K-means, Hierarchical y PAM

Carvajal Florencia Carlos Sebastián, Espín Carrasco Luis Miguel, Romero Castro Víctor Francisco, y Tocta Bonilla Tyrone Wladimir.
Universidad Técnica Estatal de Quevedo
Facultad de Ciencias de la Ingeniería Quevedo, Ecuador
carlos.carvajal2015@uteq.edu.ec , Luis.espin2015@uteq.edu.ec,
victor.romero2016@uteq.edu.ec, tyrone.tocta2016@uteq.edu.ec

Resumen – Los algoritmos de clasificación no supervisada representan una de las técnicas más ampliamente usadas en análisis de datos, con aplicaciones en estadística, biología, ciencias sociales, psicología, etc. En prácticamente cada campo científico que trate con datos empíricos, los humanos a menudo intentan obtener una primera impresión sobre los datos tratando de identificar grupos de "comportamiento similar" en esos datos.

Utilizando los términos de clasificación no supervisada, en el siguiente trabajo se realizará un análisis entre los tres algoritmos de clasificación, que permiten agrupar objetos de un conjunto de datos, sobre los cuales se miden diferentes variables o características. Así, objetos que presenten características muy similares deberán quedar agrupados en conjuntos que llamaremos agrupamientos.

Palabras Claves: clasificación, particiones, K-means, Hierarchical, PAM, patrones, clustering.

I. INTRODUCCIÓN

El término clustering hace referencia a un amplio abanico de técnicas no supervisadas cuya finalidad es encontrar patrones o grupos (clusters) dentro de un conjunto de observaciones. Las particiones se establecen de forma que, las observaciones que están dentro de un mismo grupo, son similares entre ellas y distintas a las observaciones de otros grupos[1].

Se trata de un método no supervisado, ya que el proceso ignora la variable respuesta que indica a que grupo pertenece realmente cada observación (si es que existe tal variable). Esta característica diferencia al clustering de las técnicas supervisadas, que emplean un set de entrenamiento en el que se conoce la verdadera clasificación[1] [2].

II. REVISIÓN LITERARIA

A. K-means

Los algoritmos de clustering son considerados de aprendizaje no supervisado. Este tipo de algoritmos de aprendizaje no supervisado busca patrones en los datos sin tener una predicción específica como objetivo (no hay variable dependiente)[3]. En lugar de tener una salida, los datos solo tienen una entrada que serían las múltiples variables que describen los datos[3].

K-means necesita como dato de entrada el número de grupos en los que vamos a segmentar la población. A partir de este número k de clusters, el algoritmo coloca primero k puntos aleatorios (centroides). Luego asigna a cualquiera de esos puntos todas las muestras con las distancias más pequeñas[4].

Esto generará una nueva asignación de muestras, ya que algunas muestras están ahora más cerca de otro centroide. Este proceso se repite de forma iterativa y los grupos se van ajustando hasta que la asignación no cambia más moviendo los puntos[4].

Descripción: El algoritmo divide los datos en k clusters. Su funcionamiento se basa en formar los clusters de forma que la varianza interna de cada uno sea mínima, para ello utiliza una medida virtual de la media de las observaciones del cluster [5].

¿Por qué es necesario crear grupos?

El clustering consiste en la clasificación de las muestras o de los genes según su nivel de similitud y nos permite encontrar relaciones desconocidas o verifican nuestras hipótesis [3].

¿Cuál es el objetivo del método jerárquico?

Estos métodos tienen por objetivo agrupar clusters para formar uno nuevo o bien separar alguno ya existente para dar origen a otros dos, de tal forma que se minimice alguna función distancia o bien se maximice alguna medida de similitud [6].

¿Cuál es el objetivo del método no jerárquico?

Tienen por objetivo realizar una sola partición de los individuos en K grupos [7].

¿Cuándo podemos usar K-means?

Este tipo de algoritmo de aprendizaje no supervisado es útil para explorar, describir y resumir datos de una forma distinta. Utilizar este agrupamiento de datos nos puede servir para confirmar (o rechazar) algún tipo de clasificación previa. También nos puede ayudar a descubrir patrones y relaciones que desconocíamos [6].
Por ejemplo, podemos aplicar K-means en [8]:

- Segmentación de clientes
- Agrupación de textos que hablan de temas similares
- Geoestadística
- Comunidades de redes sociales

¿Cuál es la diferencia entre K-means y Kmedoids (PAM)?

A diferencia del algoritmo K-means, en el que se minimiza la suma total de cuadrados intra-cluster (suma de las distancias al cuadrado de cada observación respecto a su centroide), el algoritmo PAM minimiza la suma de las diferencias de cada observación respecto a su medoid [8] [9].

B. K-medoids clustering (PAM)

K-medoids es un método de clustering muy similar a K-means en cuanto a que ambos agrupan las observaciones en K clusters, donde K es un valor preestablecido por el analista. La diferencia es que, en K-medoids, cada cluster está representado por una observación presente en el cluster (medoid), mientras que en K-means cada cluster está representado por su centroide, que se corresponde con el promedio de todas las observaciones del cluster pero con ninguna en particular[10].

Una definición más exacta del término medoids es: elemento dentro de un cluster cuya distancia (diferencia) promedio entre él y todos los demás elementos del mismo cluster en lo menor posible. Se corresponde con el elemento más central del cluster y por lo tanto puede considerarse como el más representativo[11].

El algoritmo más empleado para aplicar K-medoids se conoce como PAM (Partitioning Around Medoids) y sigue los siguientes pasos [12]:

- Seleccionar K observaciones aleatorias como medoids iniciales. También es posible identificarlas de forma específica.
- Calcular la matriz de distancia entre todas las observaciones si esta no se ha calculado anteriormente.
- Asignar cada observación a su medoid más cercano.
- Para cada uno de los clusters creados, comprobar si seleccionando otra observación como medoid se consigue reducir la distancia promedio del cluster, si esto ocurre, seleccionar la observación que consigue una mayor reducción como nuevo medoid.

Descripción: Este algoritmo es similar al k-means, con la diferencia de que en este la medida a partir de la cual se desarrolla cada cluster es la mediana de las observaciones del cluster (medoid) [8].

C. Hierarchical

Este algoritmo tiene una variedad de objetivos relacionados con agrupar o segmentar una colección de objetos, es decir, observaciones, individuos, casos o filas de datos, en subconjuntos o clústeres, de modo que los datos que están dentro de cada grupo están más estrechamente relacionados con unos a otros que los objetos asignados a diferentes grupos [7].

En la agrupación jerárquica, los datos no se particionan en un clúster en particular en un solo paso. En su lugar, tiene lugar una serie de particiones, que pueden ejecutarse desde un único clúster que contiene todos los objetos, hasta n clústeres que contienen un solo objeto. Hierarchical Clustering o agrupación jerárquica se subdivide en métodos aglomerativos, que proceden de una serie de fusiones de los n objetos en grupos[1] [7].

Estrategias de agrupación

Aglomerativo: Cada observación comienza en su propio grupo y los pares de grupos se fusionan a medida que se asciende en la jerarquía.

Divisivo: Todas las observaciones comienzan en un grupo y las divisiones se realizan de forma recursiva a medida que uno se mueve hacia abajo en la jerarquía Agrupación jerárquica.

Descripción: En este algoritmo los datos se van agrupando en clusters formados por parejas cuya similitud es más alta hasta que solo hay un cluster que reúne a todos los demás. Es necesario elegir un método de comparación entre los clusters [7].

III. METODOLOGÍA / PROCEDIMIENTO

Este presente trabajo se incluye una herramienta que es Rstudio que nos permite analizar el comportamiento de los datos que se genera en el desarrollo del ejercicio. Para ello se implementa tres métodos de clasificación no supervisada.

Proceso de desarrollo

Método: El resultado principal es una metodología para determinar el mejor algoritmo de agrupamiento para la manipulación de datos.

Métodos de agrupamiento: Se indican los algoritmos elegidos para realizar el proyecto, justificando su elección. También se describe el funcionamiento de cada uno de ellos.

Tratamiento y análisis de los datos: Se describe la carga y procesado de los datos. Se plantea un análisis sobre la estructura de estos.

Implementación de los métodos: En esta sección se realiza el proceso de selección de k y la implementación de los algoritmos elegidos.

Análisis comparativo de los métodos: Se realiza una comparación de los algoritmos seleccionados para determinar cuál es el mejor trabajando con los datos**.**

IV. RESULTADOS

```
#librerias globales utilizadas
library(tidyverse)
library(factoextra)
library(ggplot2)
library(ggfortify)
library(NbClust)
library(fpc)
library(cluster)
library(ggdendro)

#cargar el conjunto de datos y
estandarizarla para los tres modelos
ruta= choose.files()
datos<-read.table(ruta,header = T, sep =
';')
View(datos)
datos <- data.frame(datos, row.names =
datos[,1]); datos[,1] <- NULL

mydata <- scale(datos)
head(mydata)
```

en la Tabla 1 Datos estandarizados o normalizados se presenta los datos normalizados que se va a utilizar en este ejercicio.

Tabla 1 Datos estandarizados o normalizados

1039	-1.5773842506	-1.638756519
1003	0.8342765678	0.823295200
1012	-0.7936807112	-0.787989082
1004	0.0007532434	-0.065984344
1040	0.8112828209	0.865697418
1001	-0.1084670542	-0.059760165
1028	0.6859669004	0.648629183
1013	-0.9170804861	-0.917529803
1027	-0.1008024719	-0.141452511
1016	0.4127245417	0.380211473
1019	-0.0574975820	-0.008799701
1031	-0.5533960563	-0.514125216
1020	-1.0006244331	-1.085193619
1029	1.9854968281	1.969711129
1033	-0.4326788852	-0.432043858
1030	0.4889871355	0.503139004
1035	0.3874314201	0.408998300
1036	1.8996535064	1.966599040
1023	-0.3978050358	-0.352296568
1034	-1.3731231325	-1.372672876
1038	-0.5530128272	-0.511402138
1010	-1.3183213691	-1.285923385
1017	-1.4597329124	-1.463701491
1009	-0.5587612639	-0.518793350
1041	-0.2996983824	-0.255043774
1006	0.2027149868	0.232387227
1022	0.5901596217	0.509363182
1032	1.1282132987	1.048921681
1021	2.0744059827	2.054515565

```
#calcular la matriz de distancias
 m.distancia <- get_dist(datos, method =
"euclidean")
 fviz_dist(m.distancia, gradient = list(
low = "blue", mid = "white", high ="red"))
```

En la **Ilustración 1** nos indica una matriz de distancia entre las filas de una matriz de datos. Comparado con la función dist() estándar, soporta medidas de distancia basadas en la correlación.

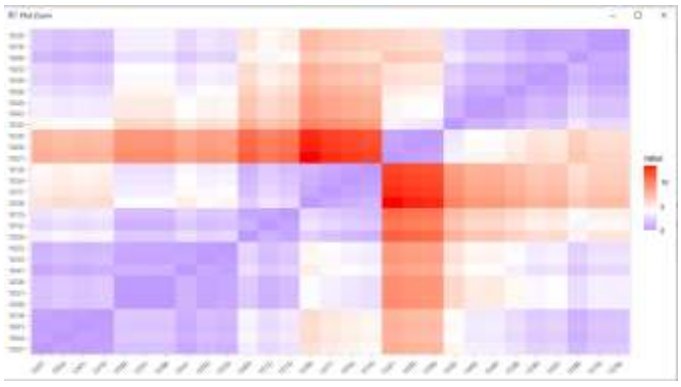

Ilustración 1 matriz de distancia con k-means

```
#estimar el número de clústers
 fviz_nbclust(mydata, kmeans, method =
"wss")
```

En la **Ilustración 2** muestran los enfoques de agrupamiento que dividen los conjuntos de datos, que contienen n observaciones, en un conjunto de k grupos. En esta parte se utiliza un método para estimar el número de clúster óptimo.

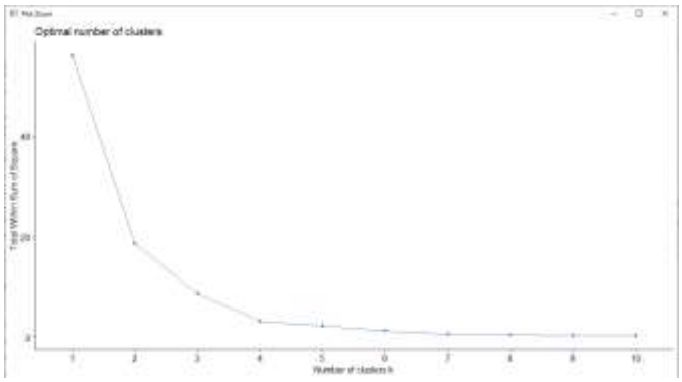

Ilustración 2 estimación del número de clústers

```
resnumclust<-NbClust(mydata, distance =
"euclidean", min.nc=2, max.nc=10, method =
"kmeans", index = "alllong")

fviz_nbclust(resnumclust)
```

En la **Ilustración 3** número de clúster más optimo según varios métodos determina y visualiza el número óptimo de agrupaciones utilizando diferentes métodos de agrupaciones, dando como resultado 4 clúster.

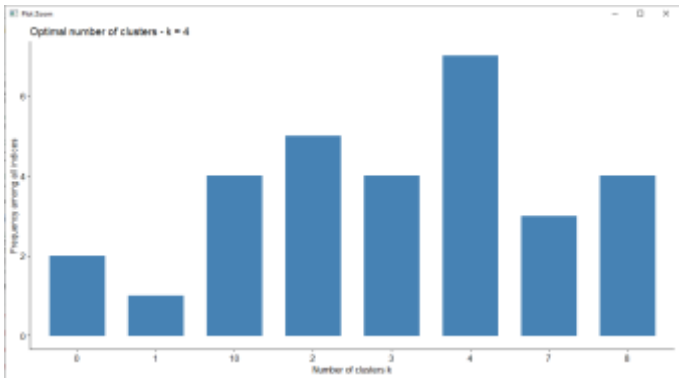

Ilustración 3 número de clúster más optimo según varios métodos

Metodo de agrupación K-Means

```
#calculamos los cuatro clústers
 clustkm <- kmeans(datos,4)
 clustkm$cluster

fviz_cluster(clustkm, data = mydata)
```

En la **Ilustración 4** nos indica las cantidades de agrupaciones mediante K-means con colores para una mejor representación.

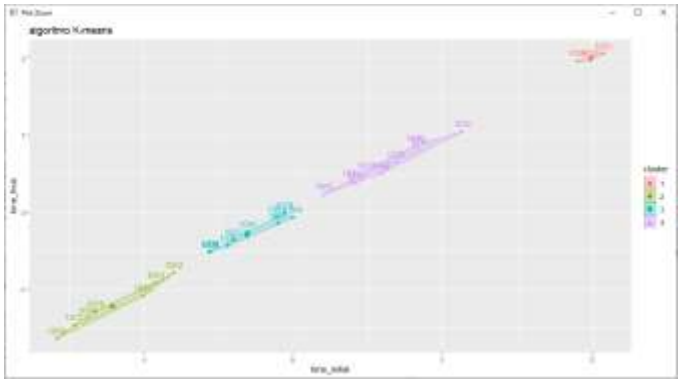

Ilustración 4 datos agrupados mediante K-means

Metodo de agrupación PAM

```
#calculamos los 4 clústers
pam4 <- pam(mydata, 4)
print(pam4)

#probamos algunas visualizaciones
fviz_cluster(pam4, data = mydata)
```

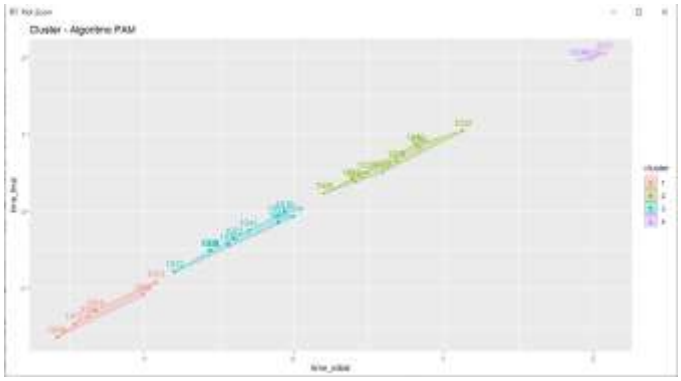

Ilustración 5 datos agrupados con el algoritmo PAM

```
res4 <- hcut(mydata, k = 4, stand = TRUE,
method = "median")
fviz_dend(res4, rect = TRUE, cex = 1,
          k_colors = "simpsons")
```

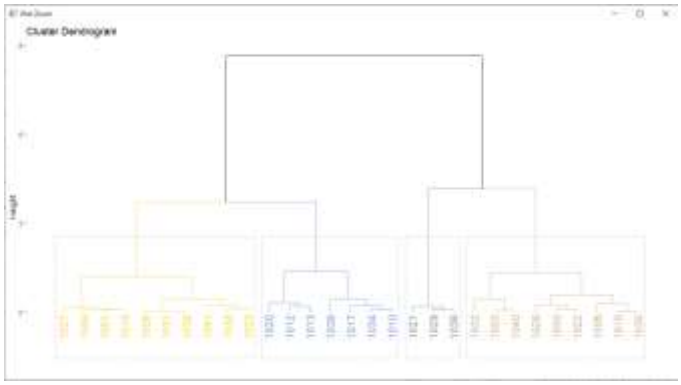

Ilustración 6 Dendograma según el algoritmo K-means

Metodo de agrupación Hierarchical

```
hc.cut <- hcut(datos, k = 4,  hc_method = "complete")
# Visualiar dendrogram
fviz_dend(hc.cut, show_labels = TRUE,rect = TRUE,cex = 1)
```

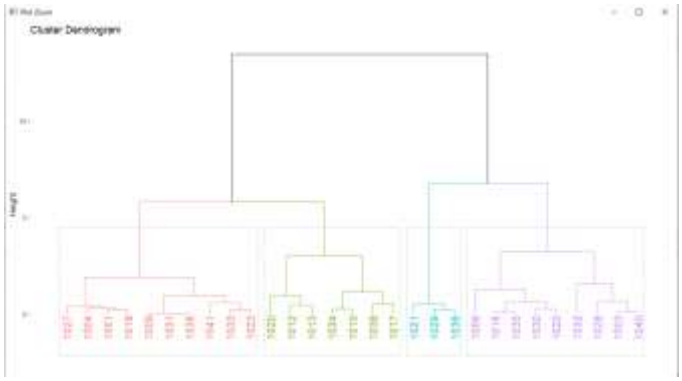

Ilustración 7 Dendograma según el número del clúster del algoritmo Hierarchical

```
# Visualize cluster
fviz_cluster(hc.cut, ellipse.type = "convex")
```

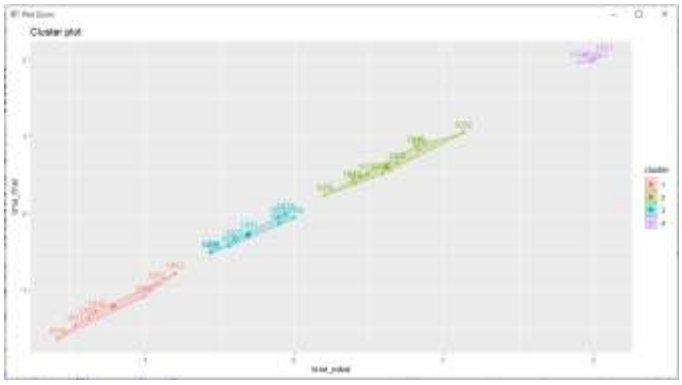

Ilustración 8 visualización de los datos agrupados en Hierarchical

Comparación de Resultados

Realizando un análisis comparativo de los tres métodos de clasificación no supervisada, y utilizando el paquete de clValid, encargado de realizar comparaciones y dando como resultado el mejor método de clasificación. K-means se ha generado partición en 4 grupos, PAM en 4 grupos y por último Hierarchical de 4 agrupaciones. En la **Ilustración 9** comparación de los algoritmos de clasificación no supervisadosIlustración 9 comparación de los algoritmos de clasificación no supervisados se detalla los resultados.

```
Validation Measures:
                                 2       3       4       5       6       7

hierarchical Connectivity   4.2425 11.0917 20.4349 23.1655 28.2437 32.8385
             Dunn           0.1866  0.2490  0.2887  0.2932  0.2932  0.3886
             Silhouette     0.5491  0.4640  0.3671  0.3392  0.3586  0.3555
kmeans       Connectivity   4.2425 12.8905 21.7290 28.2083 32.3655 39.0472
             Dunn           0.1866  0.2128  0.2236  0.2578  0.2578  0.3211
             Silhouette     0.5491  0.4596  0.3915  0.3892  0.3678  0.3587
pam          Connectivity   5.9659 12.8905 21.0821 28.2083 30.5750 33.3056
             Dunn           0.1643  0.2128  0.2044  0.2578  0.2647  0.3000
             Silhouette     0.5439  0.4596  0.4087  0.3892  0.3401  0.3298
```

Ilustración 9 comparación de los algoritmos de clasificación no supervisados

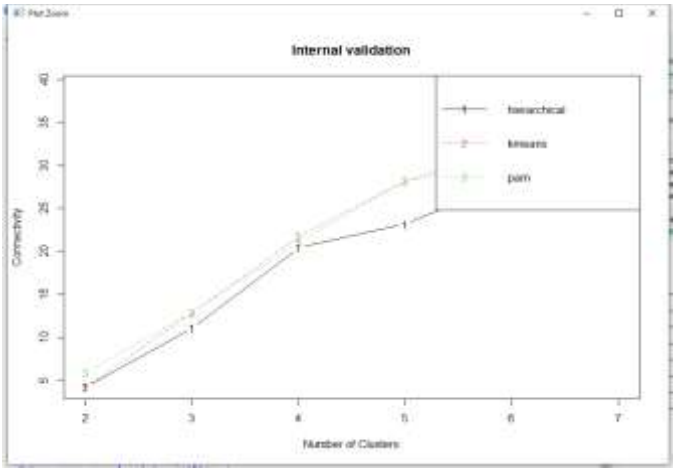

Ilustración 10 comparación de manera gráfica de los algoritmos no supervisados

V. CONCLUSIÓN

Para poder obtener el número óptimo de clúster a utilizar en los diferentes algoritmos de clasificación no supervisada, se procedió a estimar el valor con la ayuda de la función `fviz_nbclust`, haciendo uso solo del método “WSS” da como resultado K=3, mientas que si se utilizas todos los métodos que brinda esa función da como resultado K=4. Por lo tanto, se ha elegido hacer 4 grupos de separación (clúster).

Los algoritmos utilizados para realizar la clasificación no supervisada como K-means, PAM, Hierarchical, nos permite realizar agrupamientos o clústers en un conjunto de datos, cada agrupación nos indica las observaciones que están dentro de un mismo grupo, son similares entre ellas y distintas a las observaciones de otros grupos.

En este trabajo hemos presentado nuevos métodos de clasificación no supervisas y cómo clasifica u organiza sus datos dependiendo de su clasificación de las muestras o de los puntos centrales, según su nivel de similitud que nos permite encontrar relaciones desconocidas o verifican nuestras hipótesis.

VI. ANEXO

Integrantes	Actividades	
	Individual	**Todos**
Carvajal Florencia Carlos Sebastián	Análisis y estructuración del documento y verificación de resultados en Rstudio de los tres modelos.	Redacción de la documentación y Conclusiones
Espín Carrasco Luis Miguel	Investigación de clasificación no supervisada, Hierarchical y el desarrollo en Rstudio,	
Romero Castro Víctor Francisco	Investigación de clasificación no supervisada PAM y el desarrollo en Rstudio	
Tocta Bonilla Tyrone Wladimir	Investigación de clasificación no supervisada K-means y desarrollo en Rstudio	

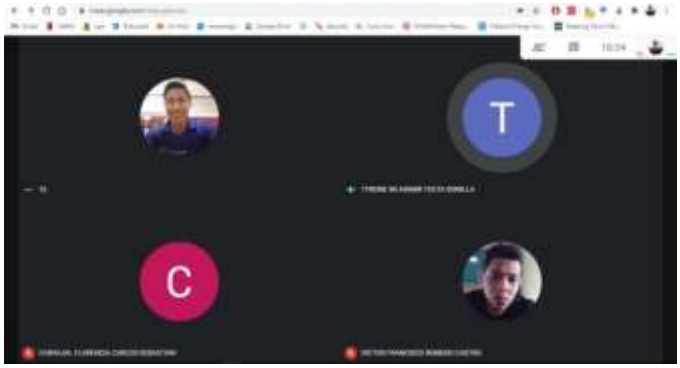

Ilustración 11 Captura de participantes

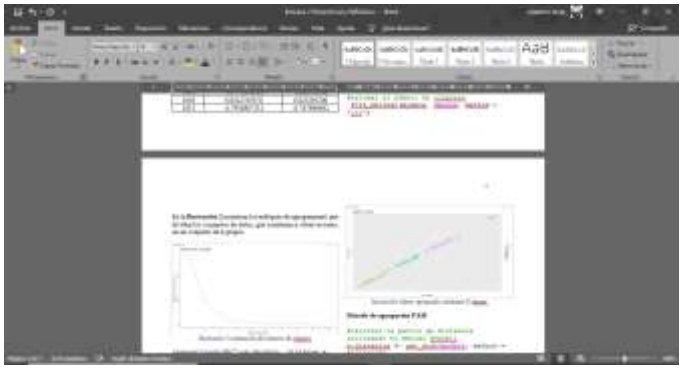

Ilustración 12 Desarrollo de la documentación compartido en One Drive

VII. REFERENCIAS

[1] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, Sep. 1967, doi: 10.1007/BF02289588.

[2] C. Romesburg, "Cluster analysis for researchers," 2004, Accessed: Feb. 24, 2021. [Online]. Available: https://books.google.es/books?hl=es&lr=&id=ZuIPv7OKm10C&oi=fnd&pg=PR5&dq=cluster+analysis&ots=7GQG3c8V89&sig=zSsVs4RF9eEiIB6Q5hG4nrJMpVc.

[3] A. Likas, N. Vlassis, and J. Verbeek, "The global k-means clustering algorithm The global k-means clustering algorithm. [Technical."

[4] P. S. Bradley, K. P. Bennett, and A. Demiriz, "Constrained K-Means Clustering," 2000.

[5] J. T. L. Wang, M. J. Zaki, H. T. T. Toivonen, and D. Shasha, "Introduction to Data Mining in Bioinformatics," in *Data Mining in Bioinformatics*, Springer-Verlag, 2005, pp. 3–8.

[6] A. Serra, "Comparación de algoritmos de clasificación supervisada MEMÒRIA Autor," Universitat Politècnica de Catalunya, Jul. 2020. Accessed: Feb. 24, 2021. [Online]. Available: https://upcommons.upc.edu/handle/2117/330482.

[7] C. C. Bridges, "Hierarchical Cluster Analysis," *Psychol. Rep.*, vol. 18, no. 3, pp. 851–854, Jun. 1966, doi: 10.2466/pr0.1966.18.3.851.

[8] K. T.- Medicine and undefined 2006, "K-means clustering tutorial," *sigitwidiyanto.staff.gunadarma.ac.id*, Accessed: Feb. 21, 2021. [Online]. Available: http://sigitwidiyanto.staff.gunadarma.ac.id/Downloads/files/38034/M8-Note-kMeans.pdf.

[9] E. Schubert and P. J. Rousseeuw, "Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms," *Springer*, vol. 11807 LNCS, pp. 171–187, 2019, doi: 10.1007/978-3-030-32047-8_16.

[10] F. De Economía, Y. Planificación, R. Miguel, and E. Vega, “UNIVERSIDAD NACIONAL AGRARIA LA MOLINA EL ALGORITMO PARTICIÓN ALREDEDOR DE MEDOIDES (PAM) CON DATOS MIXTOS" PRESENTADO POR,” 2018.

[11] C. B. Lucasius, A. D. Dane, and G. Kateman, “On k-medoid clustering of large data sets with the aid of a genetic algorithm: background, feasiblity and comparison,” Elsevier, Oct. 1993. doi: 10.1016/0003-2670(93)80130-D.

[12] A. P. Reynolds, G. Richards, and V. J. Rayward-Smith, “The Application of K-medoids and PAM to the Clustering of Rules.” Accessed: Feb. 21, 2021. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-28651-6_25.