Documentación

1. Introducción

Este proyecto se realizó para la materia de tópicos en telemática para la unidad de Big Data y data análisis. La siguiente documentación es para abordar todos los temas y herramientas que se utilizaron para cumplir con las actividades propuestas. En primer lugar, empezaremos explicando brevemente las actividades realizadas las cuales están descritas de forma específica en el README.md de este repositorio, las cuales fueron traer los datos[1] de una fuente determinada, luego la ingesta, alucinamiento, procesamiento y aplicación. A este proceso se le llama el ciclo de vida de los datos y para poder llevarlo a cabo completamente hicimos uso de las siguientes herramientas y procesos matemáticos.

2. Marco teórico

En este apartado iremos trabajando la teoría detrás de las actividades planteadas en el README.md empezando desde los algoritmos[2] utilizados para la preparación de los datos y terminando por los métodos matemáticos y de agrupamiento utilizados para procesamiento y análisis de los datos

3. Preparación

A continuación, veremos los algoritmos utilizados para el proceso de preparación de los datos. Cabe aclara que antes de la preparación de los datos se tenían tres archivos fuentes los cuales fueron unidos para crear un solo archivo con el cual se realizó toda la preparación de los datos y posteriormente el análisis.

3.1 RegexTokenizer

RegexTokenizer[3] hace parta de la familia de algoritmos provisto por Spark para la extracción, selección y transformación de datos. En nuestro caso se hizo uso de el para poder encontrar las palabras que cumpliesen con un patrón determinado el cual era los “null” que se encontraban dentro de los datos.

3.2 StopWordsRemover

StopWordsRemover[4] es otro algoritmo provisto para la trasformación de los datos que se encarga de remover los stopwords[5], dichas "palabras de paradas" son las palabras que se remueven antes de realizar un procesamiento para lenguaje natural, y son aquellas palabras más comunes en cualquier lenguaje. Por lo tanto, existe una lista definida para cada lenguaje que contiene dichas palabras.

3.3 CountVectorizer

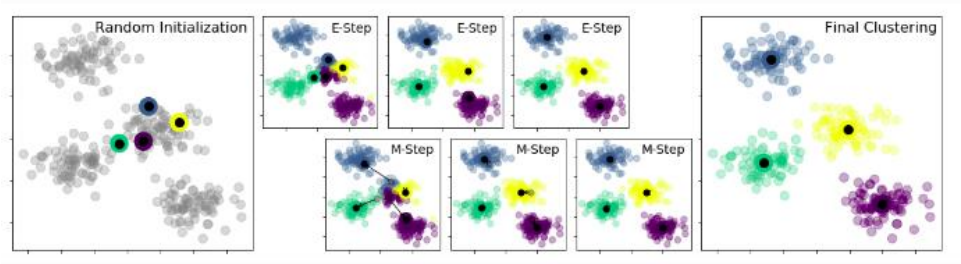
CountVectorizer[6] es el último algoritmo utilizado para la preparación de los datos y se encarga de tomar una colección de documentos en texto convertirlos en un vector de conteo de tokens, esto nos ayuda a identificar sin un diccionario previamente dado cual fue el vocabulario utilizado en los documentos. Luego más adelante este vocabulario y el conteo de la frecuencia nos permiten hacer un mapa de que tan frecuente es usada cada palabra del vocabulario en todos los documentos.

4. Procesamiento

En el apartado del procesamiento utilizaremos un método llamado Kmens en conjunto con otros dos métodos llamados Elbow y el Silhouette para poder encontrar los valores correctos con los cuales interpretar los datos.

4.1 K-means Method

El método k-means[7] es un método que tiene sus orígenes en el estudio de señales, pero ha tenido una fuerte acogida en el data mining el cual, en este caso la naturaleza con la cual se agrupan los datos. También cabe resaltar que el método K-means busca resolver el problema llamado "Expectation-maximization"[8] el cual busca resolver el comportamiento de las variables latentes, las cuales provienen de un conjunto de datos mezclados, pero en el cual no posees la distribución especifica de cada uno.



Matemáticamente se logra con el siguiente proceso:

1. Asignar un centro a cluster de datos
2. Repetir hasta que converja
   - E-Step: Asigna un nuevo punto al centro más cercano
   - M-step: Asigna el centro del cluster

Dado un conjunto de observaciones $(x_1, x_2, \cdots, x_m)$. La función objetivo es:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} ||x_i - c_k||^2$$

Dónde $w_{ik} = 1$ si x_i es en el cluster k; de otra forma $w_{ik} = 0$ y c_k es el centroide de x_i 's cluster.

Matemáticamente, k-means es un prblema de minimización con dos partes: primero, se minimiza J w.r.t w_{ik} con c_k como valor fijo; Luego se minimiza J w.r.t c_k con w_{ik} como valor fijo. Ejemplo:

E-step:

$$\rightarrow \quad w_{ik} = \begin{cases} 0, & \text{otherwise} \end{cases}$$

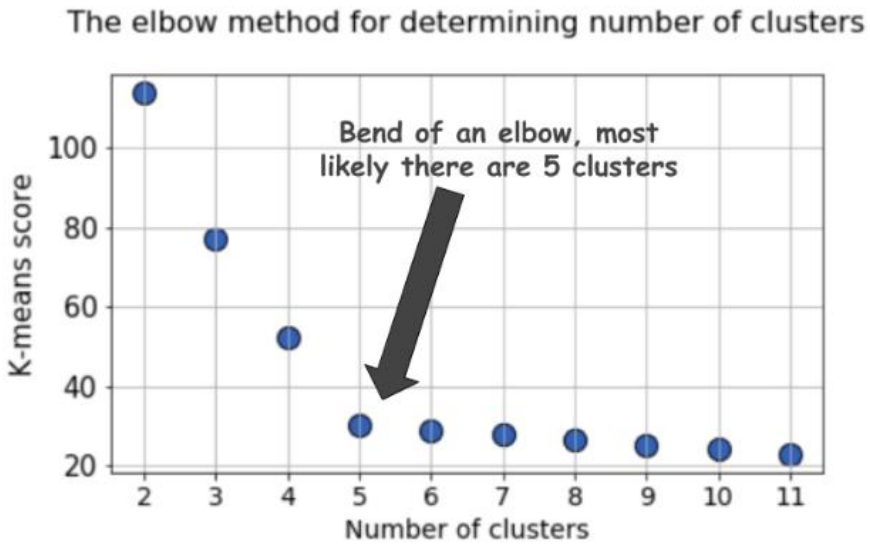
M-step:

$$\partial c_k \qquad \overleftarrow{i=1} \qquad \sum_{i=1} w_{ik}$$

4.2 Elbow Method

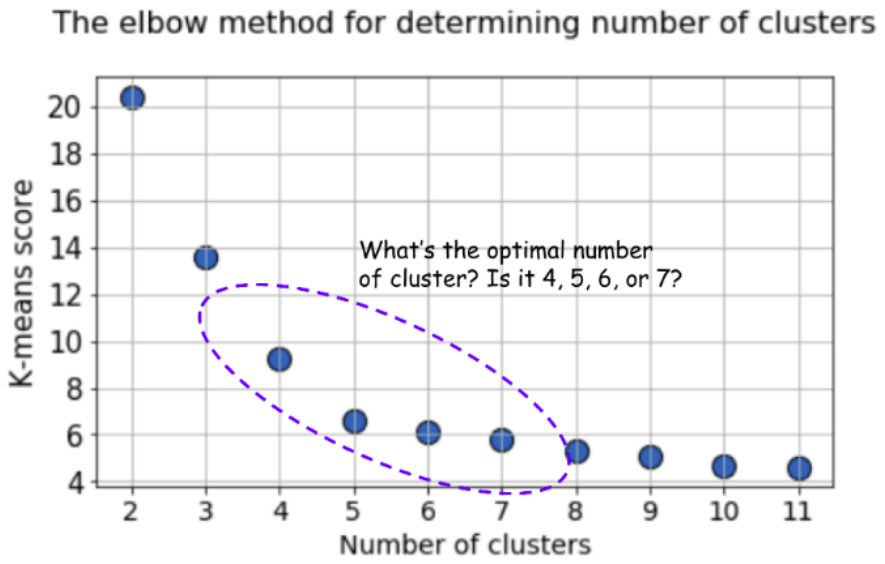
Cuando se trata de hallar el número de clústeres en el conjunto de datos o la muestra que tenemos existe la pregunta de cuantos grupos o en cuantos centros vamos a agrupar los datos. Entonces surge como respuesta el método Elbow[9]. Dicho método busca realizar correr el

método k-means con un k diferente cada vez y le asigna un puntaje a cada resultado, este basado en que también se agruparon los datos, luego de este test de grafican los resultados:



Resulta que por lo general las gráficas producen algo que más o menos parecido a un brazo y siempre hay un punto de inflexión en donde el puntaje suele empezar a ser el mismo, o sea que la tendencia de la variación en el score disminuye. Justo en ese momento se dice que allí puede estar nuestro k optimo y confidencialmente se produce donde parecería estar el codo del brazo. A pesar de ser sencillo este método no es muy fiable y muchas veces poco preciso dependiendo de la distribución de los datos y el ruido el codo no se muestra tan prominente y no se puede saber muy bien cuál es el k optimo, por lo que existe la necesidad de valernos de otras pruebas las cuales nos puedan ofrecer un segundo veredicto.

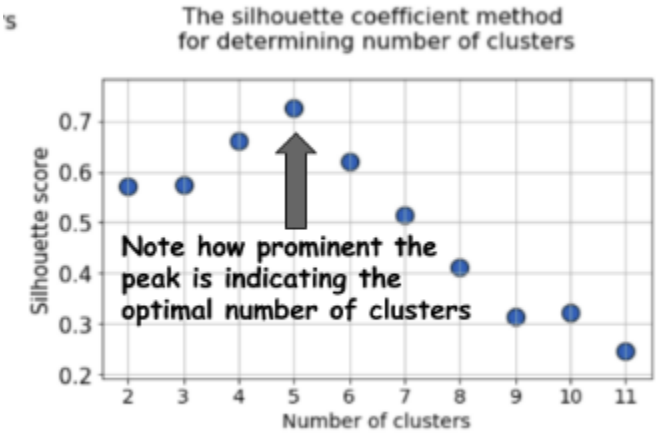
Aquí un caso cómo ejemplo donde los datos se encuentran con demasiado ruido y la gráfica no proporciona un punto de inflexión claro para poder tomar una decisión.



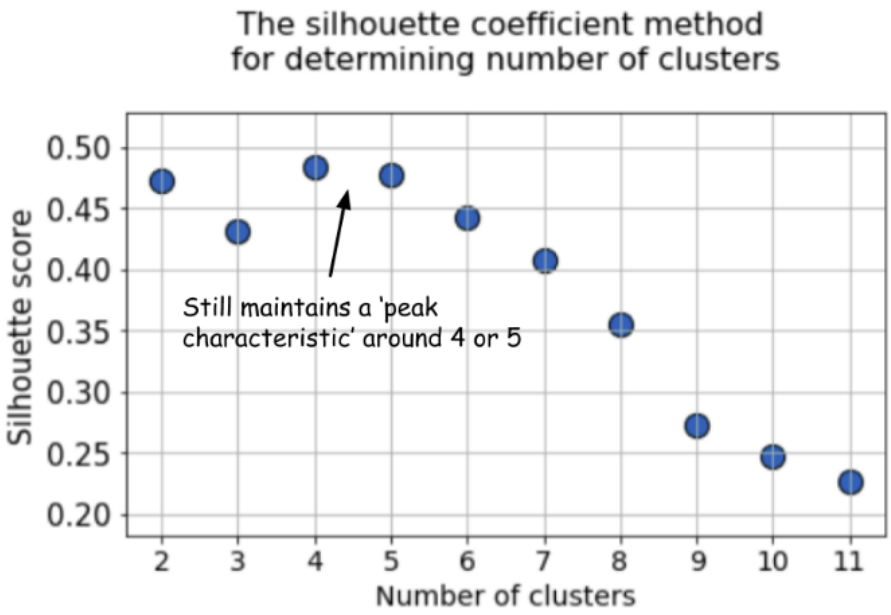
4.3 Silhouette analysis

El Silhouette analisys[10] suele ser el siguiente método para probar si nuestro primer acercamiento al k hallado por el método ELbow es correcto o si tal vez tenemos un valor para k más apropiado para nuestros datos. El método de la "Silueta" se basa en tomar cada muestra y calcular el coeficiente silhoette en el cual se toma la distancia "*mean intra-cluster a"* y la distancia "*mean nearest-cluster b"* luego el coeficiente de una muestra es (b - a) / Max (a, b). Sea calcula esto para todas las muestras y se toma como referencia para ver cuál sería el número indicado de clusters.

A continuación, un ejemplo de cómo suelen ser las gráficas de este método:



La diferencia entre el elbow y el silhoette no parece ser contundente pero cuando nuevamente agregamos ruido a los datos el silhoette muestra mejor el k óptimo para los datos:



5. Artículos propuestos para analizar

En este apartado veremos tres artículos referentes al tema de data preparación y data análisis en los cuales se aplicaron los métodos que vimos en esta práctica y muestran como son usados para lograr mejor interpretación de los datos y su uso diario el mundo del data análisis.

5.1 Refining Initial Points for K-Means Clustering.[11]

En este articulo vemos como el punto de inicio puede afectar la precisión del método k-means por lo que los autores nos presentan un método computacional para obtener un valor refinado de inicio con el cual se pueden alcanzar resultados más precisos y una mejor interpretación de los datos. Nos muestran también que esto es indiferente a la distribución de los datos, si son continuos o discretos y además mejora el tiempo de procesamiento ya que al tener un mejor arranque llegan a la respuesta y converge con mayor facilidad.

También nos presentan de forma interesante que aplicando este procesamiento logran obtener buenos resultados tomando un subconjunto de todos los datos por lo cual advierten que para problemas conde el tamaño de los datos sea incluso muy voluminoso pueden tomarse subconjunto y seguir teniendo resultados que representan la totalidad de los datos sin tener que analizarlos por completo.

5.2 Review on determining number of Cluster in K-Means Clustering [10]

En este Articulo se hace una revisión practica y teórica de los diferentes métodos que existen para calcular el k óptimo utilizando el método de k-means. Los autores apelan a este método ya que mencionan que los campos donde se están realizando hoy en día el estudio de los datos es demasiado basto, ya que va desde biología, psicología, ingeniería, bioinformática, economía, etc. Esto significa que son demasiadas arias para aplicar un problema de clustering (aglomeración de los datos por un determinado comportamiento, atributo o valor), entonces el método k-means es fácil de aplicar y entender, pero el problema resulta en cuantos clusteres particiono mis datos.

Es allí cuando en este articulo proponen el análisis de los siguientes métodos:

- By rule of thumb
- Elbow method
- Information Criterion Approach
- An Information Theoretic Approach
- Choosing k Using the Silhouette
- Cross-validation

Para concluir mencionan la importancia del conocimiento de los datos y algo relacionado a que los clústeres no están en la data sino en el ojo de quien los observa, aunque luego remarcan que los métodos son para buscar un mejor entendimiento y lograr fusionar estos dos paradigmas.

5.3 EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN

Este articulo me gusto en especial por que trataba un problema real en donde se necesitaba desarrollar una partición de clúster en un conjunto de nodos repartidos los cuales tenían una vida útil de batería limita así que debían optimizar la comunicación y agruparse de la forma más optima posible.

El articulo más que todo es practico y muestra como combino el método Elbow y LEACH, este último siendo uno de los primeros algoritmos para clisterizar datos luego eso lo une con una WSN (wireless sensor network), todo junto le da un algoritmo que el mismo creo específico para su problema alcanzando el número exacto de clúster para optimizar la red y el gasto energético.

Y al final es muy genial como muestra todos los resultados, siendo estos hechos en Matlab lo que demuestra que antes de lenguajes como python, Matlab era una de las mejores opciones para trabajar temas de Big data.

Bibliografía:

[1] Thomson, A. (2017, agosto), All the news. Retrieved from https://www.kaggle.com/snapcrack/all-the-news

[2] Spark documentation, (lastest). Retrieved from https://spark.apache.org/docs/latest/ml-features#tokenizer

[3] Spark documentation, (lastest). Retrieved from https://spark.apache.org/docs/latest/ml-features#stopwordsremover

[4] Spark documentation, (lastest). Retrieved from https://spark.apache.org/docs/latest/ml-features

[5] Ganesan, K. (year unknow) What are Stop Words?. Retrieved from https://kavita-ganesan.com/what-are-stop-words/#.XdewiVlKjeQ

[6] Spark documentation, (lastest). Retrieved from https://spark.apache.org/docs/latest/ml-features#countvectorizer

[7] Spark documentation, Retrieved from https://runawayhorse001.github.io/LearningApacheSpark/clustering.html

[8] Stephanie. (2015, September) EM Algorithm (Expectation-maximization): Simple Definition. Retrieved from https://www.statisticshowto.datasciencecentral.com/em-algorithm-expectation-maximization/

[9] Gove, R. (2017, December) Using the elbow method to determine the optimal number of clusters for k-means clustering. Retrieved from https://bl.ocks.org/rpgove/0060ff3b656618e9136b

[10] Sarkar, T. (2019, September). Clustering metrics better than the elbow-method. Retrieved from https://towardsdatascience.com/clustering-metrics-better-than-the-elbow-method-6926e1f723a6

[11] Bradley, P. S., & Fayyad, U. M. (1998, July). Refining Initial Points for K-Means Clustering. In ICML (Vol. 98, pp. 91-99).

[12] Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of Cluster in K-Means Clustering. International Journal, 1(6), 90-95.

[13] Bholowalia, P., & Kumar, A. (2014). EBK-means: A clustering technique based on elbow method and k-means in WSN. International Journal of Computer Applications, 105(9).