Practica 2: K-means & DBSCAN

Machine Learning

Castro Elvira D.
2022710168
dcastroe2100@alumno.ipn.mx

Nava Méndez E. U.
2021710144
enavam2001@alumno.ipn.mx

UPIIT: Unidad Profesional Interdisciplinaria en Ingeniería Campus Tlaxcala Instituto Politécnico Nacional, Tlaxcala, Tlaxcala, México 9000

Ingeniera en Inteligencia Artificial

30 de septiembre 2023

***Resumen*— K-means y DBSCAN son dos algoritmos populares de agrupamiento en el campo del aprendizaje automático y la minería de datos. Ambos tienen como objetivo encontrar patrones y estructuras en conjuntos de datos no etiquetados, pero se diferencian en su enfoque y aplicaciones. K-means es un algoritmo de particionamiento que divide un conjunto de datos en grupos (clusters) en función de la similitud de sus puntos. Por otro lado, DBSCAN se basa en la densidad de los datos. Identifica clusters en áreas densas de puntos y puede manejar clusters de formas arbitrarias, así como detectar puntos ruidosos.**

Palabras clave* — *K-means, DBSCAN, aprendizaje automático, centroide, cluster, densidad

I. INTRODUCCION

A. MARCO TEÓRICO

Los algoritmos de aprendizaje automático pueden clasificarse en dos categorías: aprendizaje supervisado y aprendizaje no supervisado. También existen otras categorías, como el aprendizaje semisupervisado y el aprendizaje por refuerzo. Pero la mayoría de los algoritmos se clasifican en aprendizaje supervisado o no supervisado.
La diferencia entre ellos se debe a la presencia de una variable objetivo, en el aprendizaje no supervisado, no hay variable objetivo. El conjunto de datos sólo tiene variables de entrada que describen los datos. Esto se denomina aprendizaje no supervisado.

K-Means clustering es el algoritmo de aprendizaje no supervisado más popular, se utiliza cuando tenemos datos no etiquetados, es decir, datos sin categorías o grupos definidos. El algoritmo sigue una forma fácil o sencilla de clasificar un conjunto de datos dado a través de un número determinado de clusters, fijado apriori.
El algoritmo K-Means trabaja de forma iterativa para asignar cada punto de datos a uno de los K grupos basándose en las características que se proporcionan, los puntos de datos se agrupan en función de la similitud de las características.

El clustering de K-Means es el algoritmo de aprendizaje automático no supervisado más común. Es ampliamente utilizado para muchas aplicaciones que incluyen:

- Segmentación de imágenes
- Segmentación de clientes
- Agrupación de especies
- Detección de anomalías
- Agrupación de idiomas

El clustering de K-Means se utiliza para encontrar grupos intrínsecos dentro del conjunto de datos sin etiquetar y extraer inferencias de ellos. Se basa en la agrupación por centroides.

***Centroide*:** Un centroide es un punto de datos situado en el centro de un cluster. En el clustering basado en centroides, los clusters están representados por un centroide. Se trata de un algoritmo iterativo en el que la noción de similitud se deriva de la proximidad de un punto de datos al centroide del cluster. El algoritmo de agrupación K-Means utiliza un procedimiento iterativo para obtener un resultado final. El algoritmo requiere el número de conglomerados K y el conjunto de datos como entrada. El conjunto de datos es una colección de características para cada punto de datos. El algoritmo comienza con estimaciones iniciales de los K centroides. A continuación, el algoritmo itera entre dos pasos.

1. Asignación de datos: Cada centroide define uno de los clusters. En este paso, cada punto de datos se asigna a su centroide más cercano, que se basa en la distancia euclídea al cuadrado. Así, si c_i es la colección de centroides en el conjunto C, entonces cada punto de

datos se asigna a un cluster basado en la distancia euclídea mínima.

2. Actualización de centroides: En este paso, se vuelven a calcular y actualizar los centroides. Para ello, se toma la media de todos los puntos de datos asignados al clúster de ese centroide.

A continuación, el algoritmo itera entre los pasos 1 y 2 hasta que se cumple un criterio de parada. Los criterios de parada significan que ningún punto de datos cambia los conglomerados, que la suma de las distancias se minimiza o que se alcanza un número máximo de iteraciones. Se garantiza que este algoritmo converge a un resultado. El resultado puede ser un óptimo local, lo que significa que evaluar más de una ejecución del algoritmo con centroides iniciales aleatorios puede dar un mejor resultado.

a) Elección del valor de K

El algoritmo K-Means depende de la determinación del número de conglomerados y etiquetas de datos para un valor predefinido de K. Para determinar el número de conglomerados de los datos, debemos ejecutar el algoritmo de agrupación K-Means para distintos valores de K y comparar los resultados. Por lo tanto, el rendimiento del algoritmo K-Means depende del valor de K. Debemos elegir el valor óptimo de K que nos proporcione el mejor rendimiento. Hay diferentes técnicas disponibles para encontrar el valor óptimo de K. La técnica más común es el método del codo que se describe a continuación

b) El método del codo

El método del codo se utiliza para determinar el número óptimo de conglomerados en la agrupación K-means. El método del codo traza el valor de la función de coste producido por diferentes valores de K.

II. DESARROLLO

Para ambos algoritmos se ha usado un archivo *dataset* que contiene información acerca de las reacciones que reciben distinto tipo de material en redes sociales. El archivo contiene el id de las publicaciones, su status (el tipo de material que contiene: foto, video, estado o un link), su fecha de publicaciones, el número de reacciones, numero de comentarios, número de usuarios que comparten el material.

Inicialmente se ha usado el dataset para graficar los datos considerando los atributos correspondientes al número de compartidos, reacciones y comentarios que han recibido las publicaciones en la red social y muestran un comportamiento como en las siguientes.

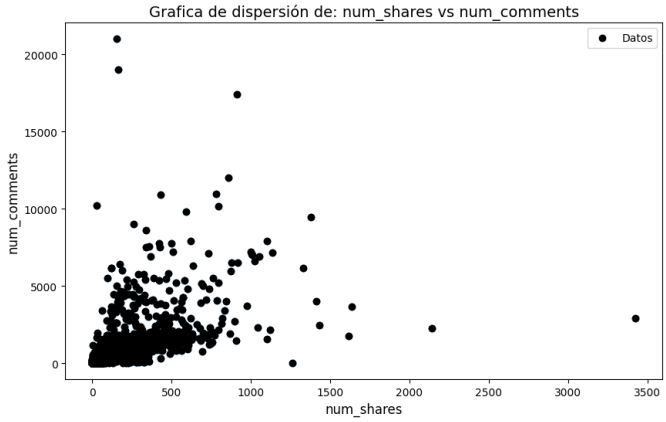


Gráfico 1. Grafica con parámetros de Compartidos y Comentarios.

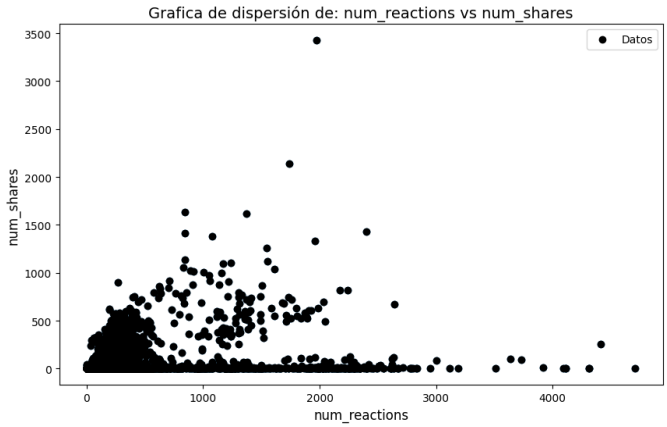


Gráfico 2. Grafica con parámetros de Reacciones y Compartidos.

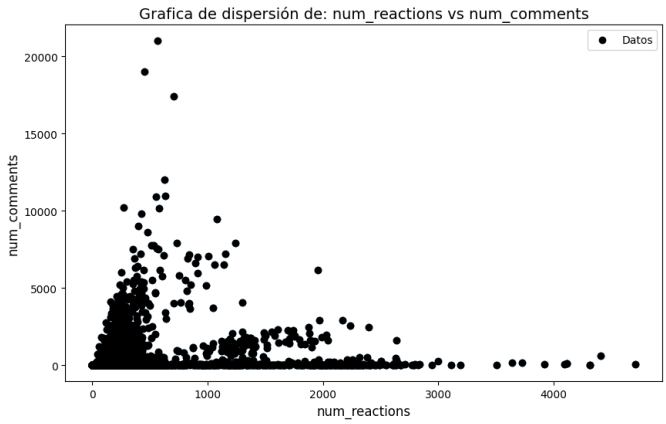


Gráfico 3. Grafica con parámetros de Reacciones y Comentarios.

Para efectos de esta práctica vamos a tratar como tres casos a las diferentes combinaciones de los atributios a tomar en cuenta.

- **Caso 1**: Los atributos a considerar son compartidos y comentarios.
- **Caso 2**: Los atributos a considerar son reacciones y compartidos.

- **Caso 3**: Los atributos a considerar son reacciones y comentarios.

A simple vista podemos observar que su comportamiento es similar en los tres casos, agrupados en la mayoría de los datos en valores pequeños para los atributos considerados.

Caso 1

No se distinguen clusters en la Grafica 1 por lo que k-medias probablemente no genere cluster3es adecuados. Para este caso en específico puede resultar más eficiente el algoritmo DBSCAN.

Caso 2

En el grafica 2, difícilmente se pueden diferenciar clusteres, sin embargo es posible que los datos en la esquina inferior izquierda pueden formar uno solo y considerar otro grupo para el resto de datos.

Caso 3

También se pueden distinguir en la tercer Grafica dos regiones o cúmulos que se paran hacia los ejes de la gráfica 3 por lo que podríamos plantear para *k* un valor igual a 2 y comenzar con la implementación del algoritmo. Es posible que estos agrupamientos indiquen una diferencia entre el tipo de interacción que recibe cada tipo de estatus en los datos.
Aunque la propuesta del valor para *k* se estableció por criterio del usuario, fue necesario implementar también el *Método del codo* para comprobar que el valor propuesto fue el más optimo.
En las gráfica podemos observar también algunos datos atípicos, que en el algoritmo *k-medias* se pueden considerar como parte de un cluster, dándole a ese cluster una forma irregular. Para evitar esto es que se ha implementado también el algoritmo DBSCAN, pues tomaría esos valores atípicos como ruido y no los considera en los clusters, además que no es necesario plantear un valor para *k*.

III. RESULTADOS

Caso 1

Cuando se grafica según el año de publicación y el tipo de post no se hay una aparente distinción de clusters.

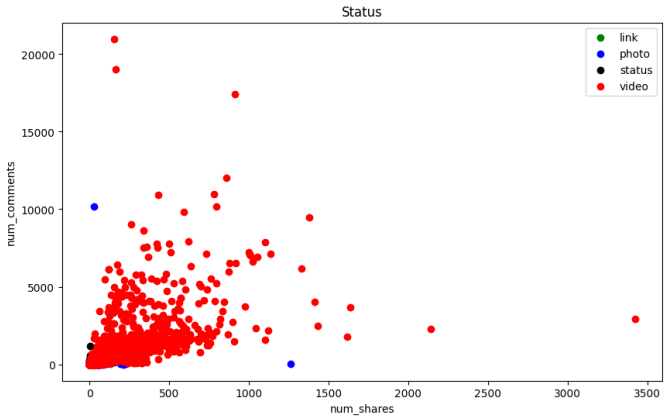


Grafico 4. Grafica (caso 1) según el status de las publicaciones.

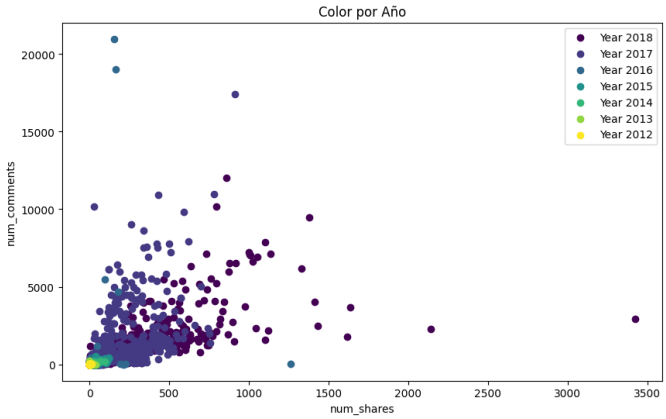


Grafico 5. Grafica (caso 1) según el año de publicacion.

Según el método del codo, el valor de k es de 2.

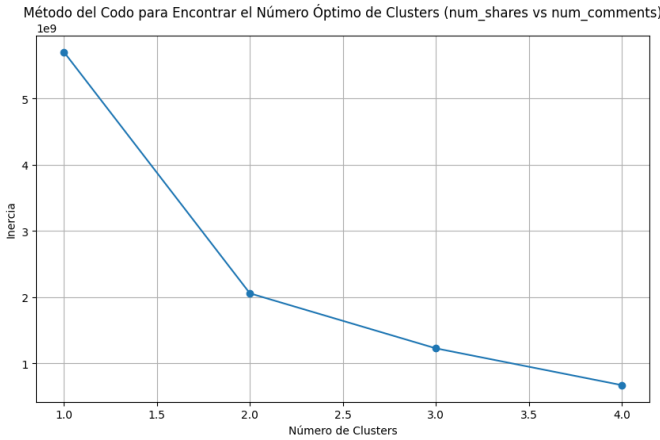


Grafico 6. Grafica (caso 1) del metodo del codo.

Entonces procedemos a ejecutar el algoritmo de k-means.

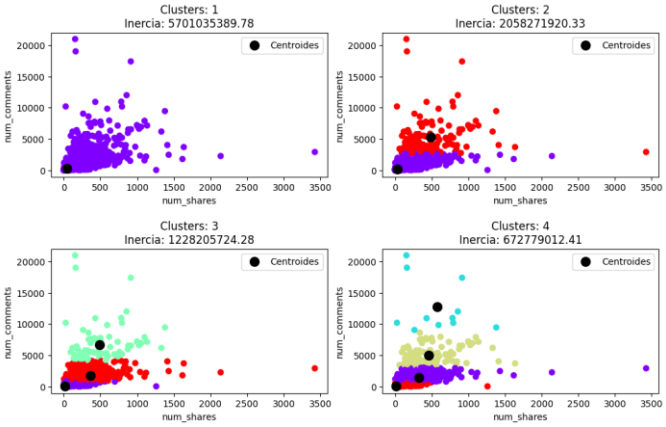


Grafico 7. Clusters del caso 1 con K-means

Los clusters sin embargo no arecen ser los mas adecuados. Pero al procesar los datos con el algoritmo DBSCAN tenemos que:

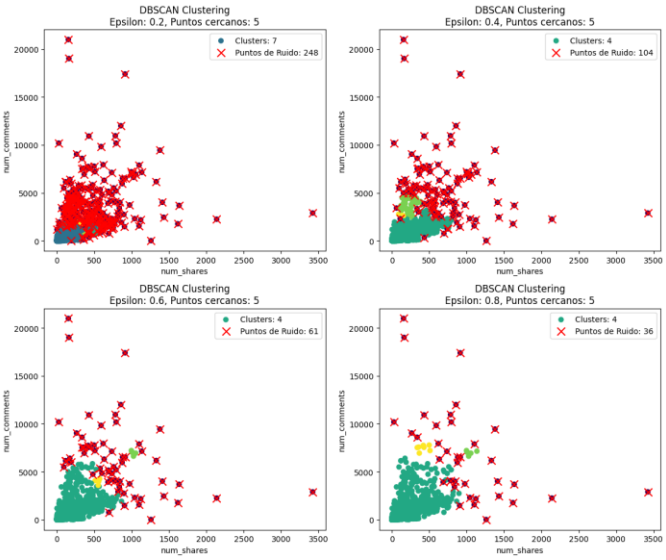


Grafico 8. Clusters generados por DBSCAN para el caso 1.

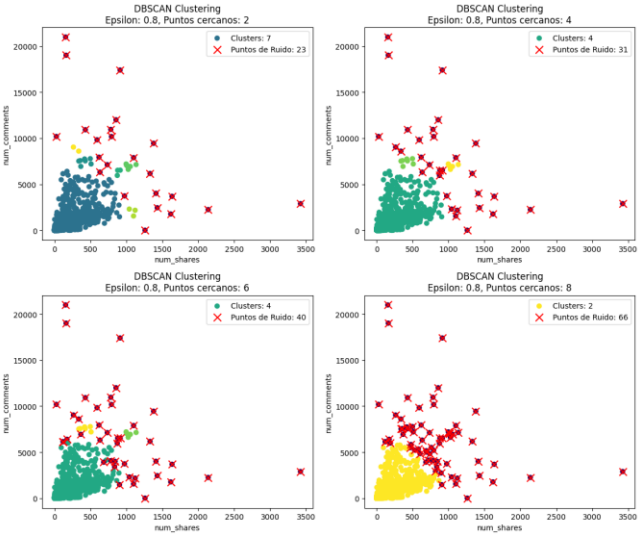


Gráfico 9. Clusters generados por DBSCAN para el caso 1.

Caso 2

Tras graficar el caso 2 por año y por status, se tiene la Grafica 10 y 11 donde se diferencian ya algunos grupos en la parte inferior, pero a nuestro criterio no son lo suficientemente distinguibles de otros datos.

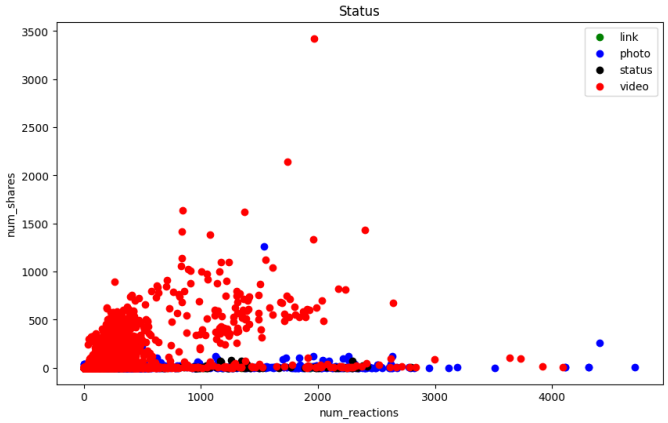


Grafico 10. Grafica (caso 2) según el status de las publicaciones

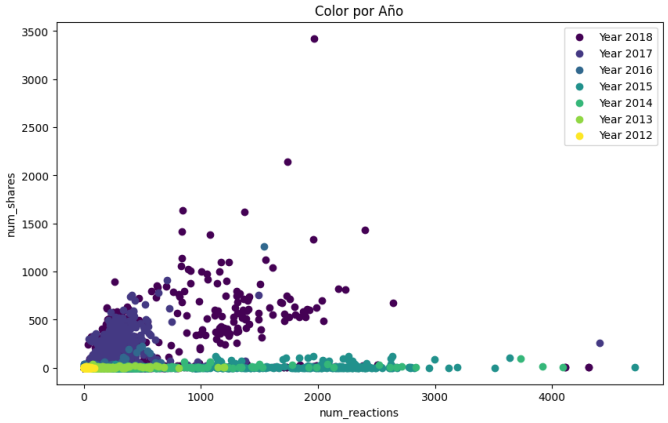


Grafico 11. Grafica (caso 2) según el año de publicación.

Como no se han reconocido clusteres significativos, se han pasado las demás graficas de este caso en la sección de anexos de este documento.

Caso 3

Como se había planteado inicialmente, en las gráficas iniciales se distinguen dos grupos que se separaban hacia los ejes de la gráfica, de lo que podríamos concluir que algunas publicaciones generaban más reacciones que comentarios y viceversa. Se pensó que podría deberse al status de los datos o debido a la fecha de registro por lo que también se decidió hacer la distinción de estos parámetros para compararlos después con el resultado de los algoritmos.

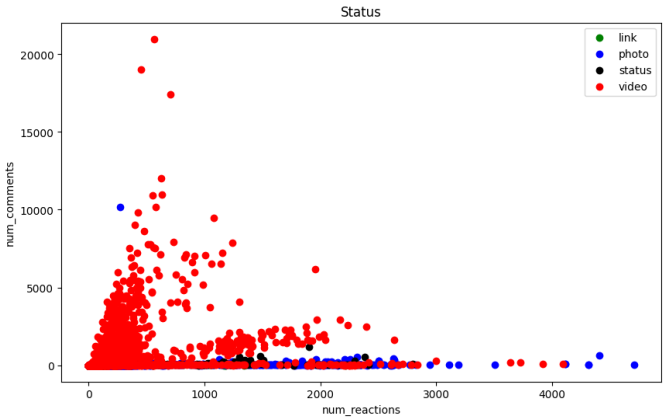


Gráfico 12. Grafica (caso 3) según el status de las publicaciones.

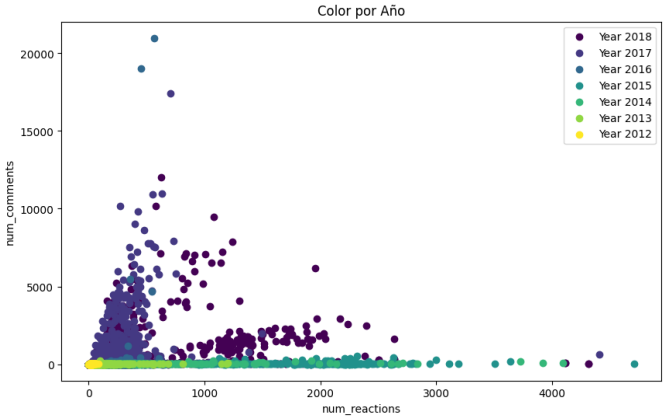


Gráfico 13. Grafica (caso 3) según el año de publicación.

Despues de aplicar el *metodo del codo* obtuvimos como resultado que el valor optimo para *k* fue igual a 2.

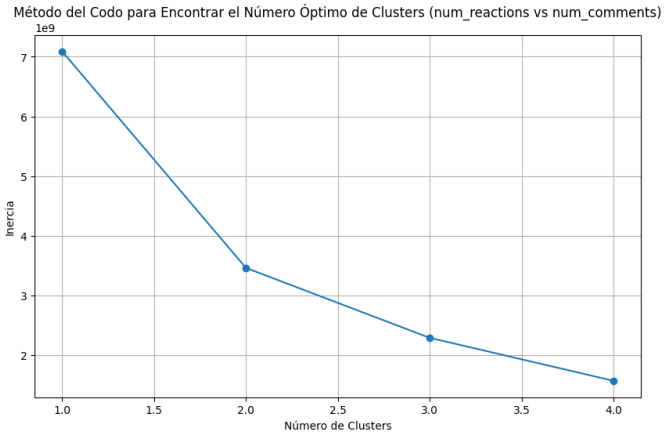


Grafico 14. Grafica (caso 3) del metodo del codo.

Al realizar los clusters mediante *k-means* obtenemos las siguientes graficas.

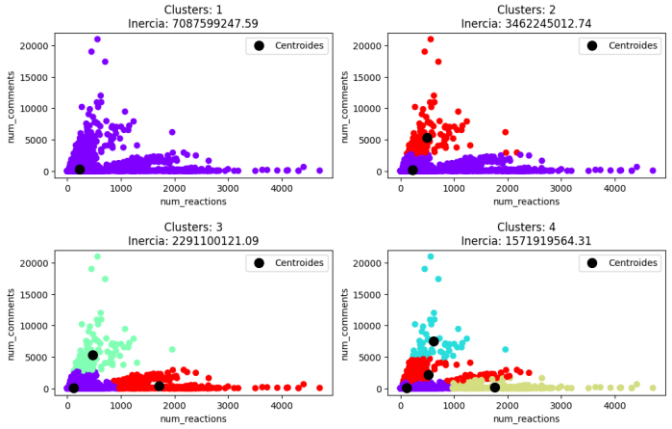


Gráfico 15. Clusters del caso 3 con K-means

Se decidio probar los clusters hasta con *k* igual a 4 pues en la grafica obtenida por el metodo del codo era hasta el valor 4 donde la pendiente del codo era mas plana. Observese que la grafica con dos clusters, son divididos tal cual se planteo al incio del problema. Para tres clusters, se encuentra un tercero en la esquina inferior izquierda que es donde se acumujlan la mayor parte de los datos.

Los cluster gnerados por el algoritmo DBSCAN se muestran a continuación.

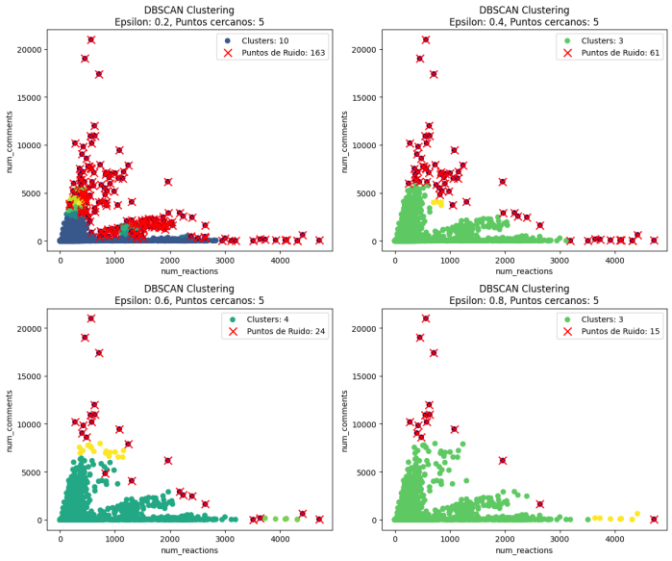


Grafico 16. Clusters generados por DBSCAN para el caso 3.

Como se visualiza en la grafica anterior se han eliminado gran parte de los datos atipicos permitiendo al algoritmo hacer las agrupaciones aunque menos distinguibles que las hechas por k-means.

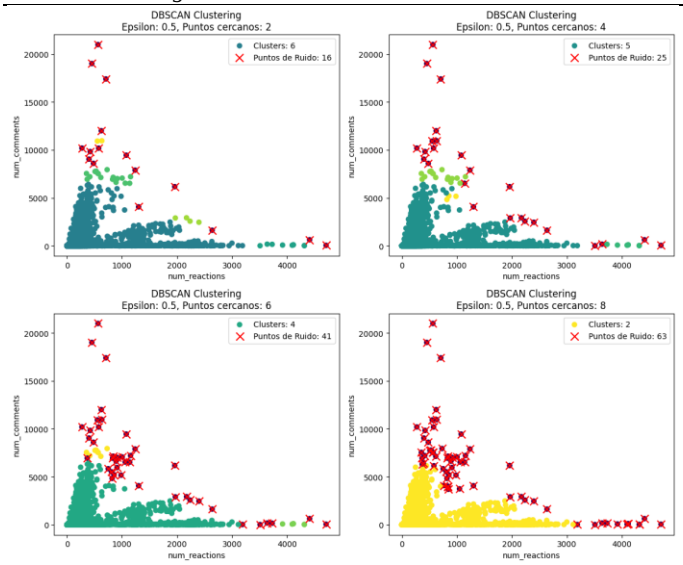


Grafico 17. Clusters generados por DBSCAN para el caso 3.

IV. CONCLUSIONES

En la gráfica 12 podemos observar que existe una mayor cantidad de videos que de cualquier otro tipo de post y que los demás tipos de estatus generan casi nada de comentarios, pero si muchas reacciones. En cuanto a la gráfica 13 podemos notar que las publicaciones con mayores reacciones fueron las ocurridas antes del 2015.

Haciendo la comparación de la gráfica 12 y 13 con los resultados del algoritmo k-means aplicados al tercer caso, identificamos que los tipos de estatus distintos a video y los más viejos se encuentran en el cluster morado de la gráfica generada para solo clusters.

Sin embargo, se aprecia una mejor distinción de características al hacer la comparación con el algoritmo k-means ara clusters en la Grafica 15.

Cabe resaltar que el algoritmo k-means ha tenido un mejor funcionamiento a las formas los clusters con la gráfica en donde se graficaron los atributos de las reacciones contra el número de comentarios. Tal no es el caso para la Grafica 1 que en este caso se encontró una mejor distribución de los cluster con el algoritmo DBSCAN en contraste con los resultados obtenidos con k-means para el mismo caso donde los clusters parecern estar apilados unos sobre otros.

Para una mejor optimización del problema es mejor usar los tres parámetros a la vez pu3es se reducen en gran medida la cantidad de gráficos que se el código tiene que hacer y los cluster resultantes estén lo mejor agrupados posibles.

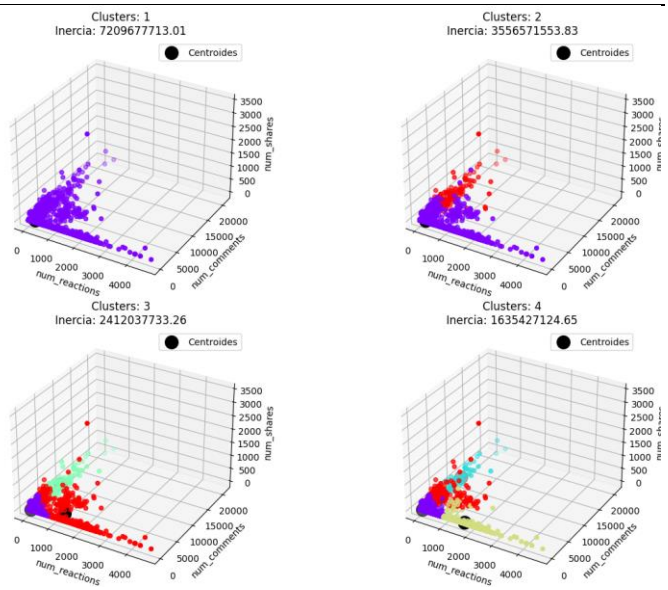


Grafico 18. Clusters generados por k-means tomando en cuenta los parámetros: comartidos, comentarios y reacciones..

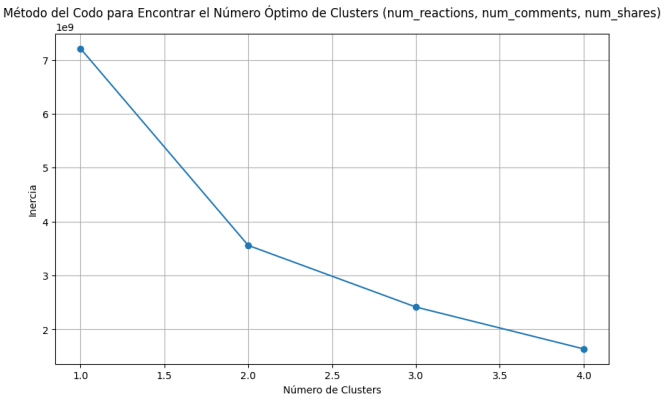


Grafico 19. Graica obtenida po rle metodo del codo para el caso de tomar los tres arametros.

V. BIBLIOGRAFIA

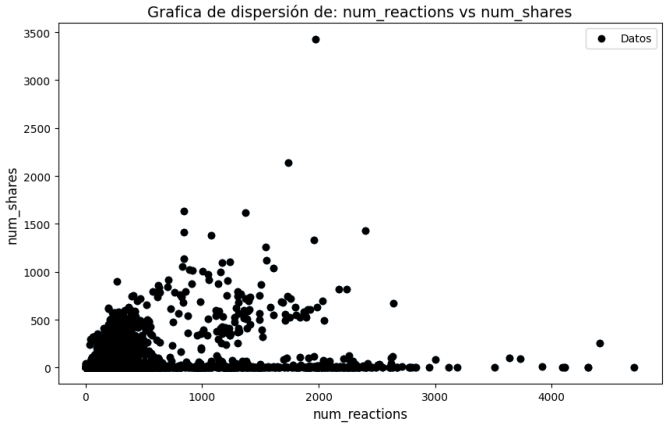
[1] Ramirez L. (5/01/2023), Algoritmo k-means: ¿Qué es y cómo funciona? https://www.iebschool.com/blog/algoritmo-k-means-que-es-y-como-funciona-big-data/

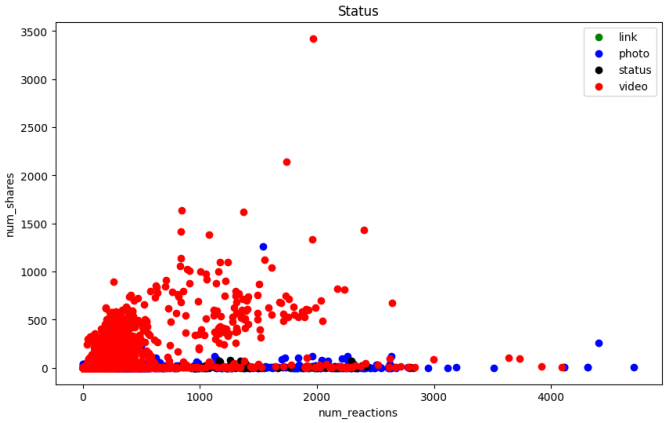
[2] Sanz F. (s. f.), Algoritmo K-Means Clustering – aplicaciones y desventajas https://www.themachinelearners.com/k-means/

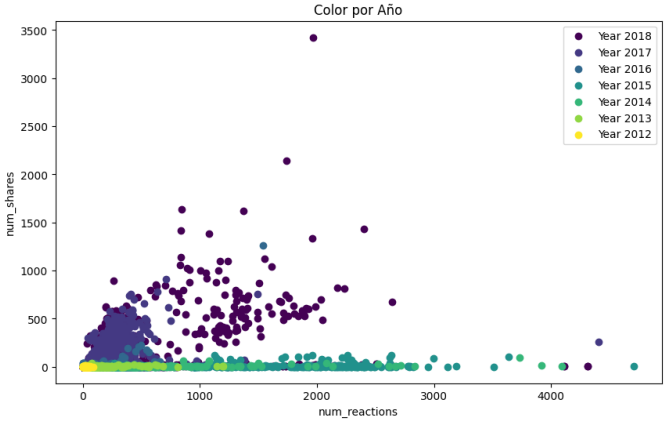
[3] DataScientest (30/11/2022) Machine Learning & Clustering: el algoritmo DBSCAN https://datascientest.com/es/machine-learning-clustering-dbscan#:~:text=El%20DBSCAN%20es%20un%20algoritmo,utilizando%20librer%C3%ADas%20como%20Scikit%2DLearn

[4] Arcgis (s. f.) Cómo funciona el clustering basado en densidad https://pro.arcgis.com/es/pro-app/latest/tool-reference/spatial-statistics/how-density-based-clustering-works.htm

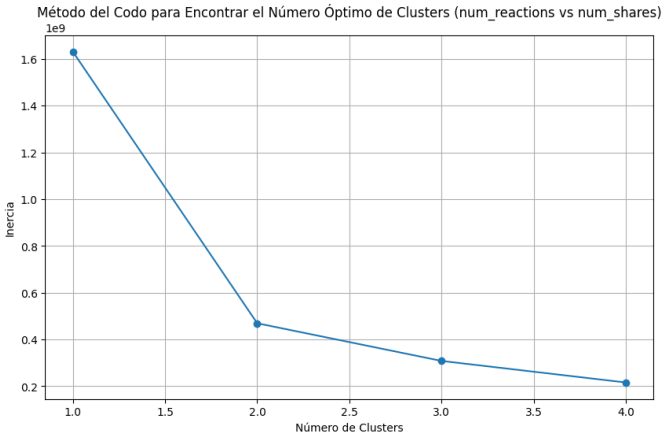
PRIMER ANEXO: SHARES VS REACCIONES

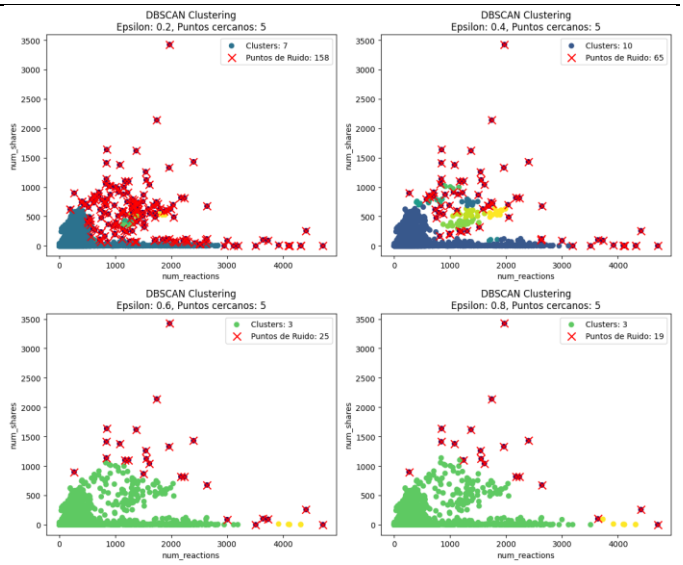




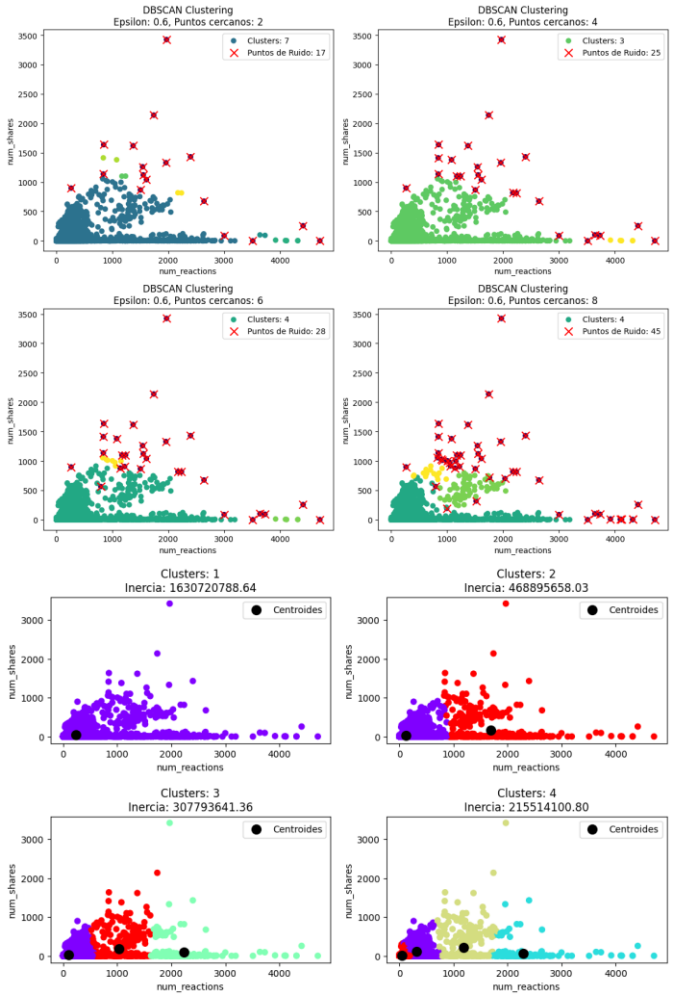


K-MEANS

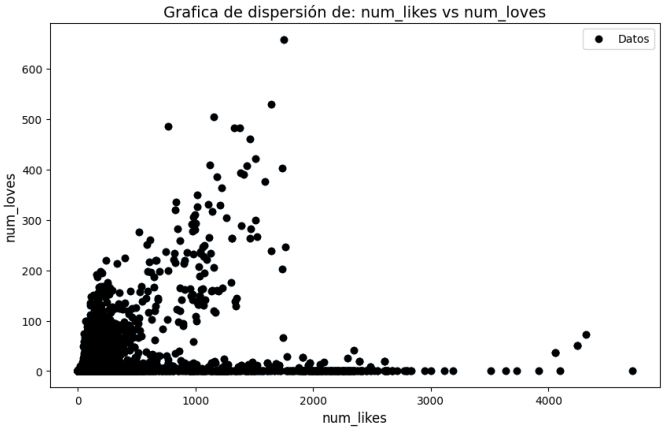


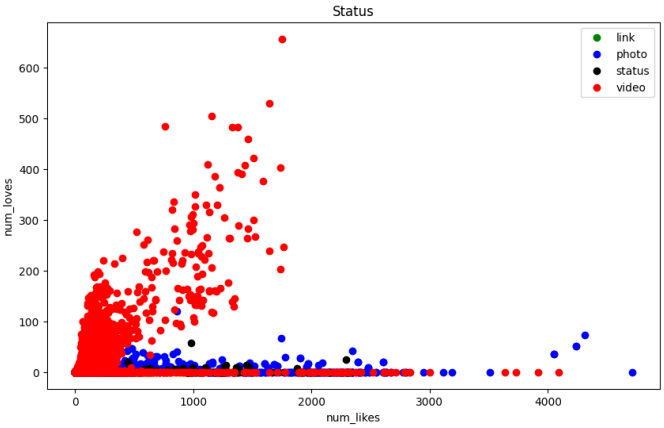


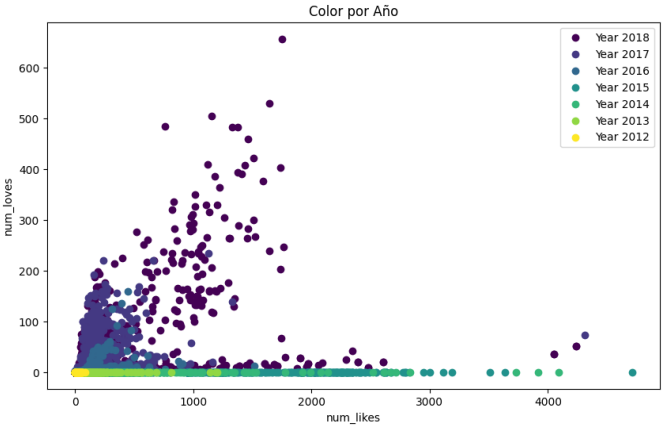
DBSCAN



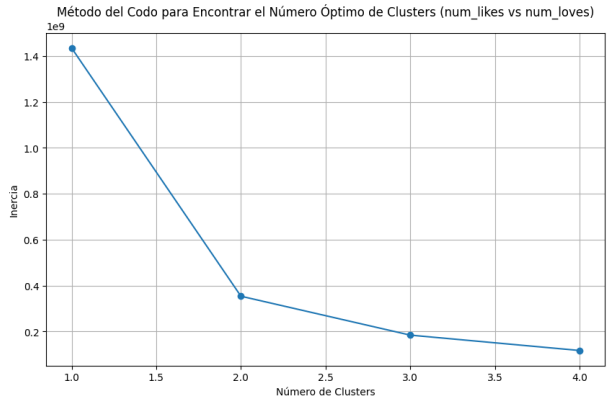
SEGUNDO ANEXO : LIKES VS LOVES

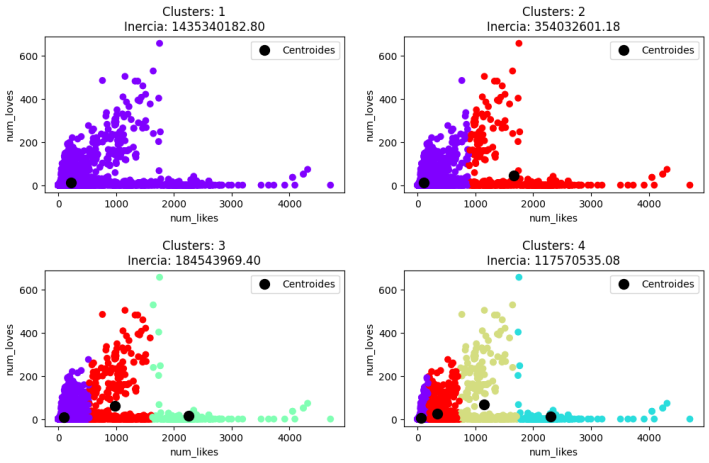




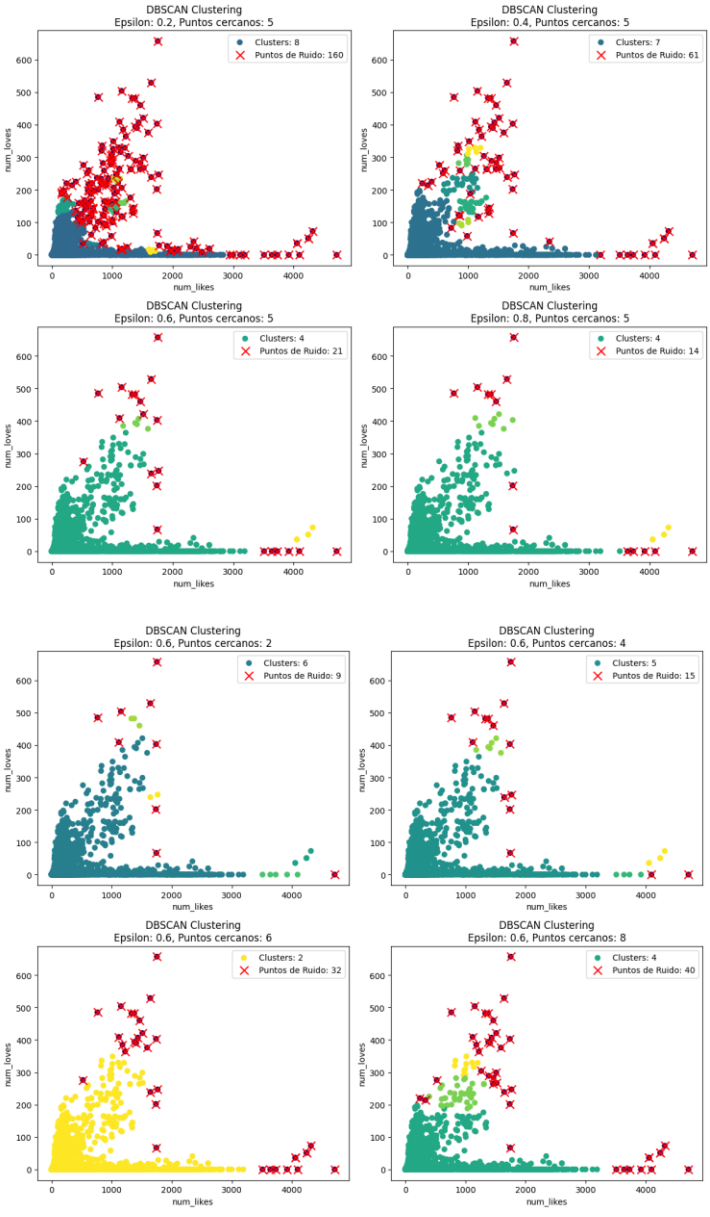


K-MEANS

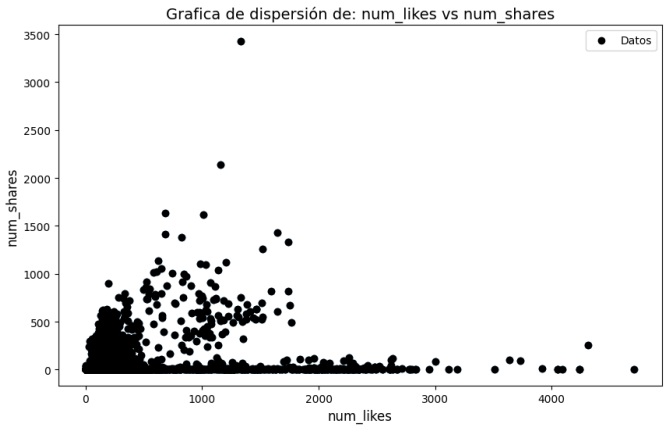


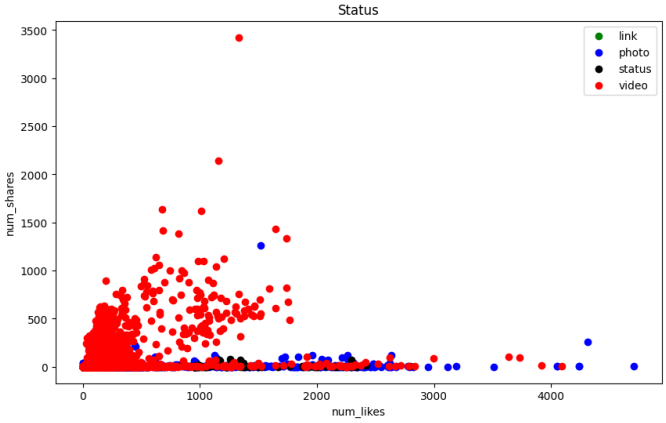


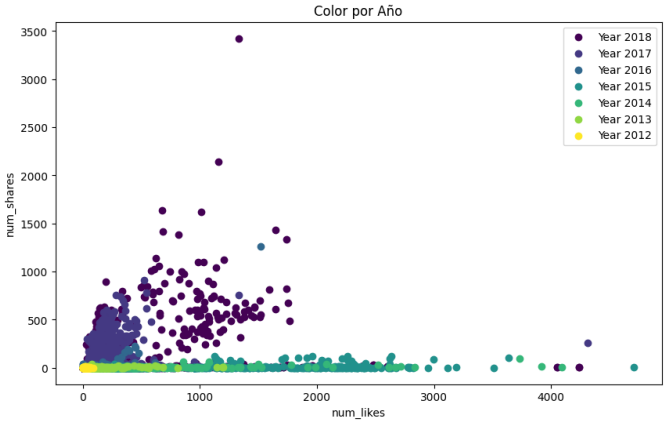
DBSCAN



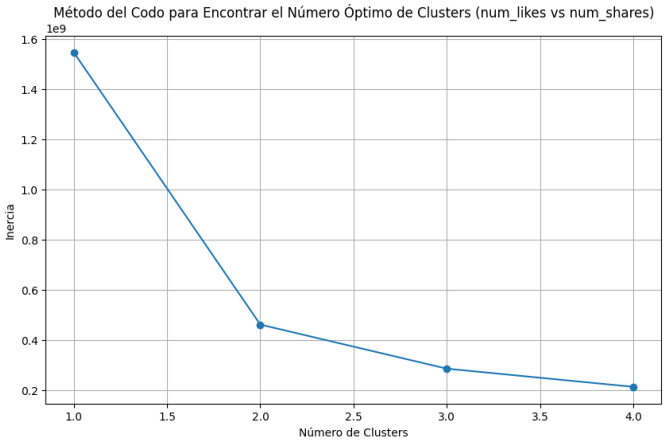
TERCER ANEXO: LIKES VS SHARES

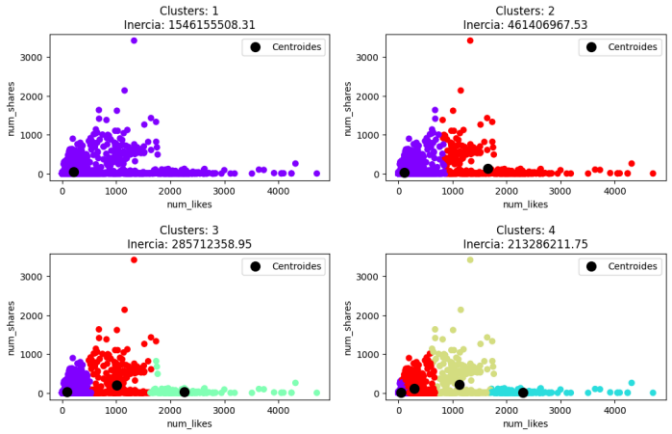




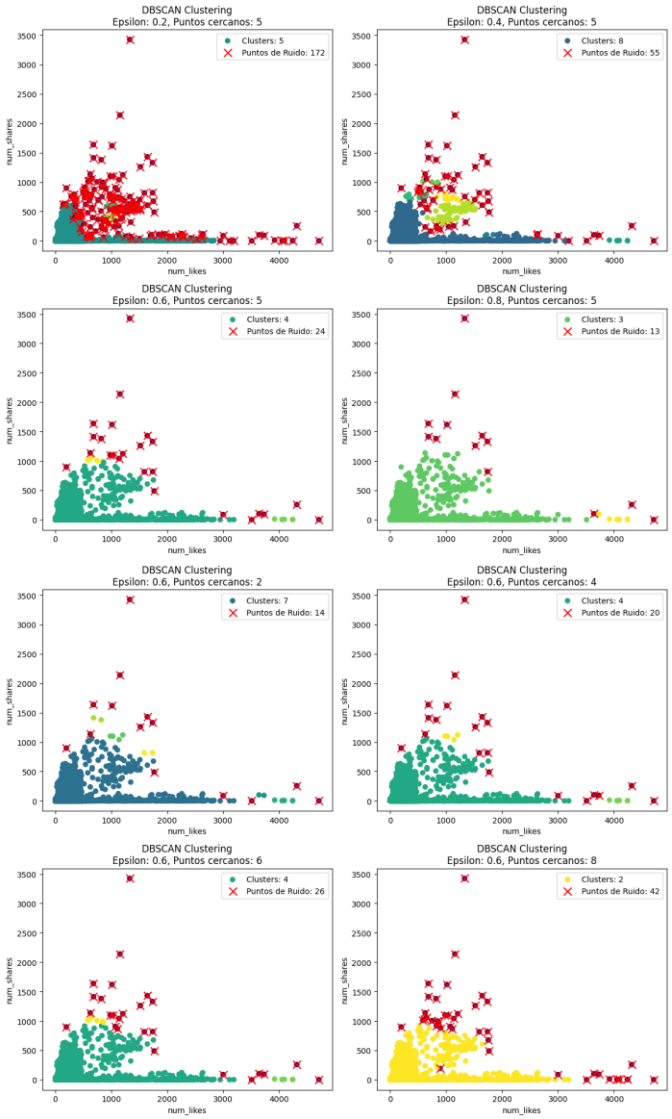


K-MEANS

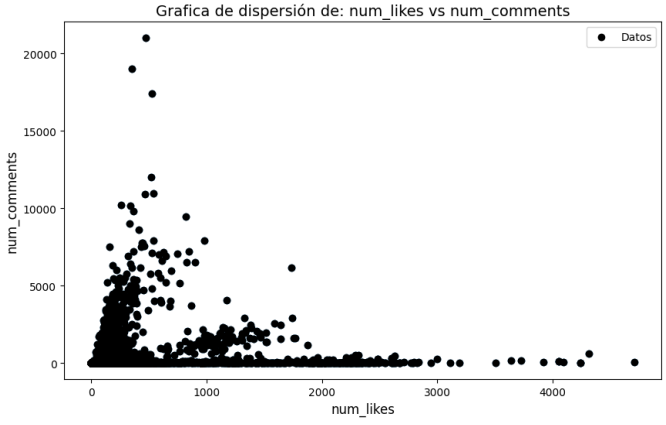


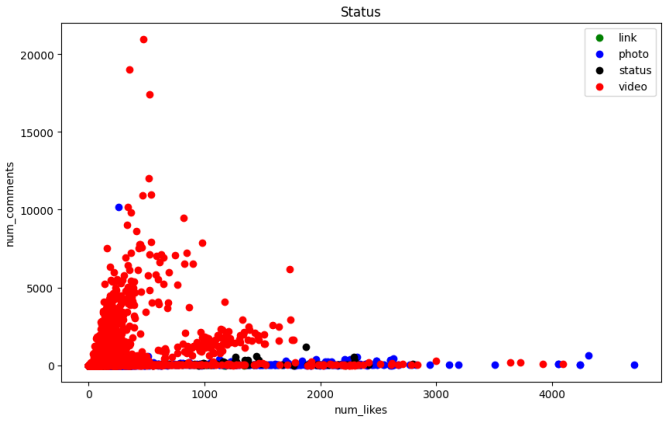


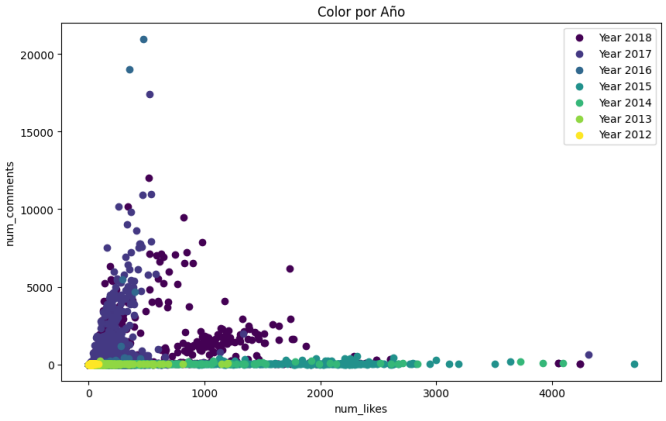
DBSCAN



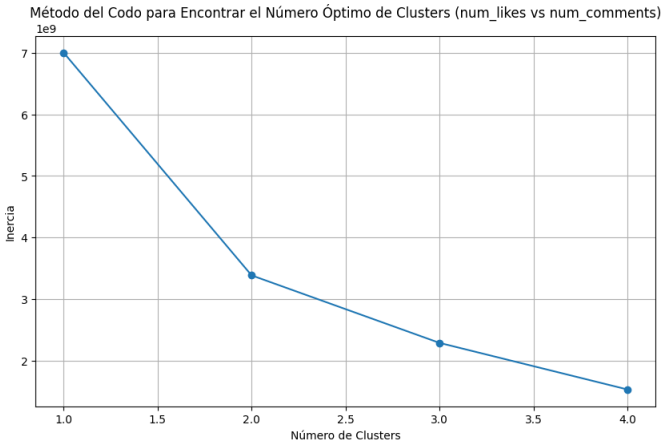
CUARTO ANEXO: LIKES VS COMMENTS

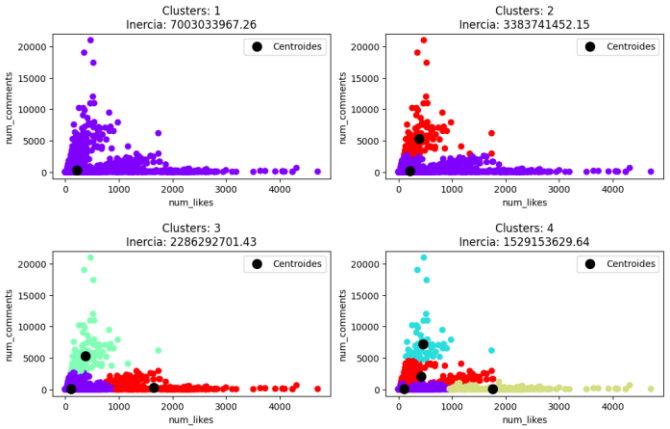




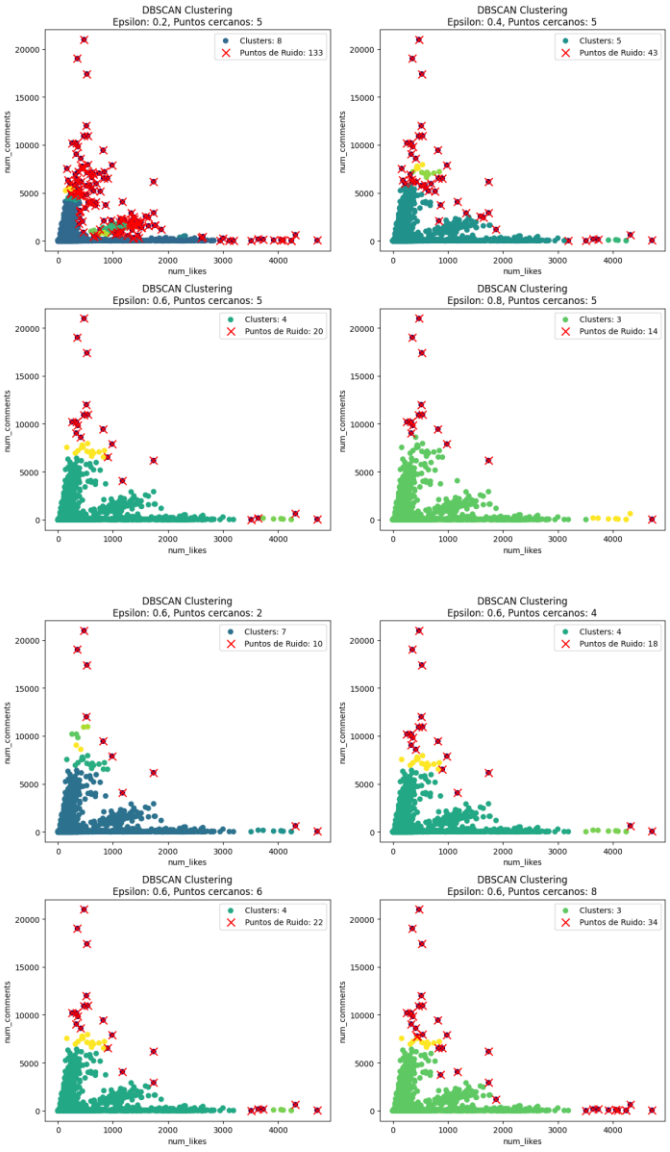


K-means

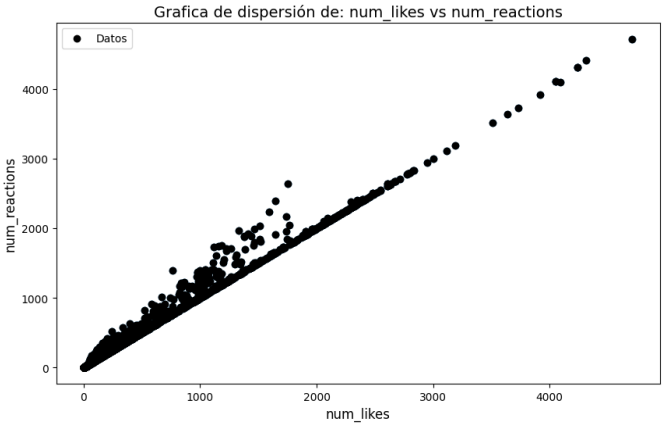


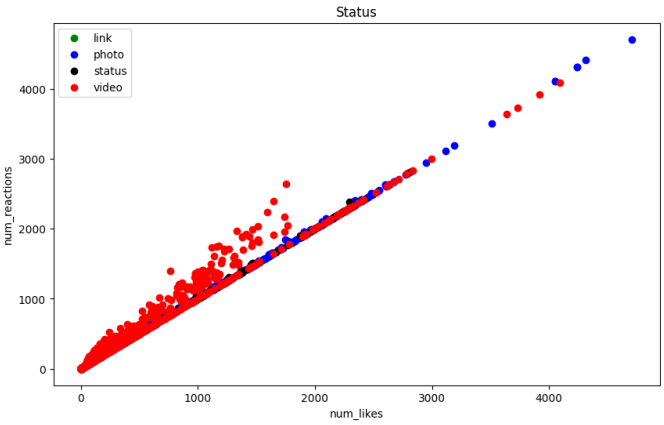


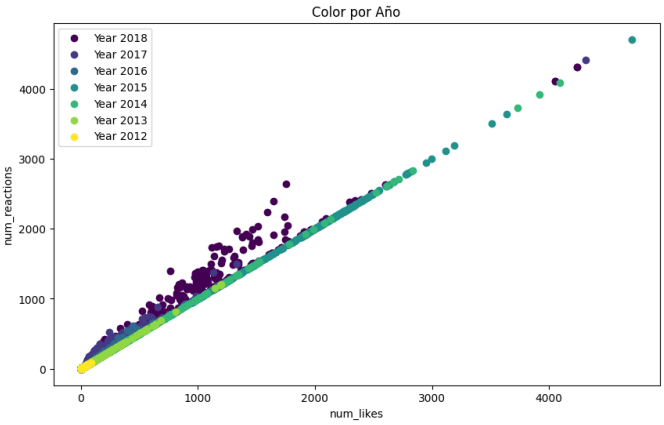
DBSCAN



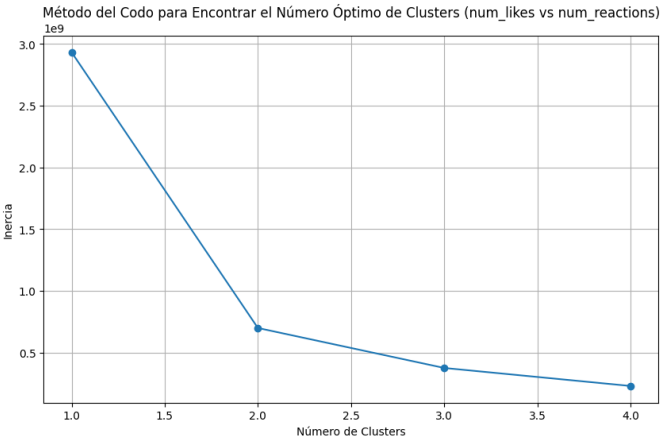
QUINTO ANEXO: LIKES VS COMMENTS

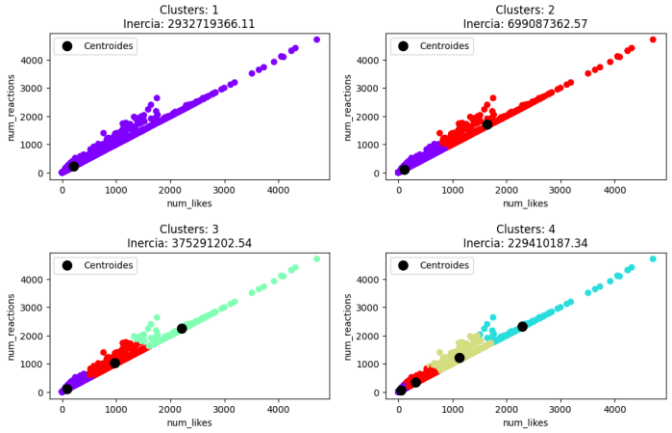






K-MEANS





DBSCAN