Proyecto práctico - Aprendizaje Automático I

Gonzalo Lope Carrasco - d21c004
Javier Pérez Vargas - d21c017

November 6, 2023

1 Abstracto

Este informe presenta un proyecto de análisis de datos centrado en la identificación de clusters significativos dentro de un conjunto de datos, junto con la posterior interpretación de estos clusters. El proyecto involucra varios pasos, que incluyen el preprocesamiento de datos, la evaluación de diferentes algoritmos de agrupación, sus resultados y la selección/descripción de una solución de cluster final. Para preprocesar los datos, elegimos llevar a cabo los pasos más comunes. cambiamos las variables categóricas a ordinales basándonos en conclusiones que explicaremos más adelante, eliminamos un valor atípico, escalamos los datos y finalmente imputamos los valores nulos. Posteriormente aplicamos un enfoque exhaustivo para probar diferentes métodos de clustering y sus diferentes parámetros, con el fin de evaluar los resultados y el efecto de las diferentes posibilidades. Finalmente, comparamos los mejores resultados con una representación de los datos en 2D (realizando PCA) y confirmamos que el mejor resultado medio de 3 clusters tiene sentido con nuestra intuición visual.

2 Introducción

La relevancia del análisis de datos es cada vez mayor en diversas áreas, desde la ciencia hasta la industria y la investigación. La capacidad de conseguir obtener patrones significativos en un conjunto de datos que a priori no los tenía se puede convertir en un componente crucial para tomar decisiones informadas y generar conocimiento a partir de unos datos.

El Clustering es una manera de agrupar objetos por similitud en grupos, de forma que todos los miembros del mismo grupo tengan características parecidas. Pertenece a lo que se conoce como **aprendizaje no supervisado**, pues, en un principio, no sabemos a qué clases o grupos pertenecen nuestras instancias, y eso es lo que queremos descubrir. Puede ocurrir incluso que nuestras instancias no pertenezcan a una "clase" real, sino que simplemente queremos ver cómo se relacionan entre ellas.

Para afrontar un problema de Clustering podemos utilizar dos enfoques: **Clustering jerárquico** y **Clustering particional**. Ambos métodos pueden proporcionar buenas y malas soluciones; será nuestra decisión adoptar un modelo u otro dependiendo del problema al que nos enfrentemos. De cada enfoque tenemos una variedad de algoritmos (K-Means, Agglomerative, DBScan, BIRCH...) y cada uno de ellos tiene sus propios parámetros. En este problema analizaremos los resultados que tienen en nuestro conjunto de datos el algoritmo K-Means y el algoritmo Agglomerative con sus diferentes variantes.

Nuestro conjunto de datos es sobre frutas. Contiene 180 instancias y 5 características para cada una. Las características son el peso, la longitud, su anchura, su simetría y la "hendidura" de cada fruta. Esta última se ha discretizado en 5 categorías.

3 Métodos

3.1 Preprocesado

Vamos a ver cómo es nuestro conjunto de datos. Para ello, ejecutamos el siguiente comando:

```
fruits = pd.read_csv("fruits.csv", index_col=0)
fruits.head()
```

	weight	length	width	regularity	cleft
1	1.205	4.603915	2.847	5.691634	Small
2	1.726	5.978000	3.594	4.539000	Large
3	1.126	4.516534	2.710	5.965993	Average
4	1.755	5.791000	3.690	5.366000	Large
5	1.238	4.666888	2.989	6.153947	Small

Table 1: Muestra del conjunto de datos

3.1.1 Datos Categóricos a Numéricos

Cuatro de las cinco columnas eran columnas numéricas continuas, y una de ellas (cleft) era categórica, con 5 posibles valores: "Muy pequeño", "Pequeño", "Promedio", "Grande" y "Muy grande".

Así, en primer lugar, mapeamos los valores de esa columna a valores numéricos. Teníamos dos opciones: OrdinalEncoder o OneHotEncoder. Optamos por la primera opción, ya que los valores de esa columna siguen un orden:

```
categories = ["Very small", "Small", "Average", "Large", "Very large"]
encoder = OrdinalEncoder(categories=[categories])
df['cleft'] = encoder.fit_transform(fruits[['cleft']])
```

OneHotEncoder es una buena opción cuando tenemos una variable categórica que no sigue un orden (ej.: soltero, viudo, casado). Utilizar este método en la columna 'cleft' llevaría a una enorme pérdida de información pues ya no existiría el orden, y la distancia entre "Very small" y "Small" sería la misma que entre "Very small" y "Very large". Como desventaja, también añadiríamos de forma innecesaria 5 columnas a nuestro conjunto de datos, lo cual sería computacionalmente más costoso si el número de instancias es muy alto.

3.1.2 Representar los datos y detectar datos atípicos

Una vez que tenemos nuestras 5 columnas numéricas, podemos hacer un boxplot de los datos y ver en qué rangos se mueven.

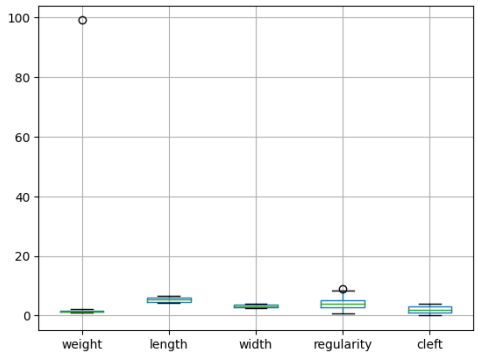

Figure 1: Boxplot de los datos sin escalar con un dato atípico

La distribución de las variables no se puede apreciar porque hay un dato atípico en la columna 'weight' con un valor de casi 100. Veamos cómo es esa instancia.

```
df[(df['weight'] == max(df['weight']))]
```

	Weight	Length	Width	Regularity	Cleft
61	99.0	6.173	3.651	2.443	4.0

Table 2: Fila del dato atípico en peso

Es la instancia 61. Hay diversas formas de tratar estos valores atípicos, pero en este caso, como es el único valor tan diferente, preferimos borrar la instancia.

Una vez borrada, veamos el boxplot de nuevo.

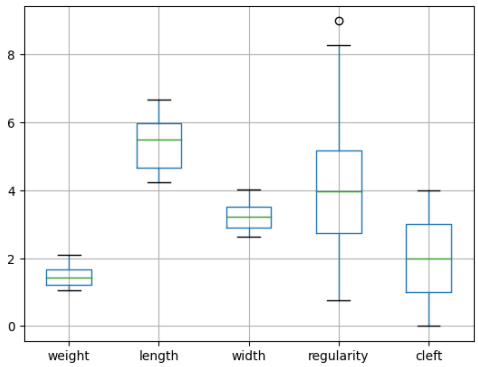

Figure 2: Boxplot de los datos no escalados sin el dato atípico

Ahora podemos observar varias cosas. Las columnas 'weight' y 'width' tienen muy poca variabilidad, por lo que esperamos que contribuyan menos que otras columnas. La 'length' tiene un poco más de varianza, pero la columna con más variabilidad es 'regularity', que incluso tiene un valor atípico, aunque al no ser muy significativo no lo eliminaremos ni lo trataremos.

3.1.3 Escalado de los Datos

Nuestro siguiente paso es normalizar los datos. Hemos decidido usar el método Standard Scaler, que convierte el cada columna en un conjunto con media 0 y varianza 1.

$$z = \frac{(x - \bar{x})}{s_x} \tag{1}$$

En Python:

```
df_scaled[df.columns] = StandardScaler().fit_transform(df[df.columns])
```

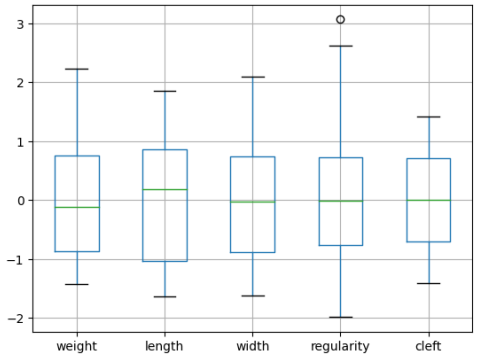

Figure 3: Boxplot de los datos escalados

3.1.4 Imputación de los datos

Echando un vistazo a nuestro conjunto de datos, vemos también que hay varios valores nulos. Veamos cuántos hay.

```
df_scaled.isna().sum()
```

Variable	Weight	Length	Width	Regularity	Cleft
NaN Data	12	24	17	10	0

Podemos observar que los datos en todas las columnas excepto en 'cleft' tienen datos nulos. Antes de tratarlos, veamos cómo están distribuidos. Para eso, nos apoyaremos en varios scatterplot.

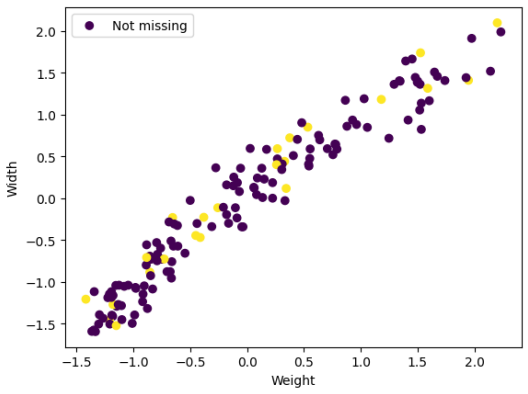

Figure 4: Scatter plot del 'width' y el 'weight'

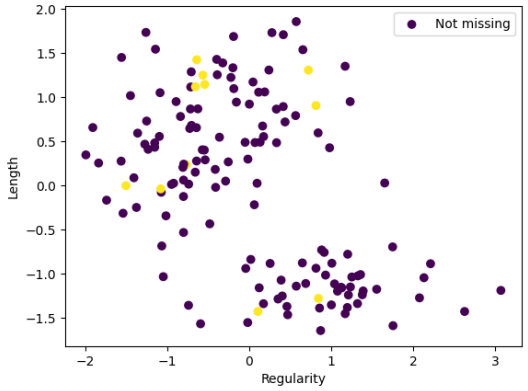

Figure 5: Scatter plot del 'regularity' y la 'length'

Claramente vemos que estos valores nulos se distribuyen de manera "Completely at Random" (CAR). Esto significa que los datos faltan por razones que no siguen un patrón concreto. Una de las soluciones para este problema es la imputación multivariante, y decidimos utilizar el KNNImputer, que estima los valores nulos en función de los valores de sus k-vecinos más cercanos.

```
imputer = KNNImputer(n_neighbors=3, weights="uniform")
df_imputted[df_scaled.columns] = imputer.fit_transform(df_scaled[df_scaled.columns])
```

Así, ahora con nuestros datos estandarizados y sin valores nulos, podemos probar los algoritmos de clustering.

3.2 Metodos de Clustering

Vamos a utilizar dos enfoques de clustering diferentes: **Particionales** and **Jerárquicos**

3.2.1 Clustering Jerárquico

El clustering jerárquico crea una estructura en forma de árbol, que se puede visualizar con un dendrograma. Las alturas en el dendrograma representan una medida de similitud o disimilitud entre los clusters que se fusionan. La elección de K no determinará el resultado final. Podemos elegir K una vez que haya terminado el algoritmo.

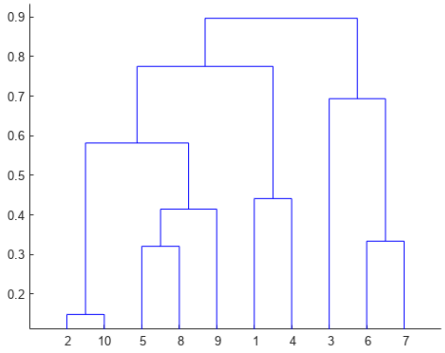

Figure 6: Ejemplo dendrograma

Vamos a estudiar el **método aglomerativo**, donde cada punto comienza como parte de un cluster y luego los clusters se fusionan de dos en dos, creando clusters más grandes. Existen diferentes métodos de enlace (linkage methods) para ello, y también podemos variar la medida de disimilitud.
Utilizaremos 4 tipos de linkage diferente, que procedemos a explicar:

- **Single**: La distancia entre dos clusters se mide por la distancia más pequeña entre sus puntos.
- **Complete**: La distancia entre dos clusters se mide por la distancia más grande entre sus puntos.
- **Average**: La distancia entre dos clusters se mide por la distancia media entre sus puntos.
- **Ward**: La distancia entre dos clusters se mide por el incremento de la dispersión dentro de la fusión de los dos clusters.

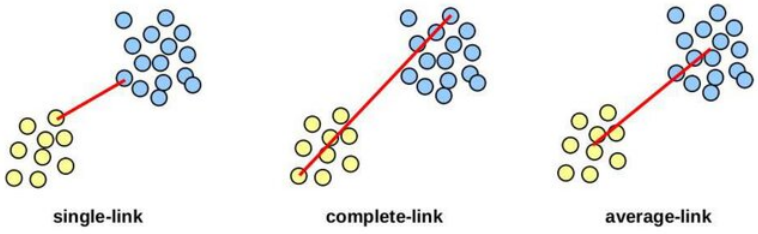

Figure 7: Single, complete y average linkage

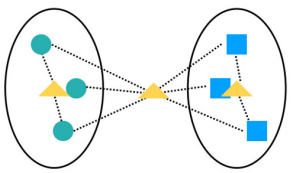

Figure 8: Ward linkage

Además, hemos utilizado dos distancias diferentes para cada método:

- **Euclídea**: Es la distancia entre dos puntos deducida a partir del teorema de Pitágoras.
- **Manhattan**: La distancia entre dos puntos se calcula sumando la diferencia absoluta entre sus coordenadas horizontales y verticales.

Figure 9: Distancia euclídea y manhattan

3.2.2 Clustering Particional

En el clustering particional, los puntos se dividen en un número fijado de clusters. Esto produce una estructura donde todos los clusters están al mismo nivel. Al contrario que en clustering jerárquico, cada punto pertenece solamente a un grupo. El método más conocido es K-Means.

El algoritmo K-Means se basa en el uso de centroides que representan a cada cluster. Expliquemos el algoritmo paso a paso:

1. Asignamos, de forma aleatoria, tantos centroides como clusters deseemos en nuestro resultado final.
2. Medimos la distancia de cada punto a todos los centroides, asignándole a cada punto el cluster del centroide que tenga más cerca.
3. Recalculamos los centroides de cada cluster tomando el promedio de los puntos del cluster.
4. Volvemos al paso 2 hasta que los centroides convergen.

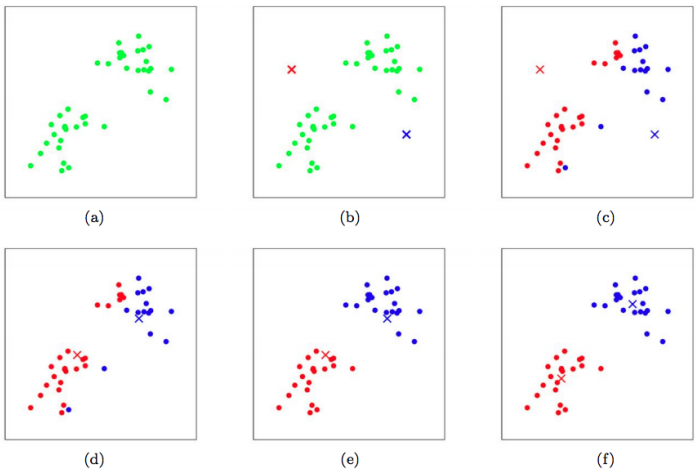

Figure 10: K-means iteración tras iteración

En la figura podemos ver cómo funciona el algoritmo con un conjunto de datos fácilmente separable. Tras 5 iteraciones el algoritmo consigue separar adecuadamente los datos, a pesar de que los centroides se inicializaron lejos de cada nube de puntos.

No obstante, los conjuntos de datos suelen ser más complejos y tener muchas variables, y la inicialización de los centroides tiene importancia. Por ello, se suele aplicar el algoritmo varias veces, y nos quedaremos con aquel resultado que reduzca más la inercia (suma de las distancias al cuadrado de todos los puntos a sus centroides respectivos).

4 Resultados

4.1 Clustering Jerárquico

4.1.1 Single Linkage

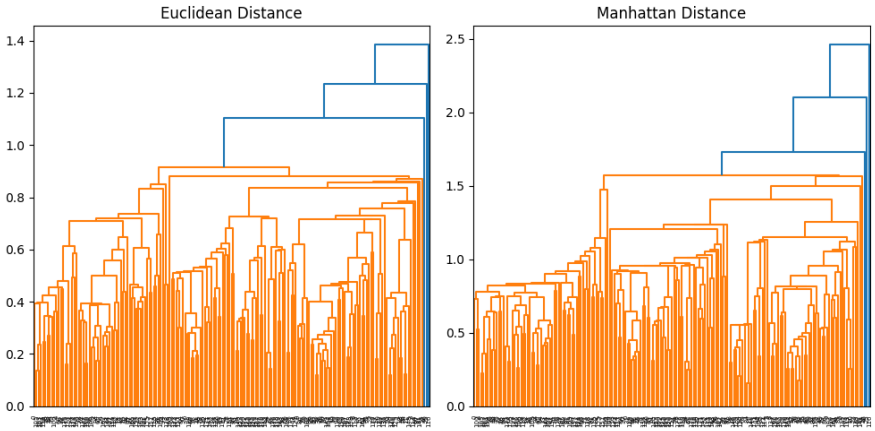

Figure 11: dendrograma para el clustering Jerárquico con el Single Linkage

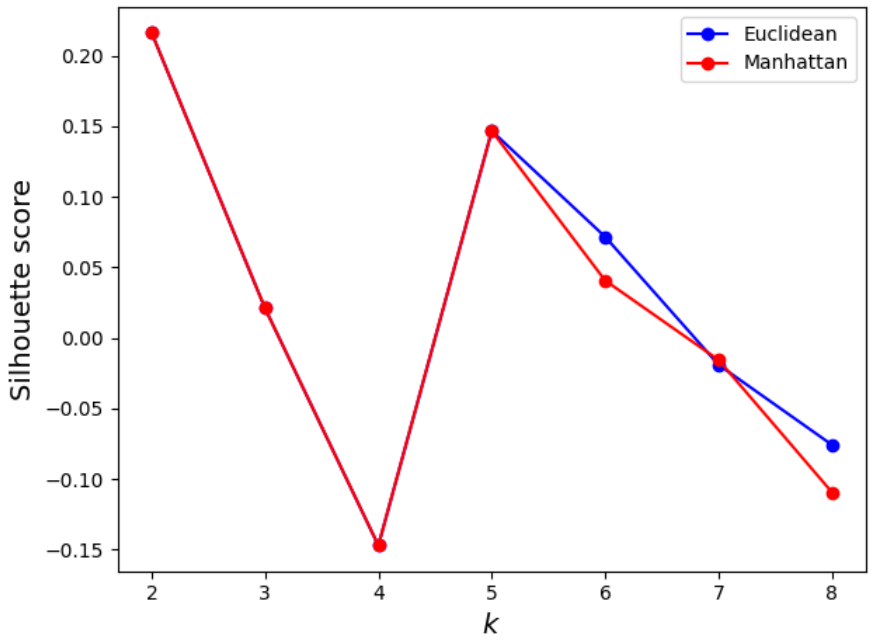

Figure 12: Coeficiente de Silhouette para distinto número de clusters y distintas distancias usando Single Linkage

4.1.2 Complete Linkage

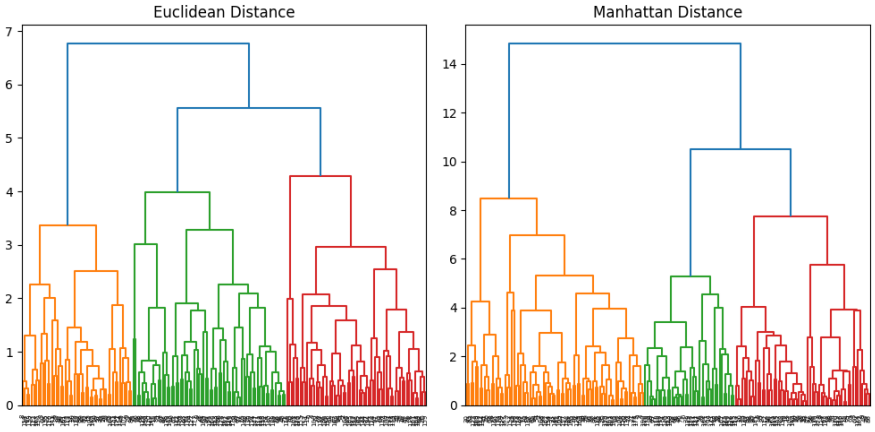

Figure 13: dendrograma para el clustering Jerárquico con el Complete Linkage

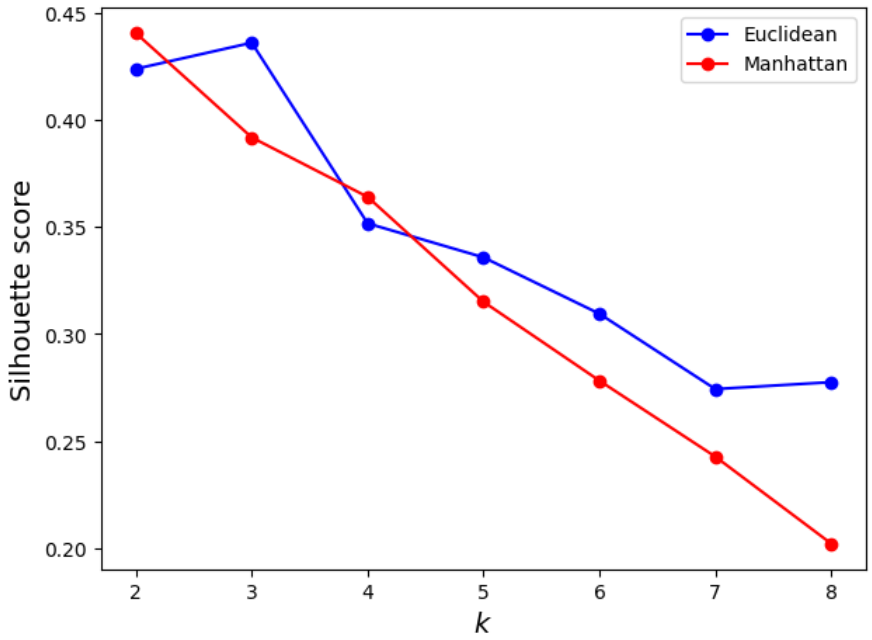

Figure 14: Coeficiente de Silhouette para distinto número de cluster y distintas distancias usando Complete Linkage

4.1.3 Average Linkage

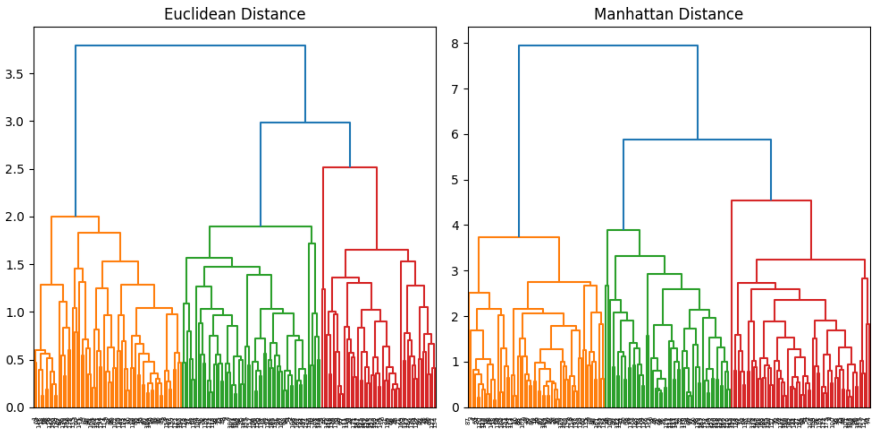

Figure 15: dendrograma para el clustering Jerárquico con el Average Linkage

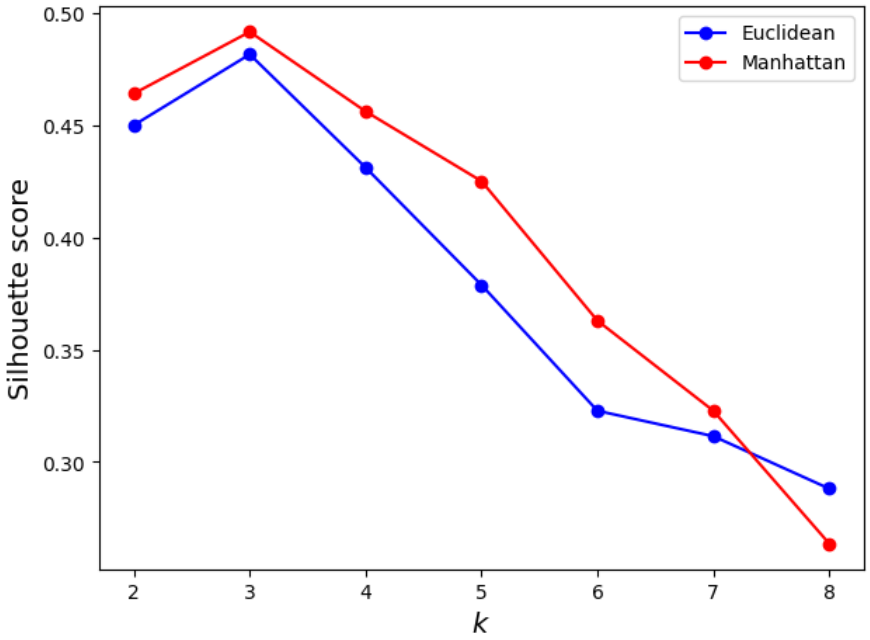

Figure 16: Coeficiente de Silhouette para distinto número de clusters y distintas distancias usando average Linkage

4.1.4 Ward Linkage

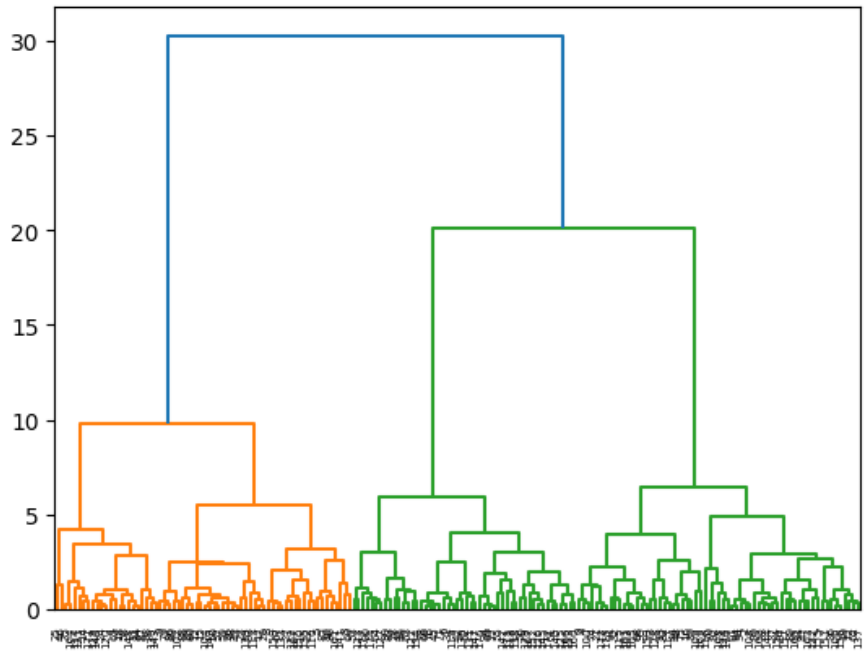

Figure 17: dendrograma para el clustering Jerárquico con el Ward Linkage

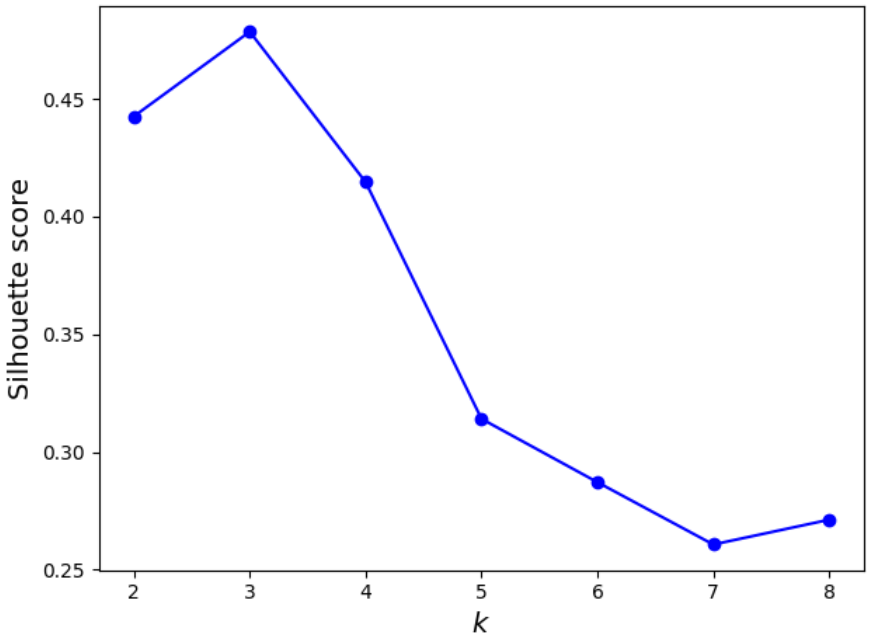

Figure 18: Coeficiente de Silhouette para distinto número de clusters usando Ward Linkage

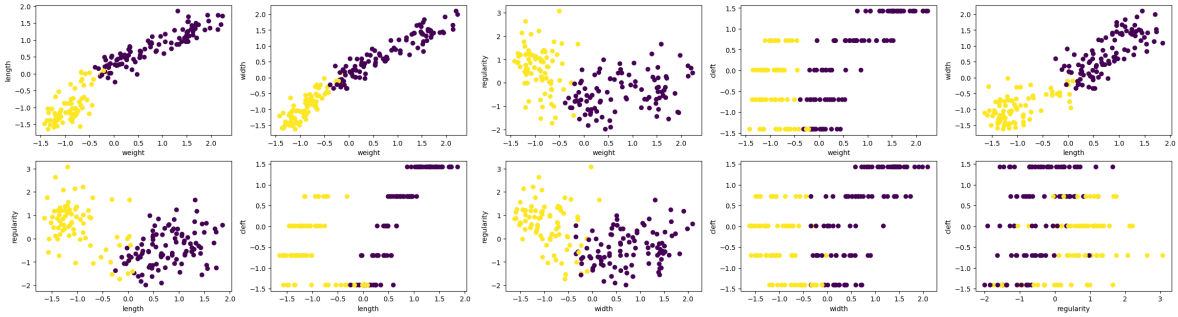

Figure 19: Visualización de 2 clusters elegidos mediante el clustering jerárquico con complete linkage y la distancia manhattan

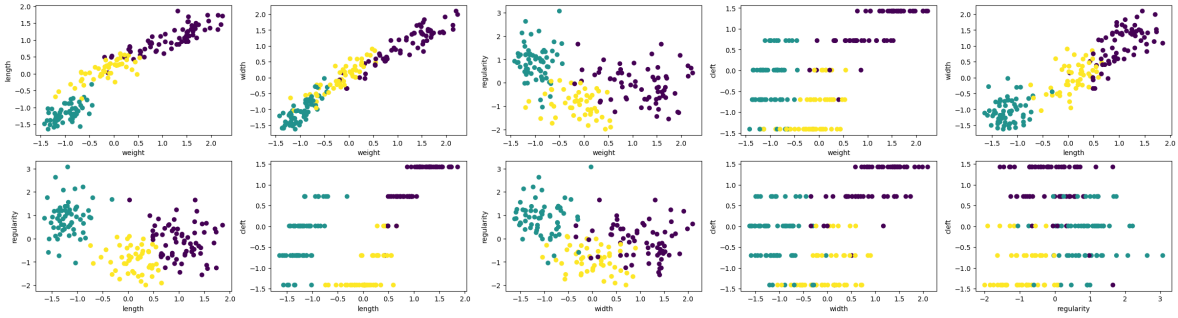

Figure 20: Visualización de 3 clusters usando el ward linkage

4.2 Partitional Clustering

4.2.1 K-Means

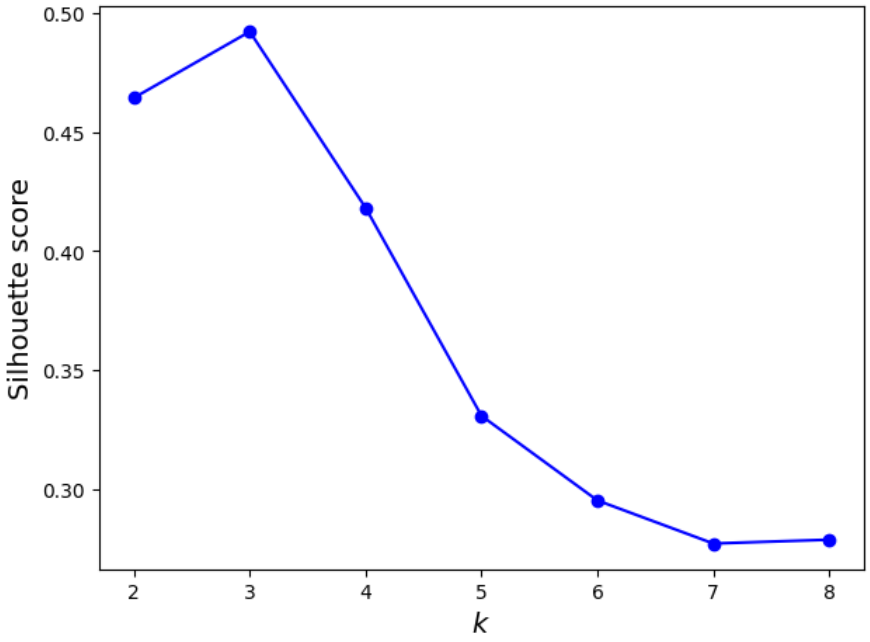

Figure 21: K-Means silhouette

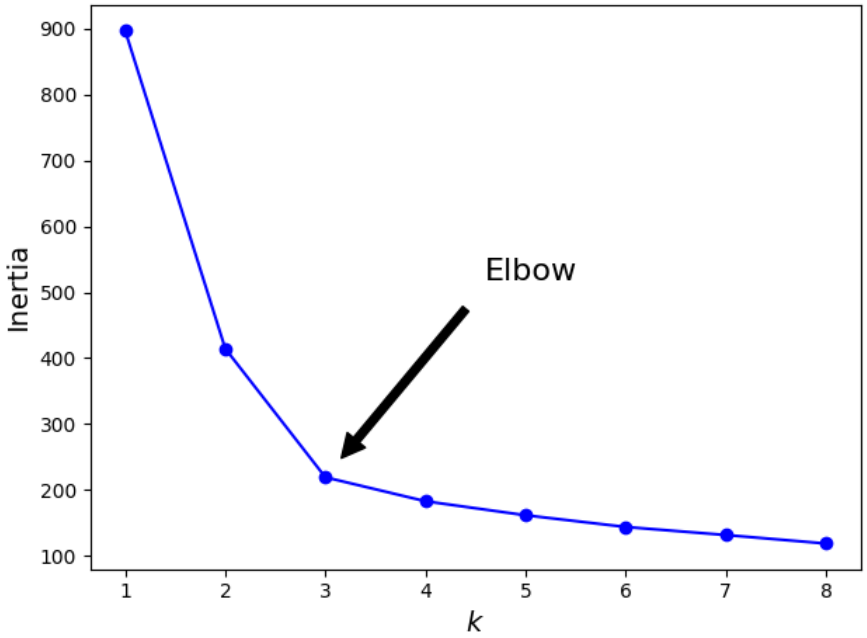

Figure 22: Codo en la gráfica de las inercias de los modelos de K-Means

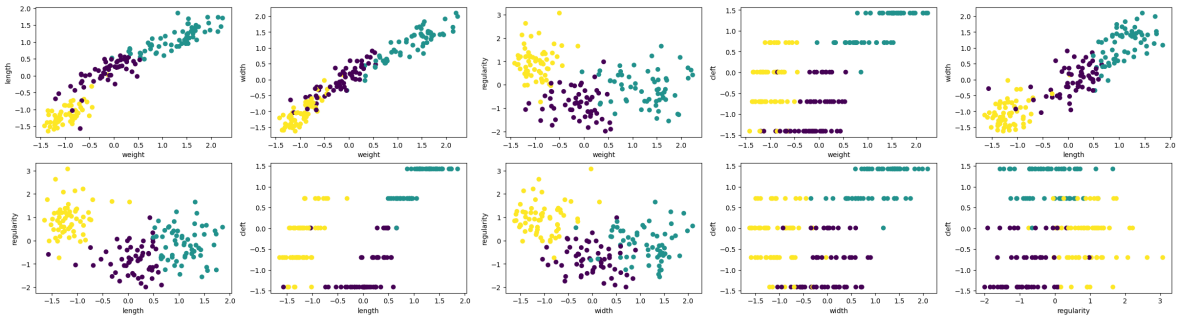

Figure 23: Visualización de 3 clusters obtenidos mediante K-Means

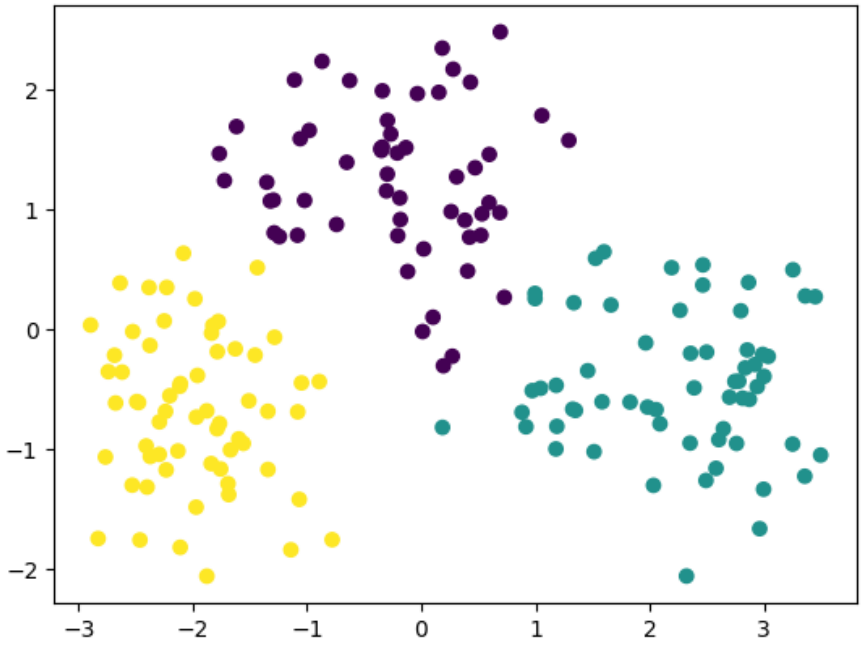

Figure 24: Visualización en 2 dimensiones obtenidas por PCA de los tres clusters obtenidos mediante K-Means

5 Discusión

5.1 Clustering Jerárquico

Como podemos ver en la figura Fig.11 al hacer clustering jerárquico con el **single** linkage este tiende a unir la mayor parte de los puntos en un solo cluster haciendo muy desbalanceados el número de puntos si eligiésemos más de un cluster. Esto se puede ver reflejado en Fig.12, donde el mayor valor del coeficiente de Silhouette es con 2 clusters. Además, es un coeficiente muy bajo (0.2), por lo que deducimos que el resultado utilizando single linkage es mejorable utilizando otros métodos.

Con el linkage de tipo **complete**, podemos observar que las conclusiones, tomando como medida de la calidad el coeficiente de silhoutte, cambian dependiendo de la distancia usada a la hora de hacer los clusters. En la gráfica del coeficiente Fig.13 vemos que con la distancia euclídea la mejor clusterización sería con 3 clusters mientras que con la distancia manhattan como medida de disimilaridad se escogerían 2 clusters en vez de 3.

Por otra parte, usando el **average** linkage, ambas distancias llevan a la misma conclusión Fig.16 y cabe destacar que los clusters obtenidos mediante la distancia manhattan obtienen mejores valores de Silhouette. Es interesante ver que no siempre la distancia euclídea es la que ofrece mejores resultados. Algo a destacar del average linkage es que consigue uno de los mejores coeficiente de Silhouette de todos los métodos que vamos a ver, llegando muy cerca de 0.5.

Finalmente para el **ward** linkage, podemos observar en el dendrograma Fig.17 que existen 3 clusters bien marcados, conclusión que podemos reforzar con la gráfica del coeficiente de Silhouette Fig.18. Pero hay que destacar que dos de estos clusters se encuentran mas cerca entre ellos que con el tercero, cosa que no han destacado los demás tipos de linkage y que debemos tener en cuenta a la hora de llegar a una conclusión.

Podemos verificar esta discrepancia entre los métodos gracias a los scatter plots de las Fig.19 y Fig.20. Vemos que existen relaciones entre variables que son claramente clasificables en dos clusters, como puede ser entre cleft y length. Por otro lado, muchas de las otras gráficas quedan mejor repartidas entre 3 clusters, viéndose de manera clara que escogiendo 2 clusters estamos uniendo dos clusters

diferentes simplemente y por tanto perdiendo información.

5.2 Clustering Particional

Al usar el método de K-Means para encontrar las mejor partición en clusters de los datos la conclusión es parecida, tanto si observamos las inercias para los distintos valores de k Fig.22 y escogemos el 'elbow' de la gráfica, como si observamos la gráfica de coeficiente de Silhoutte Fig.21 y escogemos el k con mayor valor, la conclusión es la misma: la división en 3 clusters es la mejor para estos datos.

Podemos observar esta partición tanto en los diferentes scatter plots entre variables, como al aplicar PCA para reducir el conjunto de datos a 2 dimensiones y representar la clusterización Fig.24, constatando lo visto en los gráficos de validación (Silhouette y Elbow).

6 Conclusión

En conclusión, hemos pensado que la mejor forma de dividir los datos es con 3 clusters y con el método de clustering particional K-means debido a que el coeficiente de silhoutte es de los más altos para 3 clusters. K-means, además, es un método bastante eficiente en caso de que se quisiese predecir el cluster al que pertenecen nuevos datos. También hemos visto que al aplicar PCA los grupos que se forman con los clusters de K-Means son muy compactos. Los diferentes clusters representan diferentes cosas y los describiremos haciendo referencia a su color en la Fig.23.

El primer cluster (color amarillo) representa las frutas con peso, longitud y ancho más bajos, con una simetría alta en comparación con las otras frutas y una hendidura de todos los tamaños excepto muy grande.

El segundo cluster, de color morado, es de peso, altura y ancho medio, con una simetría baja y una hendidura de tamaño desde muy pequeño a normal.

Finalmente, el tercer grupo de frutas, el de color verde, incluye a las frutas de ancho, alto y peso alto, con simetría media y con una hendidura grande o muy grande.