

TICS-411 Minería de Datos

Clase 6: Evaluación de Clusters

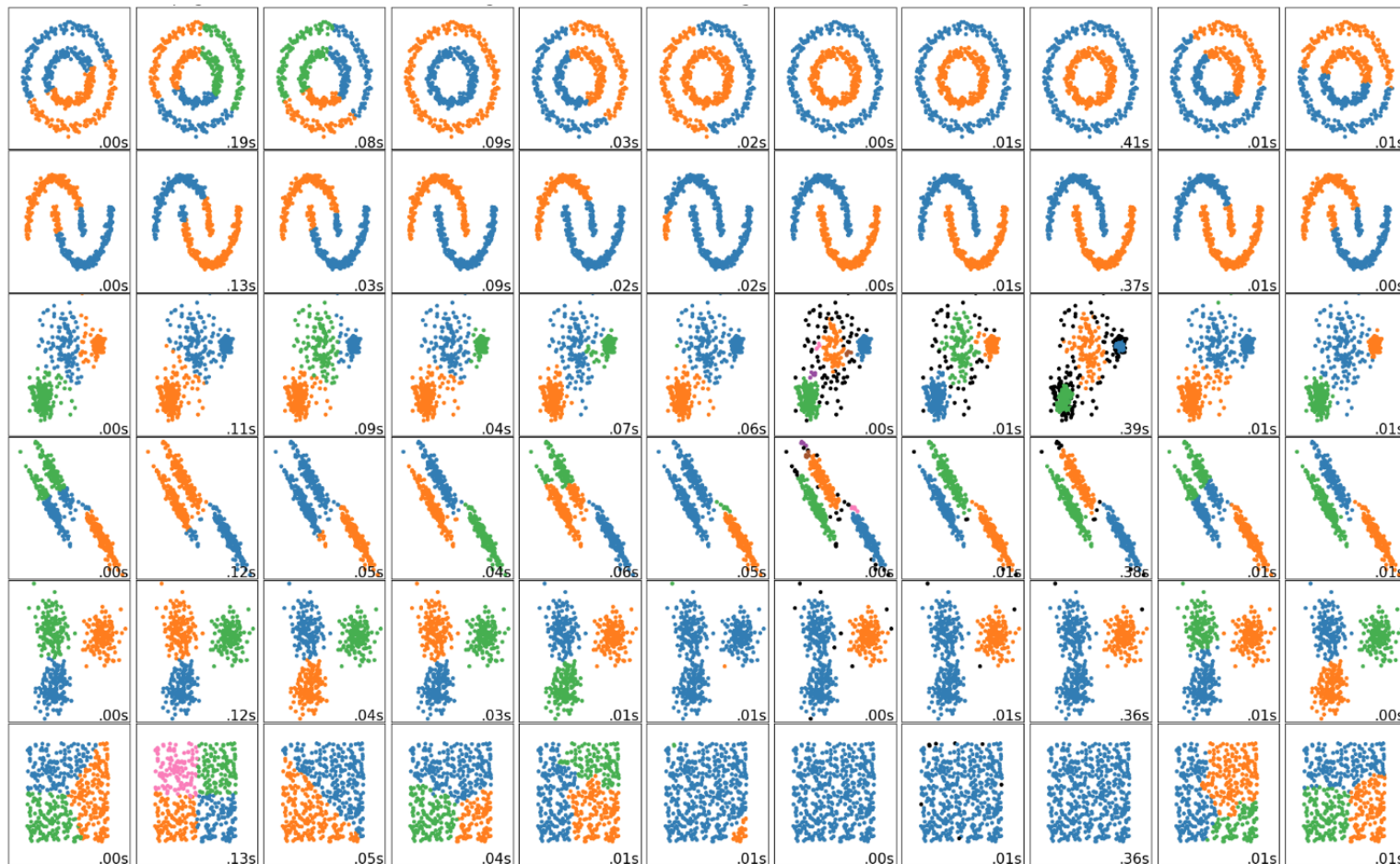
Alfonso Tobar-Arancibia

alfonso.tobar.a@edu.uai.cl

Evaluación de Clusters

Evaluación

Pensemos en la Evaluación como una medida de desempeño el cuál “*evalúa*” qué tan bien realizado está el clustering. El objetivo principal del Clustering debe ser siempre la generación de **clusters compactos** que estén **diferenciados** los unos a los otros.



¿Cuál es el Clustering que mejor describe el problema.

Objetivos de la Evaluación

Distinguir si existen estructuras de datos no aleatorias. (También llamado "Tendencia")

Evaluar el ajuste de los Clusters a los Datos

Comparar entre 2 o más algoritmos de Clustering

Determinar el Número de Clusters correctos

Tendencia: Hopkins

Estadístico Hopkins

Permite evaluar **a priori** si es que efectivamente existen clusters **antes de aplicar** un algoritmo.

$$H = \frac{\sum_{i=1}^p w_i}{\sum_{i=1}^p u_i + \sum_{i=1}^p w_i}$$

- w_i : corresponde a la distancia de un punto *aleatorio* al vecino más cercano en los datos originales.
- u_i : corresponde a la distancia de un punto *real* del dataset al vecino más cercano.
- p : Número de puntos generados en el espacio del Dataset.

```
1 from pyclustertend import hopkins
2
3 1-hopkins(X, p)
```

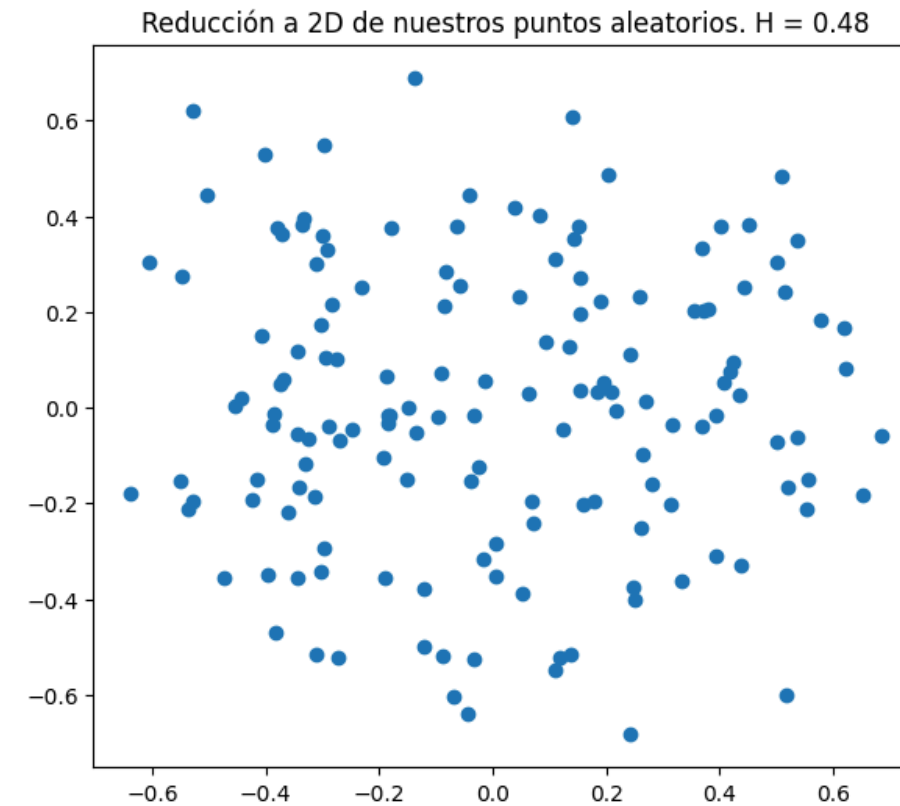
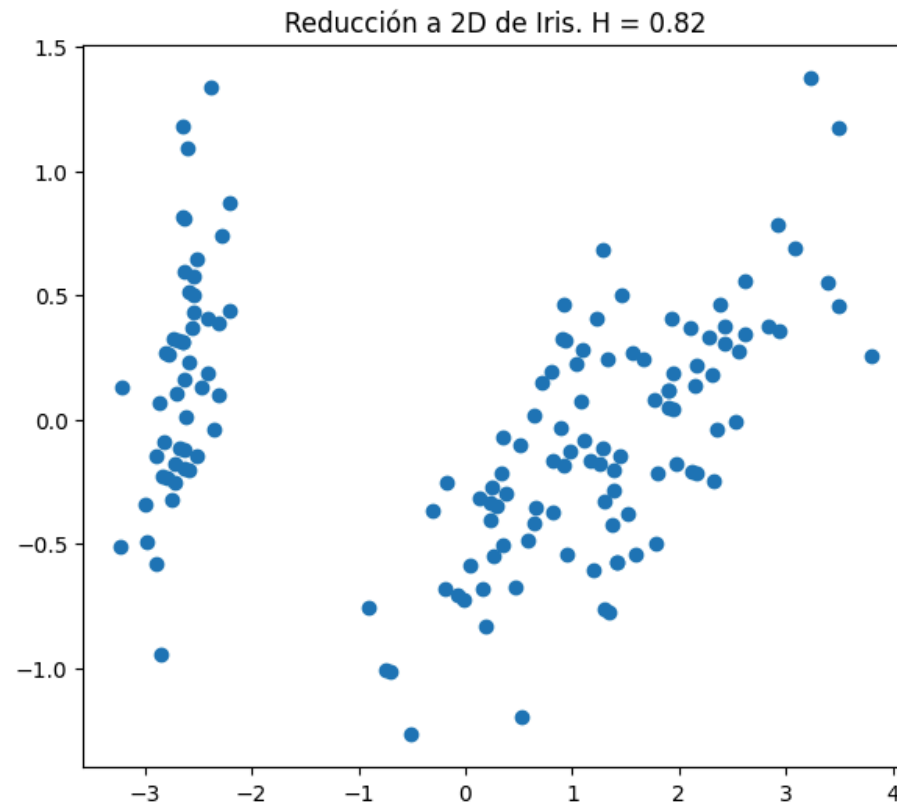


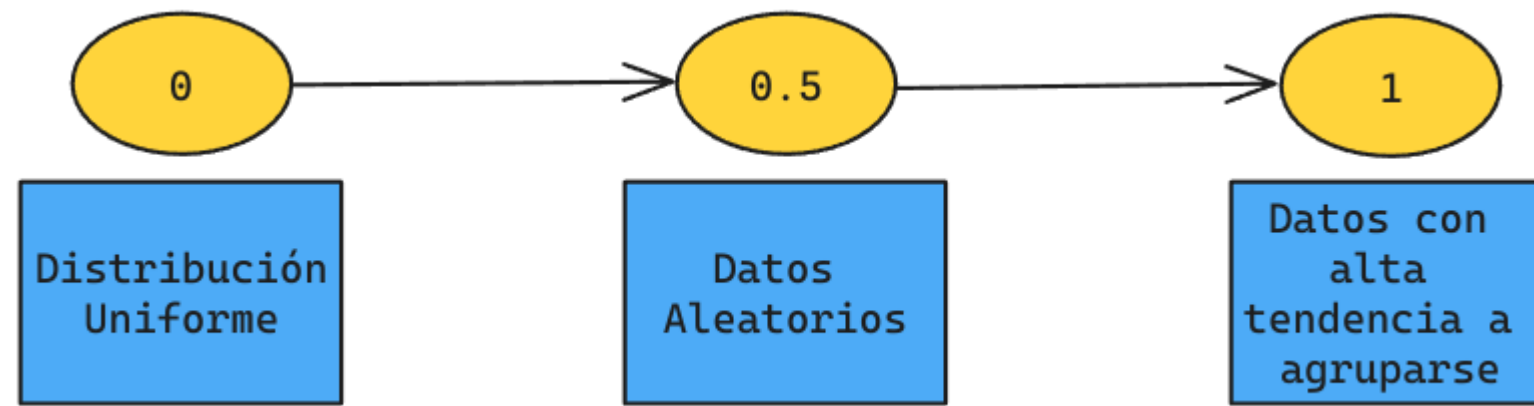
- X: Dataset al cuál se le aplica el Estadístico.
- p: Número de Puntos para el cálculo.



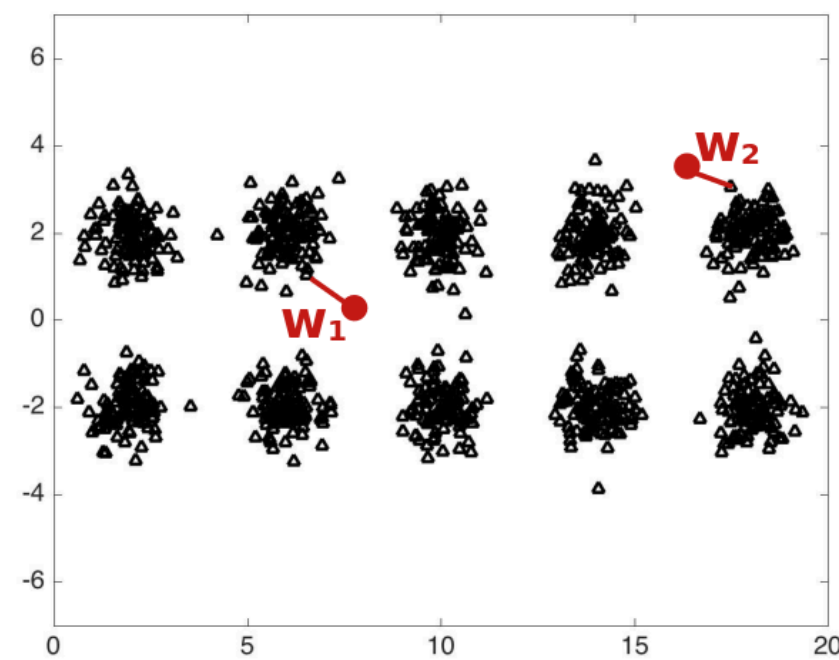
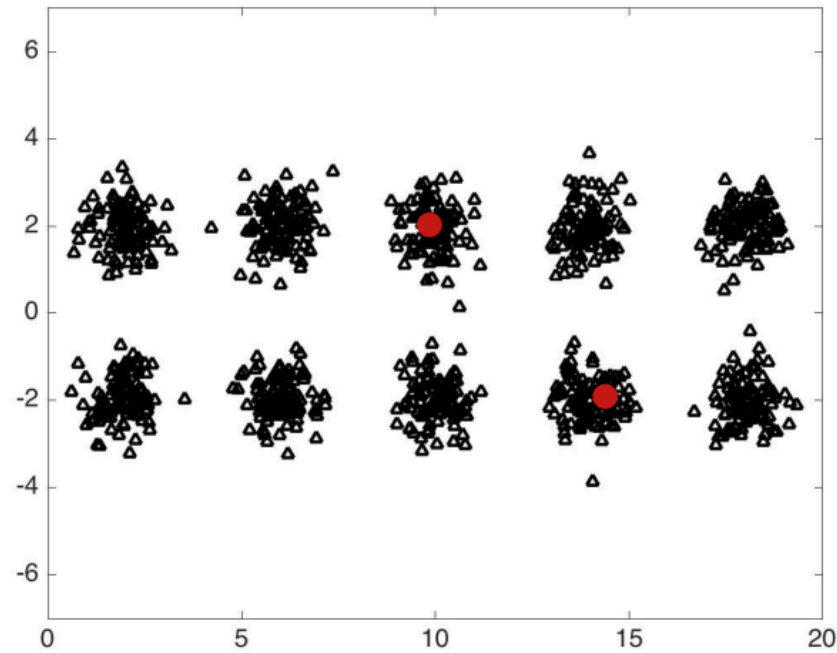
pyclustertend entrega el valor 1-H.

Tendencia: Hopkins





Cálculo Hopkins: Ejemplo p=2



Puntos obtenidos de los Datos

$$u_1 \approx 0$$

$$u_2 \approx 0$$

Puntos Aleatorios en el Espacio de los Datos

$$w_1 \approx 1.8$$

$$w_2 \approx 1.12$$

Cálculo Hopkins

$$H = \frac{w_1 + w_2}{u_1 + u_2 + w_1 + w_2}$$

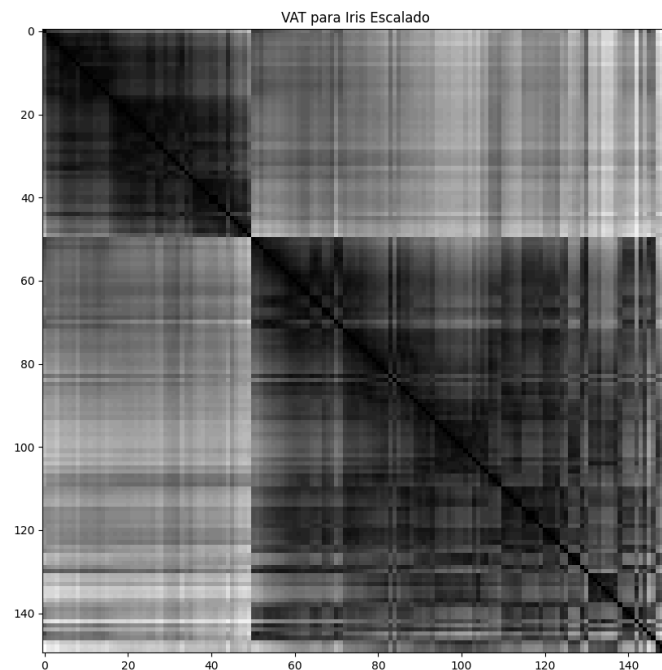
$$H = \frac{1.8 + 1.12}{0 + 0 + 1.8 + 1.12} = 1$$

Visual Assessment of Tendency (VAT)

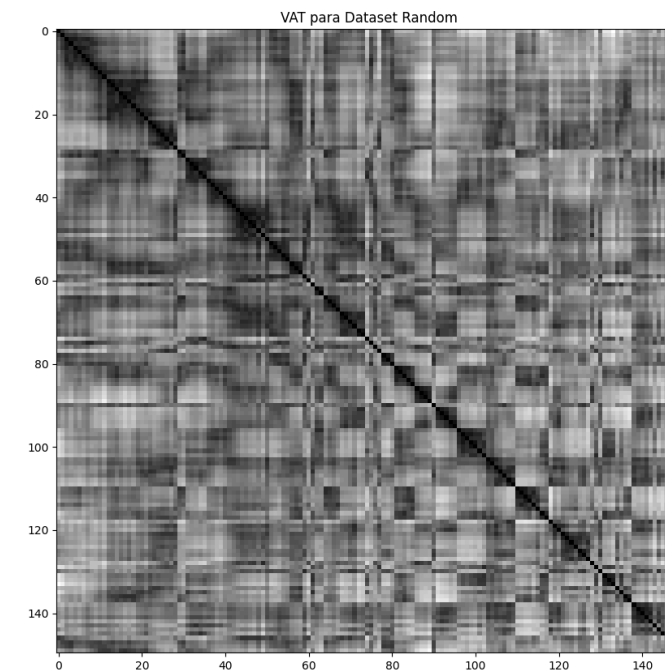
Corresponde a una inspección visual de la distancia entre los puntos (matriz de distancia). Colores más oscuros indican menor distancias entre dichos puntos lo que indica mayor cohesión.



Se pueden ver claramente dos bloques.



No es posible ver bloques importantes.



```
1 from pyclustertend import vat  
2  
3 vat(X)
```



Correlación

Procedimiento:


1. Construir una matriz de similaridad entre todos los puntos de la siguiente manera:

$$s(i, j) = \frac{1}{d(i, j) + 1}$$

2. Construir una matriz de similaridad "*ideal*" basada en la pertenencia a un Cluster.

Si i y j pertenecen al mismo cluster entonces $s(i, j) = 1$, en otro caso $s(i, j) = 0$

3. Calcular la Correlación entre la matriz de similaridad y la matriz ideal (obtenidas en los pasos 1 y 2).

 Una correlación alta indica que los puntos que están en el mismo cluster son cercanos entre ellos.

Cohesión

Cohesión

Mide cuán cercanos están los objetos dentro de un mismo cluster. Se utiliza la Suma de los Errores al Cuadrado, que es equivalente a la Inercia de K-Means (o Within Cluster).

$$SSE_{total} = \sum_{k=1}^K \sum_{x_i \in C_k} (x_i - \bar{C}_k)^2$$

- C_k corresponde al Centroide del Cluster k . Dicho centroide puede ser calculado como la media/mediana de todos los puntos del Centroide.
- K corresponde al Número de Clusters.



- No me gusta mucho este nombre, porque en realidad es como un **inverso de la Cohesión**.

Separación

Separación

Mide cuán distinto es un cluster de otro. Se usa la suma de las distancias al cuadrado entre los centroides hacia el promedio de todos los puntos. (Between groups sum squares, SSB).

$$SSB_{total} = \sum_{k=1}^K |C_k| (\bar{X}_k - \bar{C})^2$$

- $|C_k|$ corresponde al número de elementos (Cardinalidad) del Cluster i .
- \bar{X}_k corresponde al promedio de todos los puntos en el Cluster k .

Coeficiente de Silhouette (Coeficiente de Silueta)

El coeficiente de Silhouette es otra medida que combina la cohesión y la separación. Los valores varían entre -1 y 1, donde valores cercanos a 1 representan una mejor agrupación.

 Valores cercanos a -1 representan que el punto está incorrectamente asignado a un cluster.

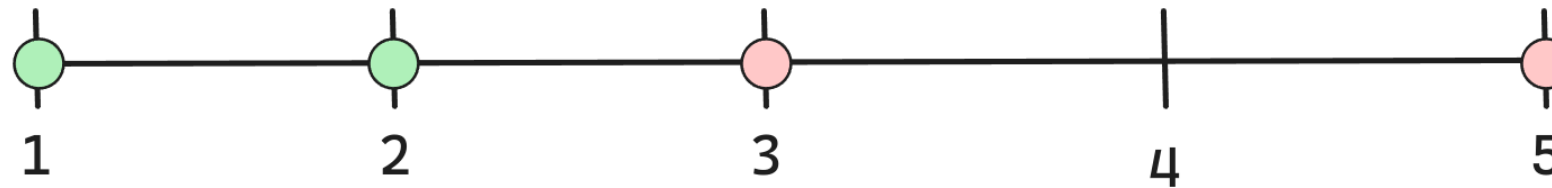
$$S_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

```
1 from sklearn.metrics import silhouette_score
2
3 silhouette_score(X, labels, sample_size = None, metric="euclidean")
```



Coeficiente de Silhouette: Ejemplo

● Cluster 1
● Cluster 2



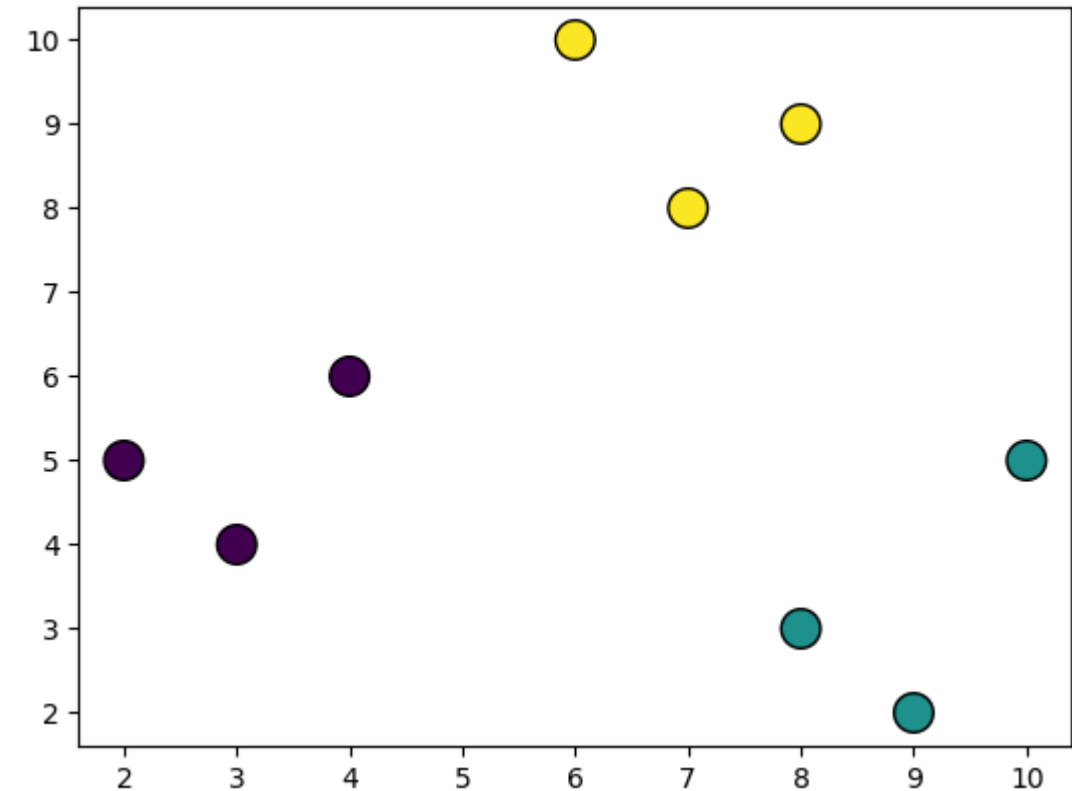
	a_i	b_{i1}	b_{i2}	b_i	s_i
1	1,0	—	3,0	3,0	$2/3$
2	1,0	—	2,0	2,0	$1/2$
3	2,0	1,5	—	1,5	$-0.5/2$
5	2,0	3,5	—	3,5	$1.5/2$

$$C_{silueta} = \sum_i s_i$$

- a_i : Distancia promedio del punto i a todos los **otros** puntos del mismo cluster. (Cohesión)
- b_{ij} : Distancia promedio del punto i a todos los puntos del cluster j donde no pertenezca i . (Separación)
- b_j : Mínimo de b_{ij} tal que el punto i no pertenezca al cluster j . (Menor Separación)

Ejercicio Propuesto

	x	y	c
0	2	5	0
1	3	4	0
2	4	6	0
3	8	3	1
4	9	2	1
5	10	5	1
6	6	10	2
7	7	8	2
8	8	9	2

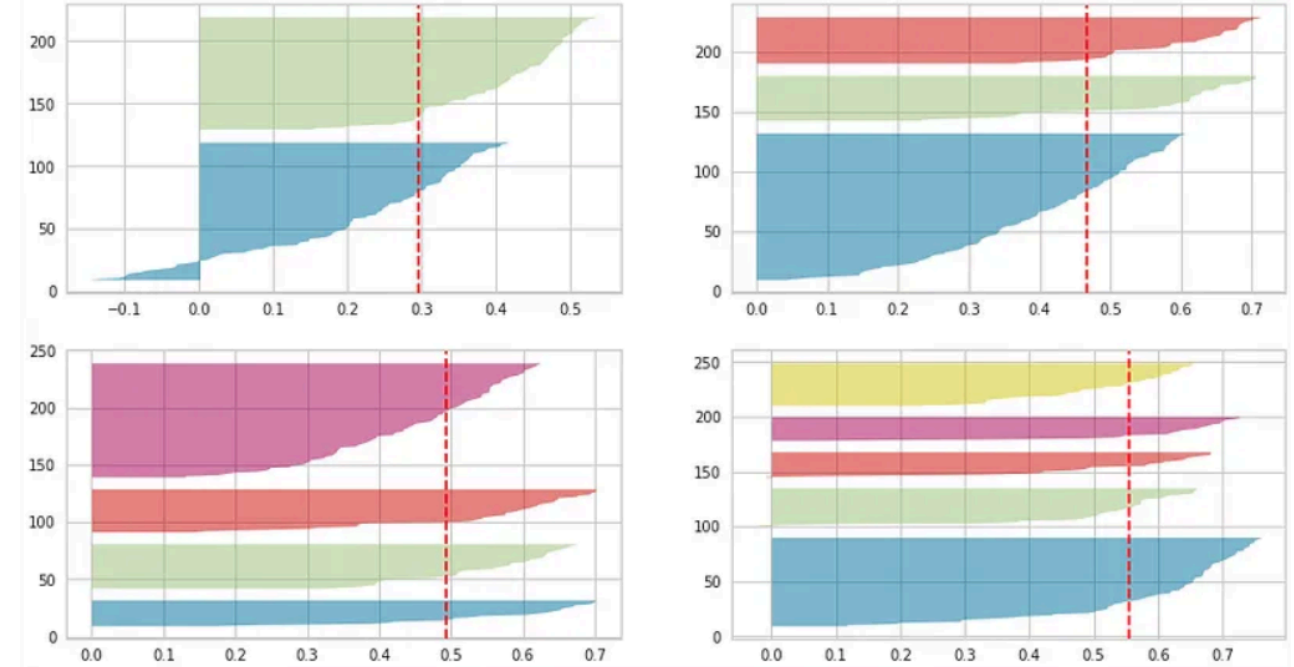
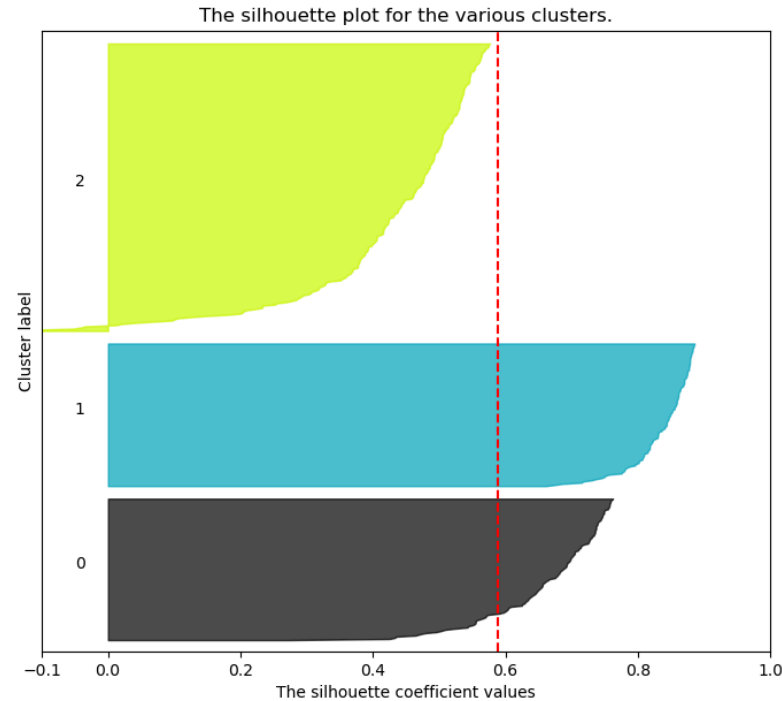


Ejercicio Propuesto

Calcule el coeficiente de Silueta. Tabla de resultado al final de las Slides.

Curvas de Silueta

Es común mostrar los resultados del coeficiente de silueta como gráficos de este estilo:



Problemas

- Siluetas negativas.
- Clusters bajo el promedio.
- Mucha variabilidad de Silueta en un sólo cluster.

Curvas de Silueta: Implementación

```
1 import scikitplot as skplt
2 import matplotlib.pyplot as plt
3
4 skplt.metrics.plot_silhouette(X, labels, metric="euclidean", title="Silhouette Analysis")
5 plt.show()
```

- **L1-2:** Importación de Librerías Necesarias. Esta implementación está en la librería Scikit-plot. (Para instalar `pip install scikit-plot`)
- **X:** Dataset usado para el clustering.
- **labels :** etiquetas obtenidos de algún proceso de Clustering.
- **metric:** Métrica a utilizar, por defecto usa “*euclidean*”.
- **title:** Se puede agregar un Título personalizado a la curva.

¡Felicitaciones! 🎉🎉🎉🎉
Terminamos Clustering

Resultados Ejercicio Propuesto

	a	b0	b1	b2	b	s
0	1.825141	0.000000	7.313443	6.481726	6.481726	0.718417
1	1.825141	0.000000	6.164881	6.478709	6.164881	0.703946
2	2.236068	0.000000	5.828629	4.359229	4.359229	0.487050
3	2.121320	5.474525	0.000000	6.126376	5.474525	0.612511
4	2.288246	6.781151	0.000000	7.313209	6.781151	0.662558
5	2.995353	7.051277	0.000000	5.039300	5.039300	0.405602
6	2.236068	5.861155	7.409079	0.000000	5.861155	0.618494
7	1.825141	5.031119	5.222072	0.000000	5.031119	0.637230
8	1.825141	6.427390	5.847735	0.000000	5.847735	0.687889

Coeficiente de Silhouette = 0.6148

⚠ Comprobar utilizando Scikit-Learn