Subject: Probability and Statistics

Class XIX: Clustering: K-means

UdeA

Bioengineering

Francisco José Campuzano Cardona

Bioengineerer, MSc in Engineering

Ahora analizaremos algunos algoritmos de aprendizaje automático no supervisado. En este tipo de métodos, no se hace uso de etiquetas en ningún momento. Se dice que se permite a los datos hablar por sí mismos.

Los primeros métodos que veremos serán los de Agrupamiento. En este caso, los métodos buscan aprender de las características de los datos una división o agrupamiento apropiado de los datos. La asignación de una clase a cada grupo, dependerá del contexto y la experticia del analista.

K-means Clustering

UdeA

El algoritmo más conocido e intuitivo de los métodos de agrupamiento es el agrupamiento k-means.

Este algoritmo busca un numero predeterminado de grupos en un set de datos multidimensional sin etiquetas. En el caso ideal, cuando se alcanzó un agrupamiento apropiado se cumple:

- El centro del *cluster*, es la media aritmética de todos los miembros del *cluster*
- Cada punto de un *cluster* está más cerca a su centro que a cualquier otro centro.

K-means Clustering

UdeA

El algoritmo más conocido e intuitivo de los métodos de agrupamiento es el agrupamiento k-means.

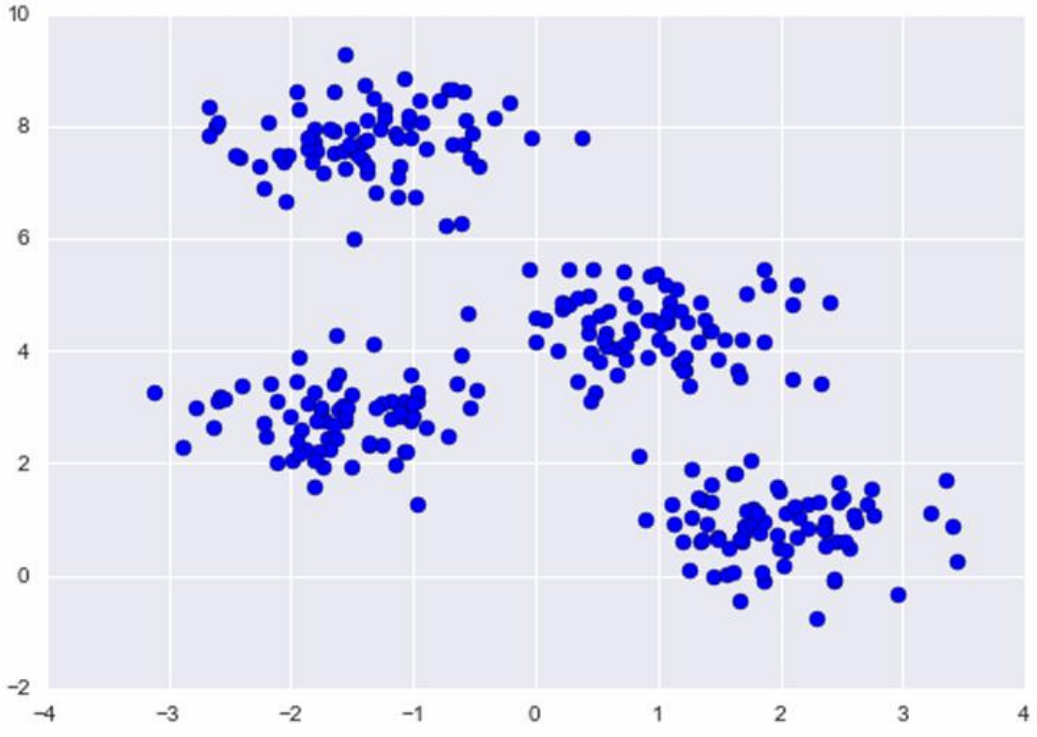
Este algoritmo busca un numero predeterminado de grupos en un set de datos multidimensional sin etiquetas. En el caso ideal, cuando se alcanzó un agrupamiento apropiado se cumple:

- El centro del *cluster*, es la media aritmética de todos los miembros del *cluster*
- Cada punto de un *cluster* está más cerca a su centro que a cualquier otro centro.

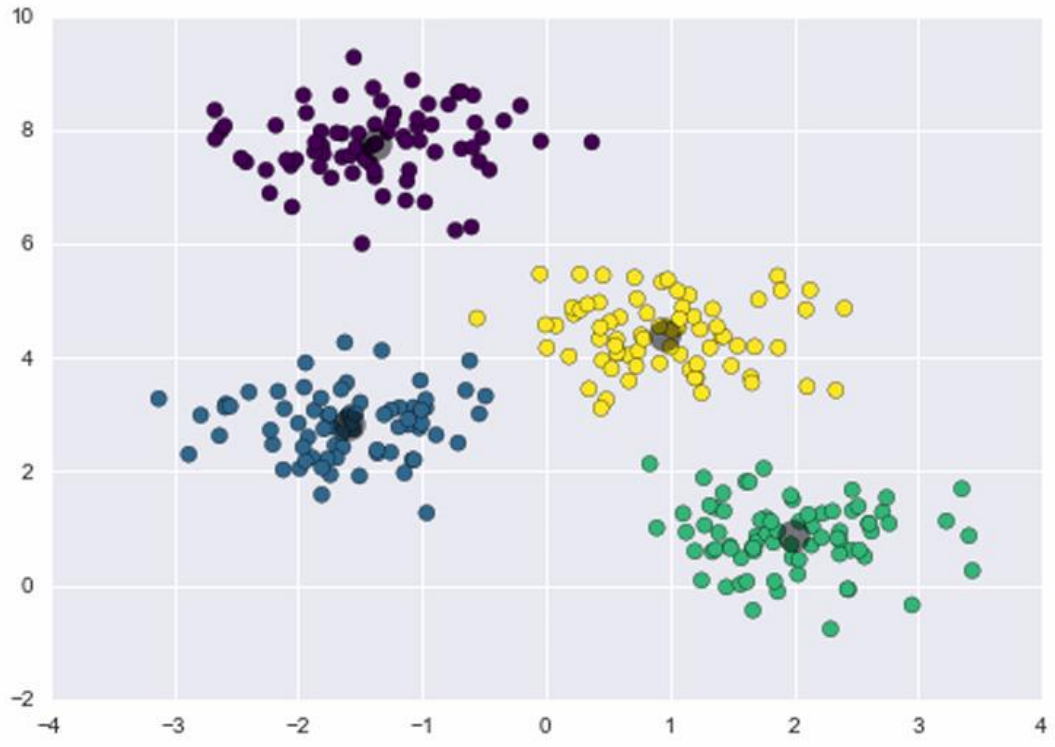
K-means Clustering

UdeA

Datos sin etiquetas



Datos agrupados



K-means Clustering

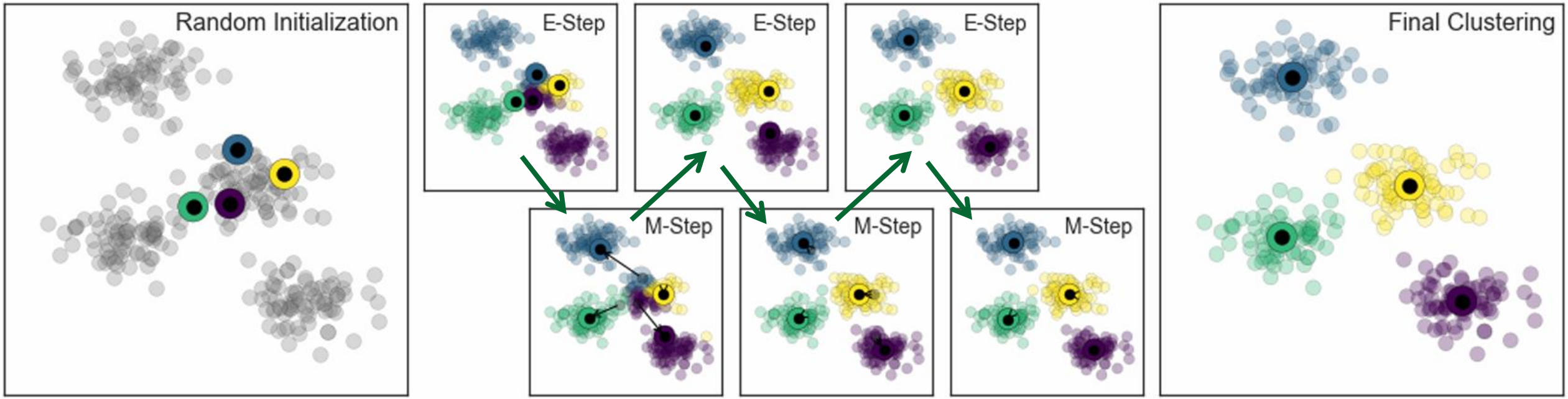
UdeA

Pero, ¿cómo funciona el algoritmo?

1. Se asumen unos centros de *cluster* aleatorios
2. Se asigna un *cluster* a cada dato en función de la cercanía a los centros definidos.
3. Se recalculan los centros calculando la media de los datos que quedaron en cada *cluster.*
4. Como los centros se movieron, entonces se reasignan los puntos a los centros, según su cercanía.
5. Nuevamente, se recalculan los centros
6. Nuevamente, los centros cambiaron, entonces se reasignan los datos a los centros …Y así sucesivamente.

K-means Clustering

UdeA

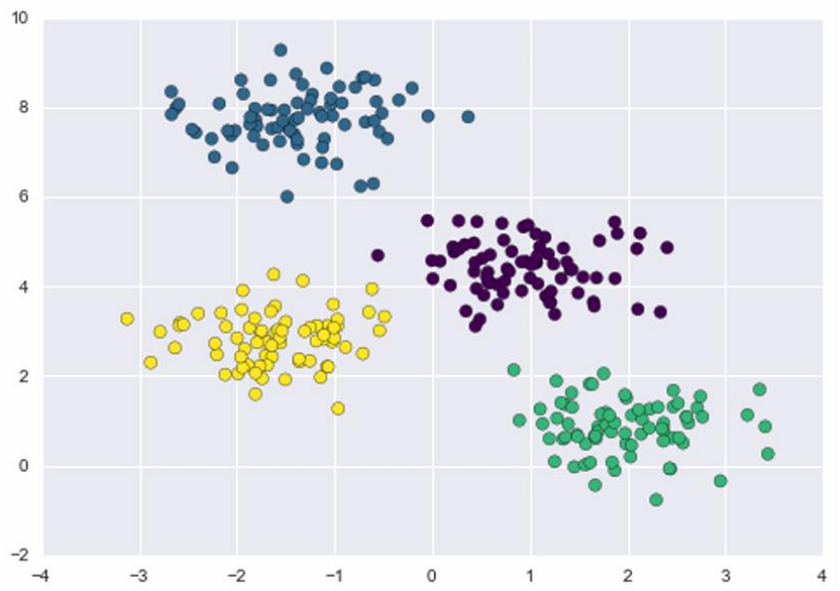


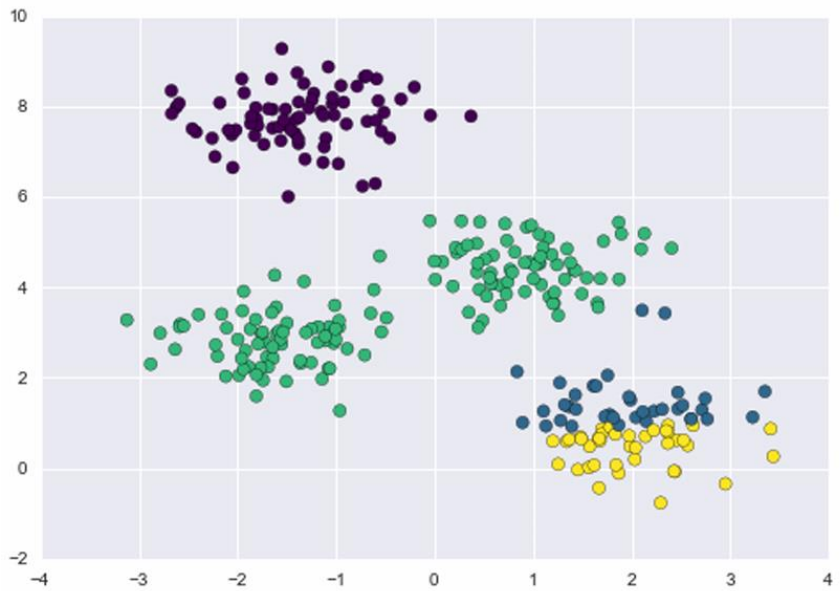
Y en cada iteración, el agrupamiento mejora. Los grupos cada vez están mejor definidos.

K-means Clustering

UdeA

Sin embargo, si bien este algoritmo asegura que cada vez mejora el agrupamiento, no asegura que el agrupamiento sea el óptimo, por ejemplo partiendo de semillas aleatorias diferentes se puede obtener agrupamientos diferentes. Por esta razón, los algoritmos suelen correrse múltiples veces con semillas diferentes y se promedian.

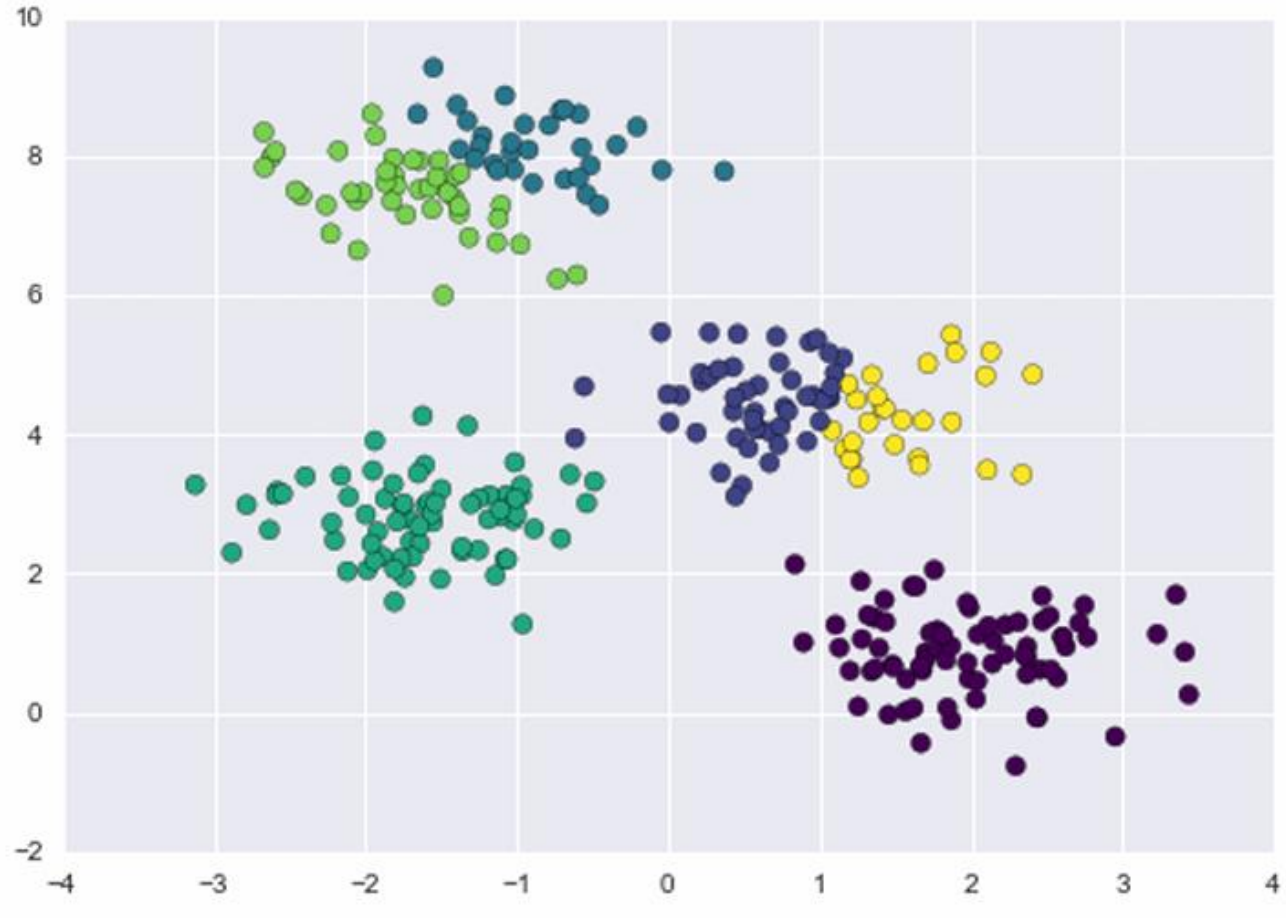




UdeA

El número de *clusters* K

Se debe informar al algoritmo cuantos *clusters* se espera tener, pues esto no puede aprenderlo el algoritmo desde los datos, por ejemplo con los datos anteriores, si se le pide encontrar 6 grupos, el algoritmo lo hará gustosamente… ahora si el resultado es significativo, no es una cuestión simple.



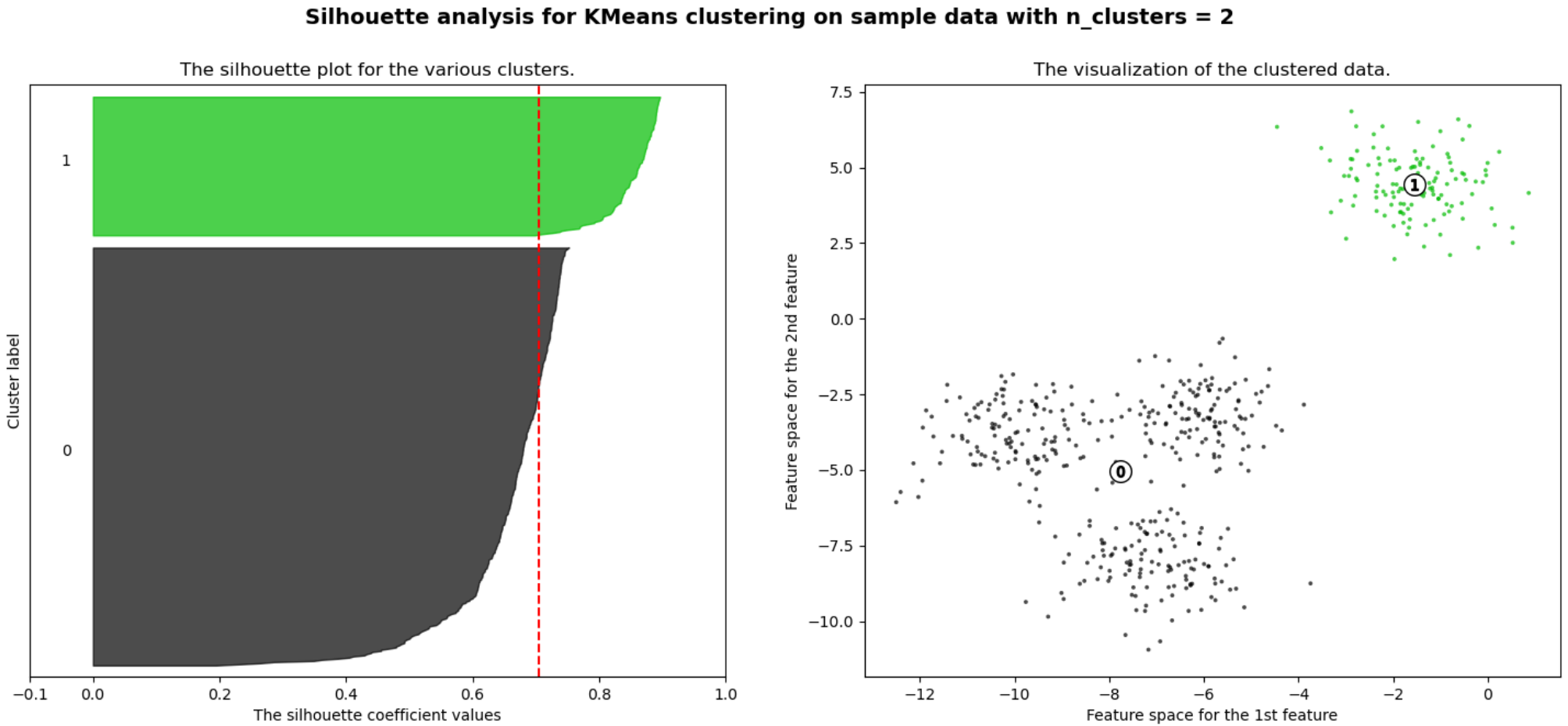
Silhouette Analysis

La selección del K puede ser guiado por este tipo de análisis, en el que se genera un gráfico que permite conocer qué tan cerca están los puntos de un *cluster,* de los puntos vecinos. Es una métrica entre [-1,1], donde valores positivos hablan de separación entre los grupos, 0 indica mucha cercanía, y valores negativos indican que hay puntos que no pertenecen a ese grupo.

Este análisis guía, pero no es suficiente, uno debe tener alguna idea de sus datos de cualquier modo.

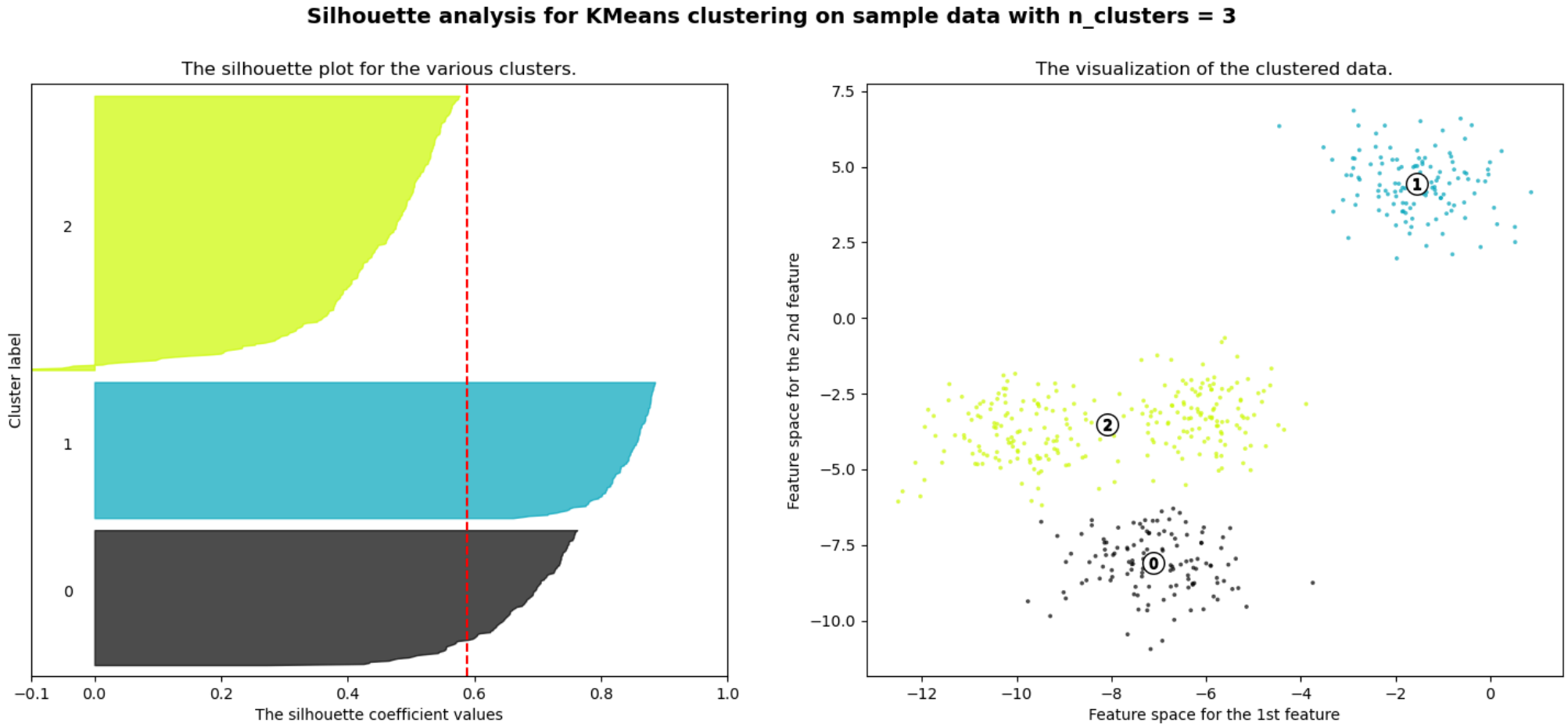
UdeA

Silhouette Analysis



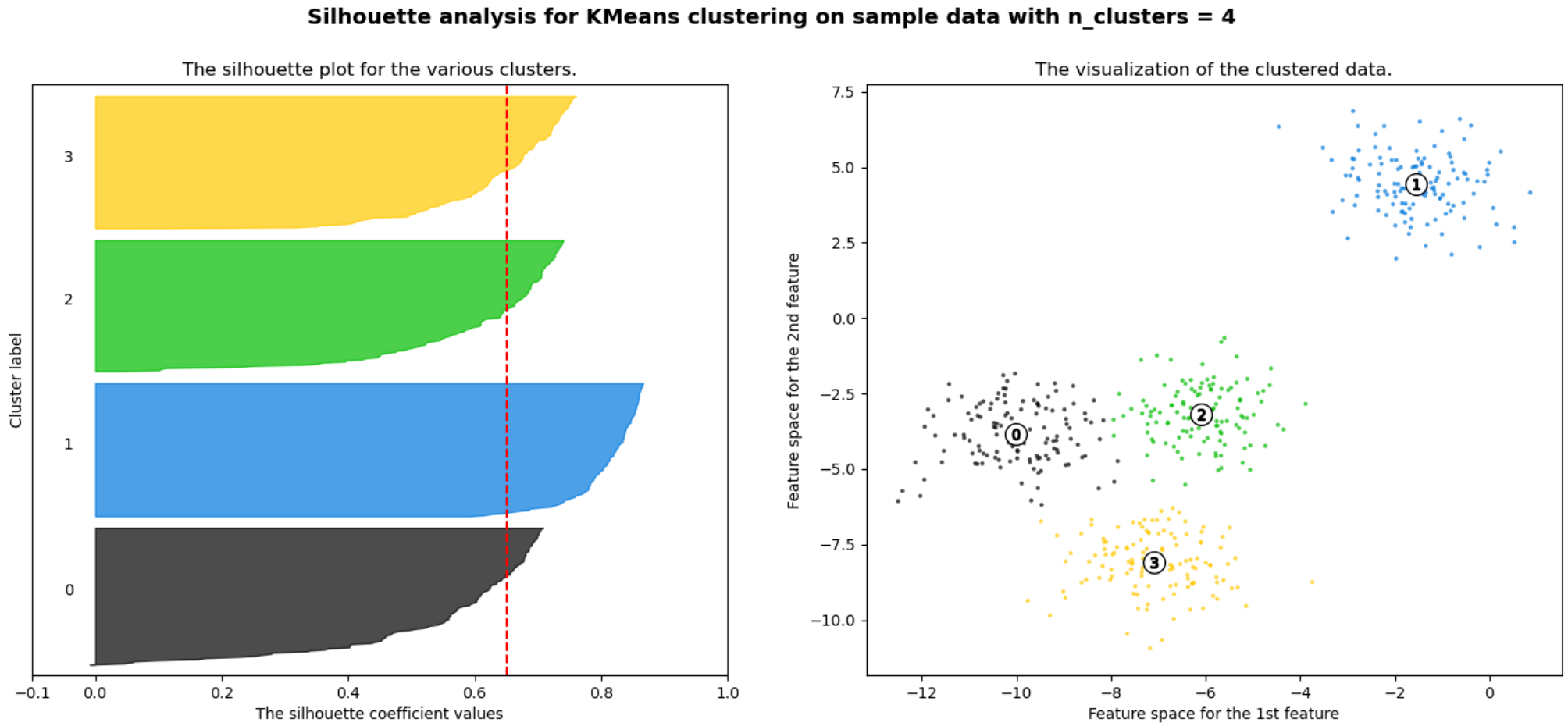
UdeA

Silhouette Analysis



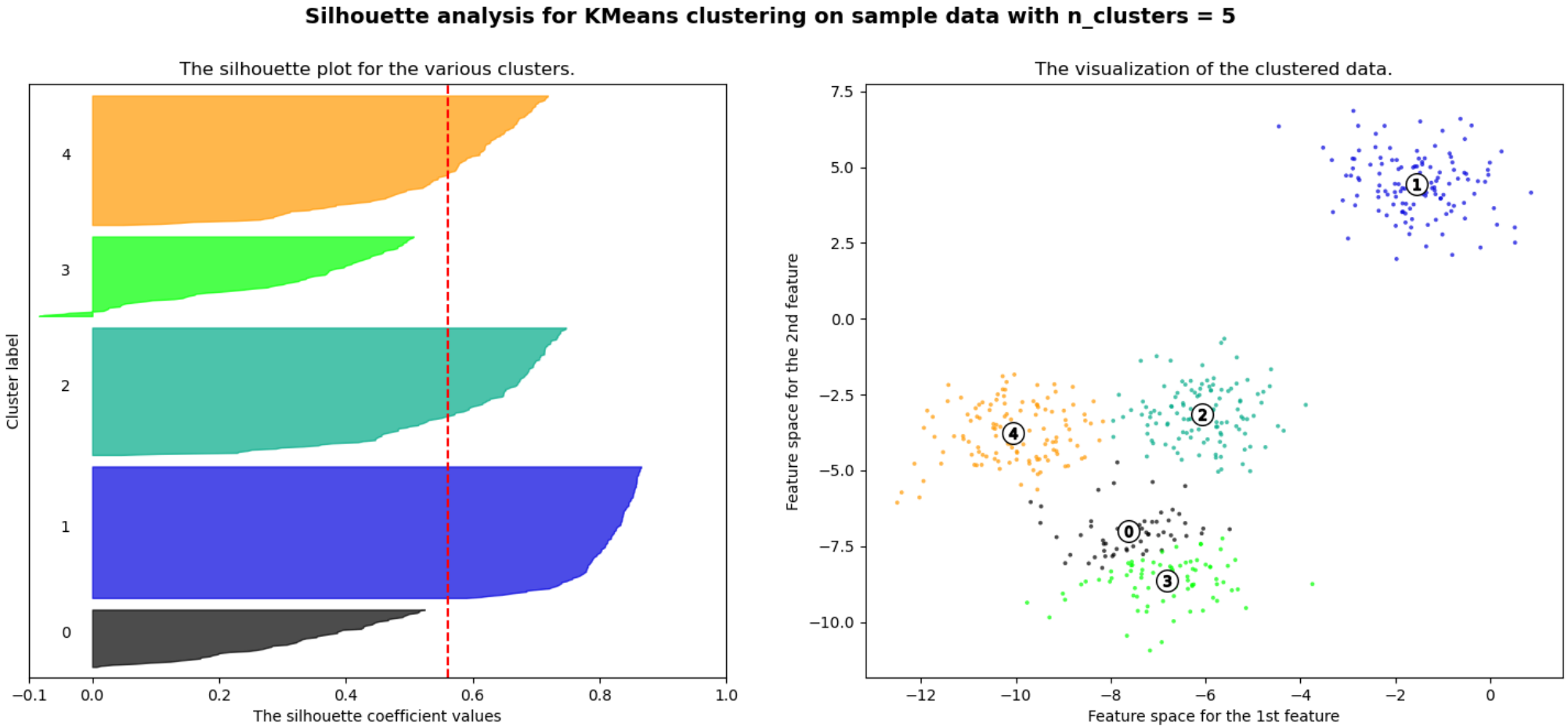
UdeA

Silhouette Analysis



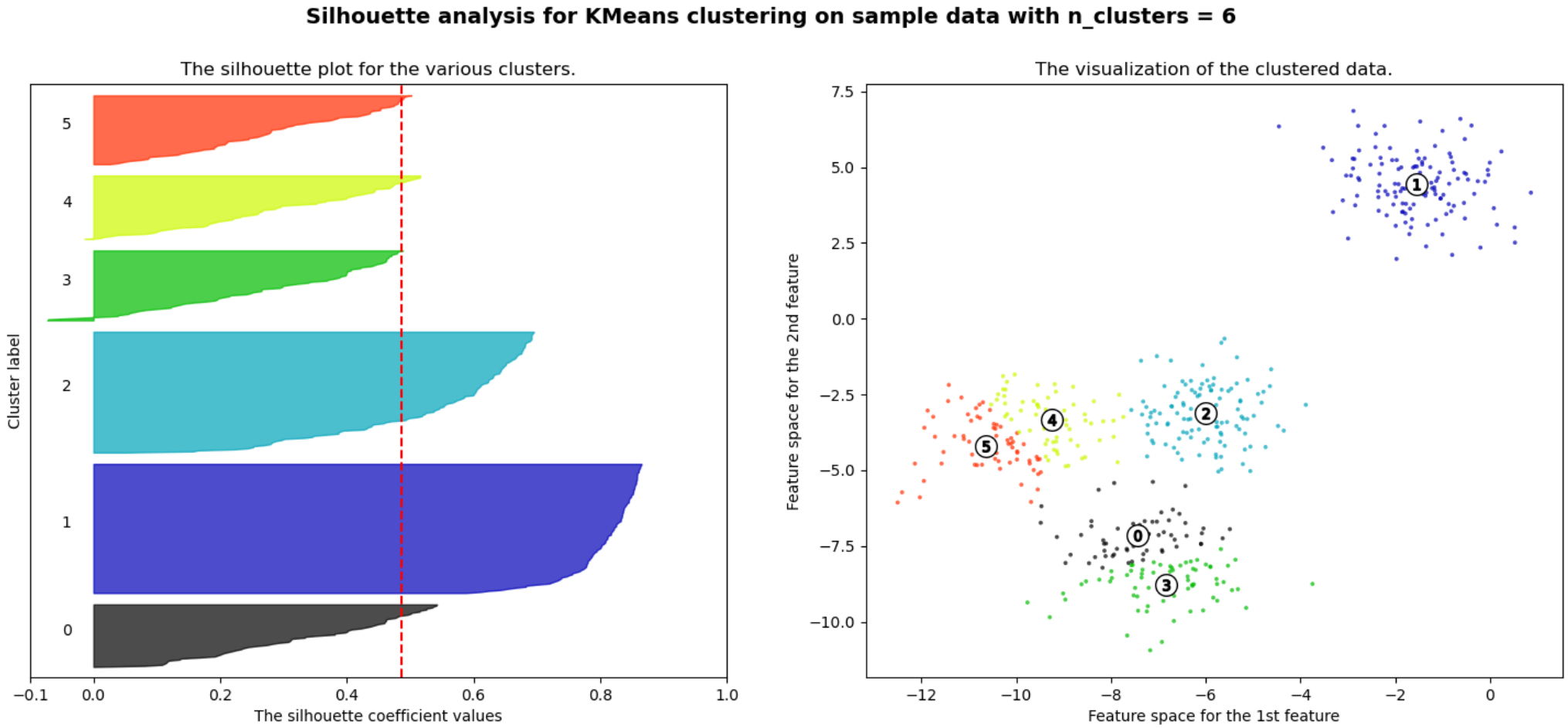
UdeA

Silhouette Analysis



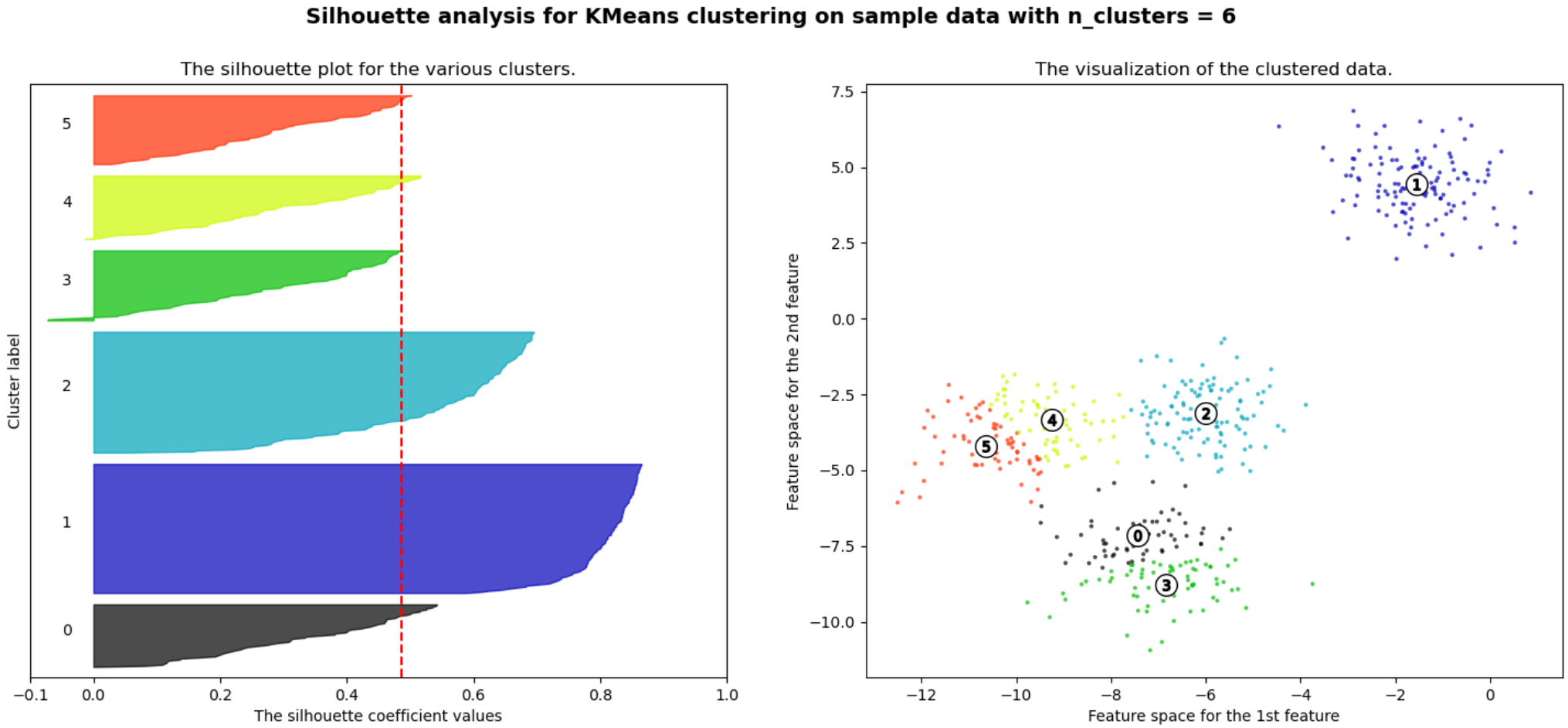
UdeA

Silhouette Analysis



UdeA

Silhouette Analysis



UdeA

En ocasiones es difícil abstraer las aplicaciones que tienen los diferentes algoritmos. En este caso seguramente solo pensamos en aplicaciones donde directamente queremos encontrar grupos para generar alguna clasificación, pero y si les digo, que queremos comprimir una imagen, es decir, que sea menos pesada, usando agrupamiento k-means ¿cómo creen que podría hacerse?

Entonces veamos:

Una imagen a color RGB está constituida por miles de pixeles, y cada pixel tiene 3 componentes, rojo, verde, y azul, y la superposiciones de estos tres en diferentes intensidades permiten millones de colores diferentes.

Cada color, RGB, tiene 256 posibles valores.

Entonces mediante la superposición de los 3 colores, tenemos 256*256*256 posibles colores, lo que es más de 16 millones de colores.

K-means para compresión de imágenes

UdeA

Por ejemplo, esta imagen que está en RGB, tiene entonces esos 16 millones de colores posible, pero posiblemente no requiere de todos para representarla apropiadamente.

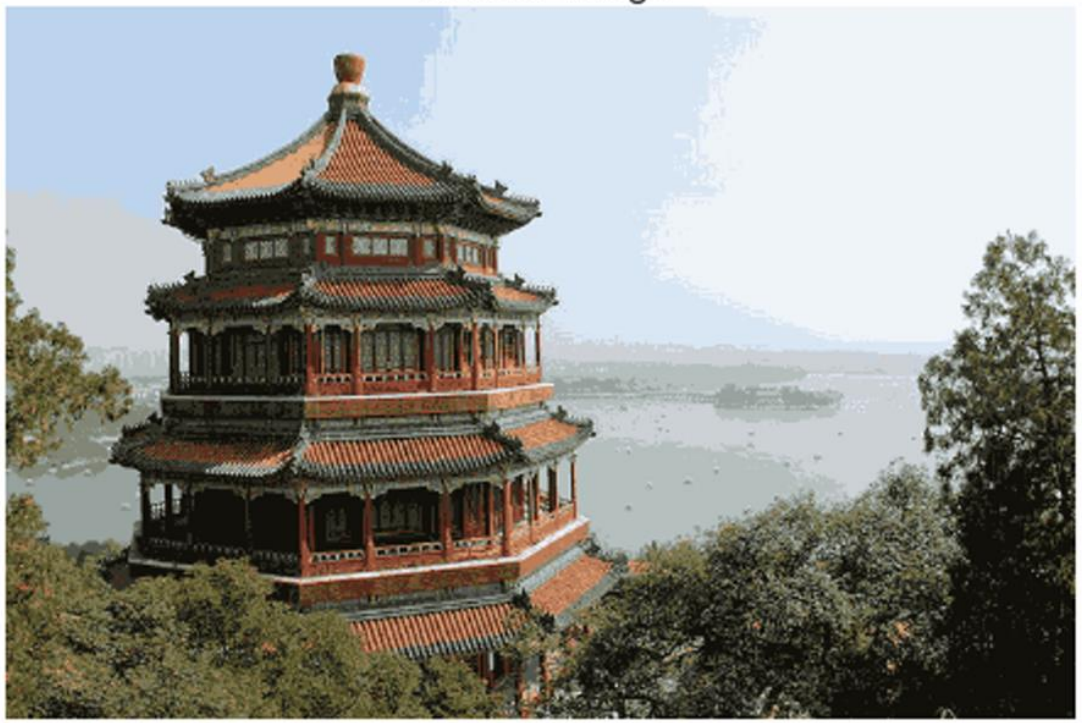
K-means para compresión de imágenes

UdeA

Entonces lo que se puede hacer es agrupar los colores de la imagen, en una cantidad menor de colores, y reconstruir la imagen con menos colores, lo que necesita menos bits, lo que causa que en consecuencia la imagen pese menos.

Original Image

16-color Image



K-means para segmentación de imágenes

UdeA

En el mismo sentido del procesamiento de imágenes tenemos lo que se conoce como segmentación, qué básicamente es sacar de una imagen un elemento específico.

Por ejemplo quitar el fondo de una imagen es una actividad de segmentación, o aislar los huesos del resto de la imagen en una radiografía también lo es.

Y esto puede hacerse usando k-means.

K-means para segmentación de imágenes

UdeA

Por ejemplo esta es una segmentación de una radiografía con k-means.

K-means puede ser lento

UdeA

Dado que este algoritmo es iterativo, y debe evaluar la distancia a todos los centros de cada punto del set de datos, puede volverse muy lento cuando los datos son muchos.

Para mejorar esto, puede actualizarse los centros en cada iteración usando solo un subset de datos, no todos. Lo cual es de utilidad en aplicaciones como las que vimos donde hay millones de datos.

Esto se implementa usando sklearn.cluster.MiniBatchKMeans()

Subject: Probability and Statistics

UdeA

Bioengineering

¡Thanks!

Francisco José Campuzano Cardona

Bioengineering. MSc in Engineering