MINERÍA DE DATOS EN UN SERVIDOR LOCAL PARA CLASIFICAR PALABRAS POR EL MÉTODO DE LOS COSENOS

DATAMINING IN A LOCAL SERVER TO CLASSIFY WORDS BASED ON COSINE METHOD

Salvador Manuel Malagón Soldara
Tecnológico Nacional de México / IT de Celaya, México
salvador.malagon@itcelaya.edu.mx

Juan Paulo Maldonado Rodríguez
Tecnológico Nacional de México / IT de Celaya, México
15030927@itcelaya.edu.mx

José Vladimir Maldonado Espino
Tecnológico Nacional de México / IT de Celaya, México
15030824@itcelaya.edu.mx

Recepción: 29/abril/2020 **Aceptación:** 29/octubre/2020

Resumen

En el presente artículo se muestra un algoritmo de minería de datos capaz de clasificar palabras. El método utilizado para la clasificación es el método de los cosenos. Adicionalmente, se utiliza una librería para eliminar los stopwords que puedan causar ruido en la clasificación. El lenguaje de programación empleado es Python en el sistema operativo Windows. Por otra parte, la base de datos es conformada por varios archivos de texto con oraciones de distintas temáticas. Los resultados obtenidos permiten obtener un grado de similitud entre los archivos, y por lo tanto, identificar temas en común entre ellos. Esta aplicación es bastante útil cuando se tienen grandes cantidades de datos, ya que se puede identificar un posible cliente entre comentarios en una página de internet. Por último, para identificar mejor las clasificaciones encontradas por la minería de datos, los resultados fueron exhibidos en diagramas de Venn, clustering de documentos y gráficos de similitud.

Palabras Clave: clasificador de palabras, método de los cosenos, minería de datos.

Abstract

This article presents a data mining algorithm capable of classifying words. The method used for classification is the cosine method. Additionally, a library is used to eliminate stopwords that can cause noise in the classification. The programming language used is Python in the Windows operating system. On the other hand, the database is made up of several text files with sentences on different topics. The results obtained allow obtaining a degree of similarity between the files, and therefore, identify common themes between them. This application is quite useful when you have large amounts of data, since a possible client can be identified between comments on an internet page. Finally, to better identify the classifications found by data mining, the results were displayed in Venn diagrams, document clustering, and similarity graphs.

Keywords**:** *classify words, cosine method, demining.*

1. Introducción

Hoy en día la minería de datos es un área de trabajo muy importante debido al big data de la población mundial. El big data es un término que describe el gran volumen de datos, tanto estructurados como no estructurados, que inundan los negocios cada día. Pero no es la cantidad de datos lo importante, lo que importa con el big data es lo que las organizaciones hacen con los datos. En general, la minería de datos (a veces llamada descubrimiento de datos o de conocimiento) es el proceso de analizar y procesar estos datos desde diferentes perspectivas, reduciéndose en información útil, es decir, información que se puede utilizar para aumentar los ingresos, reducir los costos, o ambas cosas. Aunque “minería de datos” es un término relativamente nuevo, la tecnología no lo es. Las compañías han utilizado equipos de gran alcance para tamizar a través de volúmenes de datos de escaneo de los supermercados y analizar los informes de investigación de mercado durante años [Ahmed, 2009].

La minería de datos es muy útil en los siguientes dominios:

- Análisis y gestión del mercado.
- Análisis empresarial y gestión de riesgos.

- Detección de fraudes.

Aparte de los anteriores, la minería de datos también se puede utilizar en las áreas de control de producción, retención de clientes, exploración científica, deportes, astrología y navegación web en Internet. Por ejemplo, hablando específicamente del mercado, permite que las empresas determinen las relaciones entre los factores "internos" como el precio, posicionamiento del producto, o las habilidades del personal, y factores "externos", como los indicadores económicos, la competencia, y la demografía de los clientes [Xu, 2014].

La empresa “Portal B2B Neumáticos Soledad”, empleó la minería de datos para resolver el cómo modificar el portal de compra online que usan los talleres asociados para aumentar las ventas por este canal. La solución que encontraron fue el extraer patrones de comportamiento de los usuarios sobre el motor de búsquedas del portal, analizando aquellas búsquedas que terminan en pedido y las que no. Si bien la tecnología de la información a gran escala ha ido evolucionando por separado las transacciones y sistemas de análisis, la minería de datos proporciona un enlace entre los dos.

Ahora, comprendiendo las bases fundamentales, así como los conceptos básicos de la minería de datos, en el presente artículo se pretende implementar mediante el software Python, un programa para la clasificación de datos, donde se podrá hacer una filtración de la información de interés.

2. Métodos

Python es un lenguaje de programación, scripting independiente de plataforma y orientado a objetos. Posee muchas herramientas para *data mining*. Una librería muy importante la cual fue usada para el programa es *Sklearn*. Esta librería contiene utilerías de *machine learning,* parte central del programa que se encarga de la recopilación y acomodo de datos [Middelfart, 2013], [Adil, 2018].

Las tres partes en las que se divide el programa, son:

- **Capturar datos en archivos de texto tipo .txt**: Se generó una carpeta con 100 documentos de texto codificados en UTF-8 los cuales contenían textos

de diferente índole, para posteriormente leerlos, extraerlos mediante ciclos for, y almacenarlos en listas de Python (tabla 1).

Tabla 1 Ejemplo de almacenamiento en Python.

Documento 1	El perro es un animal de 4 patas
Documento 2	El sol es considerado una estrella
Documento 3	Cada animal es un ser vivo, que siente

- **Procesamiento de la información**: Es necesario eliminar palabras irrelevantes que sólo causan ruido, llamadas "*stopwords*". Estas palabras son adjetivos, preposiciones, artículos, etc. por ejemplo: el, ella, ustedes, con, contra, cabe, entre otras. *Python* tiene un comando especial para quitar esas palabras "stopwords.words".

  Posteriormente, se genera matrices TF-IDF (*Term frequency - Inverse document frequency*). Posteriormente, se obtendrán las palabras más importantes asignándoles un valor bajo, para poder trabajar con valores pequeños y manejables, ya que TF-IDF maneja algoritmos (Tabla 2).

Tabla 2 Ejemplo de una matriz TF (term frequency).

	perro	animal	4	patas	sol	considerado	estrella	ser	vivo	siente
Doc. 1	1	1	1	1	0	0	0	0	0	0
Doc. 2	0	0	0	0	1	1	1	0	0	0
Doc. 3	0	1	0	0	0	0	0	1	1	1

  Mediante las TF-IDF se asignan diferentes pesos a los textos para así determinar qué tanta es la relación entre uno y otro. Usando la función get_features_names. Posteriormente, mediante el método de cosenos (cosine_similarity) en data mining, se obtiene una distancia al comparar dos matrices TF-IDF, la cual arrojó un valor que se interpreta como el grado de similitud entre los dos textos, donde el número 1 representa los textos iguales.
- **Visualización de la información**: El programa ofrece al usuario la posibilidad de escoger entre 4 diferentes métodos de visualización de los datos, gracias a las funciones directas de las librerías de Python, los cuales son:

- ✓ Impresión por circunferencias del método del coseno.
- ✓ Clustering de distancia entre documentos.
- ✓ Clustering de documentos en 3D.
- ✓ Similitud de documentos (dibujar distancia entre ellos).

3. Resultados

Para implementar el algoritmo, se creó una serie de documentos con información de computadoras, excepto uno (D1) que contenía información de otro tema totalmente diferente y posteriormente se realizó una comparación de documentos uno a uno (ver Gráfica 1). Donde se compara el total de palabras, sin contar las palabras consideradas "basura" como preposiciones o artículos. En general, se desprecian palabras que no muestran algún aporte importante o que no cuenten con gran peso. Gracias a "stopwords", función que Python otorga dentro de la librería de *sklearn*, se puede realizar la búsqueda de forma totalmente automática.
Para la prueba de minería de datos se tomaron en cuenta 4 gráficas:

- Impresión de similitud de documentos por método de coseno (Intersección de circunferencias).
- *Clustering* de distancia entre documentos.
- *Clustering* de distancia entre documentos en 3D.
- Similitud entre documentos (Dibujar distancia entre ellos).

En la figura 1 se encuentran dos círculos semejantes a los diagramas de Venn. Tomando en cuenta la similitud de palabras entre los dos documentos y se define el porcentaje de relación entre los dos documentos. Para ello se descartan las palabras que no se repiten y sólo es tomada en consideración su similitud.

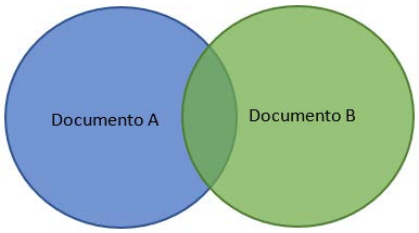

Figura 1 Intersección de circunferencias.

La figura 2 relaciona la distancia que hay entre los documentos por medio de *Clustering*. En este caso, el documento D1 es el que menos similitud tiene respecto con los otros. Por lo tanto, es el que más separado se encuentra de los demás ya que no se asemeja a los otros documentos. La gráfica está basada en una técnica llamada escalado multidimensional. Para formar la gráfica, se utilizan los ejes S y X, donde S es la medición por medio de la matriz de similitud y X representa a las disparidades.

Como se mencionó anteriormente, el documento 1 es un texto de un tema diferente a los demás, debido a esto, se observa su separación de los otros documentos (figura 3).

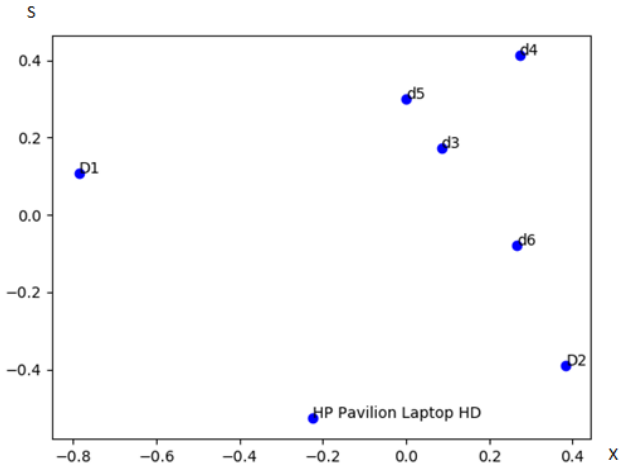

Figura 2 Clustering de distancia entre documentos (2D).

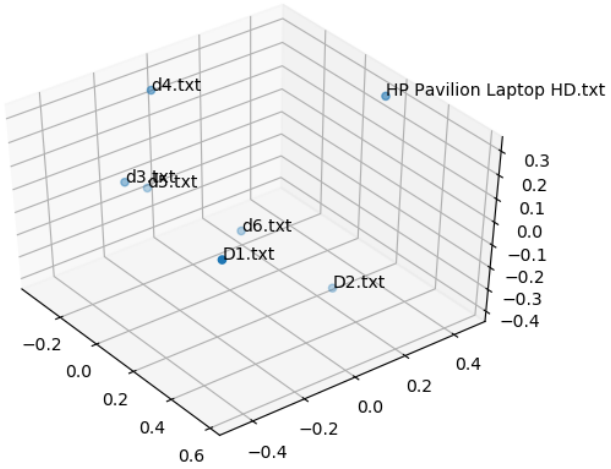

Figura 3 Clustering de distancia entre documentos (3D).

Al igual que en la gráfica de la figura 3 se observa la separación de similitud por medio de un clúster, pero en formato 3D. Este tipo de gráfica es más significativa, debido a que se muestra la medición de la matriz de similitud, la disparidad y una distancia en el conteo de palabras. En la figura 4 se encuentra la similitud de palabras. D1 no tiene similitud con ningún documento por lo que su color (azul) sobresale, asimismo se aprecia como los documentos en verde y rojo se asemejan entre ellos.

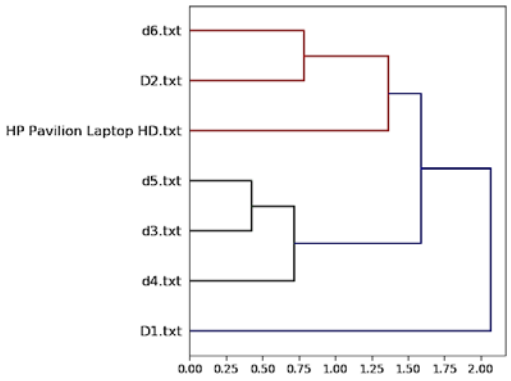

Figura 4 Similitud entre documentos.

4. Discusión

En el presente proyecto se desarrolló un programa en Python, donde por medio de algoritmos de minería de datos, se compararon diferentes textos, considerando los textos como una pequeña base de datos. El programa logró exitosamente eliminar las stopwords y utilizó una técnica de clasificación de matrices denominadas TF-IDF (Term frequency - Inverse document frequency) para asignar un valor determinado a cada texto. Los resultados obtenidos demuestran exitosas comparaciones entre los diferentes textos, mostrando un porcentaje del grado de similitud entre cada uno. Este grado está en función del peso de cada palabra de los diferentes textos.

5. Conclusiones

Los patrones de coincidencia encontrados fueron estudiados y se determinó que efectivamente pueden ser usados para diferentes fines, como, por ejemplo, como

herramienta para los profesores, donde se pueda detectar si un trabajo es genuino o copiado de algún otro alumno, o también para el marketing, al filtrar las palabras importantes y discernir las partes esenciales entre un artículo u otro.

Por otra parte, el trabajo de investigación demostró que es factible la adaptación de esquemas de minería de datos, en un entorno de programación muy usado como es Python. Donde pareciera que, en primera instancia, este lenguaje creado para ser aplicado en otros campos de investigación. Sin embargo, se consiguió de forma exitosa la extracción y clasificación de datos clave o importantes entre cada texto.

6. Bibliografía y Referencias

[1] Ahmed, C. F. Tanbeer, S. K. Jeong B. and Lee, Y. "Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases," in IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 12, pp. 1708-1721, Dec. 2009.

[2] Xu, L. Jiang, C. Wang, J. Yuan J. and Ren, Y. "Information Security in Big Data: Privacy and Data Mining," in IEEE Access, vol. 2, pp. 1149-1176, 2014.

[3] Middelfart, M. Pedersen T. B. and Krogsgaard, J. "Efficient Sentinel Mining Using Bitmaps on Modern Processors," in IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 10, pp. 2231-2244, Oct. 2013.

[4] Adil, S. H. Ebrahim, M. Raza, K. Azhar Ali S. S. and Ahmed Hashmani, M. "Liver Patient Classification using Logistic Regression," 2018 4th International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, 2018, pp. 1-5.