

Práctica 3: Aplicaciones de agrupamiento de texto (k-means)

Castro Elvira D. 2022710039, Pineda Rugerio N. 20222710240, Castaño Hernández V. J. 2022710020,
Galicia Cocoletzi N. 2022710234, Sánchez Zanjuampa M. A. 2022710029, y Nava Méndez E. U. 2021710144.

Profesor: Lauro Reyes Cocoletzi

Resumen—La clasificación de los textos es una de las tareas más representativas de los algoritmos de aprendizaje automático, los cuales están relacionados con la inteligencia artificial. Uno de los usos más populares sobre la clasificación de textos es el análisis de tendencias y búsqueda de la verdad, de modo que tener una buena organización de los corpus como archivos de texto, publicaciones en redes sociales o foros de discusión resulta útil para poder aplicar distintos análisis exhaustivos de la información presentada.

Palabras clave—K-Means, Clasificación, Algoritmos no supervisados, Coeficiente Jaccard, Clústers, Análisis de textos.

I. INTRODUCCIÓN

I-A. Marco teórico

I-A1. ¿Qué es el algoritmo K-means?:

El algoritmo de K-means (o K-medias) es un algoritmo clasificador el cual se encarga de agrupar elementos o instancias de una base de datos de acuerdo a una clase a la que pertenezcan. Es un algoritmo no supervisado, por lo que el programador no debe tener mucha intervención más que para hacer unos ajustes de parámetros como el número de clases en las que se busca clasificar o el número de Iteraciones que deben suceder antes de detener el algoritmo.

Utiliza datos no clasificados haciendo un análisis de la similaridad de sus características de cada instancia con el fin de que los elementos que se encuentren dentro de un conjunto (clúster) tengan características similares.

I-A2. Distancia de Jaccard:

La distancia Jaccard es un coeficiente que se encarga de determinar la similitud entre 2 conjuntos, esto nos habla sobre agrupaciones de datos y que nos recuerdan a los diagramas de Venn. Los conjuntos a analizarse pueden tener elementos similares y es aquí donde la distancia Jaccard hace un cálculo para obtener la similitud de estos. Utiliza una escala que va desde 0 (los conjuntos son totalmente distintos), hasta 1 (que significa que los conjuntos son idénticos).

I-A3. Health News in Twitter:

Health news in twitter (noticias de salud en twitter) es una base de datos la cual ha recopilado los tweets (tuits) de las

agencias más grandes de salud registradas en la plataforma Twitter (ahora X) durante el periodo de Enero y Diciembre del año 2015. Ésta contiene información en forma de archivos de texto en donde se encuentran los tweets de las 15 agencias más importantes de salud en donde se encuentran características (features) como la cuenta, fecha y hora del tweet y la información en forma de texto. Esta base de datos ha sido utilizada para fines de evaluación de rendimiento de modelos que analizan textos cortos. Aunque también puede ser utilizado para el agrupamiento (clustering) de datos como es el caso de esta práctica.

I-B. Objetivos

- Analizar el funcionamiento de la distancia de Jaccard
- Observar el comportamiento de oraciones que tienen o no relación entre sí

I-C. Aporte

- Utilizar el algoritmo de k means para agrupar los tweets del dataset Health News in Twitter, en clusters basados en la similitud de sus contenidos, identificando graficamente los clusters generados.
- Utilizar la distancia de Jaccard para calcular la similitud entre los perfiles de usuario en función de los hashtags, palabras clave o temas de salud que utilizan en sus tweets.
- Esto podría proporcionar información acerca de las preocupaciones de salud que ahí se abordan y los temas que están generando atención en una plataforma como lo es Twitter.

II. DESARROLLO

II-A. Análisis del código original

El código se compone de 9 funciones:

II-A1. Preprocesamiento de texto: Antes de hacer todo el proceso de clasificación, es necesario realizar un preprocesamiento de los datos; esto quiere decir que se deben eliminar los caracteres especiales (#, @, :), las URL, los espacios en blanco extra y los saltos de línea. Posteriormente se debe realizar una normalización del texto, esto significa volver todo

a minúsculas para que no haya ninguna palabra diferente que pudiera interferir con la clasificación.

II-A2. Algoritmo k-medias: La siguiente función dentro del programa corresponde al algoritmo de k-medias, este es muy conocido y sigue los siguientes pasos:

1. Inicialización: Selecciona K centroides iniciales para representar los centroides de los clusters. Esto se puede hacer de manera aleatoria, eligiendo K puntos al azar de los datos como centroides iniciales.
2. Asignación de clusters: Asigna cada punto de datos al centroide más cercano. Esto se hace calculando la distancia entre cada punto de datos y todos los centroides, y asignando el punto de datos al centroide con la distancia más corta. Esto forma K clusters iniciales.
3. Actualizar centroides: Una vez que todos los puntos de datos se han asignado a un cluster, recalcula la posición de cada centroide. Esto se hace tomando la media de todas las características de los puntos de datos asignados al cluster. Los centroides se actualizan para que representen el centro de gravedad de los puntos de datos en cada cluster.
4. Reasignación de clusters: Después de actualizar los centroides, vuelve a asignar cada punto de datos al centroide más cercano. Esto puede cambiar la composición de los clusters.
5. Convergencia: Repite los pasos 3 y 4 hasta que los centroides ya no cambien significativamente o hasta que se alcance un número máximo de iteraciones.
6. Resultado final: Una vez que el algoritmo converge, se obtienen los clústers finales. Cada punto de datos está asignado a un clúster y se pueden utilizar los centroides finales como representantes de los clústers.

II-A3. Convergencia: Ligado al código anterior, se tiene la comprobación de la convergencia. Contiene dos condiciones: una para verificar si los puntos en los clústers son los mismos y otra para verificar los centroides nuevos y los anteriores.

La función devuelve verdadero cuando alguno de los dos no cambie.

II-A4. Asignación de clúster: Esta función asigna cada tweet preprocessado anteriormente a un clúster de datos, aquí se eligen aleatoriamente los centros de entre los tweets para luego iterar sobre toda la lista de estos y hacer el cálculo de las distancias para luego hacer la asignación de dicho tweet en el cluster más cercano.

Se tiene una condición especial, en la que si la distancia es 1 (los tweets son diferentes) simplemente se asigna aleatoriamente a un clúster.

Continuando con el proceso, cada que un tweet se asigna a un clúster este se guarda en un diccionario y también se guarda la distancia que tuvo.

II-A5. Actualización de centroides: En caso de el programa no haya convergido, cada iteración se debe realizar una

actualización de los centroides. Se recibe como parámetro la tupla de valores que se obtuvieron de la función anterior.

El proceso para realizar la actualización de los centroides es bastante simple: se debe sacar la media de todas las características asociadas al clúster y esta se asigna como nuevo centroide.

II-A6. Obtener la distancia: Esta función simplemente obtiene la distancia de Jaccard entre los puntos y el centroide actual.

II-A7. Suma de errores cuadráticos: De la misma manera, esta función simplemente obtiene la suma de los cuadrados de las distancias de todos los tweets y los guarda en una variable.

II-A8. Visualización de los clúster: Esta función se implementó para tener una comprensión más visual de lo que hace el programa, se utiliza PCA para reducir las dimensionalidades de los datos a 3 y así poder obtener un gráfico 3D de los clústers obtenidos.

II-A9. Vectorización de los tweets: Para poder graficar los clústers hace falta tener una representación numérica de estos, por lo que se obtiene su TF-IDF para lograrlo.

II-B. Descripción de la distancia de Jaccard

Para calcular la distancia que existe de un tweet con otro, se hace un análisis de las palabras que estos contengan, dado un pre-procesamiento en donde se elimina aquella información no relevante y una normalización de mayúsculas a minúsculas, tenemos un texto uniforme con las mismas características.

Ambos tweets se toman como conjuntos en donde las palabras son los elementos, de este modo tenemos que el coeficiente de Jaccard para ambos conjuntos (tweets) se representa con la fórmula¹.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

Ésta relación nos representa la similitud de los 2 conjuntos en cuanto a los elementos que estos contengan. Pero como nosotros queremos calcular distancias entre instancias (tweets), se toma A como el centroide con el cual se van a calcular las distancias, y B como los tweets a clasificar.

Finalmente, se tiene que realizar una normalización para tener un cálculo adecuado en donde se pueda calcular la distancia, haciendo una normalización de 0 a 1, en donde 1 significa que ambos tweets están muy alejados (no comparten ningún elemento en común) y 0 significa que son idénticos, por lo que la normalización se llevaría a cabo con la ecuación²

$$D(A, B) = 1 - J(A, B) \quad (2)$$

II-C. Prueba con otros archivos del dataset

El dataset utilizado contiene diversos archivos de texto de otras agencias de noticias, tales como CNN y NYT. Por lo que a continuación se realizará un análisis de cada agrupación y el número de k adecuado para 7 colecciones de Tweets de las disponibles en el dataset.

Se probó con valores de $3 \leq k \leq 7$, y se midió la inercia, una métrica utilizada para evaluar la calidad de un agrupamiento. En términos simples, la inercia mide la suma de las distancias cuadradas de cada punto de datos al centroide más cercano de su respectivo grupo. Cuanto menor sea la inercia, más compactos y mejor definidos estarán los clústeres.

Esta métrica se obtiene de la función `k_means`:

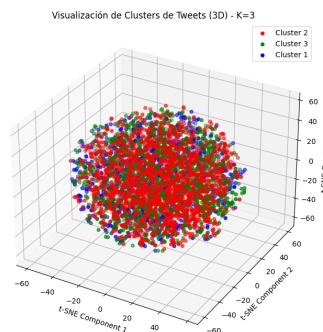
```
clusters, sse = k_means(tweets, k)
```

Donde sse es la inercia.

III. RESULTADOS

III-A. Programa original (BBC Health)

III-A1. Para un $k = 3$:

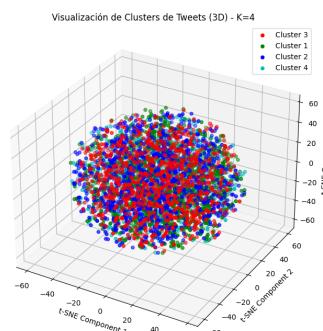


K means con un $k = 3$

Convergencia despues de 2 iteraciones

- 1: 937 tweets
- 2: 1465 tweets
- 3: 1527 tweets

III-A2. Para un $k=4$:

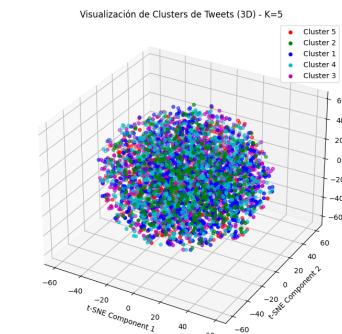


K means con un $k = 4$

Convergencia despues de 3 iteraciones

- 1: 1250 tweets
- 2: 1082 tweets
- 3: 866 tweets
- 4: 731 tweets

III-A3. Para $k=5$:

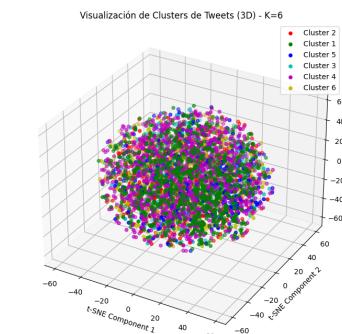


K means con un $k = 5$

III-B. Prueba con otros archivos del dataset

Convergencia despues de 2 iteraciones

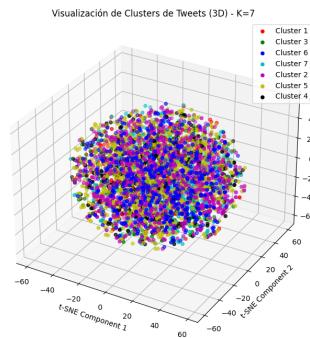
- 1: 1085 tweets
- 2: 463 tweets
- 3: 940 tweets
- 4: 786 tweets
- 5: 655 tweets



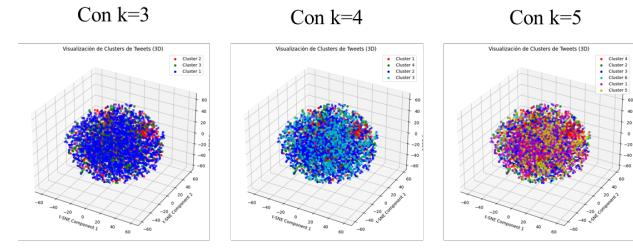
K means con un $k = 6$

Convergencia despues de 3 iteraciones

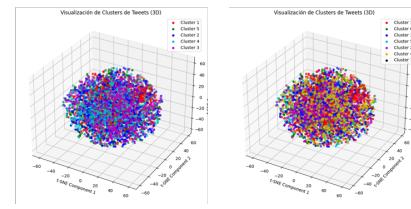
- 1: 862 tweets
- 2: 575 tweets
- 3: 377 tweets
- 4: 1155 tweets
- 5: 440 tweets
- 6: 520 tweets



K means con un k = 7



Con k=6



K means para CNN Health

Convergencia despues de 3 iteraciones

1: 365 tweets
2: 772 tweets
3: 285 tweets
4: 508 tweets
5: 992 tweets
6: 481 tweets
7: 526 tweets

En las diferentes iteraciones se encontraron las siguientes cantidades de clústeres iguales:

Iteración 1.

1: 944 tweets
2: 1750 tweets
3: 1047 tweets

Iteración 2.

1: 712 tweets
2: 577 tweets
3: 1771 tweets
4: 681 tweets

Iteración 3.

1: 558 tweets
2: 968 tweets
3: 1145 tweets
4: 322 tweets
5: 748 tweets

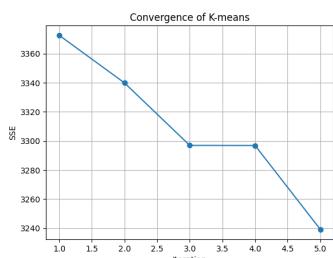
Iteración 4.

1: 600 tweets
2: 972 tweets
3: 155 tweets
4: 720 tweets
5: 784 tweets
6: 510 tweets

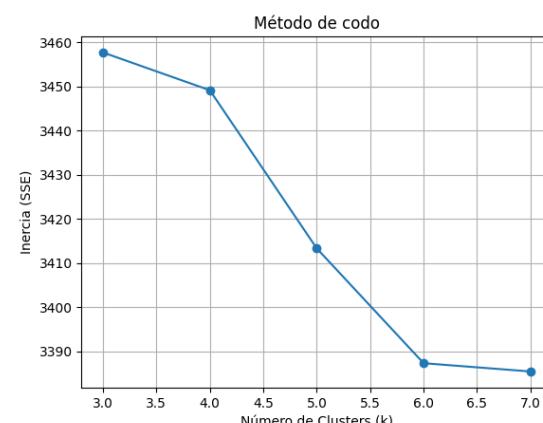
Iteración 5.

1: 220 tweets
2: 569 tweets
3: 611 tweets
4: 568 tweets
5: 497 tweets
6: 883 tweets
7: 393 tweets

Y se obtiene un valor óptimo de k entre 6 y 7.



Grafica de codo para BBC Health



III-C. Prueba con otros archivos del dataset

Las gráficas 3D muestran los siguientes clústeres.

Gráfica de inercia para CNN Health

III-C1. CNN Health.:

Las gráficas 3D muestran los siguientes clústeres.

Con k=3

Con k=4

Con k=5

Con k=6

Con k=7

K means para CNN Health

En las diferentes Iteraciones se encontraron las siguientes cantidades de clústeres iguales:

Iteración 1.

- 1: 2019 tweets
- 2: 1039 tweets
- 3: 1003 tweets

Iteración 2.

- 1: 502 tweets
- 2: 1723 tweets
- 3: 685 tweets
- 4: 1151 tweets

Iteración 3.

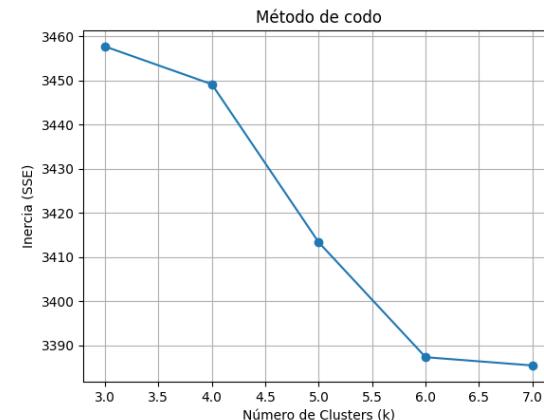
- 1: 688 tweets
- 2: 943 tweets
- 3: 617 tweets
- 4: 678 tweets
- 5: 1135 tweets

Iteración 4.

- 1: 600 tweets
- 2: 972 tweets
- 3: 155 tweets
- 4: 720 tweets
- 5: 784 tweets
- 6: 510 tweets

Iteración 5.

- 1: 957 tweets
- 2: 528 tweets
- 3: 651 tweets
- 4: 538 tweets
- 5: 529 tweets
- 6: 709 tweets
- 7: 149 tweets



Gráfica de inercia para CNN Health

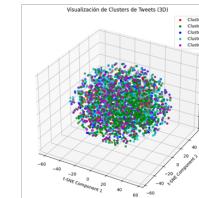
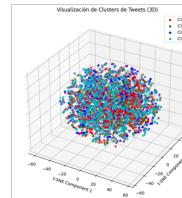
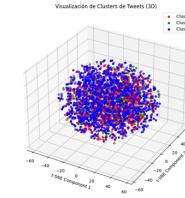
III-C2. Everyday Health.:

Las gráficas 3D muestran los siguientes clústeres.

Con k=3

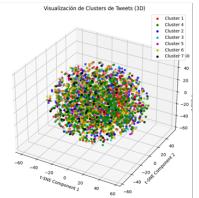
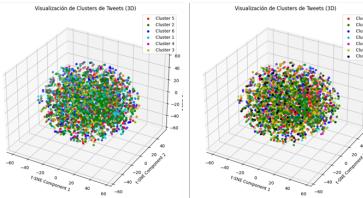
Con k=4

Con k=5



Con k=6

Con k=7



K means para Everyday Health

Iteración 1.

- 1: 604 tweets
- 2: 1214 tweets
- 3: 1421 tweets

Iteración 2.

- 1: 1039 tweets
- 2: 786 tweets
- 3: 501 tweets
- 4: 913 tweets

Iteración 3.

- 1: 952 tweets
- 2: 278 tweets
- 3: 693 tweets
- 4: 905 tweets
- 5: 411 tweets

Iteración 4.

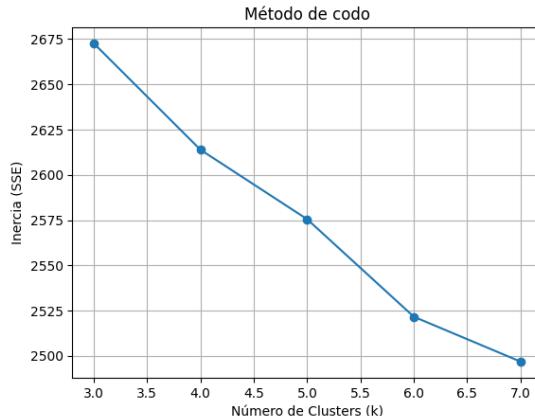
- 1: 707 tweets
- 2: 801 tweets
- 3: 189 tweets
- 4: 207 tweets
- 5: 739 tweets
- 6: 596 tweets

Iteración 5.

- 1: 593 tweets
- 2: 548 tweets
- 3: 186 tweets
- 4: 749 tweets
- 5: 386 tweets
- 6: 681 tweets
- 7: 96 tweets

Y se obtiene un valor óptimo de k entre 6 y 7.

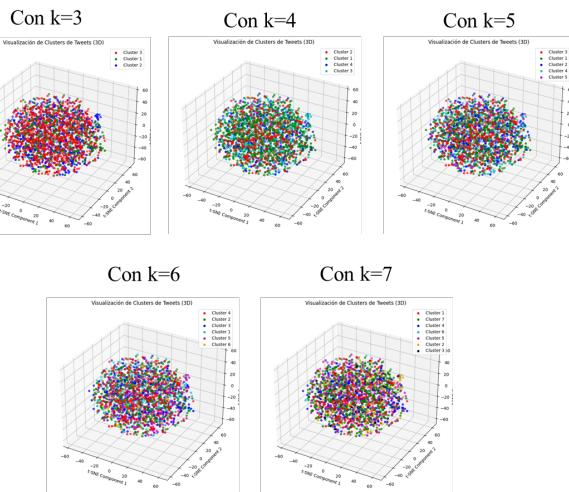
Y se obtiene un valor óptimo de k entre 6 y 7.



Gráfica de inercia para Everyday Health

III-C3. Fox News Health.:

Las gráficas 3D muestran los siguientes clústeres.



K means para Fox News Health

Iteración 1.

- 1: 537 tweets
- 2: 665 tweets
- 3: 798 tweets

Iteración 3.

- 1: 634 tweets
- 2: 385 tweets
- 3: 392 tweets
- 4: 418 tweets
- 5: 171 tweets

Iteración 5.

- 1: 265 tweets
- 2: 307 tweets
- 3: 108 tweets
- 4: 330 tweets
- 5: 419 tweets

Iteración 2.

- 1: 835 tweets
- 2: 337 tweets
- 3: 431 tweets
- 4: 397 tweets

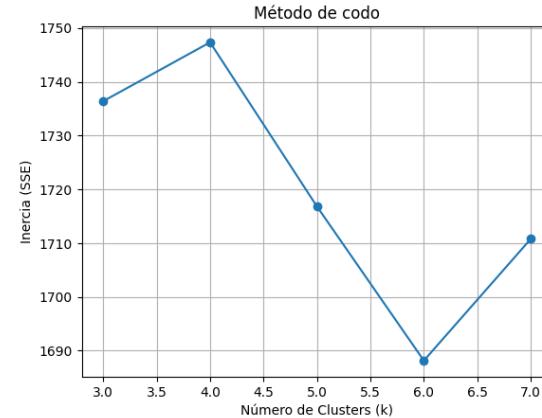
Iteración 4.

- 1: 398 tweets
- 2: 432 tweets
- 3: 331 tweets
- 4: 369 tweets
- 5: 366 tweets
- 6: 104 tweets

6: 80 tweets

7: 491 tweets

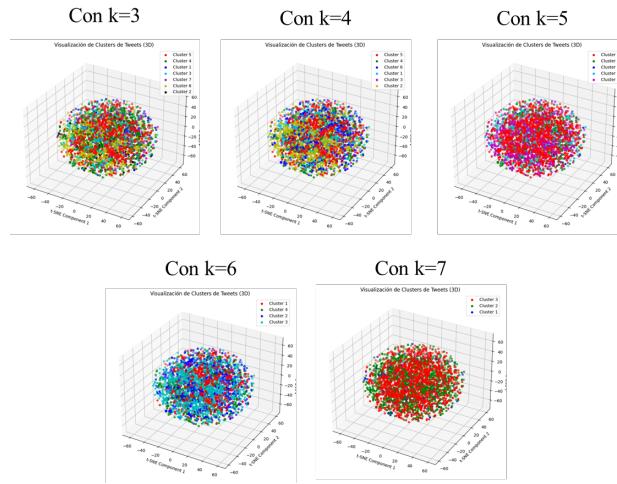
Y se obtiene un valor óptimo de k de 6.



Gráfica de inercia para Fox News Health

III-C4. GND Health Care.:

Las gráficas 3D muestran los siguientes clústeres.



K means para GND Health Care

Iteración 1.

- 1: 201 tweets
- 2: 1566 tweets
- 3: 1230 tweets

Iteración 3.

- 1: 1033 tweets
- 2: 753 tweets
- 3: 104 tweets
- 4: 586 tweets
- 5: 521 tweets

Iteración 5.

- 1: 222 tweets
- 2: 175 tweets
- 3: 374 tweets

Iteración 2.

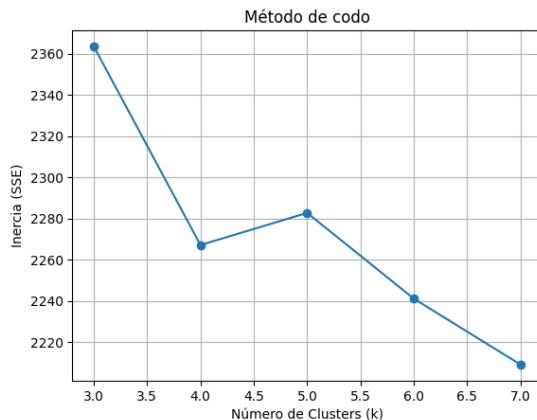
- 1: 373 tweets
- 2: 840 tweets
- 3: 952 tweets
- 4: 832 tweets

Iteración 4.

- 1: 371 tweets
- 2: 636 tweets
- 3: 151 tweets
- 4: 698 tweets
- 5: 458 tweets
- 6: 683 tweets

4: 899 tweets
 5: 564 tweets
 6: 477 tweets
 7: 286 tweets

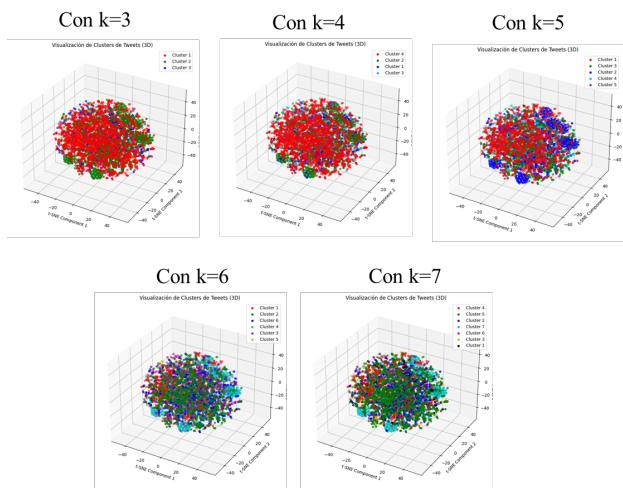
Y se obtiene un valor óptimo de k de 7.



Gráfica de incercia para GND Health Care

III-C5. NBC Health.:

Las gráficas 3D muestran los siguientes clusters.



Iteración 1.

1: 1843 tweets
 2: 1542 tweets
 3: 830 tweets

Iteración 3.

1: 1177 tweets
 2: 1033 tweets
 3: 713 tweets
 4: 958 tweets
 5: 334 tweets

Iteración 5.

1: 572 tweets

Iteración 2.

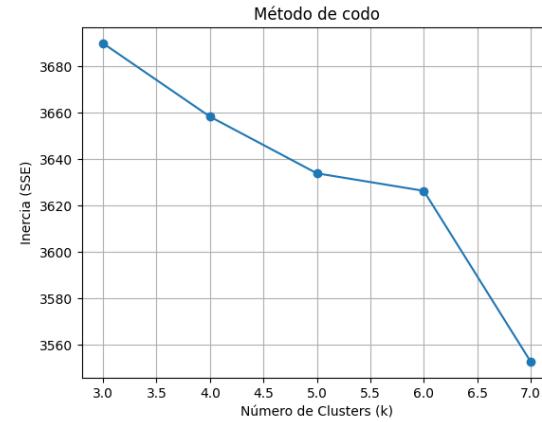
1: 602 tweets
 2: 836 tweets
 3: 963 tweets
 4: 1814 tweets

Iteración 4.

1: 590 tweets
 2: 941 tweets
 3: 831 tweets
 4: 947 tweets
 5: 346 tweets
 6: 560 tweets

2: 520 tweets
 3: 338 tweets
 4: 693 tweets
 5: 1054 tweets
 6: 239 tweets
 7: 799 tweets

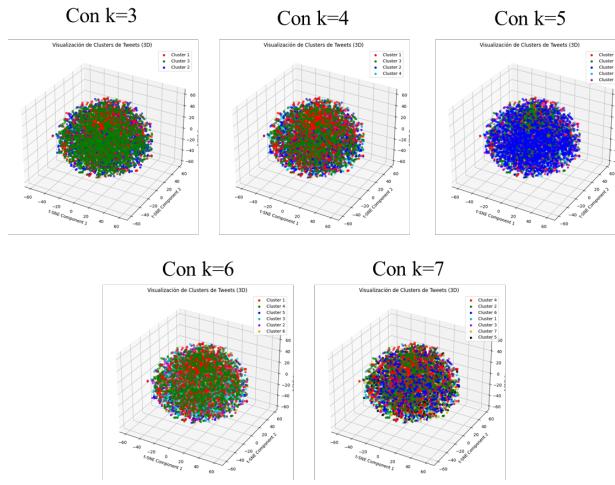
Y se obtiene un valor óptimo de k de 7.



Gráfica de incercia para NBC Health

III-C6. NY Times Health.:

Las gráficas 3D muestran los siguientes clusters.



Iteración 1.

1: 1654 tweets
 2: 2255 tweets
 3: 2336 tweets

Iteración 3.

1: 1132 tweets
 2: 1172 tweets
 3: 1077 tweets
 4: 716 tweets
 5: 2148 tweets

Iteración 5.

Iteración 2.

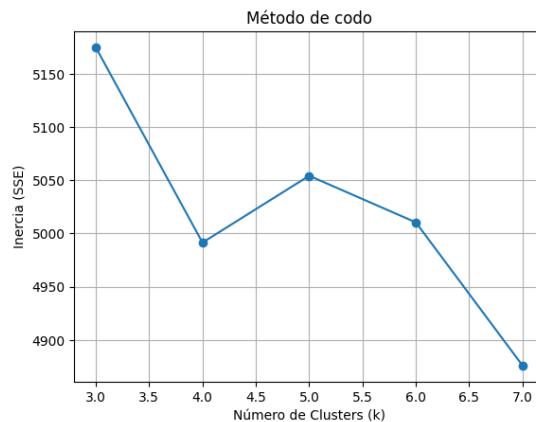
1: 1877 tweets
 2: 1880 tweets
 3: 1219 tweets
 4: 1269 tweets

Iteración 4.

1: 1467 tweets
 2: 994 tweets
 3: 1315 tweets
 4: 1238 tweets
 5: 869 tweets
 6: 362 tweets

- 1: 1197 tweets
- 2: 1040 tweets
- 3: 494 tweets
- 4: 1516 tweets
- 5: 783 tweets
- 6: 515 tweets
- 7: 700 tweets

Y se obtiene un valor óptimo de k de 7.



Gráfica de incercia para NY Times Health

IV. CONCLUSIONES

Dados los resultados obtenidos podemos observar que en un análisis de textos cortos podemos tener muchas similitudes en cuanto al contenido, esto es aún más evidente en este tipo de textos que de una forma ya están etiquetados por pertenecer a la clase de textos que hablen sobre la salud, por lo que muchos van a tener una distancia muy parecida, pero es muy raro ver que sean completamente iguales, por lo que ninguna instancia fue colocada en la misma posición en la que se encontraba su centroide.

Por otro lado, es más común encontrar distancias totalmente opuestas, los cuales significan que tanto el centroide como la instancia (ambos tweets) no tengan ninguna sola relación, interpretándolo como que no comparten ninguna sola palabra, por lo que aunque todos los textos analizados pertenezcan al ámbito de la salud, hay un gran vocabulario el cual describa la misma información.

Por último podemos decir que este nuevo cálculo de las distancias entre 2 tweets es una forma más de representar cuantitativamente información de tipo texto, lo cual no tiene mucho sentido para un lector humano pero sí para un intérprete digital, lo cual puede ayudar a realizar clasificaciones con algoritmos que calculen distancias como K-Means o DBScan.

REFERENCIAS

- [1] Abdullah, S.M., Ali, S.M., Makttof, M.A.: Modifying jaccard coefficient for texts similarity. Opción: Revista de Ciencias Humanas y Sociales (19), 28 (2019)
- [2] Ahmed, M., Seraj, R., Islam, S.M.S.: The k-means algorithm: A comprehensive survey and performance evaluation. Electronics **9**(8), 1295 (2020)
- [3] Syakur, M., Khotimah, B.K., Rochman, E., Satoto, B.D.: Integration k-means clustering method and elbow method for identification of the best customer profile cluster. In: IOP conference series: materials science and engineering. vol. 336, p. 012017. IOP Publishing (2018)