K-Means
“Íris” Dataset

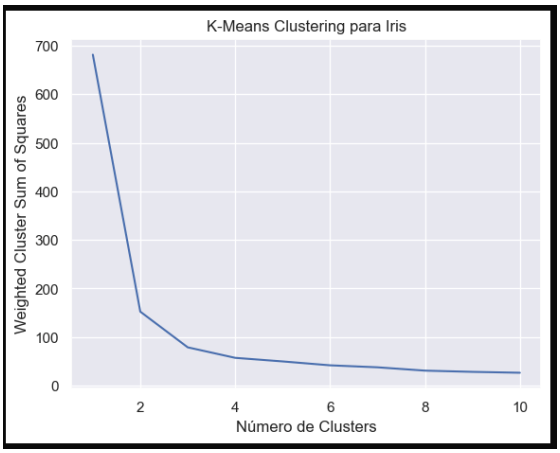
La agrupación de medias K es un algoritmo de aprendizaje no supervisado utilizado para la agrupación en clústeres de datos, que agrupa puntos de datos no etiquetados en grupos o clústeres.

Para está práctica, se comenzó realizando el agrupamiento por medio del dicho algoritmo, con el Dataset “**Íris**” sin ninguna modificación, más que eliminando la columna “Variety” de tipo “string” / “object”, para conservar únicamente las columnas de tipo numéricas.

```
values = dataset.drop("variety", axis = 1)
values
```

	sepal.length	sepal.width	petal.length	petal.width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns



El primer resultado nos muestra por medio de la graficación del “Codo de Jambú” que el número optimo de clusters es **3**.

Sin embargo, al realizar la comprobación con el “**criterio de silueta**”, está nos arroja que el número óptimo es **2**. Con un Score de **0.68**.

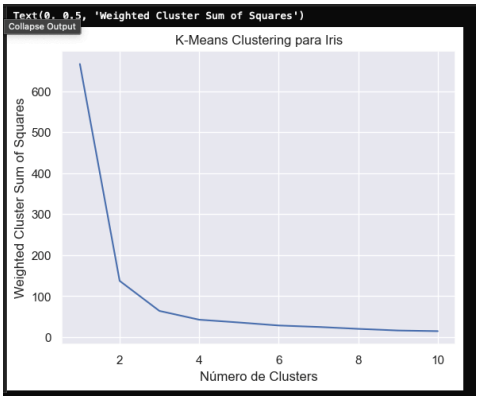
```
X = values.to_numpy()
for j in range(2, 12):
    kmeans = KMeans(n_clusters = j, random_state = 42)
    kmeans.fit_predict(X)
    score = silhouette_score(X, kmeans.labels_, metric = "euclidean")
    print("Score Silhouette: ", "k = ", j, ":", score)
```

```
Score Silhouette:  k =  2 : 0.6810461692117464
Score Silhouette:  k =  3 : 0.5511916046195922
Score Silhouette:  k =  4 : 0.4976433179321928
Score Silhouette:  k =  5 : 0.4930804067193526
Score Silhouette:  k =  6 : 0.36784649847122486
Score Silhouette:  k =  7 : 0.3542978877198852
Score Silhouette:  k =  8 : 0.34467972180562
Score Silhouette:  k =  9 : 0.315588785338977
Score Silhouette:  k =  10 : 0.30141437453251435
Score Silhouette:  k =  11 : 0.26873562164120274
```

Descomposición por PCA

Como segunda parte, se repitieron los procedimientos anteriores no sin antes realizar una descomposición de los datos, por medio de PCA, convirtiendo las 5 columnas del dataset, en únicamente 2.

Los resultados fueron muy similares en cuanto a la gráfica del codo.



Pero en cuanto al “Criterio de Silueta”, si bien muestra que el número óptimo se mantiene en **2**, el Score aumento considerablemente a un **0.70**, el cual nos indica que hay una mejor precisión en cuanto a la determinación de clusters.

```
# Criterio de Silueta
for j in range(2, 12):
    kmeans = KMeans(n_clusters = j, random_state = 42)
    kmeans.fit_predict(pca_values)
    score = silhouette_score(pca_values, kmeans.labels_, metric = "euclidean")
    print("Score Silhouette: ", "k = ", j, ":", score)
```

```
Score Silhouette:  k =  2 : 0.705670322509188
Score Silhouette:  k =  3 : 0.5976764219497546
Score Silhouette:  k =  4 : 0.5577409232179588
Score Silhouette:  k =  5 : 0.5100407194716486
Score Silhouette:  k =  6 : 0.4031973873561552
Score Silhouette:  k =  7 : 0.38682639975495786
Score Silhouette:  k =  8 : 0.440760332470465
Score Silhouette:  k =  9 : 0.40201438761056096
Score Silhouette:  k =  10 : 0.41588658014924934
Score Silhouette:  k =  11 : 0.399432551902981
```

Por último, con las 2 dimensiones que obtuvimos con este paso, procedemos a **graficar los clusters**, los cuales nos muestran una buena separación de las observaciones.

Podemos concluir que si bien el algoritmo K-means es **bastante preciso**, si se logran determinar el número de clusters de forma correcta, ya sea por medio del **gráfico del codo**, el cual se presenta como una opción más amigable en cuanto a visualización, o por medio del **criterio de silueta**, la cual parece mostrar una **mejor precisión** para este caso; lo cierto es que la **descomposición por medio de PCA** ofrece una mejora de la clasificación de los datos, por lo que es recomendable analizar e intentar siempre dentro de lo posible **realizar dicha transformación en algoritmos de clasificación** tal y como se abordó en esta practica con “**K-Means**”.