UNIVERSIDAD DE GRANADA

CLUSTERING CON KNIME

TRATAMIENTO INTELIGENTE DE DATOS

MÁSTER UNIVERSITARIO EN INGENIERÍA INFORMÁTICA

AUTOR

Pablo Valenzuela Álvarez (pvalenzuela@correo.ugr.es)

CONTENIDO

1. VINO

Para empezar con la práctica vamos a asignar nombres a las columnas según vienen dados en el documento "*wine_names.txt*".

- Col0 → Class
- Col1 → Alcohol
- Col2 → Malic acid
- Col3 → Ash
- Col4 → Alcalinity of ash
- Col5 → Magnesium
- Col6 → Total phenols
- Col7 → Flavanoids
- Col8 → Nonflavanoid phenols
- Col9 → Proanthocyanins
- Col10 → Color intensity
- Col11 → Hue
- Col12 → OD280/OD315 of diluted wines
- Col13 → Proline

1.1. IDENTIFICANDO LOS CLÚSTERS

Una vez terminado, pasamos a dibujar los dendogramas del problema aplicando el nodo "*Hierachical Clustering*".

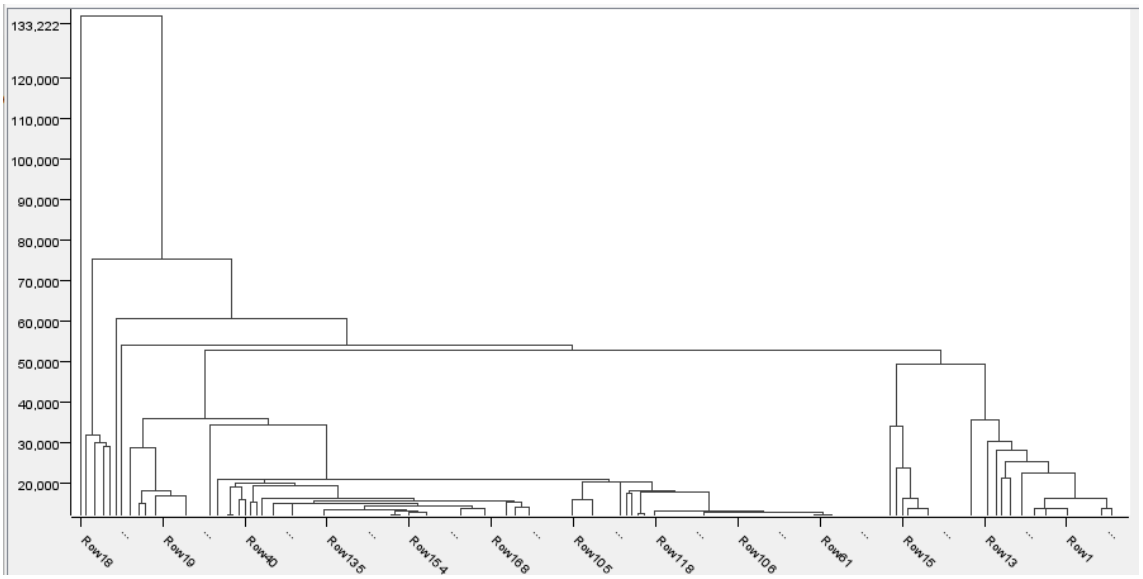


Figura 1. Dendograma jerárquico

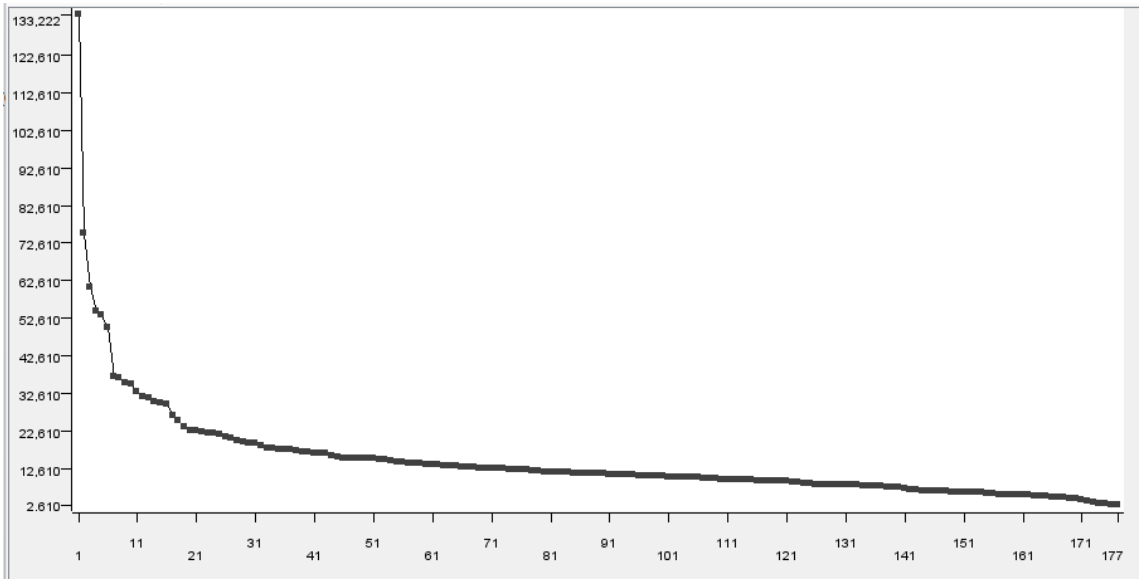


Figura 2. Vista de distancia

Analizando las figuras 1 y 2 podemos deducir que el número adecuado de clústers pueden ser **tres** o **cuatro**, sin embargo, en la figura uno podemos ver que hay ramas con pocos ejemplos y que se unen al final de la jerarquía y, ramas que mantienen la mayoría de los ejemplos. Esto puede indicar la presencia de valores anómalos y para comprobarlo vamos a aplicar el nodo "*Box Plot".*

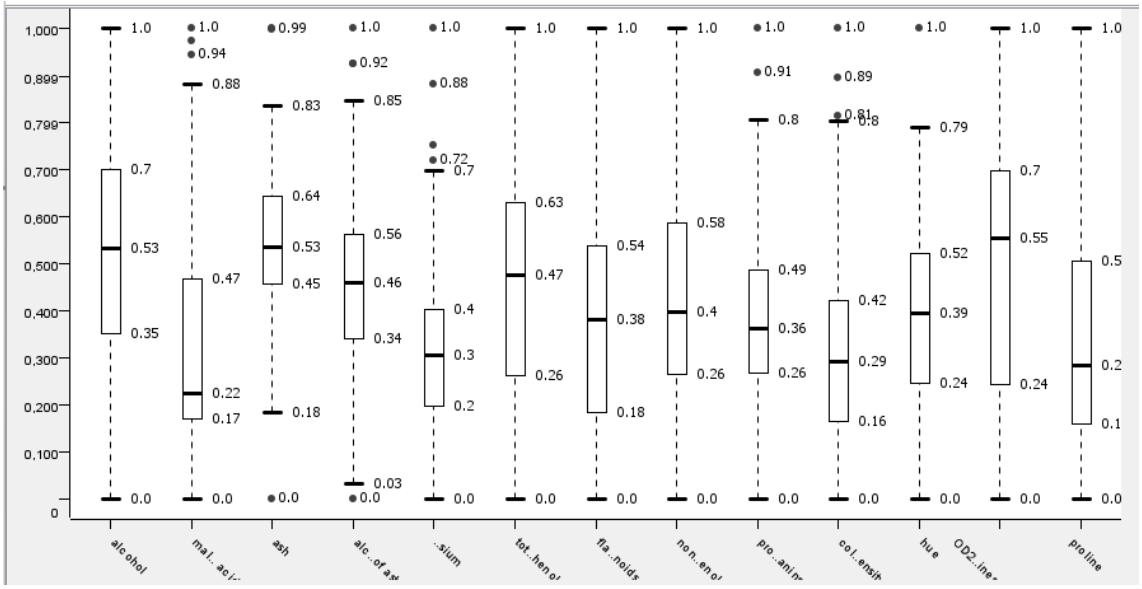


Figura 3. Diagrama de cajas

En la figura 3 podemos observar el diagrama de cajas del problema, que indican los valores que se salen de los habituales para cada atributo. En la figura aparecen varios atributos con estos valores (malid acid, ash, alcalinity of ash, magnesium, y más). Procederemos a quitarlos mediante el nodo "*Numeric Outliers".*

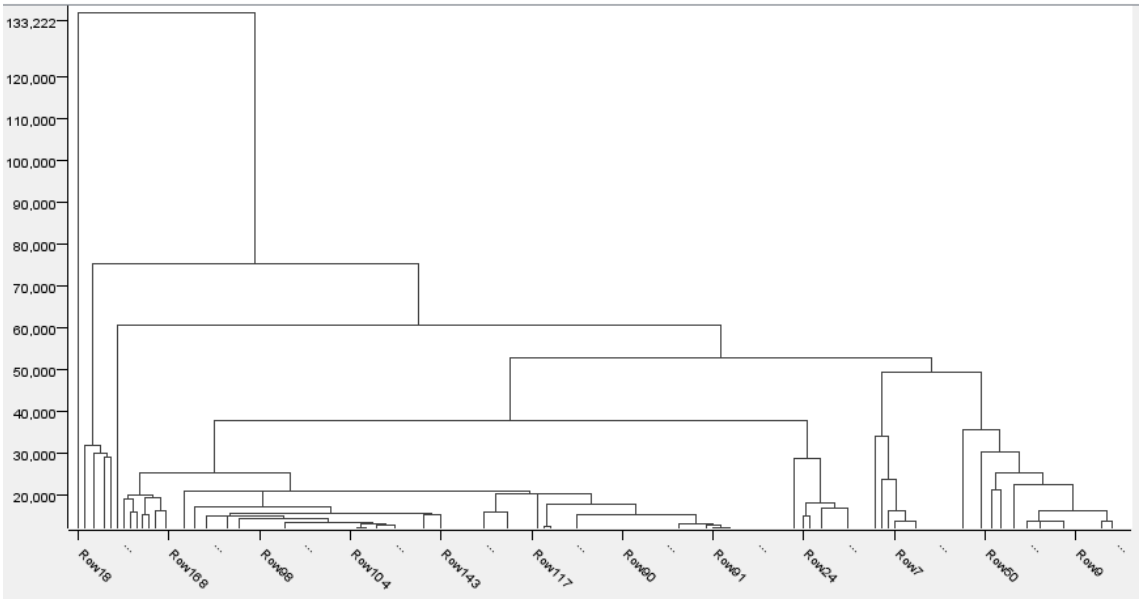


Figura 4. Dendograma jerárquico tras eliminar valores anómalos

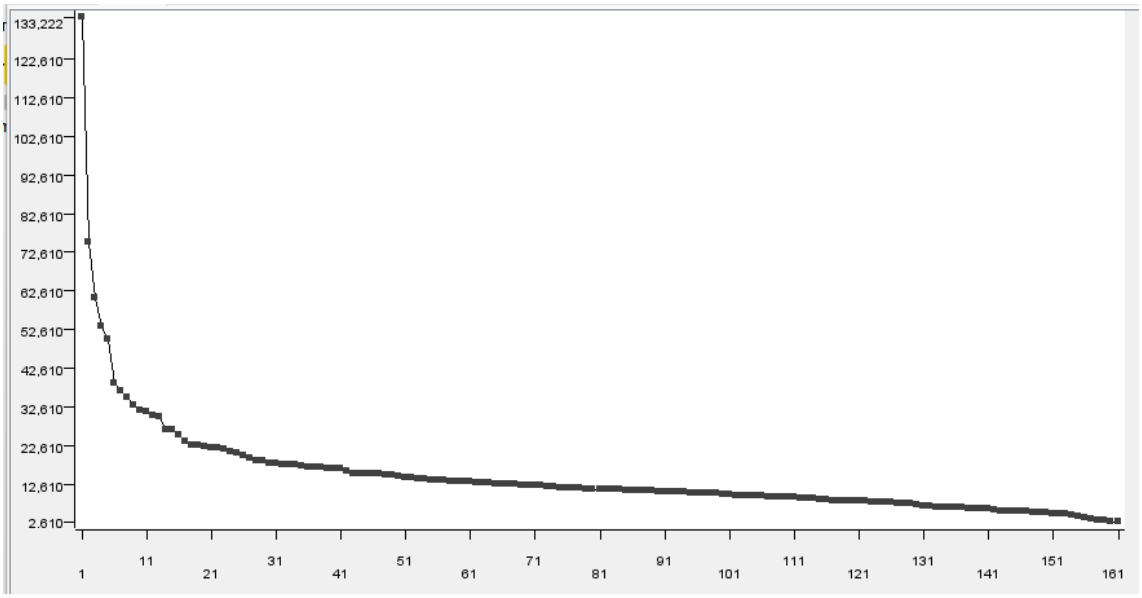


Figura 5. Vista de distancia tras eliminar los valores anómalos.

En estas figuras (figuras 4 y 5) se aprecia un cambio en la distribución jerárquica y de distancia respecto a las anteriores (figuras 1 y 2). En este caso nos recomienda el uso de **tres** a **cinco** clústers.

1.2. PRUEBAS CON K-MEANS

A continuación, se realizarán varias pruebas sobre el conjunto, utilizando el algoritmo K-Means (usado para detectar agrupaciones) con un distinto número de clústers y, también aplicaremos técnicas de reducción de características.

1.2.1. K-MEANS CON CINCO CLÚSTERS

Se empezará probando el algoritmo K-Means con cinco clústers. Como ya sabemos de antemano a la clase a la que pertenece cada caso podemos ver la distribución que tiene cada clúster de las clases.

Clúster	Class	Cantidad
0	**1**	**58**
1	**2**	**41**
2	**2**	**18**
	3	1
3	2	1
	3	**6**
4	2	1
	3	**36**

Tabla 1. Distribución de cada clase en los cinco clústers

En la distribución de la tabla 1 observamos que los clústers uno y dos contienen solo ejemplos de las clases uno y dos, pero los demás tienen ejemplos de más clases. Por lo que se le asignará al clúster el valor de la clase mayoritaria situada en él mismo.

Con estos cambios podemos separar el conjunto en dos, uno para entrenar y otra para validar, y probar la eficacia de este algoritmo.

Partición Train/Test	Accuracy
70/30	16,33 %
80/20	48,49 %
90/10	**82,35 %**

Tabla 2. Precisión del algoritmo con cinco clústers

Según los resultados de la tabla 2, no hemos obtenido mucha precisión salvo en la última partición.

1.2.2. K-MEANS CON TRES CLÚSTERS

La siguiente etapa es ver, para tres clústers, como de bien los clasifica el algoritmo K-Means. Se realizarán las mismas tablas y al final compararemos los resultados con el punto anterior.

Clúster	Class	Cantidad
0	**1**	**58**
	2	4
1	**2**	**54**
2	2	3
	3	**43**

Tabla 3. Distribución de cada clase en tres clústers

Vamos a hacer lo mismo que en el punto anterior y considerar el valor del clúster como la clase mayoritaria contenida en él.

Partición Train/Test	Accuracy
70/30	32,65 %
80/20	27,27 %
90/10	**35, 29 %**

Tabla 4. Precisión del algoritmo con tres clústers

La precisión que obtenemos según la tabla 4 es mucho peor, aunque mejora en la partición 70/30 en general es bastante mala.

Esto puede explicarse dado al número de características que tiene este problema o que distintas clases tengan características muy parecidas, lo que conlleva a un mal entrenamiento.

El algoritmo K-Means está pensado para identificar agrupaciones como hemos comentado, por lo que si varias de las agrupaciones detectadas comparten la misma clase, ya sea porque sus características son parecidas o por otra razón, provocan una mala precisión final como pasa en este caso.

1.2.3. PCA Y K-MEANS CON TRES CLÚSTERS

Como última prueba sobre el conjunto vamos a reducir la complejidad del problema a dos dimensiones y probar otra vez el algoritmo K-Means con tres clústers.

La complejidad la podemos reducir mediante el nodo "*PCA*", que nos permite bajar las dimensiones de un problema con la mínima pérdida de información. Al reducir a dos dimensiones o características, tenemos ahora un par de coordenadas que pueden ser representados mediante una gráfica de dispersión (ver figura 6).

Nota: Comentar que los dendogramas obtenidos tras realizar el clustering son idénticos a los vistos en las figuras 4 y 5.

Figura 6. Gráfica de dispersión al reducir a dos dimensiones y usando tres clústers.

En la figura 6 podemos observar como son separados en el espacio los tres clústers, lo siguiente que tenemos que realizar es la evaluación del algoritmo con este "nuevo" conjunto de datos.

Cluster	Class	Cantidad
0	**1**	**45**
1	1	13
	2	11
	3	**23**
2	**2**	**55**
	3	24

Tabla 5. Distribución de cada clase en los tres clúster.

En la tabla 5 podemos apreciar un problema con los clústers uno y dos, ambas agrupaciones contienen un número elevado de instancias con distinta clase. Aunque siempre hay una dominante puede influir en la eficiencia del algoritmo al no identificar bien las clases que se descartan.

Partición Train/Test	Accuracy
70/30	44,23 %
80/20	**74,29 %**
90/10	44,44 %

Tabla 6. Precisión del algoritmo reduciendo a dos dimensiones y con tres clústers

Aunque en la tabla 6 vemos mejores resultados que los obtenidos en la tabla 4, los resultados dictan mucho de ser buenos. Quizá los resultados de la partición 80/20 se pueden considerar buenos sabiendo los pocos ejemplos de entrenamiento y test de los que disponemos, pero en mi opinión creo que el problema está en los datos y el algoritmo usado para la predicción.

Como ya he comentado antes K-Means identifica agrupaciones y aunque este lo haga bien (ver figura 6), si hay distintas clases de datos que comparten ciertas características (ver tabla 5) pueden provocar que este algoritmo falle en la predicción como ocurre en este caso.

1.3 DBSCAN

Como parte opcional vamos a aplicar el algoritmo DBSCAN sobre el conjunto original sin reducir dimensiones para ver cómo se comporta. Este algoritmo busca agrupaciones por densidad en el cual podemos ajustar la distancia (épsilon) y el número mínimo de puntos a considerar. Como resultado obtenemos las agrupaciones obtenidas y otro tipo de datos llamado ruido (noise).

En nuestro caso, hemos considerado una distancia de 0,4 y un mínimo de 3 puntos para el algoritmo. De esta manera conseguimos tres clústers y otro adicional de ruido.

Cluster	Class	Cantidad
0	**1**	**3**
1	**1**	**51**
	2	32
2	**3**	**36**
Noise	1	5
	2	**39**
	3	12

Tabla 7. Distribución de las clases en los clústers usando DBSCAN.

En la tabla 7 podemos ver la distribución de clases, hemos incluido también el “ruido” para ver que identifica como tal.

Y tenemos un gran problema, aparte de que considera muchos ejemplos como ruido, ningún clúster identifica bien la clase dos. Los casos de esta clase se reparten entre el clúster uno y el ruido, por lo que a la hora de predecir no tendremos ningún ejemplo de esta clase, a no ser que consideremos el ruido como la clase dos al ser la clase mayoritaria.

Para finalizar, según la documentación todos los puntos que detecta el algoritmo DBSCAN como ruido representan a los datos anómalos. En este caso hay bastantes, quizá probando con otra configuración se puede refinar mejor estos datos.

2. NBA

Para la realización de este ejercicio hemos borrado manualmente la primera fila y columna del archivo de lectura ya que no me era posible hacerlo desde Knime. La primera fila no tenía valores y la primera columna solo contenía valores perdidos.

Como el ejercicio es de libre elección he decidido seleccionar las características correspondientes a los rankings ofensivos y defensivos para seleccionar a los jugadores que destaquen en estos. En este caso no vamos a eliminar los outliers porque se corresponden con los jugadores que queremos destacar.

2.1. TRATAMIENTO DE DATOS

En la figura 7 hacemos el preprocesado de los datos. Primero filtramos las características con un 90% de valores perdidos, después los tratamos de la siguiente manera:

- String → el más frecuente.
- Double → media.
- Integer → media redondeada.

Para acabar seleccionando las dos columnas que nos interesan y renombrándolas a: *Offensive Rating* y *Defensive Rating*.

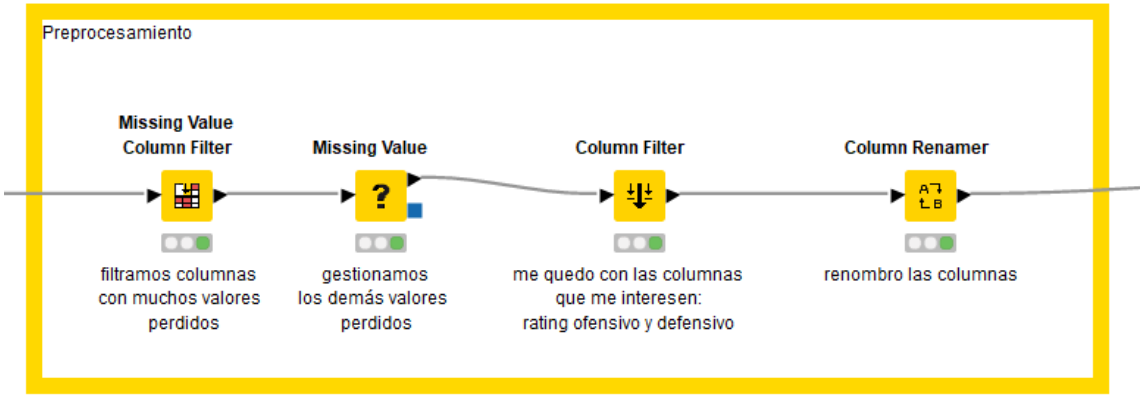


Figura 7. Tratamiento de datos

2.2. AGRUPACIONES

Antes de realizar el algoritmo de clustering, hemos realizado una selección de datos del conjunto quedándonos con los datos anómalos o este caso destacados.

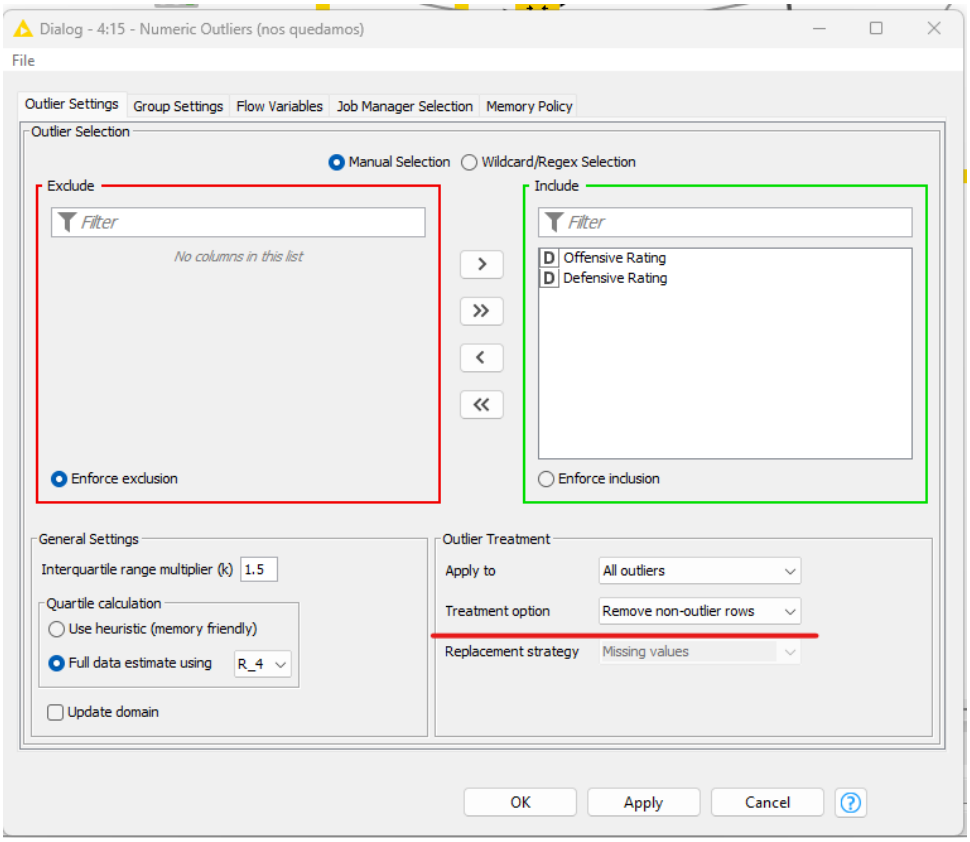


Figura 8. Configuración outliers

Del nodo *Numeric Outliers*, si tocamos la opción destacada en rojo de la figura 8 y seleccionamos "*Remove non-outliers rows",* eliminaremos los casos normales y nos quedaremos con los casos que destacan. Como he planteado antes, estos son los que nos interesan para nuestro problema.

Haciendo una vista de distancia como el de la figura 9, podemos ver que el número adecuado de grupos son de cuatro a ocho. Usaremos cuatro para que sea más simple de interpretar el resultado de las gráficas de dispersión.

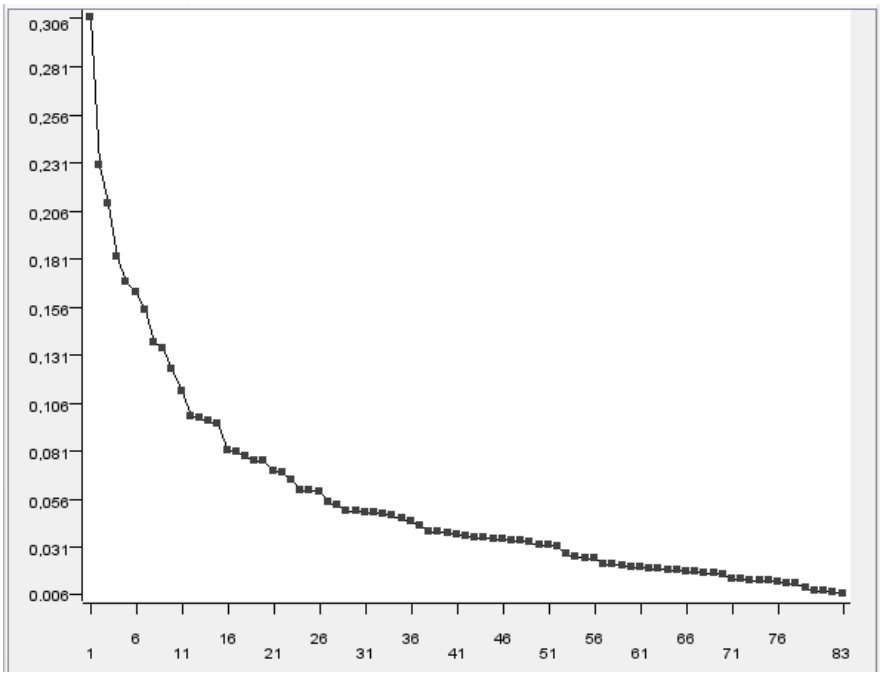


Figura 9. Vista de distancia del problema

Figura 10. Gráfica de dispersión.

De la figura 10 nos vamos a quedar con los clústers uno y dos porque son los que contienen valores más altos para las características que nos interesan.

2.3. SELECCIÓN DE DATOS

En esta sección haremos las selecciones de los jugadores que destacan en el ámbito ofensivo (ver figura 11 y 12), en defensivo (ver figura 13 y 14) y en ambos (ver figura 15 y 16). Los resultados serán almacenados en una tabla al final de la sección (tabla 8).

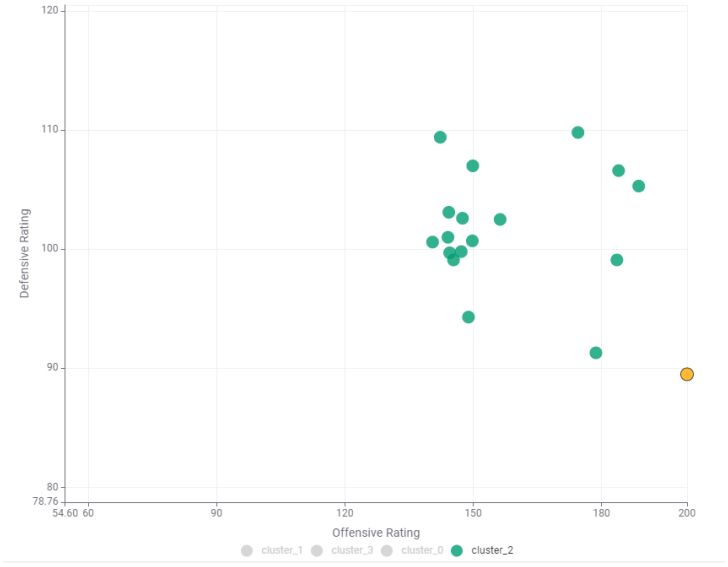
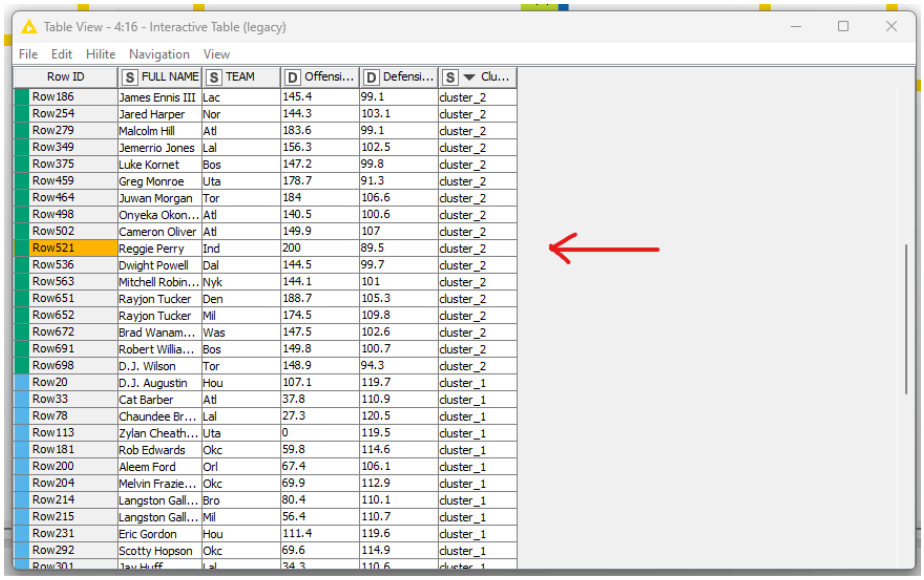


Figura 11. Selección del mejor jugador ofensivo.



Row ID	FULL NAME	TEAM	Offensi...	Defensi...	Clu...
Row186	James Ennis III	Lac	145.4	99.1	cluster_2
Row254	Jared Harper	Nor	144.3	103.1	cluster_2
Row279	Malcolm Hill	Atl	183.6	99.1	cluster_2
Row349	Jemerrio Jones	Lal	156.3	102.5	cluster_2
Row375	Luke Kornet	Bos	147.2	99.8	cluster_2
Row459	Greg Monroe	Uta	178.7	91.3	cluster_2
Row464	Juwan Morgan	Tor	184	106.6	cluster_2
Row498	Onyeka Okon...	Atl	140.5	100.6	cluster_2
Row502	Cameron Oliver	Atl	149.9	107	cluster_2
Row521	Reggie Perry	Ind	200	89.5	cluster_2
Row536	Dwight Powell	Dal	144.5	99.7	cluster_2
Row563	Mitchell Robin...	Nyk	144.1	101	cluster_2
Row651	Rayjon Tucker	Den	188.7	105.3	cluster_2
Row652	Rayjon Tucker	Mil	174.5	109.8	cluster_2
Row672	Brad Wanam...	Was	147.5	102.6	cluster_2
Row691	Robert Willia...	Bos	149.8	100.7	cluster_2
Row698	D.J. Wilson	Tor	148.9	94.3	cluster_2
Row20	D.J. Augustin	Hou	107.1	119.7	cluster_1
Row33	Cat Barber	Atl	37.8	110.9	cluster_1
Row78	Chaundee Br...	Lal	27.3	120.5	cluster_1
Row113	Zylan Cheath...	Uta	0	119.5	cluster_1
Row181	Rob Edwards	Okc	59.8	114.6	cluster_1
Row200	Aleem Ford	Orl	67.4	106.1	cluster_1
Row204	Melvin Frazie...	Okc	69.9	112.9	cluster_1
Row214	Langston Gall...	Bro	80.4	110.1	cluster_1
Row215	Langston Gall...	Mil	56.4	110.7	cluster_1
Row231	Eric Gordon	Hou	111.4	119.6	cluster_1
Row292	Scotty Hopson	Okc	69.6	114.9	cluster_1
Row301	Jay Huff	Lal	34.3	110.6	cluster_1

Figura 12. Nombre del mejor jugador ofensivo.

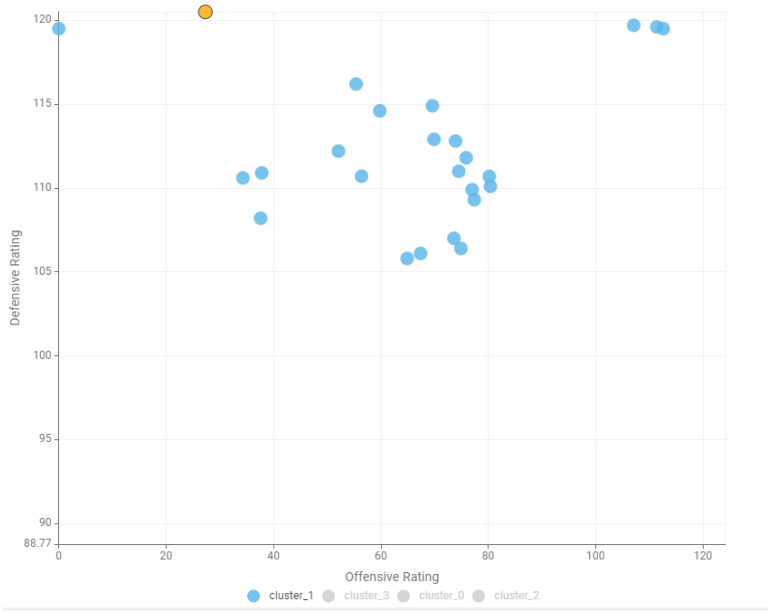


Figura 13. Selección del mejor jugador defensivo.

Row ID	FULL NAME	TEAM	Offensi...	Defensi...	Clu...
Row375	Luke Kornet	Bos	147.2	99.8	cluster_2
Row459	Greg Monroe	Uta	178.7	91.3	cluster_2
Row464	Juwan Morgan	Tor	184	106.6	cluster_2
Row498	Onyeka Okongwu	Atl	140.5	100.6	cluster_2
Row502	Cameron Oliver	Atl	149.9	107	cluster_2
Row521	Reggie Perry	Ind	200	89.5	cluster_2
Row536	Dwight Powell	Dal	144.5	99.7	cluster_2
Row563	Mitchell Robinson	Nyk	144.1	101	cluster_2
Row651	Rayjon Tucker	Den	188.7	105.3	cluster_2
Row652	Rayjon Tucker	Mil	174.5	109.8	cluster_2
Row672	Brad Wanamaker	Was	147.5	102.6	cluster_2
Row691	Robert Williams III	Bos	149.8	100.7	cluster_2
Row698	D.J. Wilson	Tor	148.9	94.3	cluster_2
Row20	D.J. Augustin	Hou	107.1	119.7	cluster_1
Row33	Cat Barber	Atl	37.8	110.9	cluster_1
Row78	Chaundee Brown Jr.	Lal	27.3	120.5	cluster_1
Row113	Zylan Cheatham	Uta	0	119.5	cluster_1
Row181	Rob Edwards	Okc	59.8	114.6	cluster_1
Row200	Aleem Ford	Orl	67.4	106.1	cluster_1
Row204	Melvin Frazier Jr.	Okc	69.9	112.9	cluster_1
Row214	Langston Galloway	Bro	80.4	110.1	cluster_1
Row215	Langston Galloway	Mil	56.4	110.7	cluster_1
Row231	Eric Gordon	Hou	111.4	119.6	cluster_1
Row292	Scotty Hopson	Okc	69.6	114.9	cluster_1
Row301	Jay Huff	Lal	34.3	110.6	cluster_1
Row303	Elijah Hughes	Por	80.2	110.7	cluster_1
Row329	Alize Johnson	Was	73.9	112.8	cluster_1
Row345	Carlik Jones	Dal	52.1	112.2	cluster_1
Row395	Damian Lillard	Por	112.6	119.5	cluster_1

Figura 14. Nombre del mejor jugador defensivo.

Figura 15. Selección de los jugadores que destacan en ambas características.

Row563	Mitchell Robinson	Nyk	144.1	101	cluster_2	
Row651	Rayjon Tucker	Den	188.7	105.3	cluster_2	
Row652	Rayjon Tucker	Mil	174.5	109.8	cluster_2	
Row672	Brad Wanamaker	Was	147.5	102.6	cluster_2	
Row691	Robert Williams III	Bos	149.8	100.7	cluster_2	
Row698	D.J. Wilson	Tor	148.9	94.3	cluster_2	
Row20	D.J. Augustin	Hou	107.1	119.7	cluster_1	←
Row33	Cat Barber	Atl	37.8	110.9	cluster_1	
Row78	Chaundee Brown Jr.	Lal	27.3	120.5	cluster_1	
Row113	Zylan Cheatham	Uta	0	119.5	cluster_1	
Row181	Rob Edwards	Okc	59.8	114.6	cluster_1	
Row200	Aleem Ford	Orl	67.4	106.1	cluster_1	
Row204	Melvin Frazier Jr.	Okc	69.9	112.9	cluster_1	
Row214	Langston Galloway	Bro	80.4	110.1	cluster_1	
Row215	Langston Galloway	Mil	56.4	110.7	cluster_1	
Row231	Eric Gordon	Hou	111.4	119.6	cluster_1	←
Row292	Scotty Hopson	Okc	69.6	114.9	cluster_1	
Row301	Jay Huff	Lal	34.3	110.6	cluster_1	
Row303	Elijah Hughes	Por	80.2	110.7	cluster_1	
Row329	Alize Johnson	Was	73.9	112.8	cluster_1	
Row345	Carlik Jones	Dal	52.1	112.2	cluster_1	
Row395	Damian Lillard	Por	112.6	119.5	cluster_1	←
Row471	Mychal Mulder	Orl	75.9	111.8	cluster_1	
Row561	Justin Robinson	Sac	37.6	108.2	cluster_1	
Row572	Matt Ryan	Bos	64.9	105.8	cluster_1	
Row579	Jordan Schakel	Was	55.4	116.2	cluster_1	
Row598	Javonte Smart	Mil	77.4	109.3	cluster_1	
Row645	Isaiah Todd	Was	74.5	111	cluster_1	

Figura 16. Nombre de los jugadores que destacan en ambas características.

Mejor Jugador	Jugador	Equipo
Defensivo	Chaundee Brown Jr.	Los Angeles Lakers
Ofensivo	Reggie Perry	Indiana Pacers
Ofensivo y defensivo	D.J. Augustin	Houston Rockets
	Eric Gordon	Houston Rockets
	Damian Lillard	Portland Trail Blazers

Tabla 8. Resultados finales.