Proyecto: Conferencias mañaneras del presidente



Asignatura: Aprendizaje de Máquina

Profesora: Galindo Duran Cristal Karina

Integrantes:

* Cortés Cirett David
* Tinoco Videgaray Sergio Ernesto
* Rangel Méndez David
* Reyes Garnelo Uziel Bruno
* García King Ricardo Neftaly

Grupo: 5BV1

[**I. Introducción:**](#_r9e4udqod3qq) **2**

[**II. Desarrollo:**](#_ckl13pll1386) **3**

[Etapa 1 . Definición del objetivo.](#_i1d6hxr85mxo) 3

[Etapa 2. Obtención de los datos.](#_ifa4auqtrhm) 3

[Etapa 3. Preprocesamiento.](#_h22y6g90s7rc) 6

[Etapa 4. Procesamiento.](#_t9vf3quy240i) 9

[Etapa 5. Post-procesamiento.](#_omp2sn5azvo0) 17

[**III. Resultados:**](#_2qmzwejk3ln0) **19**

[**IV. Conclusiones:**](#_q3as6mtkszym) **21**

[**V. Referencias:**](#_lg5jgc1kdn5f) **21**

# 

# Introducción:

El aprendizaje es el proceso en el cual se obtiene conocimiento, habilidades, valores, y actitudes en el ser humano mediante el estudio, enseñanza o por medio de la experiencia adquirida con el tiempo, mientras que en 2019 Hernández B. [[6]](#_lg5jgc1kdn5f) menciona que “El aprendizaje es un proceso, una actividad cognitiva, psicomotora y afectiva, por medio de la cual se adquieren conocimientos, que en su movilización se desarrollan habilidades, destrezas y actitudes, en su aplicación permiten resolver y crear en cualquier contexto”, con esto podemos concluir que el aprendizaje es el proceso, actividad en la cual el ser humano mediante el estímulo emocional adquiere conocimientos, habilidades, información y experiencia que permite desarrollarnos en nuestro entorno, y utilizar ese aprendizaje, mejorando los conocimientos adquiridos previamente.

Llegando al aprendizaje de máquina se parte de las aportaciones en 2015 de El Naqa, y Murphy, M.J.[[7]](#_lg5jgc1kdn5f) aseguran que el aprendizaje de máquina es *“Un proceso donde con datos de entrada para obtener un resultado en particular, siendo un campo de estudio en el cual se le da a las computadoras la habilidad de aprender sin ser explícitamente programada (hard coded)”* mientras que Mitchel T. (1997) menciona que *“Un programa de computadora aprende de una experiencia E con respecto a una clase de tareas T y medida de desempeño P, si su desempeño en las tareas en T, de acuerdo con la medida P, mejora con la experiencia E.”* por lo que se concluye que el aprendizaje de máquina es un proceso en el cual un sistema o agente artificial se les proporciona la capacidad de aprender y mejorar de manera automática, a partir de ejemplos y/o experiencia, transformando los datos obtenidos en información que puede ser utilizada para diversos propósitos.

El Web Scrapping es un proceso utilizado en el aprendizaje de máquina realizando la función de extracción de datos por parte de sitios web de manera automática, simulando como un ser humano navegaria una página web, reconociendo la estructura de un código HTML, navegando a través de ella, minando y recopilando la información para posteriormente, aplicar preprocesamiento de los datos para analizarlos, realizando esta tarea de forma automática, teniendo una gran cantidad de usos.

El Aprendizaje No Supervisado es el modelo de aprendizaje agrupamiento o clustering, y es un modelo de aprendizaje que se basa en la búsqueda de similitudes entre los datos, con el fin de crear grupos para poder etiquetar los datos.

# Desarrollo:

## Etapa 1 . Definición del objetivo.

El objetivo del proyecto es agrupar las conferencias que realiza el presidente de México, Andrés Manuel López Obrador llamadas Mañaneras.

**Planteamiento**

Como etapa inicial se requiere la extracción de los diálogos de los videos y posteriormente guardarlos en archivos de texto, para esto originalmente se propuso descargar todos los videos de la mañanera existentes desde su inicio en 2018 hasta la fecha, utilizando una herramienta llamada **Pytube**[**[1]**](#_lg5jgc1kdn5f), una vez descargados estos vídeos procedía la extracción de diálogos mediante alguna herramienta de procesamiento natural del lenguaje para concluir esta primera etapa. Sin embargo, **Pytube** no obtenía correctamente los subtítulos de los videos volviendo poco práctica esta solución.

Con ese planteamiento en mente refinamos la idea para replicar la misma solución pero sin tener que descargar los videos, se usó una herramienta llamada **Youtube2text** que mediante los enlaces URL de los videos se descargaría únicamente el texto de los diálogos generados automáticamente o manualmente, sin embargo también generó una problemática ya que **Youtube2text**[**[2]**](#_lg5jgc1kdn5f) lleva mucho tiempo sin soporte y durante la ejecución esté dejaba de funcionar dejando incompleta la extracción.

Finalmente durante nuestro periodo de investigación se encontró la página oficial del presidente Andrés Manuel Lopez Obrador[[3]](#_lg5jgc1kdn5f), la cual contiene las transcripciones de todas las mañaneras que el presidente ha realizado durante su sexenio por lo que en esta ocasión se concluyó utilizar una técnica llamada **web scraping**[**[4]**](#_lg5jgc1kdn5f), proceso mediante el cual se “raspa” una página web para descargar información.

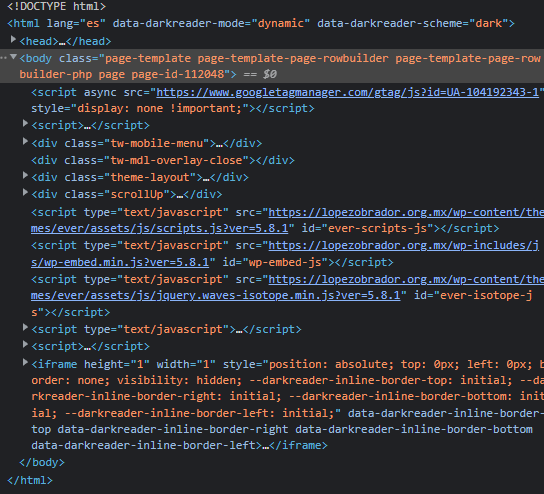
## 

## 

## 

## Etapa 2. Obtención de los datos.

El web scraping en python puede ser realizado mediante distintas librerías, en este caso fue usada la librería **beautifulsoup**. Cómo primer paso para la minería de los datos, se hizo un análisis de la estructura de la página web a minar, abriendo la herramienta **inspeccionar elementos** que los navegadores poseen se pudo visualizar parte del código necesario para el web scraping, se tuvo en cuenta un detalle importante para la extracción de datos y fue comprender la estructura de los componentes de la página en **HTML5**, leyendo el código observamos que necesitamos extraer la información de toda la etiqueta *<body>****,*** pero no bastaba con saber eso, ya que para el web scraping necesitábamos un identificador, tal como un **ID** o una **clase** según lo establece HTML5, en este caso usamos la **clase**, como se muestra en la figura 1.

****

*Figura 1. Consola del navegador.*

Esto fue bastante importante ya que realizamos un filtro usando la **clase** para no obtener información no deseada. Sin embargo esto no descartó la necesidad de preprocesar los datos, limpiarlos y obtener lo deseado de estos.

En la sección de web scrapping se comenzó importando las librerías **beautifulsoup** y **request**, este último nos ayudó a enviar solicitudes a la página web y con esto obtener el código HTML de la página web mientras que beautifulsoup es la herramienta para el análisis de la página HTML.

Si la página contiene paginación, y se necesita información de todas las páginas se necesita un contador para moverse dentro de la paginación, aumentando el contador hasta el final, esto puede hacerse como se muestra en la figura 2.

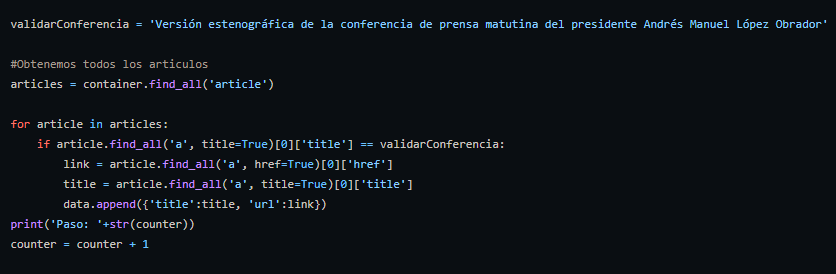


*Figura 2. Contador de paginación, con las 277 páginas en total.*

Ahora se buscó dentro de los contenedores, en dónde se encuentra la información requerida, ocupando una comparación del título para filtrar los datos de la página ya que este obtendrá todos los datos si no se hace esto, la comparación debe ser “Versión estenográfica de la conferencia de prensa matutina del presidente Andrés Manuel López Obrador”, este título es la validación que nos permite filtrar los

datos de baja importancia de los datos importantes, por lo que el código únicamente

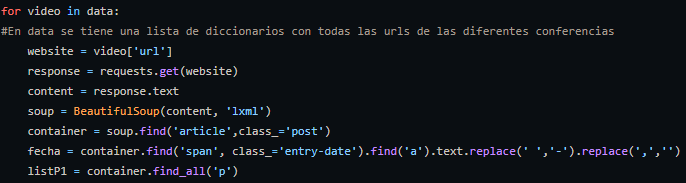
buscará y entrará a los contenedores que tengan esta validación como se ve en la figura 3.



*Figura 3. Validación del título.*

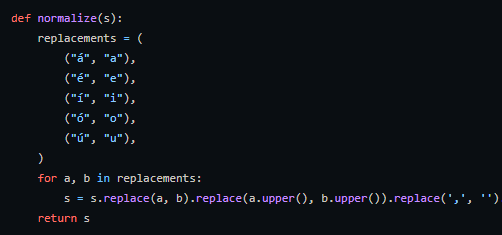
Si la validación da como resultado True, guardará el título en la variable title y el link de donde se obtuvo la coincidencia de la validación, finalmente añadimos los resultados en nuestra lista data, creando así una lista de diccionarios con link y título, finalmente se aumenta el contador para cambiar la página y realizar la búsqueda nuevamente.

Para obtener el texto dentro de los links se realiza un proceso similar al de la obtención de los títulos y links, con esto obtendremos las fechas y contenido de las mañaneras, las fechas al contener un espacio será reemplazado por un guión, y cualquier signo de coma será eliminado, como se muestra en la figura 4.

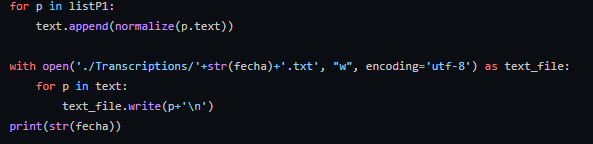


*Figura 4. Obtención de fechas.*

Finalmente obtenemos el texto dentro del contenedor de la página, en esta parte se aplica un pre preprocesado, reemplazando signos de puntuación, comas, acentos, signos de interrogación, exclamación entre otros más, finalmente se crea una carpeta y se guardan todos los archivos en formato txt, para su posterior preprocesado como se muestra en la figura 5 y 6.



*Figura 5. Normalización del texto*



*Figura 6. Obtención del texto normalizado*

## 

## Etapa 3. Preprocesamiento.

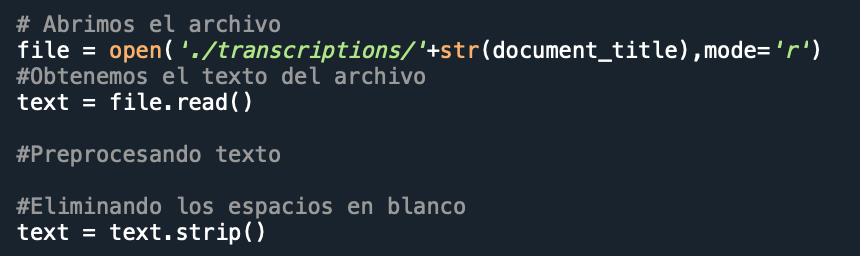
Para la etapa de preprocesamiento se utilizaron las librerías NLTK y spacy. NLTK es una librería que funciona para remover stopwords y spacy para la tokenización y lematización.

Primeramente, lo que se realizó fue obtener todos los paths de los archivos con el texto ya normalizado y se mandaron, uno por uno, a nuestra función de preprocesamiento de texto.



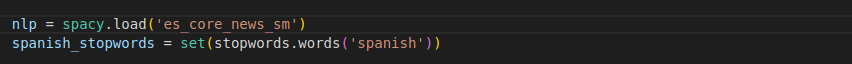
*Figura 7. Obtención de paths de archivos con textos normalizados y llamada a función de preprocesamiento.*

En la función de preprocesamiento, se comenzó abriendo el archivo, leyéndolo y eliminando espacios del inicio y del final.

**

*Figura 8. Lectura de archivo*

Posteriormente, se crea una instancia de Spacy en lenguaje español para el procesamiento de texto y se crea un Set con las StopWords que genera NLTK.

  
*Figura 9. Instanciación de Spacy y creación de set de stopwords*

Una vez que se tiene lo anterior, se tokeniza el texto en español utilizando la instancia de Spacy y se inicia una lista vacía. Una vez que se hace esto, se itera sobre los tokens.

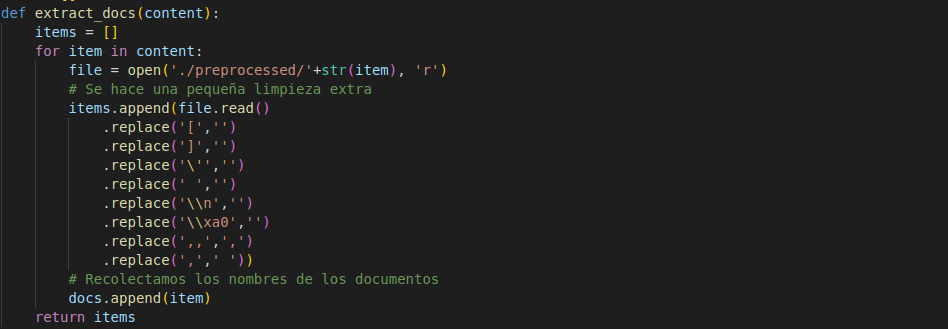
En caso de que el token sea una puntuación o una stopword, pasa a la siguiente iteración. En caso de que no lo sea, se agrega a la lista que se creó al inicio.

Una vez que termina con todos los tokens, escribe todos los que no fueron stopwords o puntuaciones en un nuevo archivo en la carpeta de “preprocessed”.



*Figura 10. Remoción de stopwords y puntuaciones y escritura de archivo.*

Una vez que se escriben los archivos se realiza una última limpieza en la que se eliminan otras puntuaciones que pudieron no haberse eliminado en la primera.



*Figura 11. Remoción de últimas puntuaciones sobrantes.*

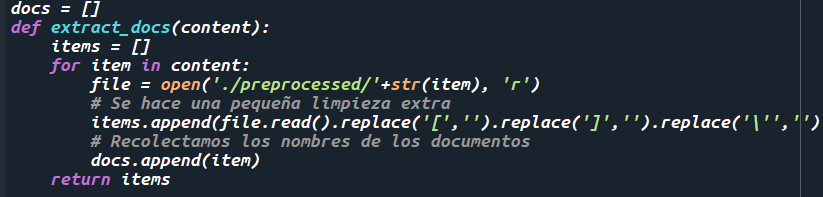
## 

Una vez realizado el preprocesamiento anterior, aún falta remover algunos caracteres especiales propios de los archivos de texto. Algunos de estos símbolos, se pueden observar en la figura 12.



*Figura 12. Archivo preprocesado (no final).*

Se puede observar que efectivamente las palabras junto con sus conjugaciones y variantes fueron reducidas hasta el punto de tener su forma canónica o lema, pero aun se puede distinguir que los espacios son considerados en esta parte por lo que la limpieza aún no se ha completado del todo. Importando la librería os, permitirá al código manipular archivos del sistema, entrar a carpetas, etcétera, con los datos previamente preprocesados, se procede a realizar el corpus, dicho de otra forma, se procede a realizar una colección de todo el texto en nuestro idioma para ser analizado, dicho en pocas palabras son todos los datos obtenidos en el preprocesamiento que servirán para el análisis en cuestión, primero se crea una lista dondé se realizará la limpieza faltante, eliminando corchetes, comillas, etcétera, esta lista guardará los nuevos documentos limpios, listos para procesar, la limpieza se encuentra en una función, como se ve en la figura 13.



*Figura 13. Limpieza de los datos preprocesados.*

## 

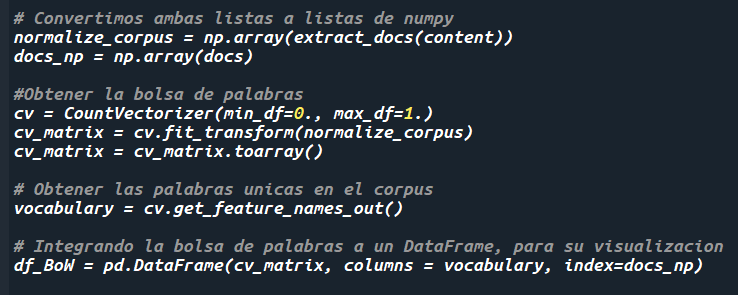
La función anterior será llamada y convertida en una lista de numpy, igualmente se convierten los nombres de los documentos en una lista, con esto se procede a realizar el procesamiento.

## 

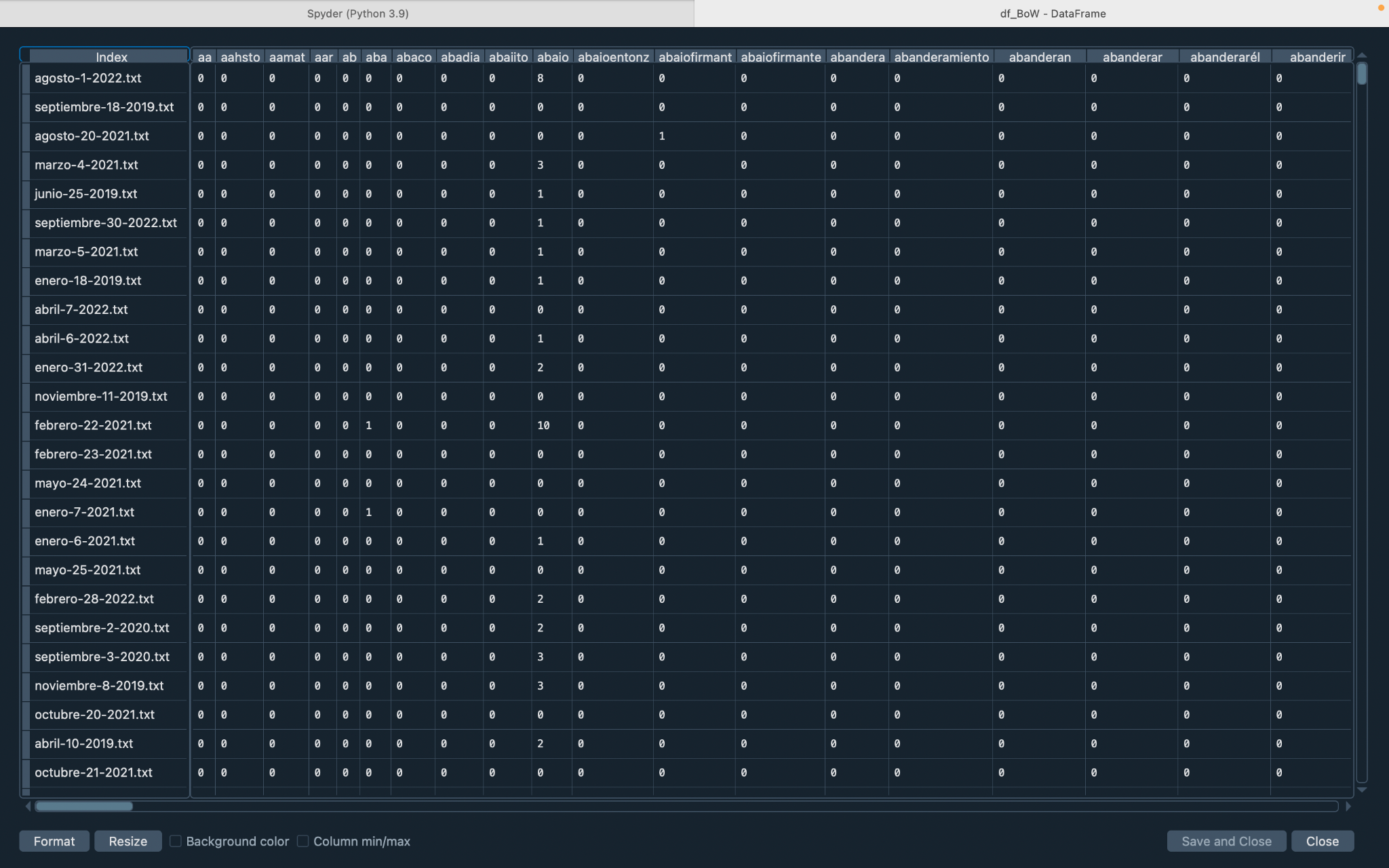
## 

## Etapa 4. Procesamiento.

Para comenzar esta etapa se computará la bolsa de palabras, contando las palabras iguales, transformándose en una matriz, scikit devuelve como resultado una lista en la que coinciden los índices de cada una de las palabras de la lista con la matriz. Una vez obtenida la bolsa de palabras se integra en un dataframe para visualizar la información de una mejor manera, como se muestra en las figuras 14 y 15.

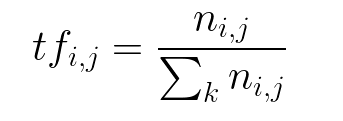


*Figura 14. Obtención de bolsa de palabras.*



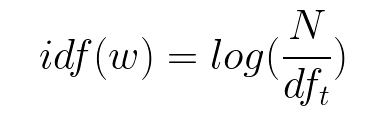
*Figura 15. Bolsa de palabras. (Bag of Words).*

Una vez obtenida la bolsa de palabras se comienza a implementar la frecuencia de término (TF) y frecuencia de documento inversa (IDF), con esto mejorando la comprensión en el enfoque de la bolsa de palabras, dónde el TF obtiene la proporción del número de veces que la palabra se encuentra en los datos comparado con el total del número de palabras en todo el documento, la siguiente figura muestra la fórmula que el TF utiliza.



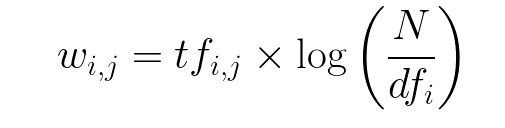
*Figura 16. Fórmula de la frecuencia de término.*

Mientras que el IDF obtiene el peso de las palabras más raras dentro de los datos en el corpus, esas palabras raras en el corpus tienen un peso más grande, calculando el peso como se muestra en la figura 17.



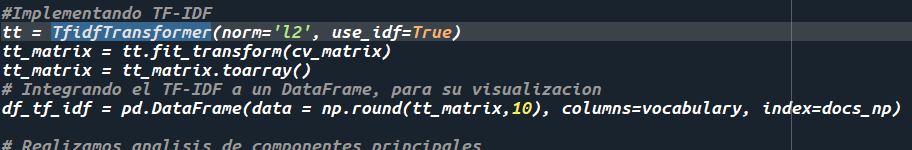
*Figura 17. Fórmula de la frecuencia de documento inversa.*

La combinación de ambas fórmulas como un producto da como resultado el peso de cada una de las palabras dentro de los datos, como se muestra en la figura 18.



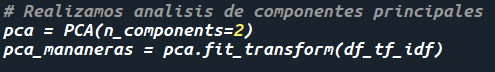
*Figura 18. Fórmula TF-IDF.*

Con ***TfidfTransformer*** convertimos la colección de datos en una matriz normalizada, como se muestra en la figura 19.



*Figura 18. TF-IDF Transformer.*

Una vez completado el proceso anterior sigue el análisis de componentes principales como se muestra en la figura 19.



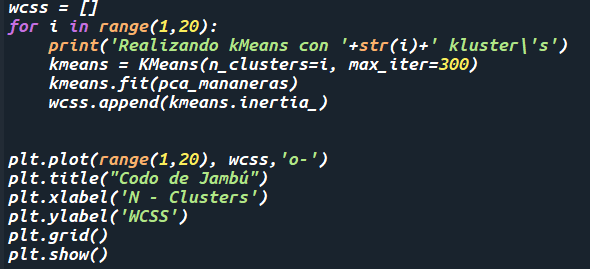
*Figura 19. ACP.*

Con esto se puede realizar el análisis de componentes principales, para poder obtener los clusters que son los óptimos para realizar el proceso de agrupamiento.

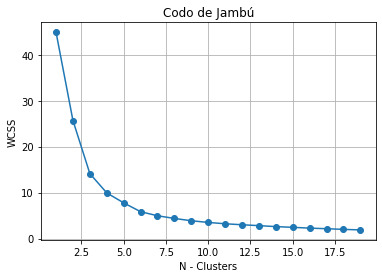
**Métodos estimadores clusters para cada algoritmo.**

Método del codo para KMeans.

Con el método del codo obtendremos el número de clusters para Kmeans como se puede ver en la figura 20 y 21.



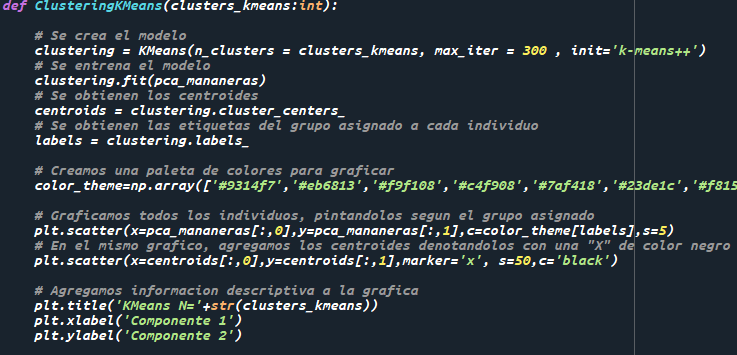
*Figura 20. Codo de Jambú.*



*Figura 21. Gráfica Codo de Jambú.*

Analizando la gráfica se puede obtener que entre más clusters disminuye el valor de WCSS, siendo WCSS la suma de los cuadrados de las distancias, y que como su nombre lo dice al momento de que WCSS disminuye su valor forma en la gráfica una especie de codo, esto cuando el número de clusters aumenta.

Con esto podemos decir que lo óptimo es realizar 4 clusters, sin embargo, para este caso se toman los valores en un rango de 3 a 5, dado a que se puede observar que en el codo no se encuentra con un error tan notable en este rango. Una vez obtenido el criterio de para obtener el número de clusters, podemos implementar en KMeans, mediante una función como se muestra en la figura 22 y 23.

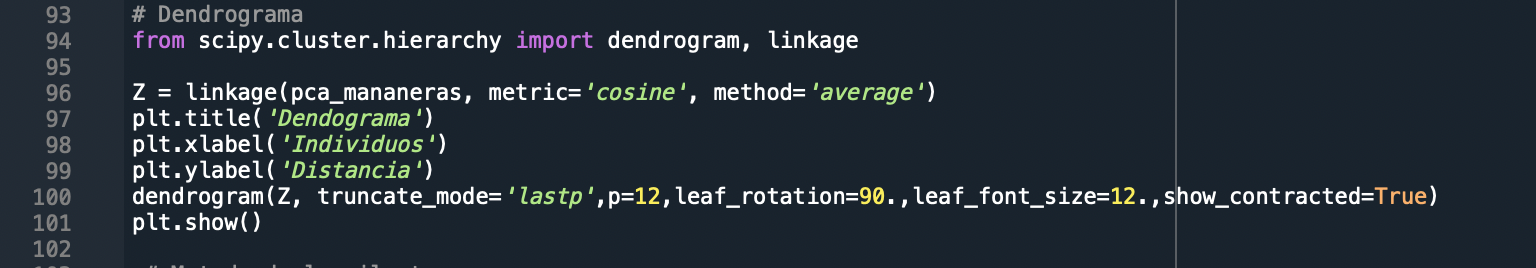


*Figura 22. Código Kmeans.*

Dendrograma para clustering jerárquico.

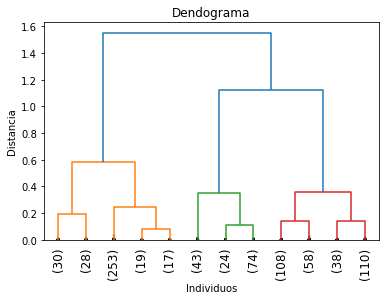
Para la creación del dendrograma, se usó la métrica de distancia coseno el método de promedio, porque son las métricas que se recomiendan para el manejo de textos.

El siguiente fragmento de código muestra cómo se realizó el dendrograma.



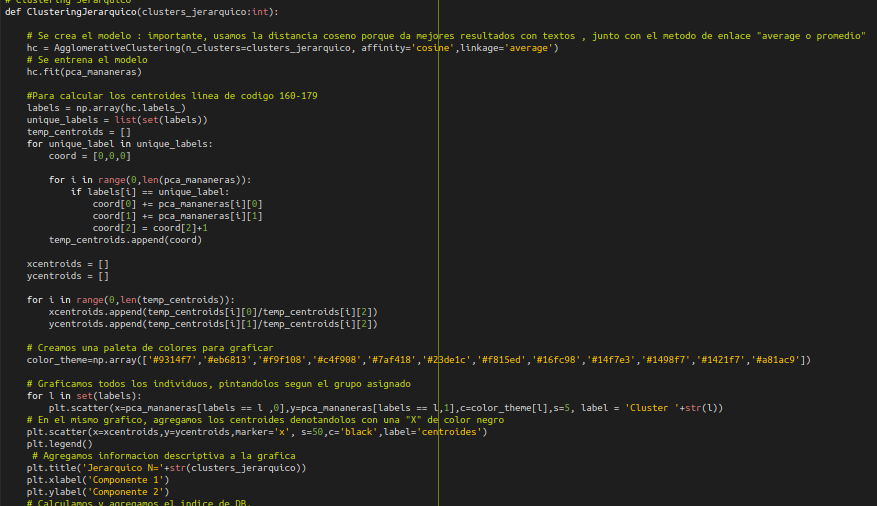
*Figura 23. Codificación del dendograma*

El resultado se muestra en la figura 24, donde después de un análisis, se determinó que el número de clusters recomendados para la agrupación jerárquica es 3.



*Figura 24. Gráfica del Dendograma*

Para concluir con el agrupamiento jerárquico, se creó el código que permite realizarlo y se muestra en la figura 25.

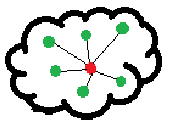


*Figura 25. Codificación del algoritmo de agrupación jerárquico.*

**Método de la silueta para el modelo de mezclas gaussiano.**

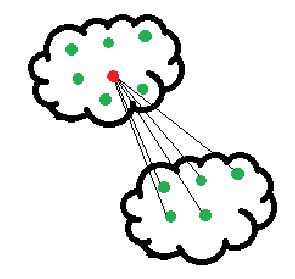
Por último para el método de mezcla gaussiano (Gaussian Mixture Metods, GMM), siendo un método basado en distribución, utilizando métodos probabilísticos y no utilizando un centroide, para esto se utiliza el método de la silueta, donde se mide la distancia de separación de los clusters, indicando que tan cerca se encuentra cada punto del cluster de los puntos cluster vecinos, donde se obtiene en valor y este valor entre más alto indica un cluster mejor, el cálculo se realiza mediante dos medidas, donde:

1. a(x): Cohesión indica la distancia promedio de “x” y todos los puntos que pertenecen a la misma clase, es decir el promedio de distancia dentro del cluster, esto se puede visualizar en la figura 26.



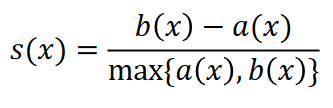
*Figura 26.Cohesión de a(x).*

1. b(x): Separación, es la distancia promedio de “x” con todos los puntos del siguiente cluster más cercano, es decir se busca el cluster más cercano y se computan las distancias , como se muestra en la figura 27.



*Figura 27. Separación b(x).*

Dado esto, el coeficiente de la silueta se obtiene como muestra la figura 28.

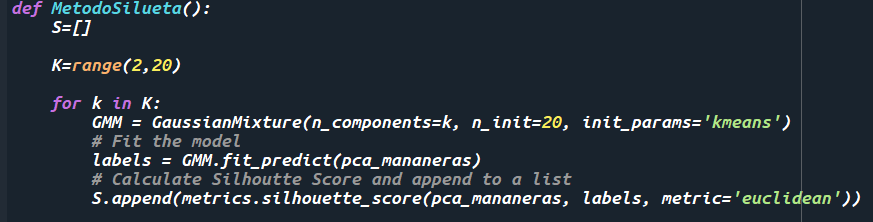


*Figura 28. Método de la silueta.*

Tomando valores entre 1 y -1, dónde:

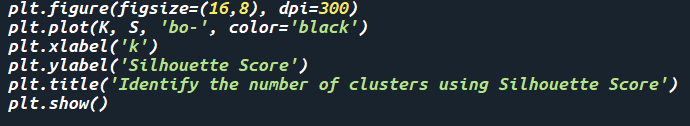
* -1 es considerado un mal agrupamiento (el cluster no está correctamente realizado).
* 0 se considera indiferente (no se han logrado separar de manera correcta los datos en grupos representativos).
* 1 en un buen agrupamiento (es lo que se busca generalmente).

Con esto se procede a realizar un método para calcular el método de la silueta, primero se establece un número de clusters, en este caso se realizan de 2 a 20 clusters, se crea una lista vacía para almacenar el valor del coeficiente de la silueta para cada valor del cluster, con un ciclo for se realiza el cómputo de los datos, agrupando para la cantidad de clusters que indica el ciclo, extrayendo las etiquetas y almacenando los el coeficiente de la silueta para el número de clusters en particular, como se muestra en la figura 29.

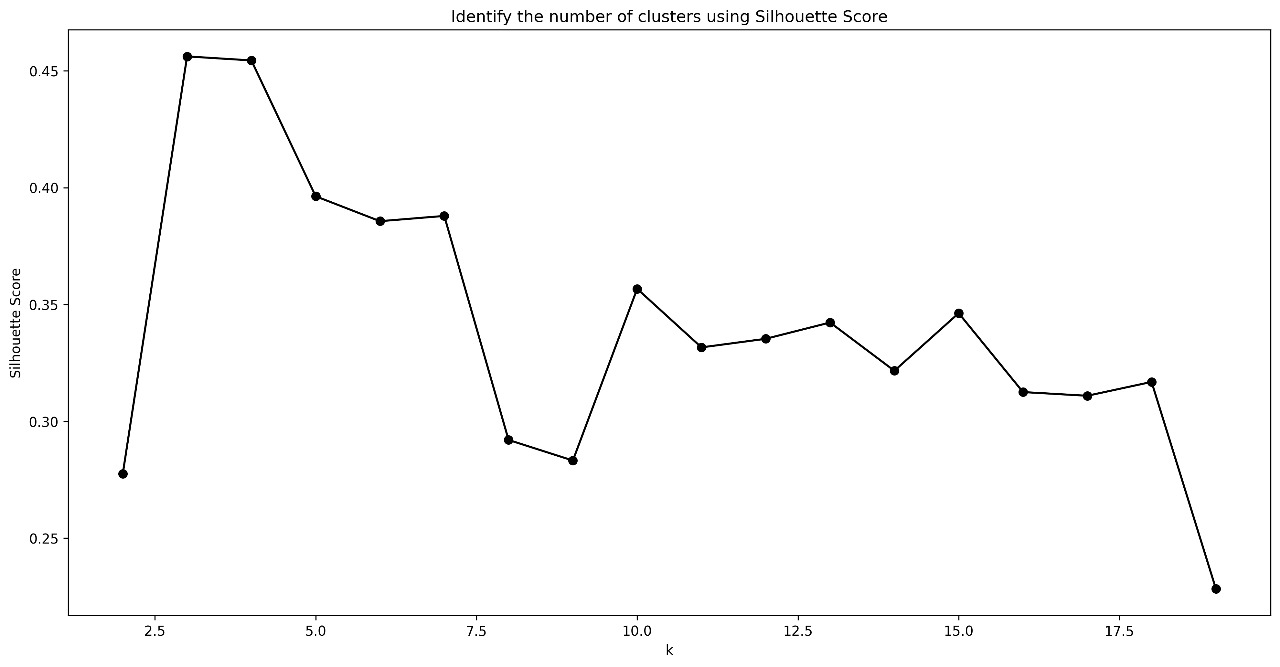


*Figura 29. Obtención de los coeficientes de la silueta.*

Finalmente procediendo a graficar lo obtenido y analizar el resultado, como se muestra en las figuras 30 y 31.



*Figura 30. Método de graficación.*



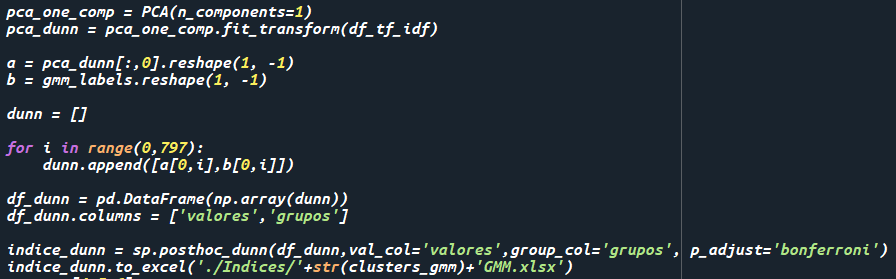
*Figura 31. Método de la silueta.*

Una vez analizada la gráfica se puede ver que el cluster óptimo se encuentra en 3 y 4, sin embargo, en este punto se tomó en cuenta el método del codo, para finalmente se tomaron en cuenta los valores de 5 y 6.

Con esto se define el GMM como se muestra en la figura 31 y 32.



*Figura 31. Código GMM.*

**

*Figura 32. Código GMM-Dunn.*

## Etapa 5. Post-procesamiento.

El objetivo de esta etapa es intentar determinar qué tan buenos o no, fueron los agrupamientos. En este caso se usó índice de Dunn y Davies-Bouldin(DB) para determinar este resultado.

El índice de Davies-Bouldin(DB) indica clústeres compactos, y cuyos centros están bien separados unos de los otros y el índice de Dunn [[8]](#_lg5jgc1kdn5f).

El índice de dunn mide la separación entre dos grupos genéricos C h y C h′ se mide por la distancia euclidiana mínima observable entre los documentos más cercanos pertenecientes a cualquier par de grupos [[9]](#_lg5jgc1kdn5f).

Con esto en mente, se comienza la codificación de estos índices.

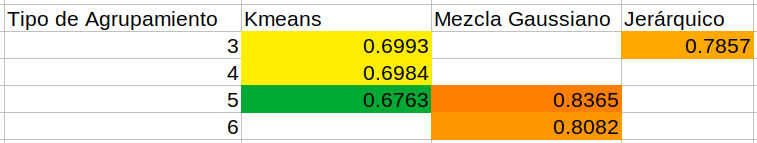
Para el cálculo del índice de Davies-Bouldin, se usa la librería de sklearn.metrics, llamando la función: davies\_bouldin\_score, que pide como parámetros los datos agrupados y el grupo asignado.



*Figura 33. Obtención de Db*

Dentro de las funciones de clustering (KMeans, Jerárquico y GMM) se usó el mismo procedimiento.

El índice del resultado de cada uno de los agrupamientos, se muestra en sus respectivas gráficas, pero se recopilaron todas en una tabla (véase en la tabla 1), para una mejor presentación.

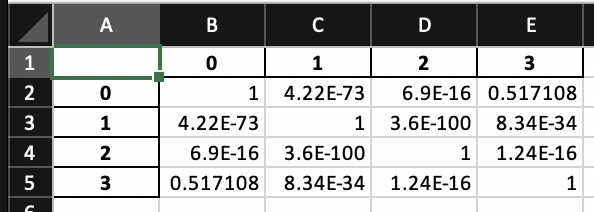
**

Lo mismo se realizó con el índice de Dunn, sin embargo fue necesario convertir los datos a una sola componente, por lo que de nuevo se utilizó la función de obtención de componentes principales con la diferencia de que en los parámetros se específico 1 componente. Con esto se creó un arreglo bidimensional con los valores de la componente y los grupos asignados.



*Figura 34. Proceso de obtención de índices Dunn y DB en KMeans*

Mientras que para el índice de Dunn se exportó a un archivo de excel, resultando la figura 36.



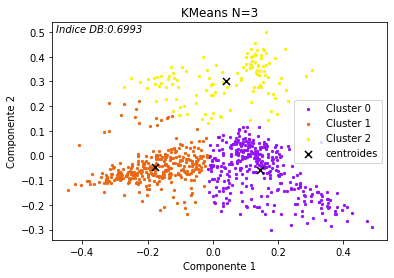
*Figura 36. Índice Dunn en formato .xlsx de KMeans, 4 clústeres*

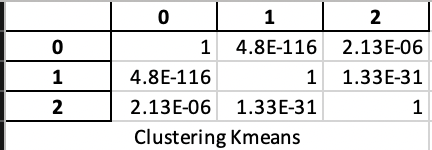
Se ejemplifico con clustering KMeans sin embargo los resultados completos con las diferentes funciones de clusterizado y N cantidad de clústers se mostrará en la sección de **resultados**.

# Resultados:

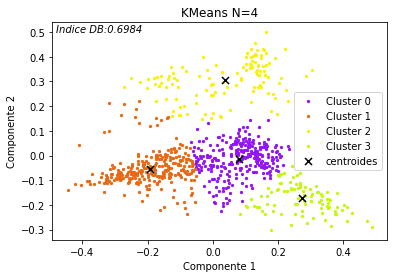
A continuación se presentan gráficamente los resultados de los clusters además de los índices tanto de DB como de Dunn, separando las secciones por el tipo de clustering aplicado.

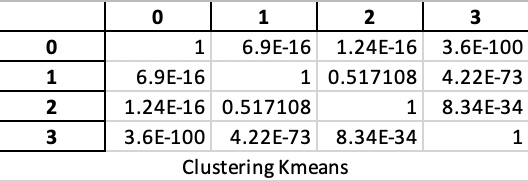
**-KMeans**

*****KMeans con 3 clusters e índice de DB*

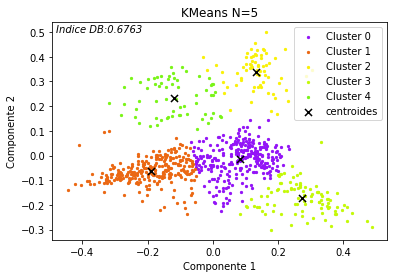


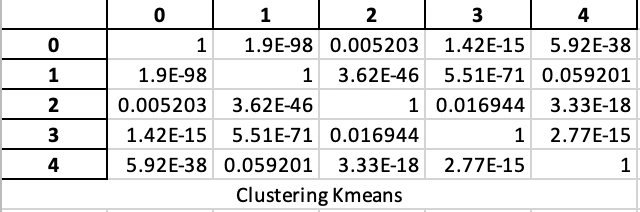
*Índice de Dunn con 3 clusters*

*KMeans con 4 clusters e índice de DB*



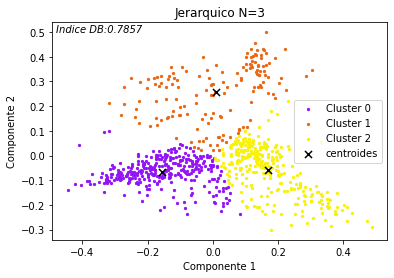
*Índice de Dunn con 4 clusters*

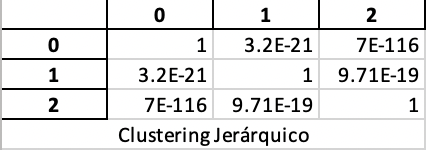
*KMeans con 5 clusters e índice de DB*



*Índice de Dunn con 5 clusters*

* **Jerárquico**

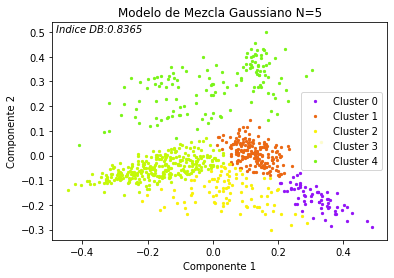


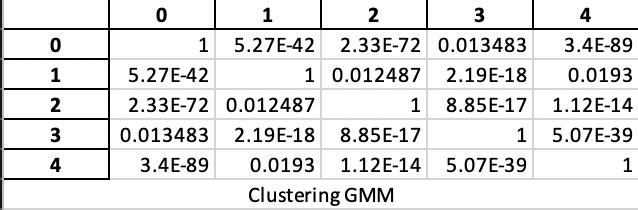


*Índice de Dunn con 3 clusters*

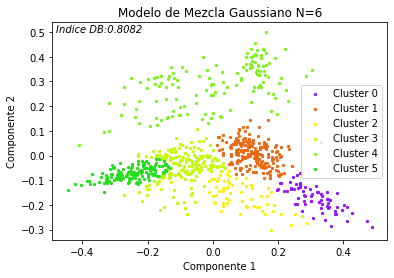
*Jerárquico con 3 clusters e índice de DB*

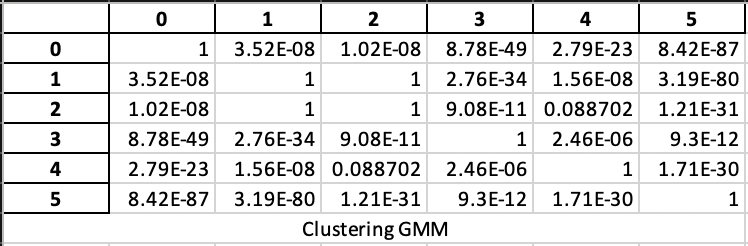
* **Gaussian Mixture Models**

*GMM con 5 clusters e índice DB*



*Índice de Dunn con 5 clusters*

*GMM con 6 clusters e índice DB*



*Índice de Dunn con 6 cluster*

Aunque se aplicaron diferentes métodos de agrupamiento, en base a los índices, y de forma visual, el algoritmo que da mejores resultados para el caso de uso de esta aplicación es el K Means, más en concreto con un número k=5 , cinco grupos.

# Conclusiones:

Aunque sería ideal que las máquinas pudieran entender el lenguaje humano sin más, la realidad es que aún estamos más que lejos de eso. Por este motivo se buscó la forma de expresar las palabras mediante un modelo numérico, tratando de medir la “distancia” entre ellos, lo que permite contextualizarlos y darle sentido, esto con el fin de ser procesados por los modelos generados.

Ahora, para poder generar este tipo de modelos, primero es necesario el procesamiento pues, sin esto, existiría un gran sesgo en la información y se estaría procesando mucha información que, en realidad, la mayoría de las veces no aporta. Sin el uso de las técnicas de preprocesamiento, los modelos que existen actualmente no podrían tener los resultados que nos generan. Es por eso que el preprocesamiento es tan importante como el procesamiento.

# Referencias:

[[1]](#_i1d6hxr85mxo) *pytube — pytube 12.1.0 documentation*. (n.d.). Retrieved October 23, 2022, from <https://pytube.io/en/latest/>

[[2]](#_i1d6hxr85mxo) *youtube2text*. (2022, March 24). PyPI. Retrieved October 23, 2022, from <https://pypi.org/project/youtube2text/>

# 

[[3]](#_i1d6hxr85mxo) EDITOR, EDITOR, EDITOR, EDITOR, EDITOR, EDITOR, EDITOR, EDITOR, EDITOR, & EDITOR. (n.d.). *AMLO – Sitio Oficial de Andrés Manuel López Obrador, Presidente de México*. Retrieved October 21, 2022, from <https://lopezobrador.org.mx/>

[[4]](#_i1d6hxr85mxo) Colaboradores de Wikipedia. (2022, August 18). *Web scraping*. Wikipedia, La Enciclopedia Libre. Retrieved October 21, 2022, from <https://es.wikipedia.org/wiki/Web_scraping>

[[5]](#_mzkyjk2dt54f) Skiena, S. S. (2017). *The Data Science Design Manual*. Springer Publishing.

[[6]](#_r9e4udqod3qq) Brambila Hernández, A. (2019). *La Enseñanza Aprendizaje de la Historia en el Currículo 2011 en Escuelas Secundarias del Estado de Tlaxcala*. Retrieved October 20, 20 C.E., from <https://posgradoeducacionuatx.org/pdf2019/D053.pdf>

[[7]](#_r9e4udqod3qq) Naqa, I. E., Li, R., & Murphy, M. J. (2015). *Machine Learning in Radiation Oncology: Theory and Applications*. Springer Publishing.

[[8]](#_omp2sn5azvo0) León Guzmán, E. (n.d.). *Métricas para la validación de clustering*. Universidad Nacional De Colombia. Retrieved October 23, 2022, from <https://disi.unal.edu.co/~eleonguz/cursos/mda/presentaciones/validacion_Clustering.pdf>

[[9]](#_omp2sn5azvo0) Michelangelo Misuraca, Maria Spano & Simona Balbi (2019) BMS: An improved Dunn index for Document Clustering validation, Communications in Statistics - Theory and Methods, 48:20, 5036-5049, <https://www.tandfonline.com/doi/full/10.1080/03610926.2018.1504968>