UNIVERSIDAD DE GRANADA

Clustering con Knime

Tratamiento Inteligente de Datos
Master Universitario en Ingeniería Informática

En esta práctica veremos el uso de algoritmos de agrupamiento o clustering utilizando Knime. Se trabajará con un conjunto de datos sobre el que se emplearán diferentes técnicas de clustering y a la luz del conocimiento descubierto se podrán concluir diferentes aspectos sobre los datos. Se valorará la interpretación de los resultados, la complejidad de los experimentos realizados, y la organización y redacción del informe.

1. Clustering en KNIME

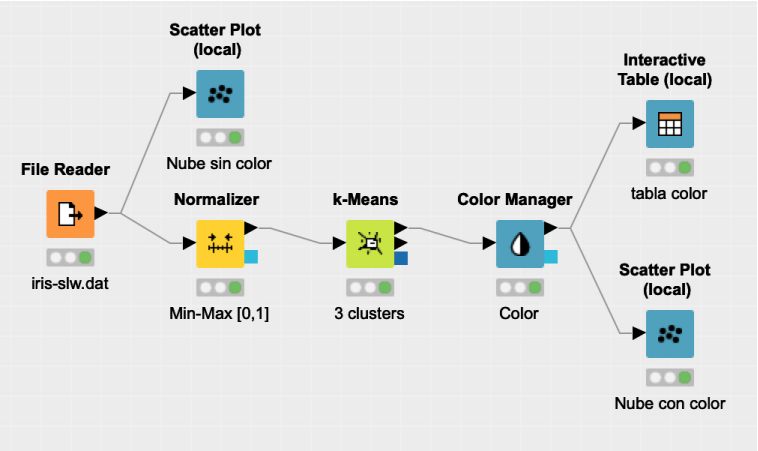
Los algoritmos de clustering en KNIME se encuentran en la carpeta `Analytics > Mining > Clustering` del repositorio de nodos. Aunque, como pasa con los nodos de clasificación, existen nodos de Weka para clustering en la carpeta `Analytics > Mining > Integrations > Weka > Weka (3.7) > Clustering Algorithms`. Sigue los pasos siguientes para ver un pequeño ejemplo de su uso.

Crea un proyecto en KNIME con los siguientes nodos que nos permitan agrupar los datos de `iris-slw.dat`. Este fichero contiene los datos de instancias de plantas de la familia iris (lirios) a las que se les ha calculado longitud y anchura del sépalo. Este fichero es una manipulación de `iris.data` en el que cada instancia contiene cuatro características (longitud y anchura del sépalo, y del pétalo) y también el tipo de planta (la solución al agrupamiento). Por simplicidad y una mejor visualización, se ha utilizado esta versión reducida de la base de datos, pero se puede utilizar como ejercicio el archivo original. Basta eliminar de las instancias la última columna para realizar verdaderamente un aprendizaje no supervisado y comparar después con los agrupamientos verdaderos.

Se han utilizado los siguientes nodos.

- Un nodo para leer el fichero.
- Un nodo `Manipulation > Column > Transform > Normalizer` para normalizar los datos, ya que KNIME implementa k-Means sin normalizar previamente las variables.
- Un nodo `Analytics > Mining > Clustering > k-Means` para realizar el clustering. Este nodo añadirá una columna Cluster, indicando el agrupamiento asignado a cada tupla de nuestro conjunto de datos. En su configuración, debemos indicar los atributos que se usarán para establecer los clusters. En nuestro ejemplo, usaremos ambos.
- Un nodo `Views > Property > Color Manager` para colorear los datos correspondientes a la columna Cluster del nodo anterior.
- Un nodo `Views > Local > Interactive Table` para ver los resultados.
- Dos nodos `Views > Local > Scatter Plot` para ver la nube de puntos original y la coloreada con los clusters obtenidos.

El diagrama resultante debe quedar de la siguiente manera:

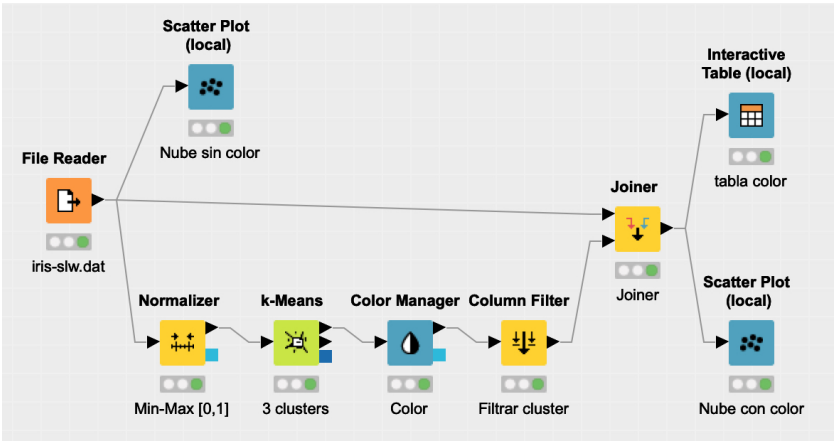


Para ver los centroides obtenidos, podemos fijarnos en la pestaña `2: Clusters` del nodo `k-Means`, que muestra las coordenadas de los centroides de cada clase. Por otro lado, con los nodos `Interactive Table` y `Scatter Plot` podemos seleccionar puntos del gráfico de dispersión y verlos en la tabla interactiva, o al contrario.

Observa que, en nuestro proyecto actual, tanto la tabla como la nube de puntos nos muestran los datos normalizados. Si queremos ver los datos originales, basta hacer lo siguiente:

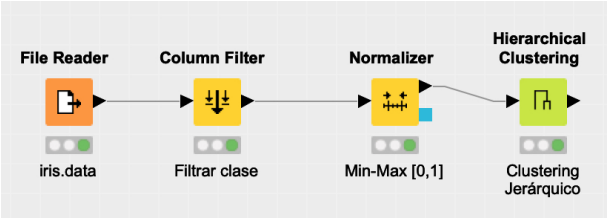
- De la salida de `k-Means`, mantenga únicamente el atributo Cluster y el identificador de fila (RowID). Esto lo podemos hacer con un nodo de tipo `Column Filter` aplicado sobre la salida de `Color Manager`.
- A continuación, combina el resultado del `Column Filter` con los datos originales mediante un nodo `Joiner` (eligiendo el RowID para ambos inputs en la pestaña Join Column).

Así quedaría nuestro proyecto KNIME tras realizar estas modificaciones:

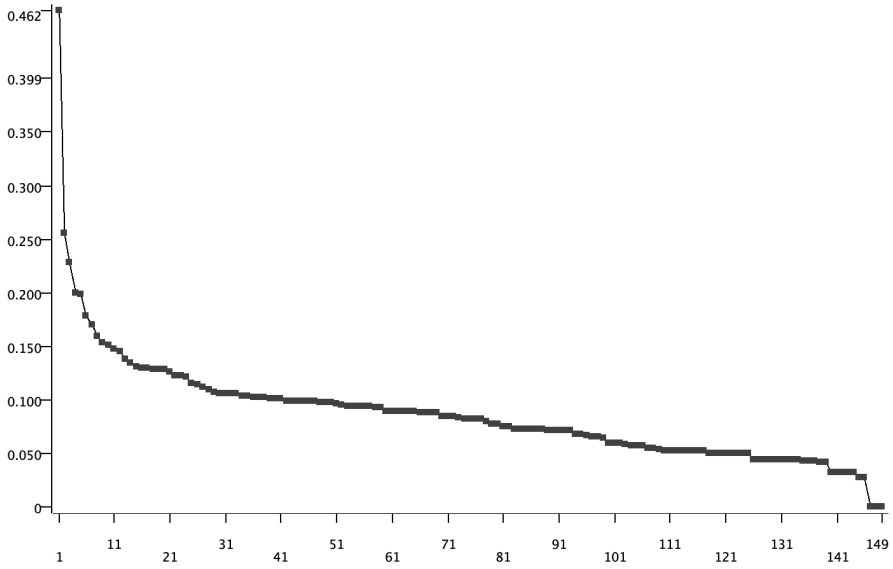


Este fichero de datos es parte de un conjunto de datos muy conocido que se utiliza normalmente como ejemplo de clasificación. Las instancias corresponden a tres categorías diferentes: Iris Setosa, Iris Versicolor e Iris Virginica. Por lo tanto, es lógico que hayamos utilizado tres cluster en el algoritmo k-medias. Sin embargo, a priori, sin conocer esta información, no está claro el número de cluster a utilizar. Para intentar determinarlo, podemos realizar un clustering jerárquico usando el nodo `Hierarchical Clustering` sobre el archivo `iris.data` que

contiene el conjunto de datos completo de atributos e instancias.

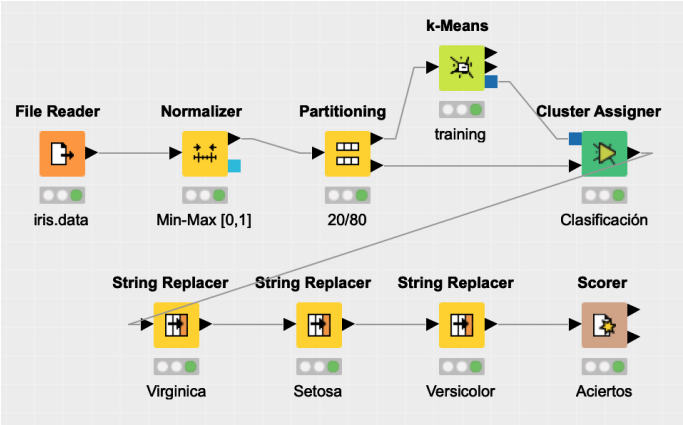


Hemos filtrado el atributo clase puesto que es la solución al problema. En la configuración del nodo `Hierarchical Clustering` seleccionamos, por ejemplo, la distancia euclídea y el tipo de enlace `SINGLE`. Entonces obtenemos una distancia según clusters como especifica el siguiente gráfico (se puede obtener pinchar con el botón derecho en `Hierarchical Clustering` y seleccionar `View: Dendrogram /Distance View`)



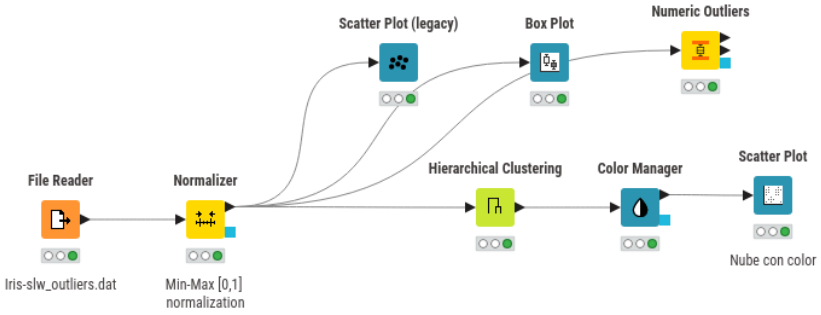
Siguiento la regla del codo, el gráfico parece sugerir que el número adecuado de clusters debería ser tres, cuatro o cinco, que es el momento a partir del cual la distancia deja de dar saltos y se va reduciendo de forma lenta y continua.

Por otro lado, los algoritmos de clustering pueden utilizarse para clasificación. En KNIME esta tarea se puede realizar con el nodo `Cluster Assigner`. Por ejemplo, vamos a considerar de nuevo el archivo `iris.data` y vamos a dividir los datos en dos conjuntos: uno de entrenamiento y otro de validación.

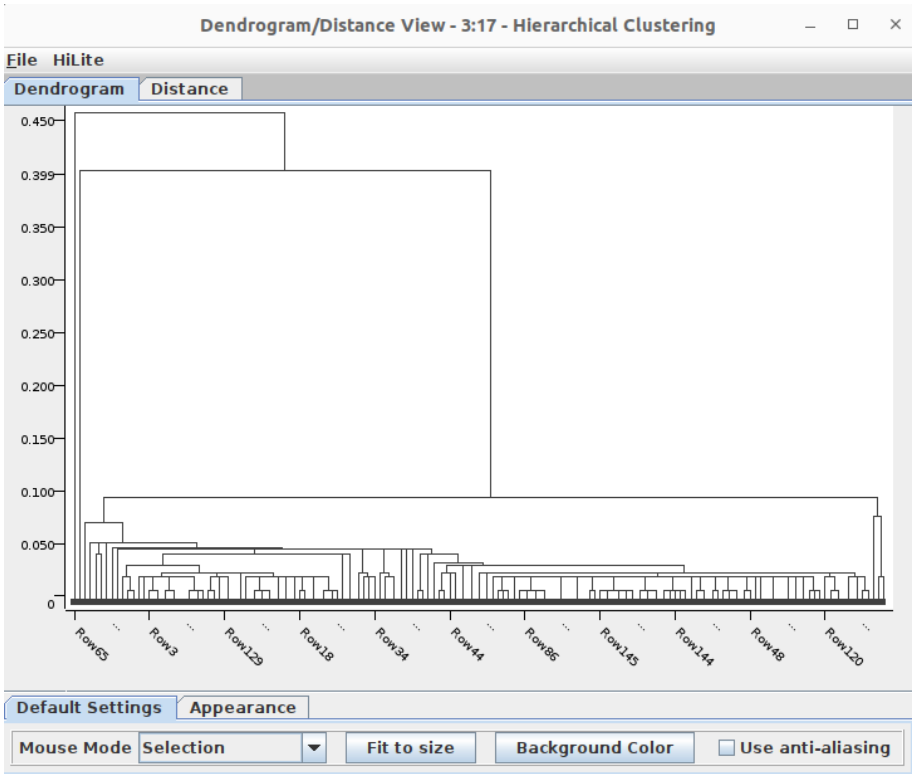


Con el conjunto de entrenamiento, utilizamos el algoritmo k-medias (para tres cluster con la distancia Euclídea) para crear el modelo de prototipos. Comparando con los datos reales, vemos que el modelo nombra como `cluster_0` al cluster formado por `iris-virginica`, `cluster_1` al cluster formado por `iris-setosa` y `cluster_2` al cluster formado por `iris-versicolor`. Una vez clasificado el conjunto de validación, cambiamos los nombres con el nodo `String Replacer` para poder comparar con los datos reales. Este proceso nos da un acierto del 85.8 %, proporcionado por el nodo `Scorer`.

Por último, los algoritmos de clustering sirven también para detectar outliers. Los outliers o puntos anómalos podrían identificarse a partir de técnicas de visualización y análisis estadístico visto en prácticas anteriores, con nodos como `Scatter Plot` y `Box Plot`.



Si usamos un algoritmo de clustering jerárquico de tipo aglomerativo, como el implementado en el nodo `Hierarchical Clustering`, en el que empezamos con muchos clusters pequeños que se van combinando en clusters más grandes, un outlier se unirá a aquel cluster que esté más cercano a él en las fases finales del agrupamiento jerárquico. Los outliers serán aquellos valores que, de forma aislada, se unen a algún cluster en las últimas iteraciones del algoritmo jerárquico. Esto puede observarse en el dendrograma como las líneas más altas que nacen desde abajo y se conectan con otros clusters muy arriba. También se puede elegir visualmente el número de clusters óptimo, analizando el dendrograma de un algoritmo jerárquico, y trazando una línea horizontal cuando hay saltos verticales grandes y poco numerosos (que significan distancia entre clusters grande).



También se puede elegir visualmente el número de clusters óptimo, analizando el dendrograma de un algoritmo jerárquico, y trazando una línea horizontal en la parte alta del mismo, que corresponde a pocos clusters. El número de líneas verticales que se corten, es el número de clusters. Como regla general, se debe cuando elegir un número de clusters lo más pequeño posible (lo más arriba posible del dendograma), y con una distancia vertical suficiente hasta un agrupamiento con más clusters (más abajo).

Considera el archivo `iris-slw_outliers.dat`.

- Realiza un clustering jerárquico similar a lo anteriormente explicado.
- Analiza la existencia de outliers analizando el dendrograma.
- Haz un diagrama de dispersión en el plano y compara los outliers obtenidos con los sugeridos por el dendrograma.
- En el caso de existir outliers, elimínalos y repite el estudio. Compara ambos experimentos.

Aunque la base de datos clásica de Iris funciona bien con el algoritmo *k-Means*, podemos encontrar problemas para realizar clustering con bases de datos más complejas. Opcionalmente, como ampliación puede seguirse el workflow ejemplo de la web de Knime utilizando el algoritmo *k-Medoids*.

2. Vino (8 puntos)

El archivo `wine.data` contiene los datos reales de 178 vinos de una misma región de Italia. Cada instancia está compuesta por trece atributos numéricos más una clase (la primera columna) que determina el nivel de alcohol del vino (tres tipos; 1, 2 y 3), y es la solución al proceso de clustering (por lo que se debe eliminar para realizar la tarea de minería de datos). La descripción de los atributos se encuentra en el fichero `wine_names.txt` (por algún motivo, falta la descripción de una de las columnas). Se deben realizar las siguientes actividades:

- Realizar un algoritmo de clustering jerárquico para analizar en cuántos clusters diferentes podríamos agrupar los datos.
- Analiza la existencia de outliers y elimínalos si consideras que existe alguno. Repite el clustering jerárquico y vuelve a analizar el número de cluster a considerar.
- Aplica el algoritmo k-medias al archivo del punto anterior con el número de clusters elegido. Compara cómo se distribuyen los clusters respecto a las clases de la primera columna.
- Aplica algún tipo de reducción de dimensionalidad que consideres oportuno (filtrando columnas, correlacción, análisis de componentes principales, etc...) y aplica el algoritmo k-medias para tres cluster. De nuevo, compara cómo se distribuyen los clusters respecto a las clases de la primera columna.
- Opcionalmente, aplica el algoritmo DBSCAN basado en la densidad al archivo original, para varios parámetros de radio (epsilon) y puntos mínimos. Compara cómo se distribuyen los clusters respecto a la primera columna ¿Se pueden identificar los outliers?

3. NBA (2 puntos)

Este ejercicio se deja como un problema abierto consistente en desarrollar una agrupación/clustering de un dataset de jugadores de baloncesto que jugaron algún partido de la NBA durante la temporada regular 2021-2022. Concretamente, se pretende aprovechar las técnicas de clustering para examinar qué diferentes tipos de jugador podemos encontrarnos en los equipos de la NBA. El dataset se llama `NBA_RegularSeason2021_2022.xlsx`, y es recomendable realizar una análisis exploratorio y un preprocesamiento de datos para agilizar los algoritmos de clustering.