**
- 2. Distancias y Similaridades**
- 3. Familias de Algoritmos Aglomerativos**
- 4. Elección de número de grupos**

Problema...



Problema...

- ❖ Jugadores federados tienen posición en el campo..
- ❖ Esas clases, emergen de las características generales de juego?
- ❖ Son muchas clases



Problema...

Todas estas etiquetas son muchas, si queremos que emergan grupos generales, podríamos esperar observar los siguientes grupos

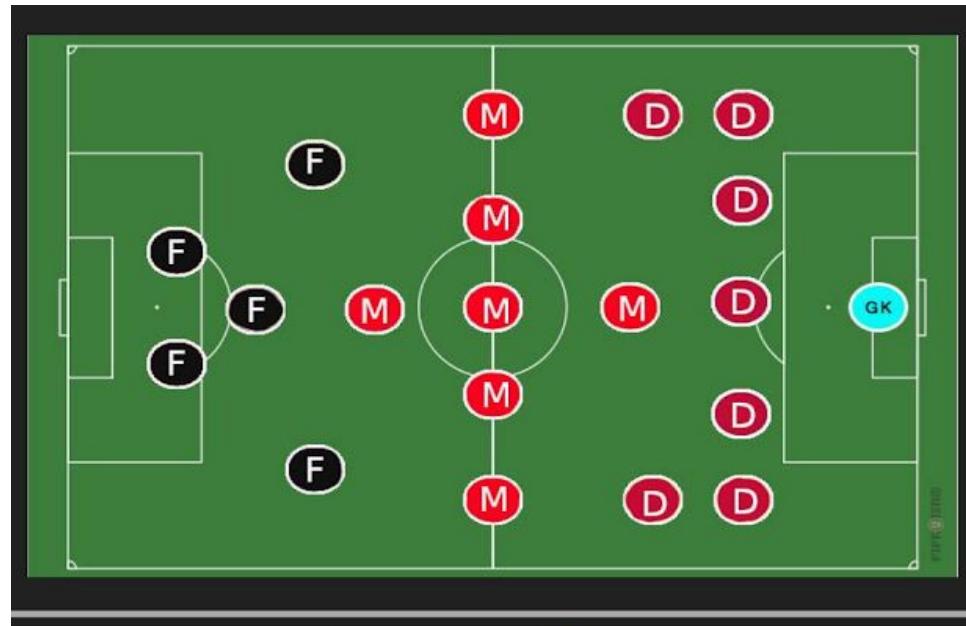
- ❖ Defensa
- ❖ Mediocampo
- ❖ Ataque
- ❖ Arqueros



Problema...

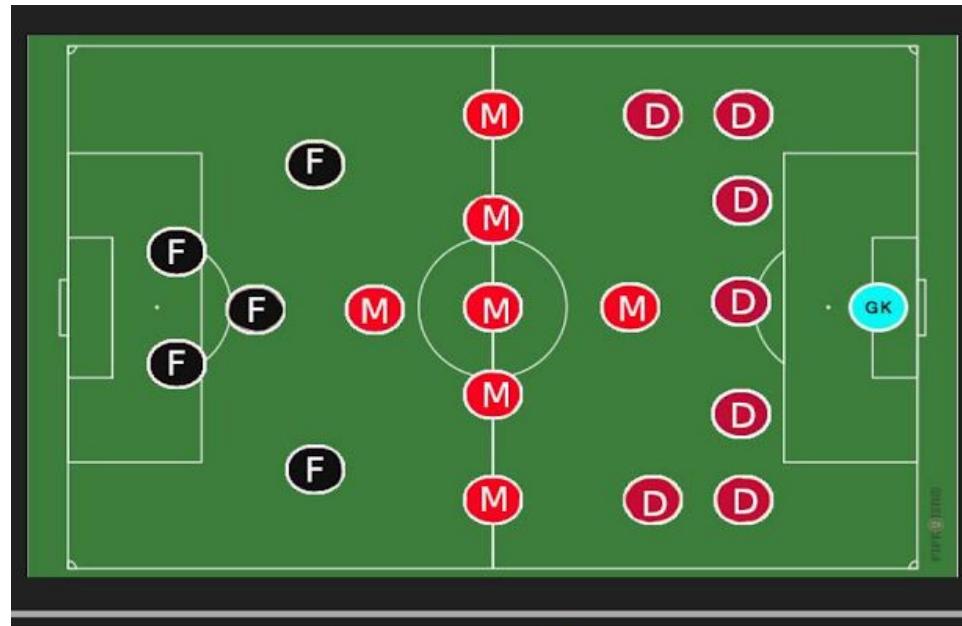
Todas estas etiquetas son muchas, si queremos que emergan grupos generales, podríamos esperar observar los siguientes grupos

- ❖ Defensa
- ❖ Mediocampo
- ❖ Ataque
- ❖ Arqueros



Objetivo...

- ❖ Este es mi objetivo!! agrupar jugadores usando características generales en cuatro grupos que representen posiciones preferidas de juego
- ❖ Defensa
- ❖ Mediocampo
- ❖ Ataque
- ❖ Arqueros



Datos...

- ❖ 18278 Jugadores
- ❖ 104 Características

Agrupamiento latente

- ❖ Defensa
- ❖ Mediocampo
- ❖ Ataque
- ❖ Arqueros



Pre-procesamiento

- ❖ Atributos continuos
 - Para evitar que unas variables dominen sobre otras los valores de los atributos se estandarizan a priori
 - estandarización (llevar las variables a un mismo rango de valores)
 - normalización (llevar las variables al rango $N(0,1)$)
 - `from sklearn.preprocessing import MinMaxScaler`
 - `from sklearn.preprocessing import StandardScaler`
- ❖ Atributos categoricos
 - encoding mediante transformaciones
 - <https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing-categorical-features>

Distancias: datos continuos

Distancia de Minkowski

$$d_r(x, y) = \left(\sum_{j=1}^J |x_j - y_j|^r \right)^{\frac{1}{r}}, \quad r \geq 1$$

- Distancia de Manhattan ($r=1$) / city block / taxicab

$$d_1(x, y) = \sum_{j=1}^J |x_j - y_j|$$

- Distancia euclídea ($r=2$):

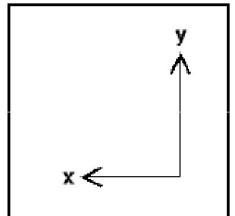
$$d_2(x, y) = \sqrt{\sum_{j=1}^J (x_j - y_j)^2}$$

- Distancia de Chebyshev ($r \rightarrow \infty$) / dominio / chessboard

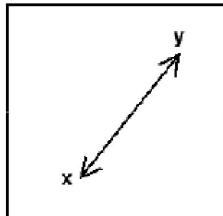
$$d_\infty(x, y) = \max_{j=1..J} |x_j - y_j|$$

Distancias: datos continuos

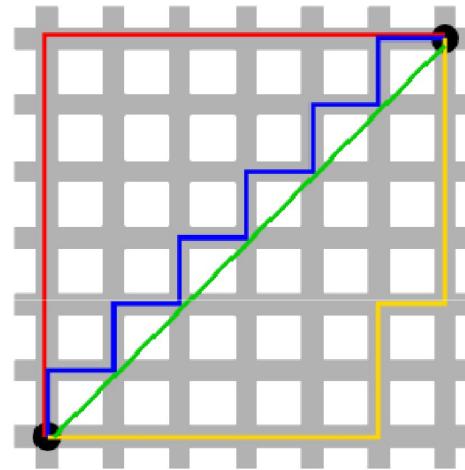
Distancia de Minkowski



Manhattan



Euclidean



- Distancia de Manhattan = 12 (roja, azul o amarilla)
- Distancia euclídea ≈ 8.5 (verde - continua)
- Distancia de Chebyshev = 6 (verde - discreta)

Distancias: datos continuos

Distancia de Chebyshev

$$d_\infty(x, y) = \max_{j=1..J} |x_j - y_j|$$

También conocida como distancia de tablero de ajedrez (chessboard distance): Número de movimientos que el rey ha de hacer para llegar de una casilla a otra en un tablero de ajedrez.

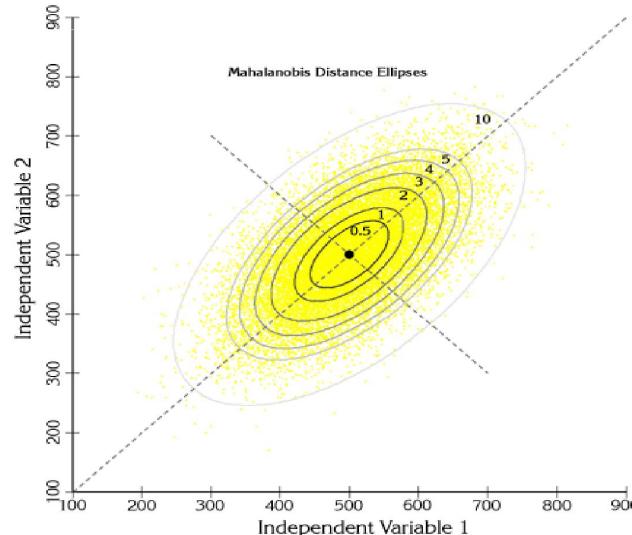
	a	b	c	d	e	f	g	h
8	5	4	3	2	2	2	2	2
7	5	4	3	2	1	1	1	2
6	5	4	3	2	1	1	1	2
5	5	4	3	2	1	1	1	2
4	5	4	3	2	2	2	2	2
3	5	4	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4
1	5	5	5	5	5	5	5	5
	a	b	c	d	e	f	g	h

Distancias: datos continuos

Distancia de Mahalanobis

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y})}.$$

- Considera las correlaciones entre variables.
- No depende de la escala de medida.



Distancias: datos discretos

Distancia de edición = Distancia de Levenshtein

Número de operaciones necesario
para transformar una cadena en otra.

$$d(\text{"data mining"}, \text{"data minino"}) = 1$$

$$d(\text{"efecto"}, \text{"defecto"}) = 1$$

$$d(\text{"poda"}, \text{"boda"}) = 1$$

$$d(\text{"night"}, \text{"natch"}) = d(\text{"natch"}, \text{"noche"}) =$$

Aplicaciones: Correctores ortográficos, reconocimiento de voz,
detección de plagios, análisis de ADN...

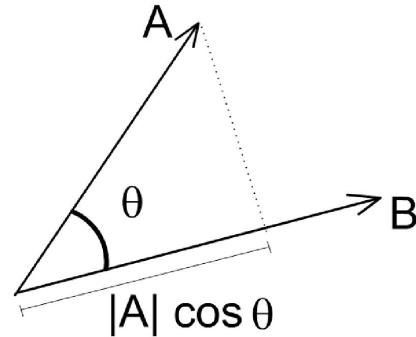
Para datos binarios: Distancia de Hamming

Similaridades

Medidas de correlación

- Producto escalar

$$S.(x, y) = x \cdot y = \sum_{j=1}^J x_j y_j$$



- “Cosine similarity”

$$\cos(\vec{x}, \vec{y}) = \sum_i \frac{x_i \cdot y_i}{\sqrt{\sum_i x_i^2} \cdot \sqrt{\sum_i y_i^2}}$$

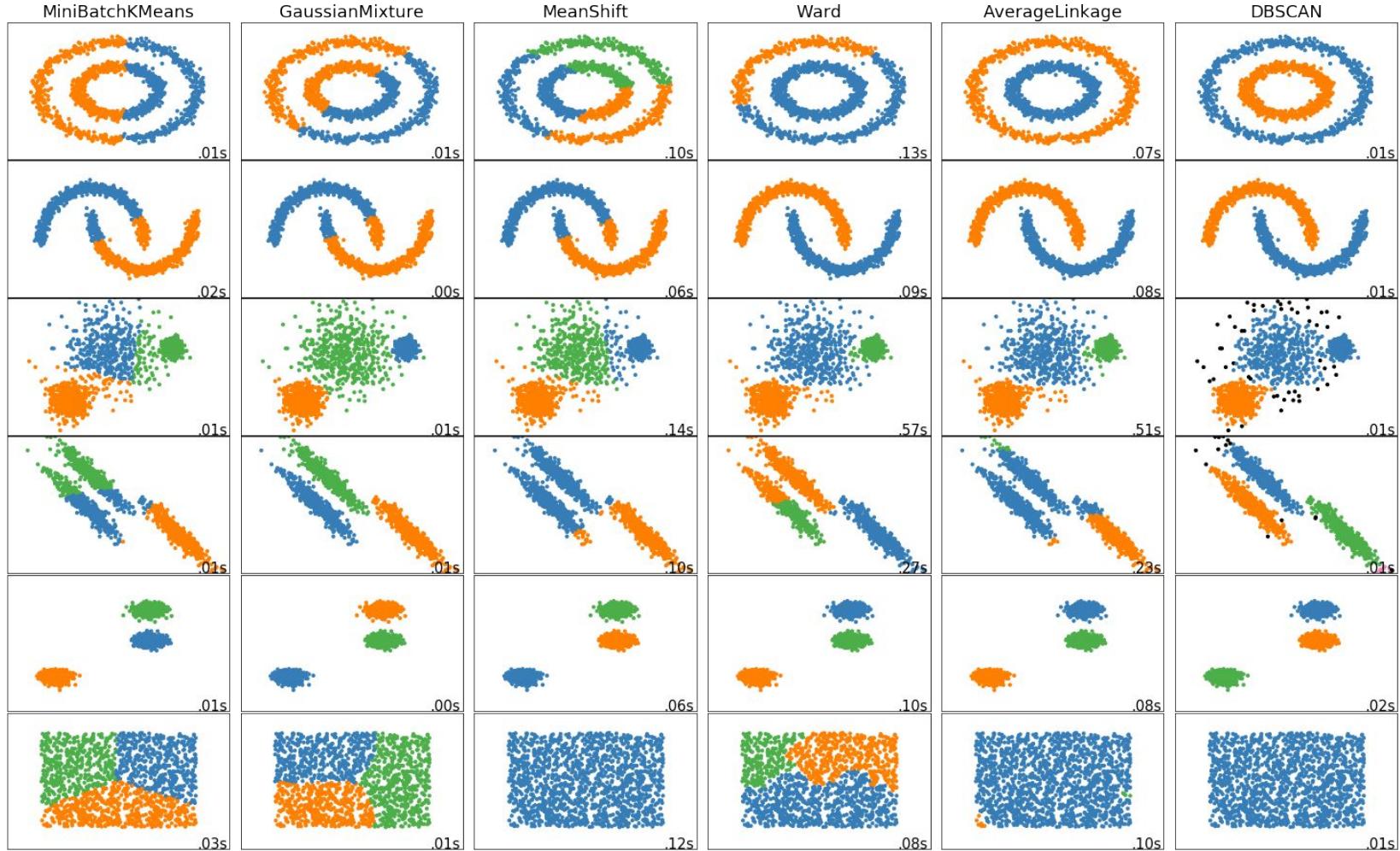
- Coeficiente de Tanimoto

$$s(\vec{X}, \vec{Y}) = \frac{\vec{X}^t \cdot \vec{Y}}{\vec{X}^t \cdot \vec{X} + \vec{Y}^t \cdot \vec{Y} - \vec{X}^t \cdot \vec{Y}},$$

Familias de algoritmos de clustering

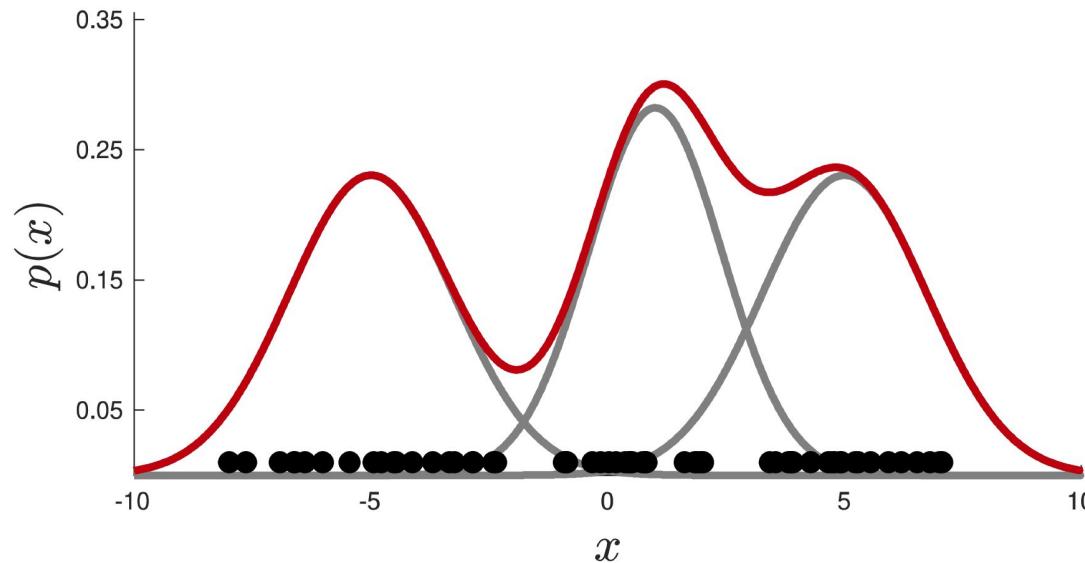
- ❖ Métodos generativos
 - Mezcla de gaussianas, MeanShift
- ❖ Agrupamiento por particiones
 - k-Means, PAM/CLARA/CLARANS
- ❖ Métodos basados en densidad
 - DBSCAN, Optics, DenClue
- ❖ Clustering jerárquico
 - Ward, Diana/Agnes, BIRCH, CURE, Chameleon, ROCK
- ❖ Note_fig1.ipynb crea la figura siguiente

Note_fig1.ipynb



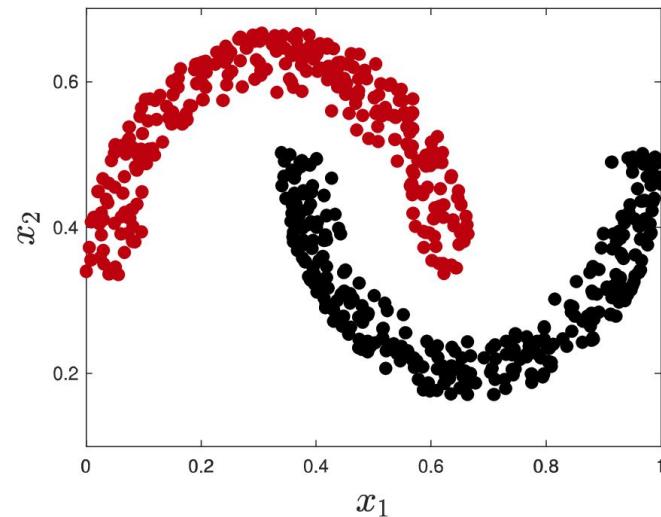
Mezcla de Gaussianas

- ❖ Supongamos tener alguna información
 - Consideremos que estos datos son reales,
 - puedo trabajar con la distancia Euclídea.
 - datos producidos por una densidad mezcla de Gaussianas,

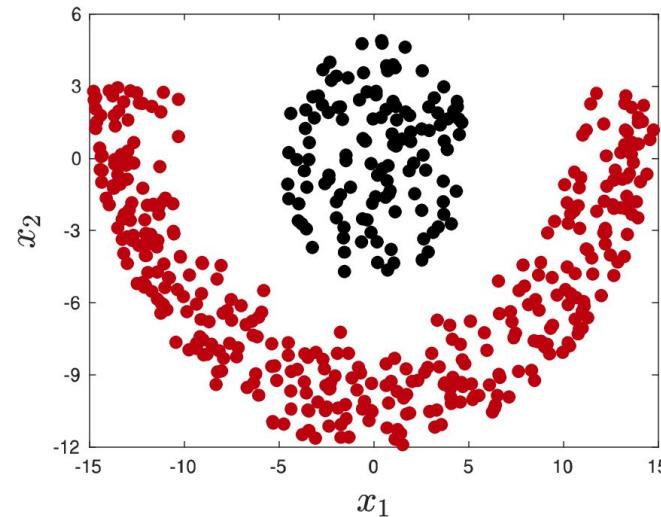


Mezcla de Gaussianas

- ❖ Cualquier dato puede ser modelado con una mezcla de gaussianas?



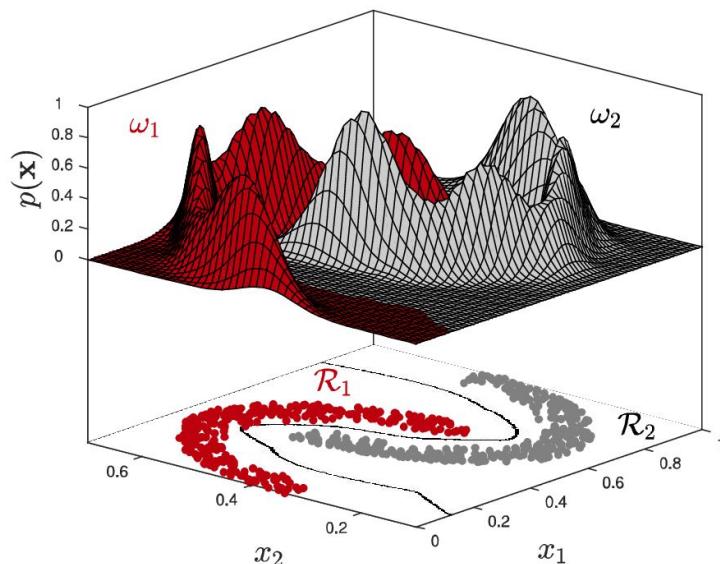
(a)



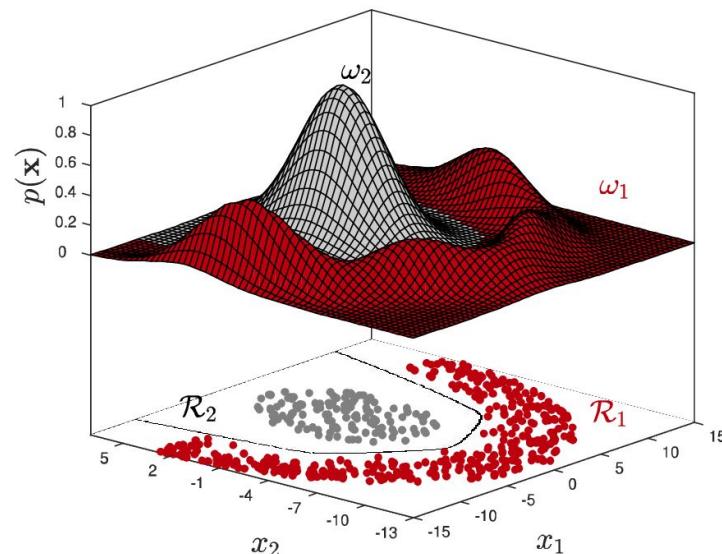
(b)

Mezcla de Gaussianas

- ❖ No todos, pero muchos si se pueden modelar, si uno conoce la cantidad de gaussianas que forman la mezcla
- ❖



(a)



(b)

Como funciona el GMM?

- ❖ Si uno fija la cantidad de gaussianas que uno considera que hay en la mezcla,
 - se estiman los parámetros de cada gaussiana y los parámetros de representación
 - se imputa cada dato como proveniente de la una de las componentes de la mezcla.
 - La estimación se realiza mediante el algoritmo Expectation Maximization.
-

Como funciona el GMM?

Expectation Maximization Algorithm

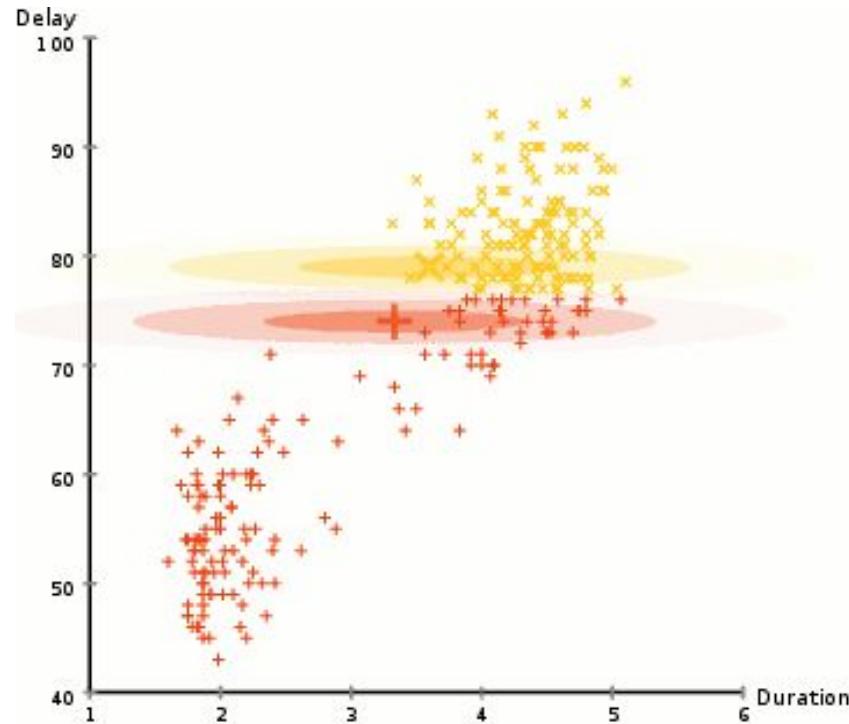
Input: $\mathbf{X} = \{\mathbf{x}_i | i = 1, \dots, n\}$ datos, m =numero de componentes, ϵ tolerancia

Output: $\hat{\Theta}$

- ## $\hat{\Theta}^0 \leftarrow \left[\left(\hat{\theta}_1, \hat{p}_1 \right)^0, \dots, \left(\hat{\theta}_m, \hat{p}_m \right)^0 \right]$
 - ## $t \leftarrow 0$
- ## do
- ## $t \leftarrow t + 1$
 - ## **Paso-E:** $Q(\Theta; \hat{\Theta}^t) \leftarrow \mathbb{E} \left[\sum_{i=1}^{\infty} \ln \left(p(\mathbf{x}_i | j; \hat{\Theta}_j^t) p_j^t \right) \right]$
 - ## **Paso-M:** $\hat{\Theta}^{t+1} \leftarrow \arg \max_{\Theta} Q(\Theta; \hat{\Theta}^t)$
 - ## until $|Q(\Theta; \hat{\Theta}^t) - Q(\Theta; \hat{\Theta}^{t+1})| < \epsilon$

Como funciona el GMM?

- ❖ Comenzamos con una partición aleatoria de la cual se sacan los parámetros de inicio y desde allí se itera

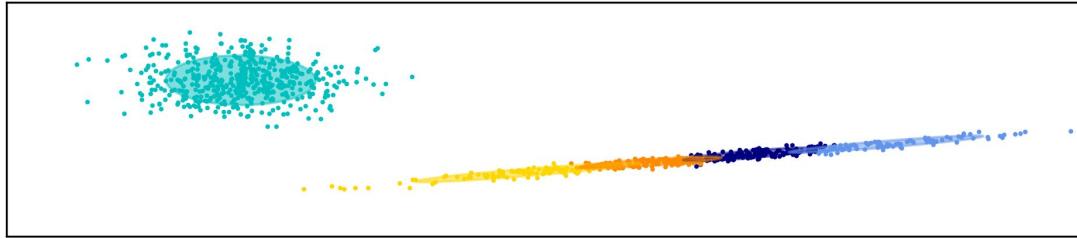


Parámetros

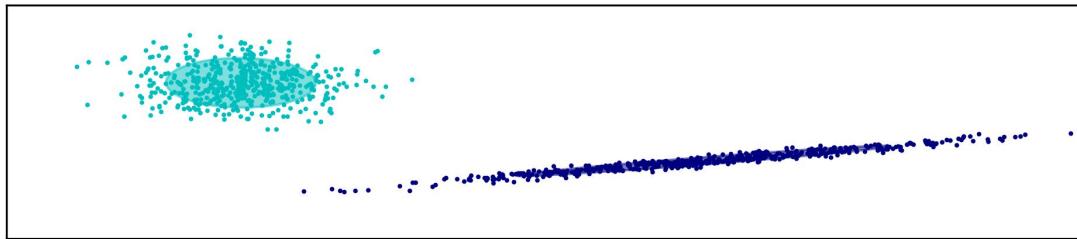
- ❖ Gran problema de GMM es la determinación del número de componentes de la mezcla
- ❖ Si no se elige un buen número, el modelo parte de forma aglutinada pero los clusters pueden no tener sentido.
- ❖ La otra característica que puede ser forzada de inicio es el tipo de matriz de varianza covarianza.
- ❖ Este ejemplo (Note_fig2.ipynb) ha sido realizado modelando matrices de covarianza full usando el módulo sklearn.

Note_fig2.ipynb

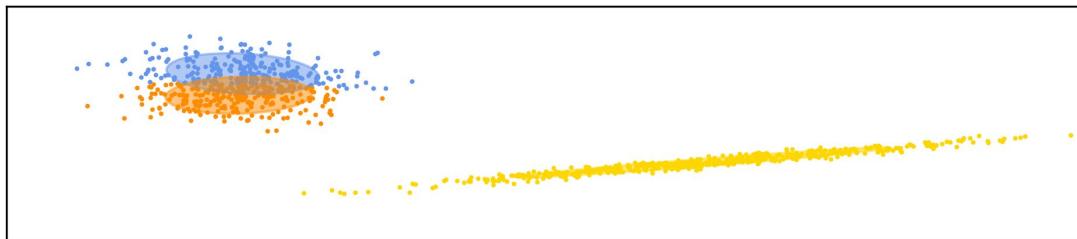
Gaussian Mixture K=5



Gaussian Mixture=2



Gaussian Mixture=3



Detección automática de k

- ❖ Bayesian Information Criterium (BIC) da un score al modelo con m parámetros.

$$BIC = -2 \cdot \log(L(\Theta)) + \log(n)m$$

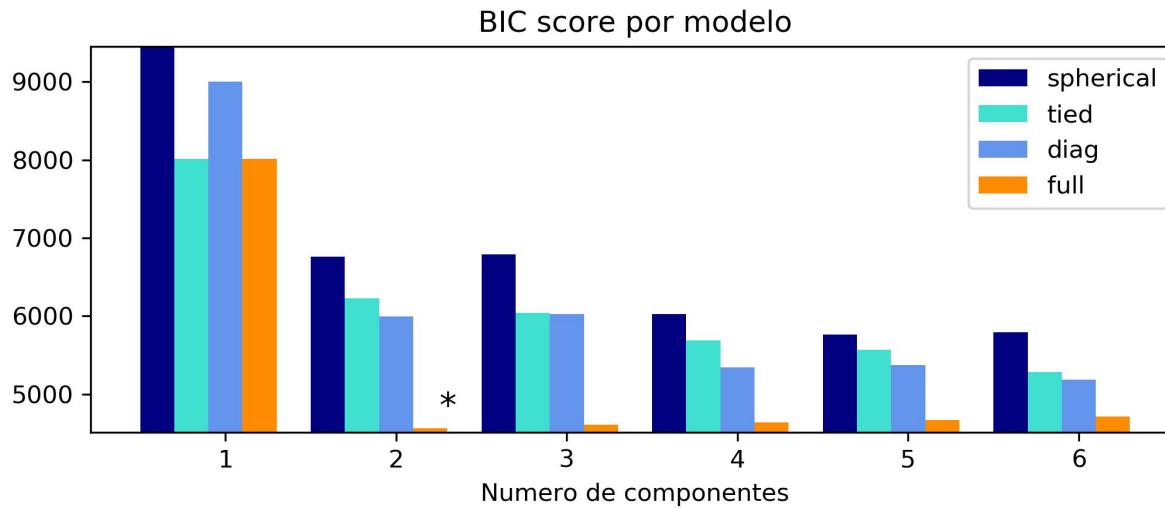
- ❖ Puede usarse otro índice llamado Akaike Information Criterium (AIC)

$$AIC = -2 \cdot \log(L(\Theta)) + 2m$$

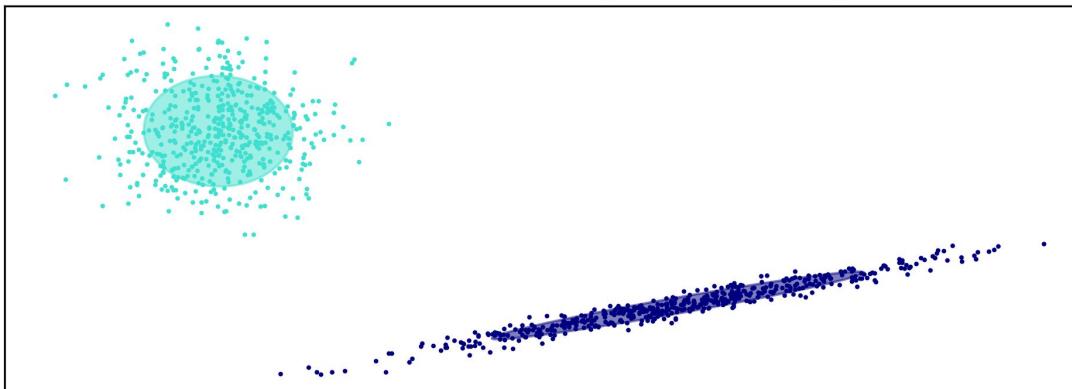
donde $L(\Theta)$ es la verosimilitud, n el número de datos, y m el número de parámetros estimados, (k , el número de componentes, más las medias y entradas de la matriz de varianza covarianza.)

- ❖ La figura siguiente está generada por el script Note_fig3.ipynb

Note_fig3.ipynb



GMM Seleccionado: modelo completo con 2 componentes



Mean Shift Algorithm

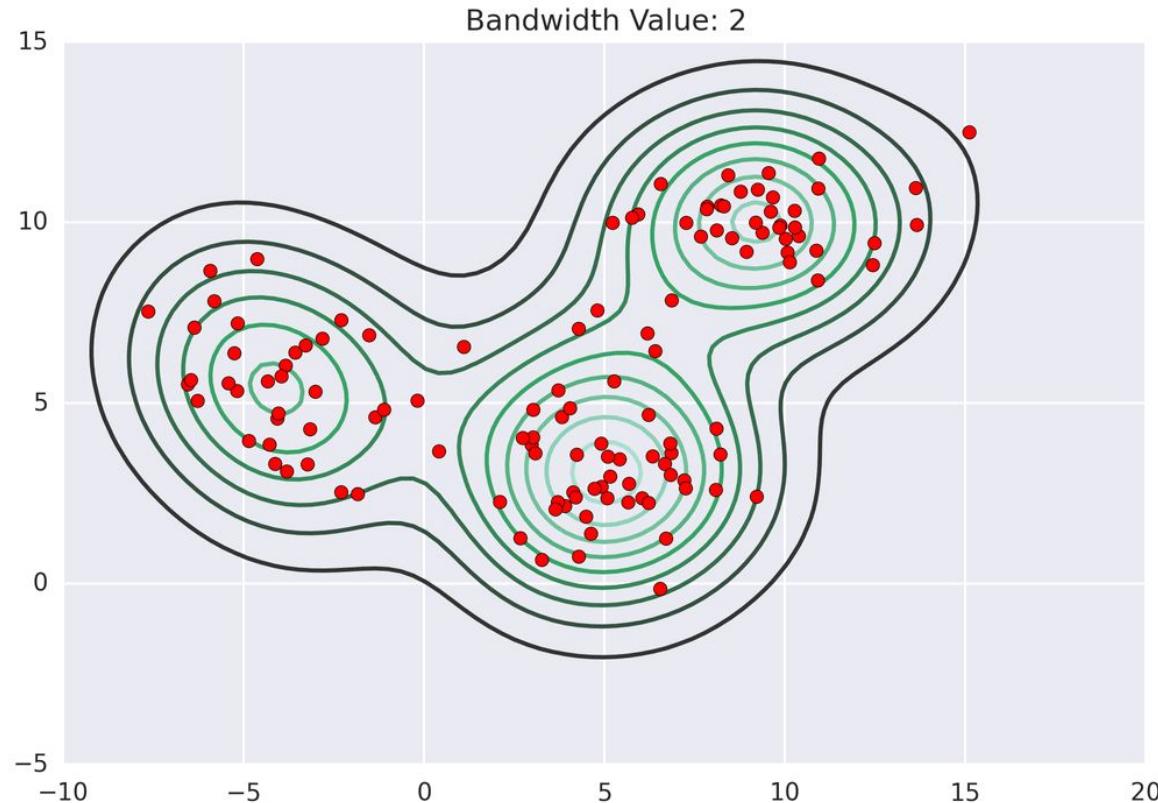
- Mean shift se basa en el concepto de kernel density estimation (KDE)
- Si los datos se suponen muestrados de una distribución de probabilidad, KDE es un estimador no paramétrico de la densidad asociada a dicha distribución.
- KDE aplica un kernel, esto es, una función de peso, en una ventana alrededor del punto con un ancho de banda (bandwidth) determinado. Sumando todos las estimaciones individuales se obtiene el estimador de la densidad.

Mean Shift Algorithm

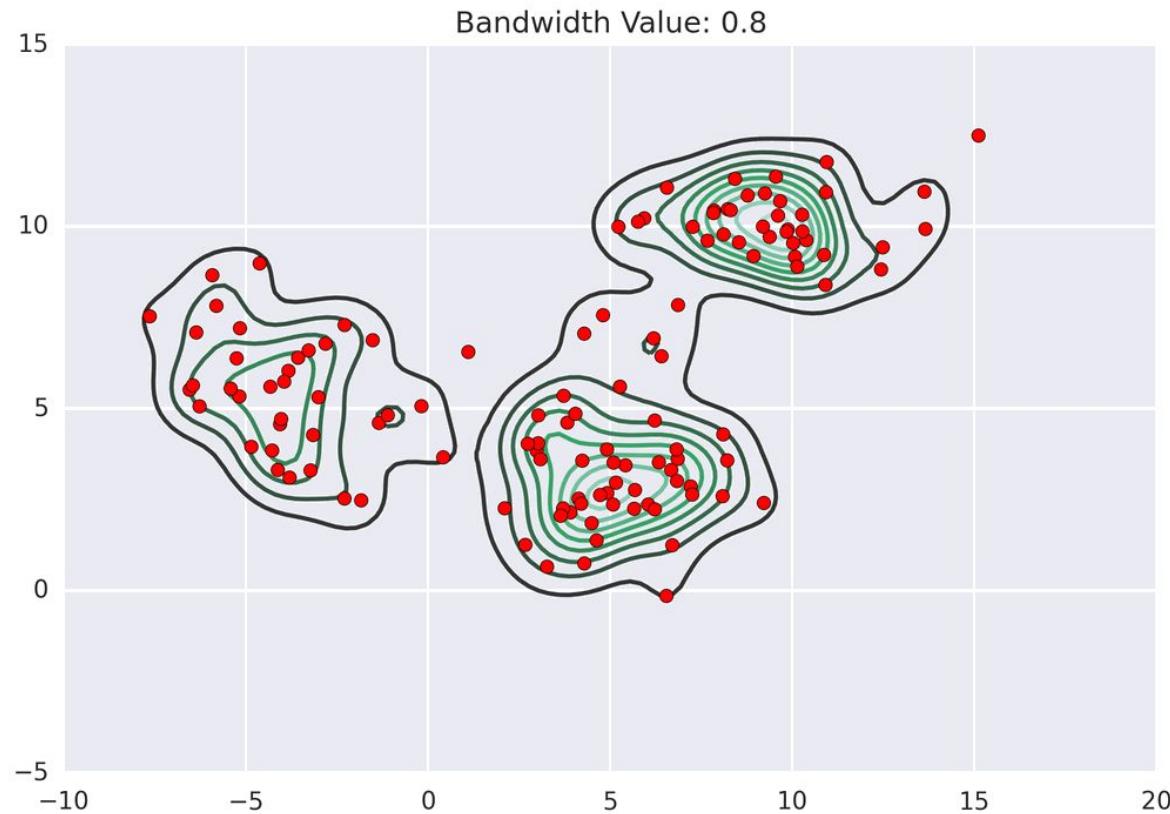
- Para generar la partición, el algoritmo Mean-Shift Clustering va deslizando la ventana y computando el promedio de los datos pesados por el kernel, para localizar las áreas de alta densidad.

- Cada moda de la densidad va a ser considerada un centroide, y los puntos de la partición van a ser asignados al centroide más próximo

Mean Shift Algorithm



Mean Shift Algorithm



Mean Shift Algorithm

- ❖ Parámetro bandwidth puede ser fijado a priori.
- ❖ No tiene sentido usar BIC o AIC pues no se está fijando el modelo paramétrico
- ❖ Puede ser estimado utilizando la teoría no paramétrica, dependiendo de que kernel se use.
- ❖ Note_fig4.ipynb tiene un ejemplo de Mean Shift automático y con k fijo.

Mean Shift Algorithm

Note_fig4.ipynb tiene un ejemplo de Mean Shift automático.

Estimated number of clusters: 3

Homogeneity: 0.941

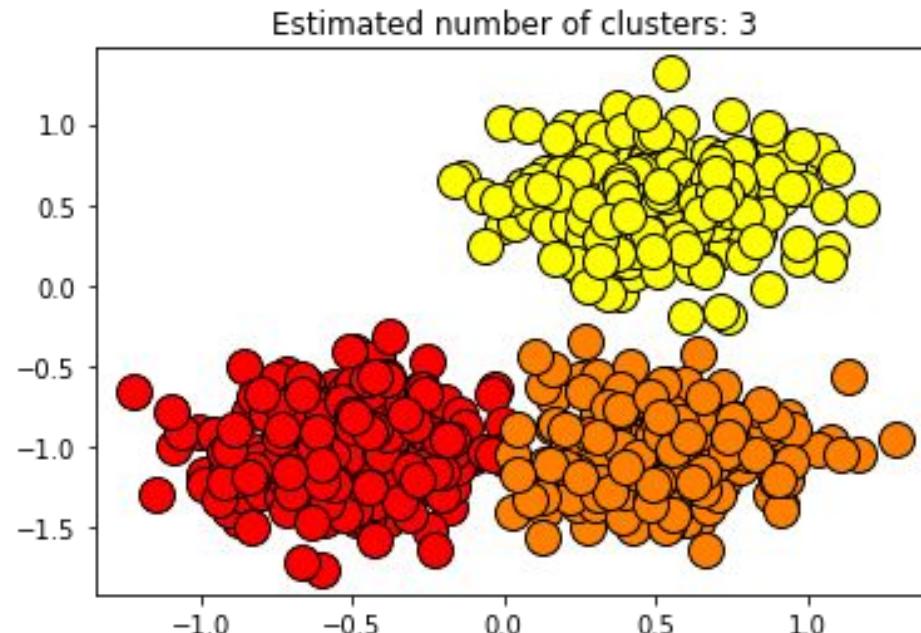
Completeness: 0.941

V-measure: 0.941

Adjusted Rand Index: 0.961

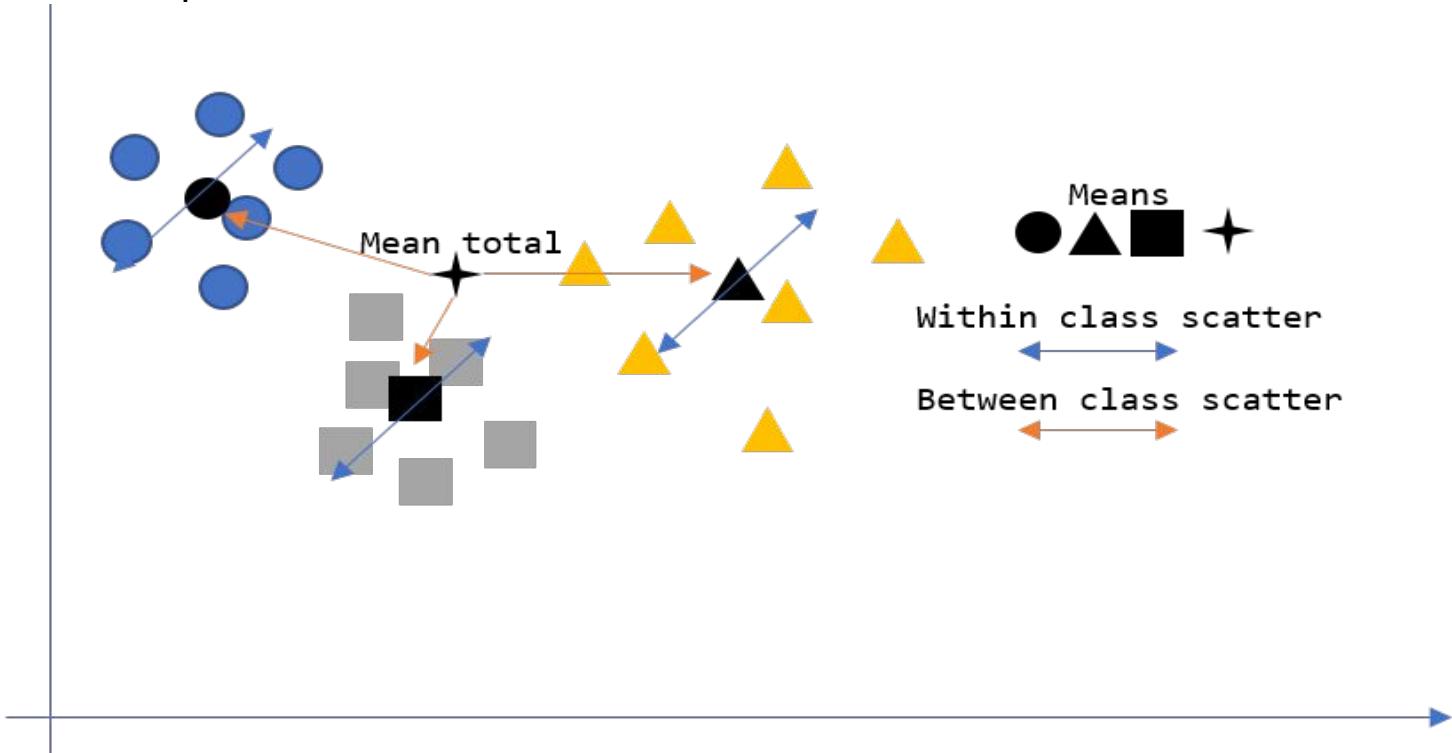
Adjusted Mutual Information: 0.941

Silhouette Coefficient: 0.640



K-means

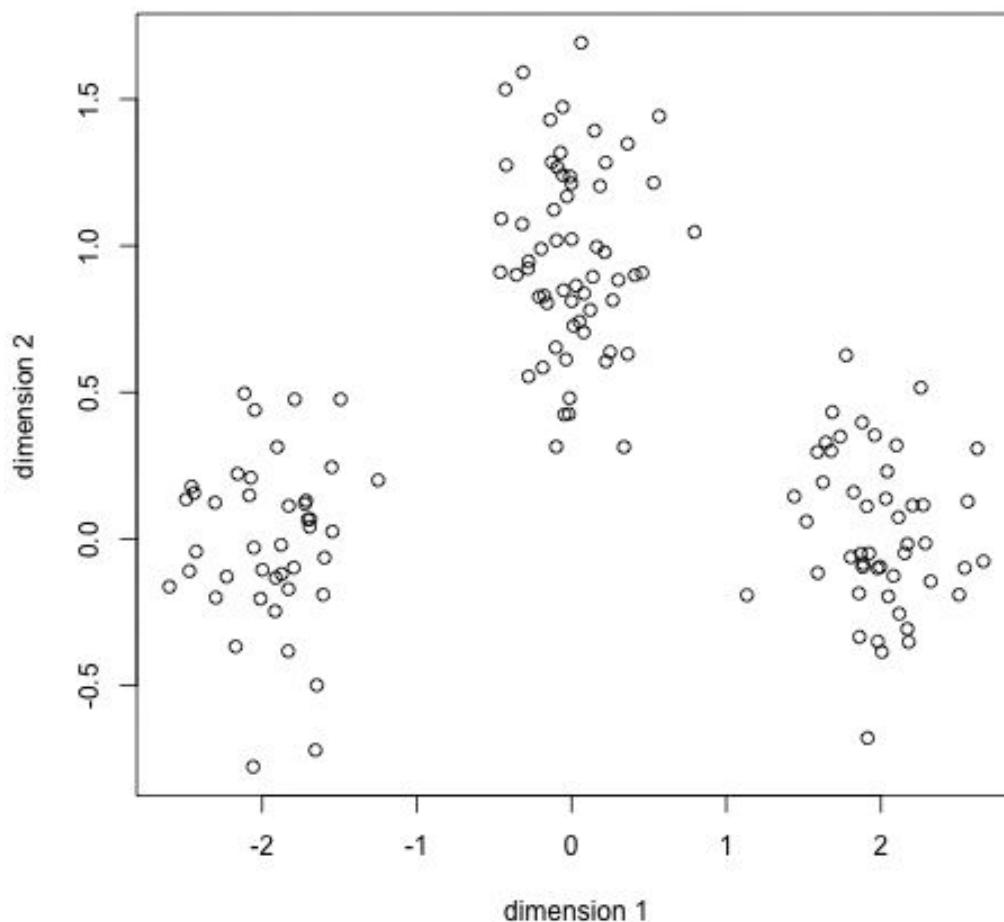
- Pensemos solo en particionar usando distancia, sin pensar en densidades ni distribuciones de probabilidad.



K-means

- ❖ K medias es el algoritmo más usado para aglomerar datos.
 - K medias comienza por elegir k centros aleatorios.
 - Después, todos los puntos son asignados al centro más cercano basado en la distancia euclídea, lo cual genera una partición del espacio.
 - Luego los centros son re calculados usando la nueva partición y el ciclo comienza nuevamente.
 - Este proceso continúa hasta que no haya más cambios en la partición entre iteraciones.
- ❖ Este algoritmo genera una partición similar a la de la mezcla de Gaussianas Esféricas, esto es, con una matriz de varianza Covarianza múltiplo de la Identidad.

step 0



K-means

Problemas

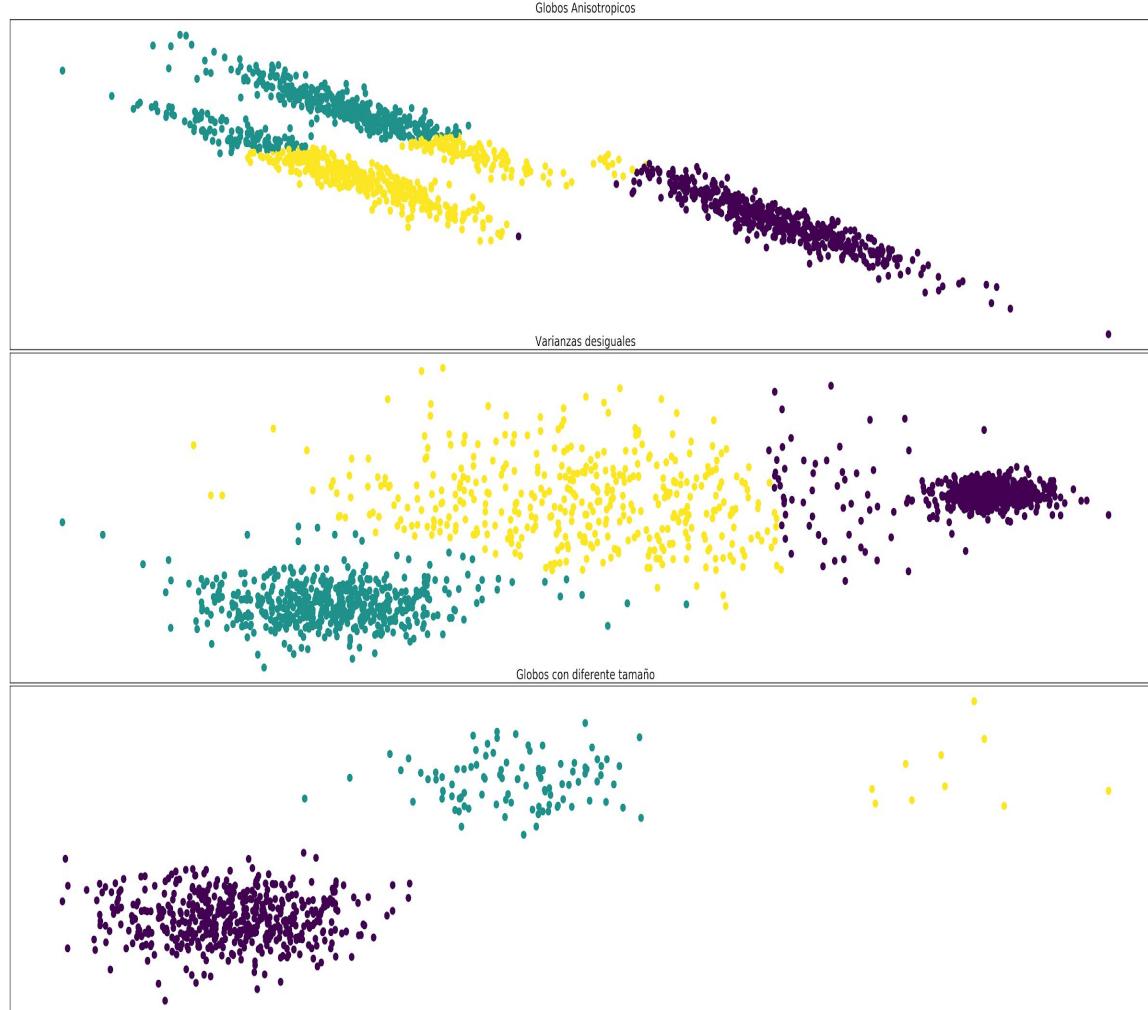
- Inestabilidad
- Mínimos locales (mucha sensibilidad a las semillas)
- Soluciones globales → sensibles a outliers
- El número de clusters k suele ser desconocido, Note_fig5.ipynb

Parámetros

- Inicialización
- número de veces que se vuelven a tirar las semillas
- cuántas iteraciones hasta que termina la búsqueda

K-means

Note_fig5.ipynb



K-means: como definimos k?

- ❖ No podemos usar BIC, o MDL o AIC porque no usamos un modelo de verosimilitud para ajustar.
- ❖ Pero si podemos comparar entre diferentes modelos en función de k el valor de la inercia del modelo, esto es, la suma de distancias cuadradas dentro de cada cluster de la partición final.

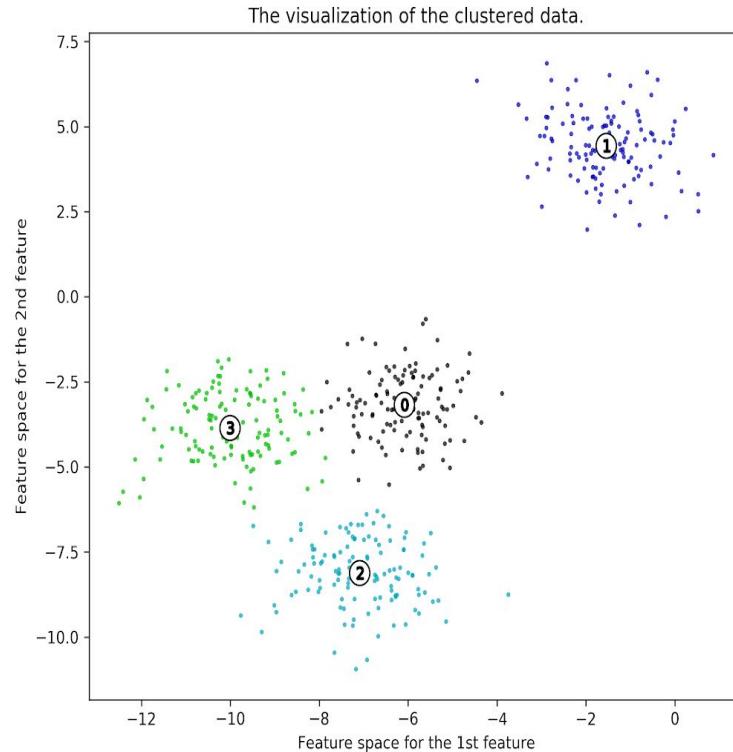
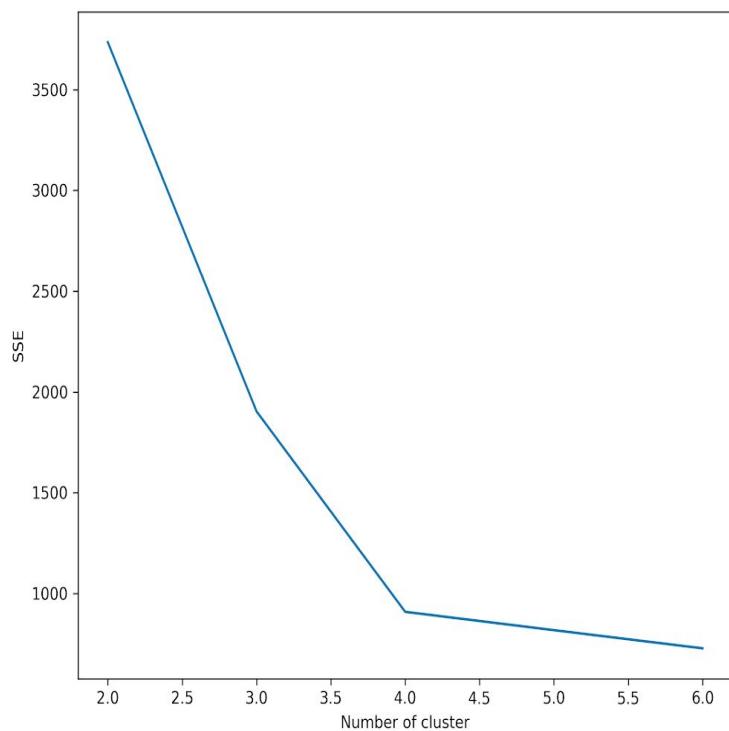
$$Inercia = SSE = \sum_{i=1}^K \sum_{x \in C_i} d^2(m_i, x)$$

K-means: como definimos k?

- ❖ La inercia se considera una medida de cuán coherentes los clusters son,
- ❖ si se hace un gráfico de la inercia en función del k, se considera heurísticamente que el mejor valor se da cuando se desacelera la reducción de la inercia.
- ❖ La note_fig6.ipynb muestra cómo elegir el k mas apropiado con el método del codo.

K-means

Elbow method for KMeans clustering on sample data



K-means: Análisis de siluetas

- ## Para cada punto $i \in C_k$, se crean los indices $a(i)$ de similaridad promedio y $b(i)$ de disimilaridad minima promedio

$$a(i) = \frac{1}{|C_k| - 1} \sum_{j \in C_k, i \neq j} d(i, j) \quad b(i) = \min_{k \neq l} \frac{1}{|C_l|} \sum_{j \in C_l} d(i, j)$$

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

K-means: Análisis de siluetas

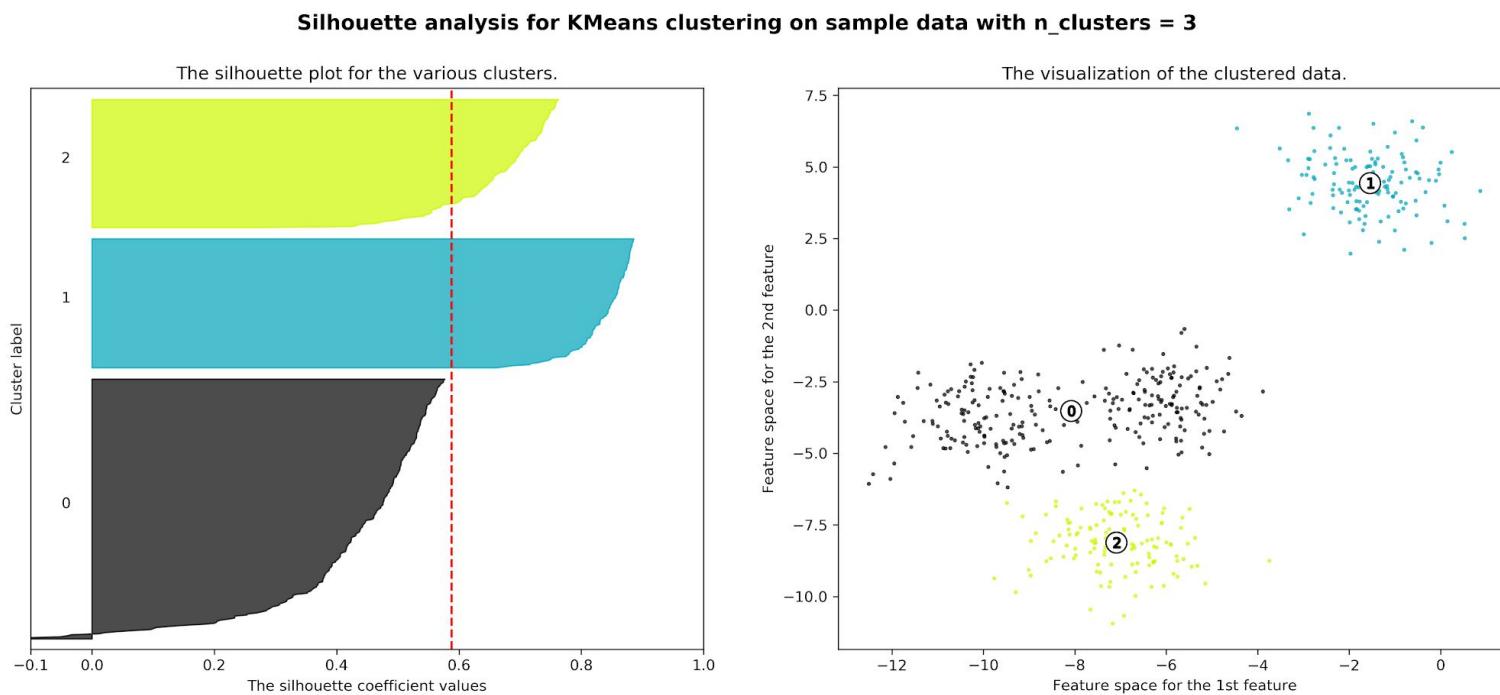
- ❖ La medida $s(i)$ tiene valor entre -1 y 1.
- ❖
- ❖ Los coeficientes $s(i)$ cercanos a +1 indican que la muestra está lejos de los clusters vecinos.
- ❖
- ❖ El valor 0 indica que la muestra está muy cerca del borde de decisión entre los clusters.
- ❖
- ❖ Un valor negativo indica que esos puntos deben haber sido asignados al cluster equivocado.

K-means: Ejemplo

- ❖ Se simula un grupo de datos con cuatro gaussianas.
- ❖ Se calcula el gráfico de silueta para particiones de k medias con K=2,3,4,5,y 6.
- ❖ El gráfico de silueta para los valores de k =3, 5 and 6 muestran que esos k son una mala elección, dado que hay clusters por debajo del valor de silueta promedio y clusters con valores negativos.
- ❖ Note_fig6.ipynb

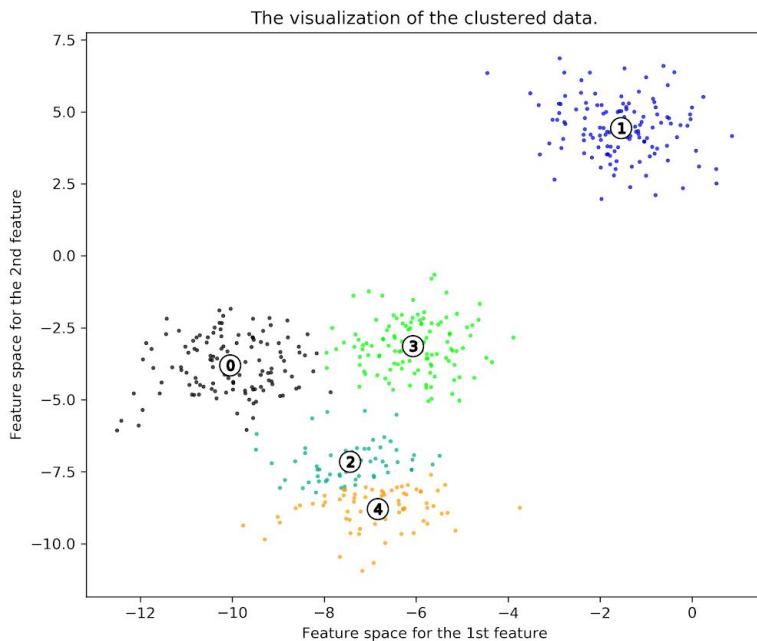
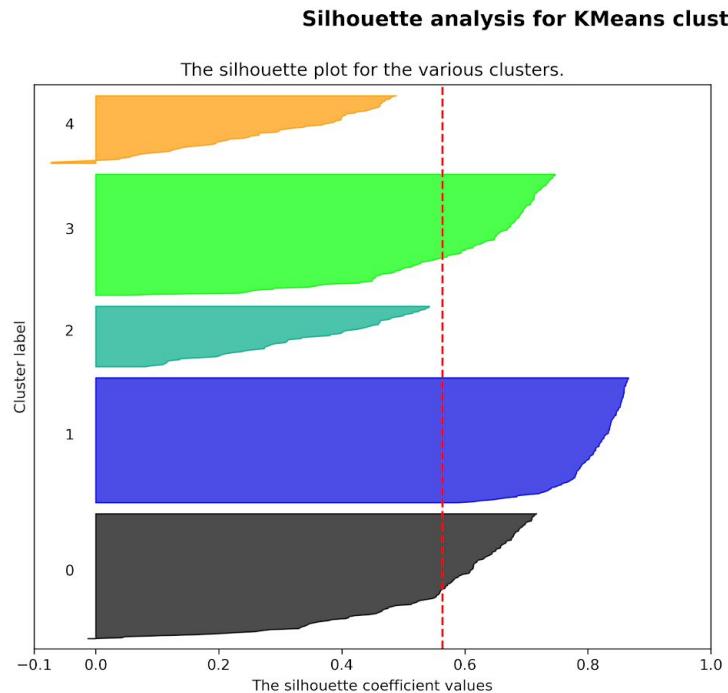
K-means: Ejemplo

Note_fig6.ipynb



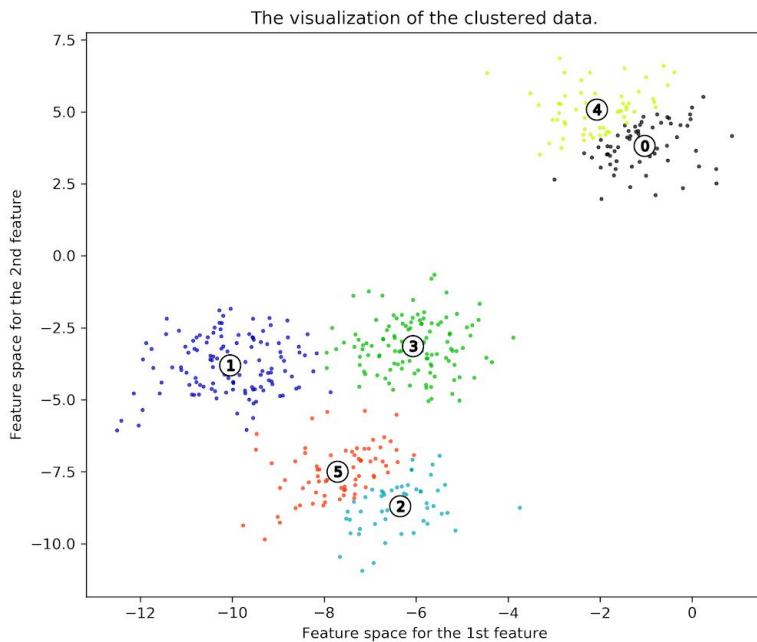
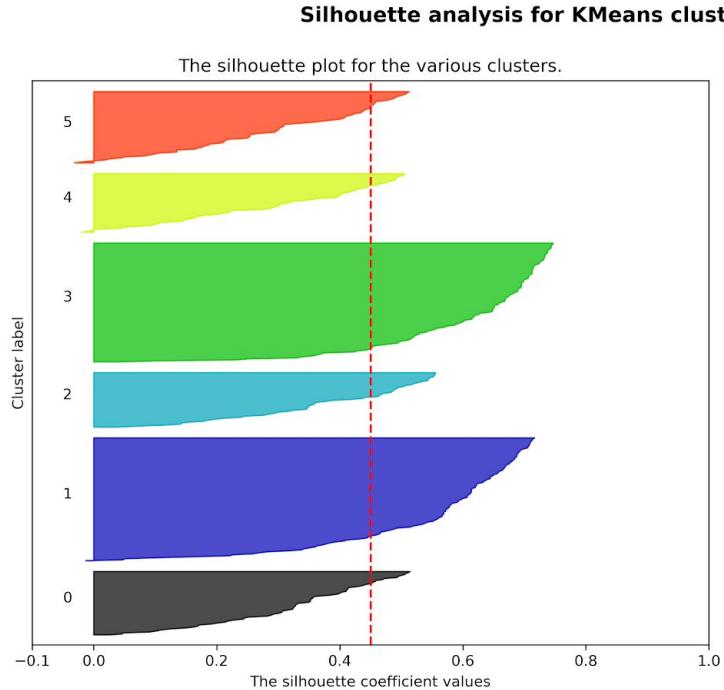
K-means: Ejemplo

Note_fig6.ipynb



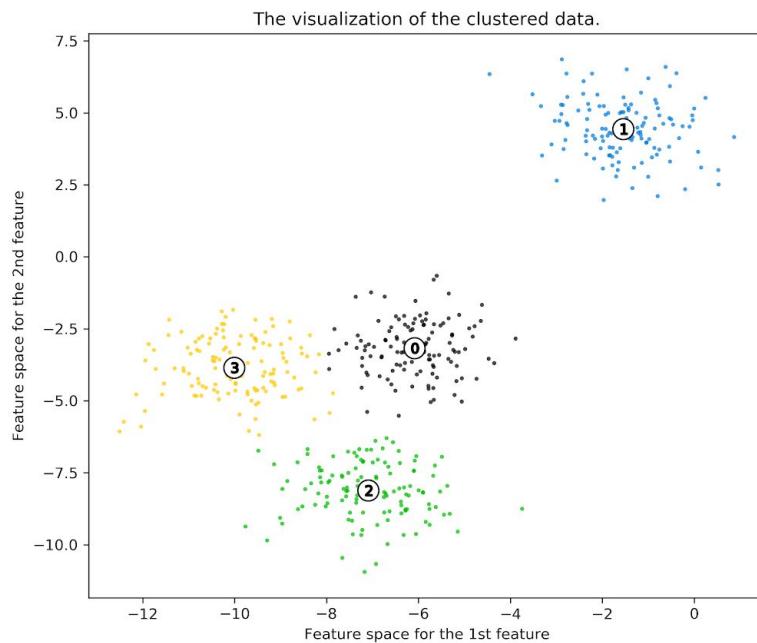
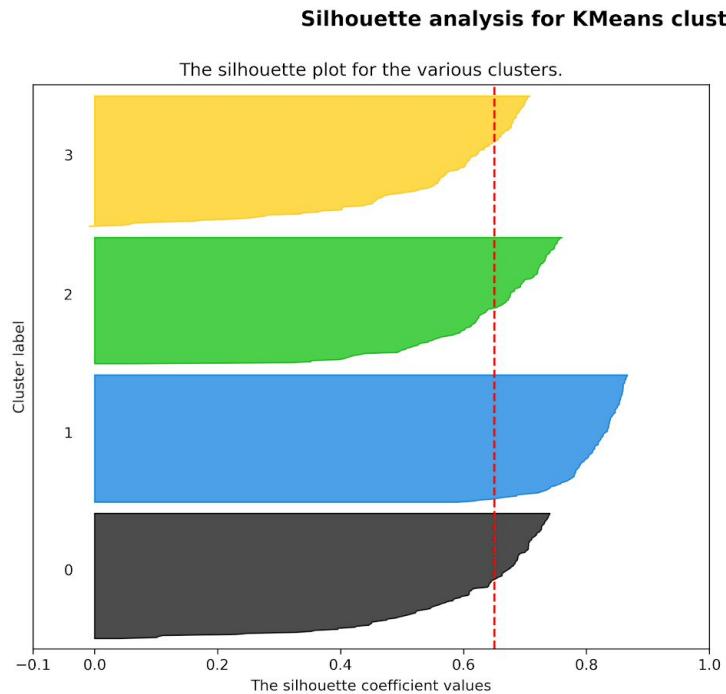
K-means: Ejemplo

Note_fig6.ipynb



K-means: Ejemplo

Note_fig6.ipynb



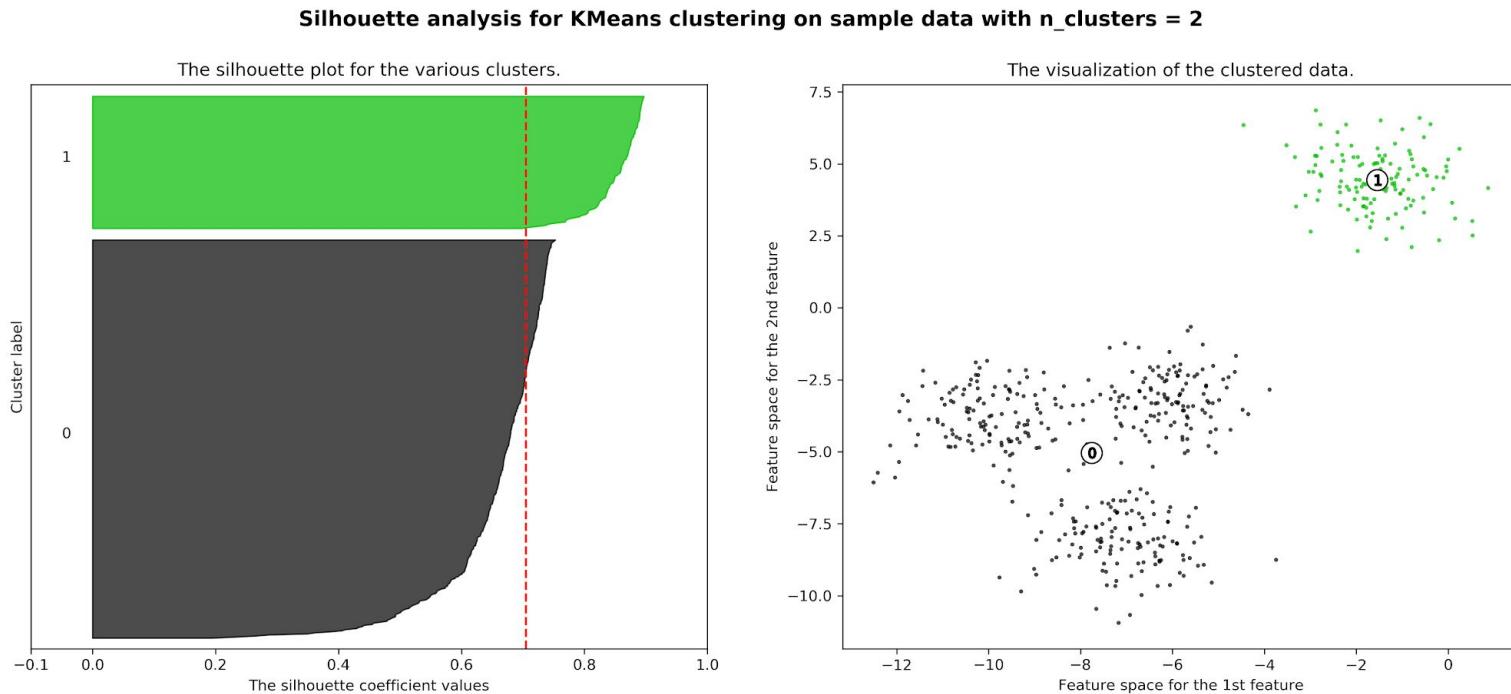
Silhouette analysis for KMeans clustering on sample data with `n_clusters = 4`

The silhouette plot for the various clusters.

The visualization of the clustered data.

K-means: Ejemplo

Note_fig6.ipynb



K-means: Ejemplo

- ❖ El problema de coeficientes negativos no ocurren el el caso de k=2 y 4
- ❖ Si se estudia el grosor de gráfico de silueta se ve que k=2 produce una partición muy desbalanceada, dado que uno de los clusters absorbe tres clusters diferentes.
- ❖ Cuando k=4 las siluetas están balanceadas, por lo cual este es el mejor k

K-means: sklearn y yellowbrick

