**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**FACULTAD DE CIENCIAS**

# RECOMENDACIÓN AUTOMÁTICA DE NOTICIAS USANDO AGRUPAMIENTO

**T E S I S**

**QUE PARA OBTENER EL TÍTULO DE:**

**LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN**

**P R E S E N T A :**

**CARLOS ROMERO SANTA ANA**

**DIRECTOR DE TESIS:**
**DR. GUSTAVO DE LA CRUZ MARTÍNEZ**

**CIUDAD DE MÉXICO, 2020**

## Agradecimientos

*A mis padres Gabriela y Carlos, por haberme convertido en quien soy, haberme criado y haberme dado todo, sin ustedes nada de lo que he logrado existiría.*
*A mi hermano Alejandro, por ser mi compañero y amigo siempre.*
*A Estefanía por acompañarme, apoyarme constantemente e impulsarme en la vida.*
*A toda mi familia que me ha enseñado acerca del apoyo y soporte que tengo a mi alrededor.*
*A mis amigos que me han apoyado siempre y con quienes he pasado grandes momentos.*
*A Gustavo De la Cruz, por su paciencia y guía durante clases y el desarrollo de este trabajo.*

# Capítulo 1

# Introducción

## 1.1 Antecedentes

El objetivo principal de un sistema de recomendaciones es predecir el posible interés que un usuario pudiera tener por un elemento específico de una colección de elementos que el sistema le puede ofrecer. La popularidad de estos sistemas se ha ido incrementando en los últimos años, debido a la creciente cantidad de información disponible en las aplicaciones y sitios web que buscan ofrecer sus productos a los usuarios, de acuerdo con los intereses y necesidades de cada usuario.

De acuerdo con esto, los sistemas de recomendación son sistemas de cómputo que se encargan de entregar a los usuarios los elementos que sean de su interés, a partir de la información del usuario, conocido como perfil, el cual describe sus gustos o preferencias (Ghorab *et. al.,* 2013). Algunos de los sistemas de recomendación más comunes recomiendan películas, series, música o noticias. Algunos de los ejemplos más conocidos de dichos sistemas son: Youtube, Netflix y Amazon.

Estos sistemas hacen uso de técnicas de minería de datos, para el procesamiento de los elementos disponibles en las bases de información. Una vez procesados los elementos, se identifican aquellos que puedan resultar del interés para un usuario en específico y, de esta forma, se deben mostrar los elementos que tengan características en común con su perfil (Su y Khoshgoftaar, 2009).

Existen diferentes técnicas para identificar los elementos de posible interés. De manera general, podemos decir que se han propuesto 3 métodos para el funcionamiento de los sistemas de recomendaciones, los cuales serán descritos más adelante en el trabajo (Mobasher, 2007):

Sistemas basados en reglas.
Sistemas basados en filtrado de contenido.
Sistemas basados en filtrado colaborativo.

Este trabajo se centrará en los sistemas de recomendaciones de noticias basados en contenido, estos buscarán encontrar las noticias relevantes para un usuario a partir de un modelo de sus intereses.

Como se presenta más adelante, han existido diversas propuestas para la implementación de los sistemas de recomendación basado en contenidos, con ventajas y desventajas cada una, y no existe una que sea considerada como la solución óptima. Se puede observar que la mayoría

de los sistemas de recomendación utilizan algoritmos de clasificación, en el presente trabajo, se explora el uso de algoritmos de agrupamiento como técnica para el manejo de la información.

Si bien los algoritmos de agrupamiento suelen ser ejecutados con grandes cantidades de información, el reto de este trabajo es hacer recomendaciones a partir de la información que publica un periódico de un día cotidiano, lo cual implica una cantidad menor de información.

## 1.2 Objetivo

El objetivo principal de este trabajo es construir un sistema de recomendaciones de noticias para un usuario utilizando sus preferencias personales, obtenidas a partir del análisis de noticias previas que fueron de su interés.

## 1.3 Hipótesis

Para la construcción del sistema propuesto se explorará la utilización de un modelo vectorial (Salton *et. al.,* 1975) para representar el contenido textual de las noticias. Posteriormente se explorará una técnica de agrupamiento para determinar las noticias que tengan más afinidad con el perfil del usuario.

Se espera que con esta estrategia el sistema propuesto sea capaz de discriminar las noticias de interés para el usuario y pueda generar un grupo de recomendaciones para él que sean potencialmente de su interés de acuerdo con su perfil.

## 1.4 Organización del trabajo

El trabajo presente se divide en 5 capítulos, para profundizar en los elementos que componen el sistema propuesto.

El siguiente capítulo presentará los siguientes temas: los diferentes modelos de representación de documentos de texto, algunas técnicas usadas en la minería de datos y, los diferentes tipos de sistemas automáticos de recomendaciones. Explicando sus características y, así definir cuáles de estas técnicas fueron utilizadas en el sistema propuesto.

El tercer capítulo explicará el diseño del sistema automático de recomendación de noticias construido como parte de este trabajo. Se describirá el funcionamiento general de dicho sistema y el desarrollo general del mismo.

El cuarto capítulo concentrará un análisis de los resultados obtenidos de la realización de ejecuciones del sistema desarrollado.

Finalmente, se presentan las conclusiones y el trabajo futuro a desarrollar.

## Capítulo 2

# Sistemas automáticos de recomendaciones y minería de datos para la web

En este capítulo se analizará qué son los sistemas de recomendaciones de noticias y su funcionamiento.

Lo primero que se realizará es una revisión a los diferentes métodos de recuperación y representación de información, los cuales nos permiten estructurar la información en algún formato específico y uniforme para poderlo analizar, ya que la información que se obtiene puede provenir de diversas fuentes y en diferentes formatos.

A continuación, se revisarán algunas técnicas de minería de datos que son fundamentales para agrupar la información obtenida en conjuntos más pequeños y más manejables para que finalmente se realice una serie de recomendaciones de elementos de alguno de los grupos que tenga mayor similitud con el perfil del usuario.

## 2.1 Recuperación de información

La recuperación de la información se refiere a las técnicas utilizadas para la ubicación y obtención de información almacenada en diferentes medios, que pueden encontrarse en diferentes formatos como: imágenes, texto, audio, video, hojas de cálculo y muchos otros. Esta información puede provenir de diversos lugares como dispositivos de almacenamiento, bases de datos y también de sitios web. (Lingras y Akerkar, 2008)

Debido a la gran cantidad de fuentes de información y al volumen de la misma, no es viable que una persona se dedique al análisis manual de la información para su almacenamiento y procesamiento, por lo que las técnicas de recuperación y análisis automático de información son cada vez más importantes.

Una vez que la fuente de información fue identificada es posible extraer los elementos importantes para aprovecharlos de mejor forma. Dependiendo del tipo de fuente y del tipo de información se puede seleccionar uno o diferentes métodos de extracción. En cuanto la información fue extraída, debe ser almacenada para su procesamiento.

### 2.1.1 Representación de documentos

Como fue anteriormente mencionado, la información original puede estar en diferentes formatos, es por eso que existen distintas formas de representar la información, lo cual facilitará su procesamiento y aprovechamiento.

Un ejemplo muy conocido en la representación de información de texto, es la utilización de palabras claves. Esta consiste en representar el texto por medio de palabras seleccionadas del texto en cuestión que sean descriptivas y relevantes para el mismo, asociando el texto original a una colección de palabras representativas del mismo.

La ventaja de esta representación es que se puede hacer más manejable el procesamiento de una colección grande de textos. La desventaja es que usualmente esta representación, requiere de una persona que realice esta tarea, seleccionar las palabras clave o representativas, lo que hace que el proceso no sea completamente automático.

Para el procesamiento automático de textos, existe una aproximación a la representación de textos que trata de asignar un valor a cada término de acuerdo con su relevancia dentro del texto. La relevancia se calcula a partir de la frecuencia de aparición del texto.

Una estrategia para optimizar esta representación, se basa en la eliminación de lo que es conocido como *stop words* en inglés o *palabras vacías* en español. Las palabras vacías son aquellas palabras que se considera que no aportan nada al contexto de una colección de palabra, estas palabras usualmente son artículos, pronombres, preposiciones y cualquier otra palabra que pueda estar dentro de un texto pero no contenga información relevante al tema central del texto.

Una vez eliminadas estas palabras, el texto se mantiene con palabras que son mucho más relevantes.

Un ejemplo de dicha estrategia es, dadas 2 frases:

```
D1: La minería de datos es un análisis de información
D2: Un análisis de información puede revelar datos importantes
```

Se realiza la eliminación de palabras vacías obteniendo las siguientes dos colecciones de palabras:

```
D1: minería datos es análisis información
D2: análisis información puede revelar datos importantes
```

Como se puede observar a ambas frases, se les retiraron los artículos y preposiciones, obteniendo frases más manejables para su procesamiento.

### 2.1.2 Matrices de términos

Una vez reduciendo el texto, quitando las palabras vacías del texto, es significativamente más corto, lo cual. Lo hace más manejable, pero los algoritmos de procesamiento que se utilizan comúnmente en procesos de minería de datos, requieren que la información sea representada de una forma para que estos algoritmos puedan procesarla.

Una forma conocida para la representación de los documentos de texto son las *matrices de términos*.

Una matriz de términos es una representación bidimensional de una colección de textos, en la que las columnas representan los diferentes términos dentro de los textos y las filas representan los diferentes documentos. Los valores de la matriz se suelen representar de formas numéricas para el procesamiento más sencillo por algoritmos.

La representación más común de los valores de una matriz de valores es representando los valores como la frecuencia de los términos en cada documento.

Utilizando las mismas frases anteriores, la matriz se forma tomando todos los términos diferentes en ambos textos. Hay que tener presente que estos términos se toman una vez, aunque aparezca en más de un texto: {minería, datos, es, análisis, información, puede, revelar, importantes}.

| Textos / Términos | minería | datos | es | análisis | información | puede | revelar | importantes |
|---|---|---|---|---|---|---|---|---|
| D1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| D2 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

**Figura 2.1**: Matriz de frecuencia de términos.

Como se puede observar, la matriz representa el número de veces que cada término aparece en cada uno de los dos textos del ejemplo anterior.

Ya se mencionaron algunos ejemplos de diferentes representaciones de documentos haciendo énfasis en la representación de documentos de texto, que serán utilizados más adelante en este trabajo.

### 2.1.3 Algoritmo de Porter

Cuando se toman las palabras de un texto, en muchas ocasiones se tienen palabras diferentes que hacen referencia a una misma idea, por ejemplo: *computadora* y *computación*. A pesar de que las palabras son diferentes vienen de un mismo concepto y dentro del contexto de un documento seguramente ambas palabras hacen referencia a ideas muy similares. Igualmente pasa con verbos conjugados que independientemente de la persona y el tiempo de conjugación vienen del mismo verbo y por lo tanto refieren a la misma idea en cualquier contexto.

Para refinar la representación de los documentos tomando en cuenta estas condiciones, existen diferentes propuestas, siendo uno de los más utilizados el algoritmo de Porter.

El objetivo de este algoritmo es recortar los sufijos de los términos de un texto para acercarlas a una raíz. Por ejemplo en las palabras *caballo*, *caballos* y *caballería* al eliminar el sufijo del plural se obtiene una raíz común que es *caball*, esto reduce las palabras a procesar porque las 3 palabras están asociadas al mismo término al ser derivadas de la misma palabra raíz.

El funcionamiento del algoritmo se basa en conjuntos de reglas, cada conjunto de reglas hace referencia a alguna terminación específica que se pueda dar en el idioma y cómo tomar esta terminación y transformarla por otro elemento o eliminarla completamente. Cada palabra se va a someter a una serie iteraciones buscando dichas terminaciones a identificar y se detendrá cuando la palabra tenga un tamaño muy chico para procesar o cuando no se detecte ninguna terminación relevante que se puede modificar o eliminar.

Cada conjunto de reglas independientemente de la terminación que se esté tratando se construye de la misma manera:

1. Se identifica un sufijo.
2. Se designa el texto para reemplazar dicho sufijo dependiendo de la regla (no todos los sufijos se reemplazan, los plurales por ejemplo suelen solamente eliminarse).
3. Se contabiliza el tamaño del sufijo.
4. Se contabiliza el tamaño del texto de reemplazo.
5. Se calcula el tamaño de la palabra resultante después de un reemplazo para evitar procesar palabras muy cortas.
6. Se realiza una función de validación.

Tomando como ejemplo una de las 3 palabras anteriores: *caballos* se puede observar que el primer sufijo a retirar es una terminación para formar el plural de *caballo.* Por lo tanto se puede identificar la regla como plural y se identifica el sufijo a eliminar que es *s*, para la regla que contenga la formación de plurales regulares no se requiere un texto de reemplazo ya que el plural en palabras regularmente es agregando alguna de las siguientes terminaciones: *as, es, is, os, us* o simplemente *s.*

Dado que no hay término de reemplazo y la palabra tiene una longitud válida para eliminación se elimina dicha terminación obteniendo entonces la palabra *caballo*.

El algoritmo entonces procederá a analizar nuevamente la palabra para identificar si existen sufijos a tratar, en este caso simplemente será la letra *o,* igual que anteriormente para esta regla no hay texto de reemplazo y el tamaño sigue siendo válido por lo que el resultado será la palabra *caball.* Esta palabra como tal no es una palabra existente en el vocabulario del español, pero el algoritmo identifica la letra *o* como terminación a eliminar ya que por reglas gramaticales del idioma de la palabra *caballo* pueden derivar muchos sufijos al eliminar la *o* y agregando otras terminaciones para crear otros sustantivos, como por ejemplo: *caballería, caballeriza, caballerango* y muchos otros. Observando estas palabras, se nota que todas tienen la misma raíz hasta *caball,* una vez pasando esta palabra lo demás son sufijos agregados a cada palabra independientemente.

Utilizando nuevamente los ejemplos vistos anteriormente:

```
D1: minería datos es análisis información
D2: análisis información puede revelar datos importantes
```

Después de utilizar el algoritmo de Porter las frases resultantes quedan de la siguiente manera:

```
D1: min dat es analisis inform
D2: analisis inform pued revel dat import
```

El uso del  algoritmo reduce el vocabulario de cualquier texto dramáticamente lo cual permite que el procesamiento de las palabras sea mucho más sencillo para cualquier aplicación de recolección y procesamiento de información, pero también tiene algunas desventajas, ya que el algoritmo fue ideado específicamente para el idioma inglés. La implementación original tiene condiciones que no aplican para el español y también hay condiciones que sólo existen en español que en inglés nunca fueron contempladas, por eso las implementaciones en español han tenido que agregar y eliminar condiciones para tener un funcionamiento similar. A pesar de estos inconvenientes, las adaptaciones a diferentes idiomas tienen un funcionamiento satisfactorio en la mayoría de las ocasiones.

### 2.1.4 Modelos de representación

Una vez que el texto ha sido reducido en su mayoría a la raíz de cada palabra, es aún más sencillo manejar la información.

Pero como ya fue mencionado, los algoritmos de procesamiento de la información requieren de modelos de representación específicos para poder trabajar con los datos.

### 2.1.4.1 Modelo booleano

La matriz de términos que ya se mencionó anteriormente, simplemente indica de forma numérica la cantidad de apariciones que tiene un término en cada documento.

Un representación más simple que una matriz que sólo calcule la frecuencia de cada palabra es el modelo booleano. Este modelo de representación muestra simplemente si la palabra tiene apariciones o no dentro de cada documento, dejando de lado la cantidad de veces que esta pueda aparecer, ya que el hecho de que aparezca con frecuencia no significa que sea más relevante que otras palabras.

Utilizando las frases del ejemplo que se ha venido trabajando tenemos las siguientes frases:

```
D1: min dat es analisis inform
D2: analisis inform pued revel dat import
```

La lista de términos diferentes es la siguiente: {min, dat, es, analisis, inform, pued, revel, import}

Utilizando la matriz de modelo booleano el resultado será el siguiente:

| Textos / Términos | min | dat | es | analisis | inform | pued | revel | import |
|---|---|---|---|---|---|---|---|---|
| D1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| D2 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

**Figura 2.2**: Matriz de representación de modelo booleano.

Como se puede observar, el resultado de esta matriz es similar al resultado de la matriz anterior. Pero en esta matriz independientemente de la cantidad de veces que un término pudiera aparecer en cualquiera de los textos, solamente se refleja un valor booleano colocando un *uno* si el término aparece al menos una vez o colocando un *cero,* si el elemento no aparece en el texto.

Este modelo resulta de mucha utilidad si lo único que se desea es buscar directamente términos dentro de una colección de textos, de esta forma se entregan como resultado solamente aquellos que contengan dichos términos, este modelo también es el más sencillo y rápido de implementar.

### 2.1.4.2 Modelo vectorial

Para poder llegar a realizar un procesamiento de minería de datos es necesario procesar aún más la información ya que no es suficiente saber solamente qué términos contiene cada documento si no es necesario un análisis más profundo por lo que el modelo booleano resulta insuficiente, para eso se puede ocupar el modelo vectorial.

El modelo vectorial busca dar una representación más completa que el modelo booleano donde simplemente se expone si un término apareció o no en un texto. Todas las matrices que se han mostrado anteriormente son esencialmente modelos vectoriales ya que cada documento es representado por medio de un vector de valores numéricos cada uno que representa algo en la matriz.

El modelo vectorial más completo busca que el valor numérico que aparece en los vectores sea más representativo por eso el modelo propone el cálculo del peso de cada término utilizando las frecuencias de cada uno en el documento.

Una forma sencilla de construir un modelo vectorial es utilizando la normalización de los pesos de cada término, eso se obtiene de la siguiente manera.

$$P_{ij} = \frac{frec_{ij}}{Max\{k=1, m\}(frec_{ik})}$$

"Donde *Pij* es el peso, y *frecij* es la frecuencia del término número *j* en el documento número *i*. Se asume que hay *m* términos en toda la colección de documentos, es decir, que el número de columnas en la matriz es *m."* (Lingras y Akerkar, 2008)

Dado que el ejemplo que se había trabajado previamente no muestra ninguna diferencia entre la matriz de frecuencias con los términos ya procesados por el algoritmo de Porter y la matriz de representación booleana, ya no se ocupará este ejemplo.

En lugar del ejemplo previo se usará como ejemplo la siguiente matriz donde cada D es un documento y cada T es un término que aparece en alguno de los documentos de la colección.
El ejemplo proviene directamente de (Lingras y Akerkar, 2008).

| Textos / Términos | T1 | T2 | T3 | T4 | T5 | T5 | T7 |
|---|---|---|---|---|---|---|---|
| D1 | 0 | 0 | 5 | 2 | 0 | 1 | 2 |
| D2 | 4 | 1 | 1 | 0 | 0 | 0 | 0 |

**Figura 2.3**: Matriz abstracta de términos y documentos. (Lingras y Akerkar, 2008)

Utilizando este ejemplo las operaciones para obtener una matriz de pesos normalizados requerirá los siguientes pasos: para el primer documento, se dividirá cada término entre 5 y para el segundo, se dividirá cada término entre 4. Ya que son los dos términos con más apariciones en sus respectivos documentos. Obteniendo como resultado la siguiente matriz.

| Textos / Términos | T1 | T2 | T3 | T4 | T5 | T5 | T7 |
|---|---|---|---|---|---|---|---|
| D1 | 0 | 0 | 1 | 0.4 | 0 | 0.2 | 0.4 |
| D2 | 1 | 0.25 | 0.25 | 0 | 0 | 0 | 0 |

**Figura 2.4**: Matriz de pesos de términos y documentos normalizada. (Lingras y Akerkar, 2008)

Este modelo es más representativo con lo que ocurre con cada término respecto al documento en el que se encuentra. Se puede observar que cada valor cercano al 0 es un término que no aparece ninguna ocasión o muy pocas ocasiones, y un valor muy acercado al 1, es un término que aparece demasiadas veces, y como se mencionó previamente, un término con demasiadas apariciones, no necesariamente es muy relevante para el contexto del documento.

El objetivo de haber realizado un modelo que representara los documentos como vectores de pesos fue para realizar posteriormente un análisis de similitudes entre los diversos documentos para poder realizar agrupaciones por características en común.

Un método usual para medir la similitud entre documentos que tengan esta representación es mediante el coseno del ángulo generado entre los 2 vectores que se desee encontrar la similitud, esto es de la siguiente manera:

Tomando los dos vectores utilizados como ejemplo *D1(0,0,1,0.4,0,0.2,0.4)* y *D2(1,0.25,0.25,0,0,0,0)* se define la función de similitud entre estos documentos de la siguiente forma:

$$sim(D1, D2) = \frac{(D1 \bullet D2)}{(|D1| \times |D2|)}$$

El producto punto de ambos vectores dividido entre el producto cruz de las normas de los mismos, como

$$D1 \bullet D2 = |D1| \times |D2| \times cos(\Theta)$$

se tiene que

$$sim(D1, D2) = \frac{(|D1| \times |D2| \times cos(\Theta))}{(|D1| \times |D2|)} = cos(\Theta)$$

De esta forma:

$$sim(D1, D2) = cos(\Theta) = \frac{(\Sigma D1 \times D2)}{(\sqrt{(\Sigma D1^2)}) \times (\sqrt{(\Sigma D2^2)})}$$

(Lingras y Akerkar, 2008)

Sustituyendo los valores de los vectores del ejemplo se tiene:

$$sim(D1, D2) = \frac{(0x1+0x0.25+1x0.25+0.4x0+0x0+0.2x0+0.4x0)}{\sqrt{(0^2+0^2+1^2+0.4^2+0^2+0.2^2+0.4^2)} \times \sqrt{(1^2+0.25^2+0.25^2+0^2+0^2+0^2+0^2)}}$$

$$= \frac{0.25}{\sqrt{(1+0.16+0.04+0.16)} \times \sqrt{(1+0.0625+0.0625)}} = \frac{0.25}{\sqrt{1.36} \times \sqrt{1.125}} = \frac{0.25}{1.166 \times 1.06} = \frac{0.25}{1.235} = 0.2024$$

El resultado de la función de similitud entre ambos documentos es 0.2024, lo que esto dice es que los ángulos entre ambos vectores están separados, la forma en la que se puede agrupar diversos documentos es realizando esta misma operación entre todos los pares de vectores posibles y ordenarlos de mayor similitud a menor, entre mayor sea el resultado del ángulo se puede ver que los vectores son muy similares. Al contrario de aquellos con menor resultado del ángulo.

### 2.1.4.3 Representación usando TF-IDF

A pesar de que la representación matricial ya es una representación adecuada, siempre se puede optimizar, para eso existe la medida *TF-IDF (Term frequency - inverse document frequency)* que como su nombre lo menciona toma en cuenta la frecuencia del término y lo que se denomina la frecuencia inversa de documento. (Lingras y Akerkar, 2008)

Esta medida permite representar de forma optimizada el texto existente tomando en cuenta cada término existente y cada documento disponible, esto hace que el valor de cada término tome en cuenta no solo su texto si no los demás textos.

La medida se calcula de la siguiente manera.

$$TFIDF = tf * idf$$

Los valores se obtienen de la siguiente manera:

$$TF = \frac{c(t,d)}{|d|}$$

donde c(t,d) es la cantidad de veces que aparece el término *t* en el documento *d* y |*d*| la cantidad de términos totales que tiene el documento *d*.

$$IDF = log(\frac{N}{|d \in D : t \in d|})$$

donde *N* es la cantidad total de documentos y |*d* ∈ *D* : *t* ∈ *d*| es la cantidad de documentos que contienen el término *t*.

Como en su nombre se puede observar, cada valor segmento del nombre hace referencia a lo que se calcula, las siglas *TF* en el nombre representan la frecuencia de un término, calculada con base en el texto en el que se están calculando los términos; mientras que las siglas *IDF* determinan la relevancia de un texto con respecto a una palabra clave específica.

Este valor representa la importancia de un término basándose en las apariciones de este término en la colección de documentos. Si el término aparece en todos los documentos al menos una vez se puede observar que no es un término que describa a ningún documento ya que todos lo tienen y pueden ser documentos muy diferentes, este tipo de palabras podrían ser artículos, pronombres o preposiciones que son palabras muy repetidas en un lenguaje y que al eliminar se realiza una simplificación del texto.

Es por esto que la relevancia se calcula mediante un logaritmo, un logaritmo de cualquier base de un término muy frecuente dará un resultado cercano 0 lo que significa que un término que aparece en todos los documentos existentes no es de ninguna importancia, esto significa que en una colección grande de textos donde pueden haber términos repetidos constantemente pueden existir vectores con varios términos con valores en 0.

Retomando los textos que se tenían en los ejemplos anteriores:

```
min dat es analisis inform
analisis inform pued revel dat import
```

La lista de términos diferentes es la siguiente: {min, dat, es, analisis, inform, pued, revel, import}.

El primer término se calcula de la siguiente manera:

Para el valor TF se calcula la cantidad de apariciones del término en el texto 1 por lo que se tiene

$$TF = \frac{1}{5} = 0.2$$

Para el valor IDF se calcula el logaritmo del resultado obtenido al dividir la cantidad total de textos entre la cantidad de aquellos que tienen al término deseado, en este caso *min* por lo que se tiene:

$$IDF = log(\frac{2}{1}) = log(2) = 0.69314$$

Entonces se tiene

$$TF * IDF = 0.2 * 0.69314 = 0.1386$$

De esta misma forma se calcula este algoritmo para cada término en cada texto y se obtiene una representación matricial con la misma estructura que el modelo booleano pero con valores diferentes.

| Textos / Términos | min | dat | es | analisis | inform | pued | revel | import |
|---|---|---|---|---|---|---|---|---|
| Texto 1 | 0.1386 | 0 | 0.1386 | 0 | 0 | 0 | 0 | 0 |
| Texto 2 | 0 | 0 | 0 | 0 | 0 | 0.1155 | 0.1155 | 0.1155 |

**Figura 2.5**: Matriz de pesos de términos y documentos tratada con TF-IDF. (Lingras y Akerkar, 2008)

Estos valores representan en forma de vectores numéricos cada texto asignado y un peso específico a cada término existente en la colección de textos.

Como se puede observar los términos *analisis, dat* e *inform* tienen 0 en ambos vectores esto se debe a que estos términos tienen apariciones en ambos textos, lo cual hace que el cálculo del algoritmo los considere como términos poco relevantes para el análisis de la información. Por otro lado los demás términos tienen valores numéricos mayores a 0 ya que son considerados más relevantes, dependiendo de la frecuencia de cada término respecto a cada texto es el valor que tendrá, entre más alto sea el valor se considera que ese término es más relevante en el texto y en la colección de textos existentes.

Esta representación resulta más óptima que las anteriores para el siguiente paso que será calcular las similitudes entre dichos textos para poder realizar agrupaciones entre los elementos.

## 2.2 Minería de datos de la web

La minería de datos se refiere a las técnicas aplicadas en la búsqueda y análisis para obtener información comprensible y útil a partir de grandes cantidades de datos. (Lingras y Akerkar, 2008)

La información de la que se parte no se presenta siempre en el mismo formato y en ocasiones mucha de esta información no tiene ninguna relevancia para el análisis que se está realizando, por lo que se tienen que utilizar diversas herramientas para el procesamiento de la misma.
Para la obtención de información útil se emplean métodos como modelos estadísticos, probabilísticos, algunas implementaciones de inteligencia artificial o redes neuronales.

La minería de datos se utiliza de diferentes maneras, siendo la más común la minería en grandes bases de datos, pero para este trabajo se abordará la minería de datos de la web, que se especializa en aplicar las técnicas mencionadas a los contenidos que se encuentran en la web. La minería de datos de la web tiene también como característica el hecho de que el contenido en la web viene en grandes cantidades y mucho de este contenido puede ser irrelevante a un análisis que se desee hacer (Lingras y Akerkar, 2008), es por eso que al tratar de obtener contenido de la web, la selección de las fuentes es importante, asimismo la

información cuando viene obtenida de diferentes fuentes tiende a venir en diferentes formatos por lo que hay procesos que tienen que transformarla antes de realizar cualquier análisis.

### 2.2.1 Tipos de minería en la web

La minería de datos en la web se divide en 3 tipos dependiendo del tipo de información que se desee procesar y de donde se vaya a obtener dicha información (Lingras y Akerkar, 2008):

**1. Minería del contenido de la web:** esta se refiere a la información obtenida dentro de los sitios web, es decir, la que públicamente se visita de un sitio web. Esta puede ser cualquier tipo de información como texto, videos, música o imágenes, siendo los textos la información más obtenida y analizada por sistemas de minería de datos web.

**2. Minería de la estructura de la web:** esta se refiere a la extracción de información de la estructura y conexiones de los sitios web visitados. A su vez se divide en 2 tipos de minería de datos de estructura: patrones de ligas en el sitio y obtención de estructuras de tipo árbol en HTMLs o XMLs.

**3. Minería del uso de la web:** esta se refiere a la información obtenida del servidor al navegar en los sitios web. Para lograrlo se analiza a los clientes usando el sitio web y se toma la información de los servidores para analizar los comportamientos que puedan tener los usuarios en ciertos sitios web, también incluye la información obtenida de perfiles de los usuarios para tener acceso a información demográfica o geográfica para realizar un análisis de los tipos de usuario que realizan ciertas actividades específicas en los sitios web.

En la figura 2.6 se tiene un diagrama de las clases de minería de la web en el cual se observan las 3 clasificaciones: minería del contenido de la web, minería del uso de la web y minería de estructura de la web, de las cuales se hace un énfasis en las secciones de contenido y de uso de la web.

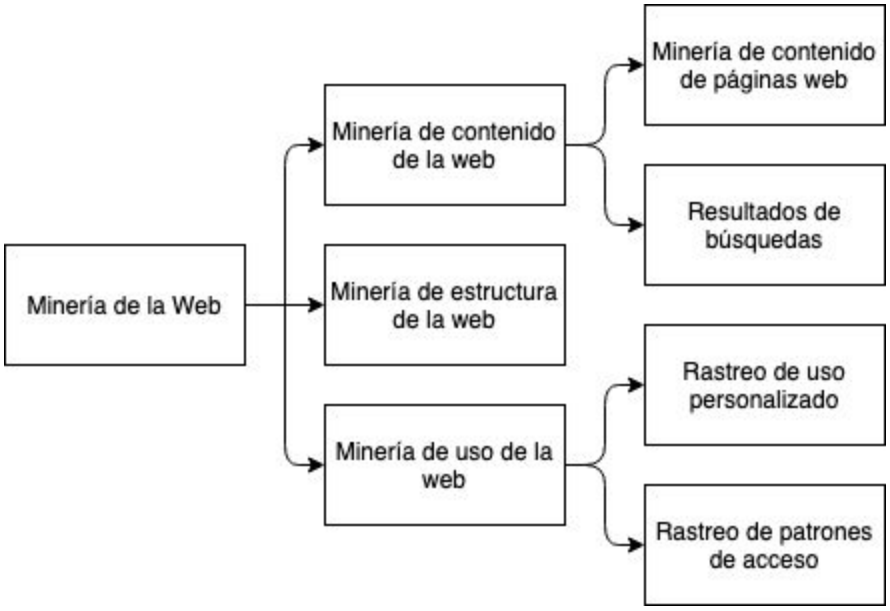


**Figura 2.6**: Diagrama de tipos de minería de datos en la web. (Lingras y Akerkar, 2008).

Como la minería de la web es la aplicación de las técnicas generales de la minería de datos, los métodos que se usan son los mismos. En la siguiente sección se presentan algunos de estos métodos.

### 2.2.2 Métodos en la minería de datos

Existen muchos métodos diferentes para realizar minería de datos, dependiendo de qué tipo de información se esté procesando y la meta con la que esta se procesa será entonces el método con el que se abordará el problema.

Algunos de los métodos utilizados son: modelos estadísticos, aprendizaje de máquina y algunos otros.

> **Métodos estadísticos:** Los métodos estadísticos estudian las colecciones de datos para realizar predicciones. Un modelo estadístico es una serie de fórmulas matemáticas que describen el comportamiento de los objetos en la colección de datos.
> Estos modelos se pueden utilizar para luego realizar predicciones basadas en los modelos. Utilizando las fórmulas matemáticas aplicadas a la información que se tiene se pueden predecir nuevos eventos que deriven de los ya computados, a esto se le llama hipótesis estadística.
>
> Los modelos estadísticos pueden ser de una complejidad muy amplia, sobretodo para conjuntos muy grandes de información, estos modelos se complican aún más para aplicaciones web que realicen consultas constantes como motores de búsqueda, que

tienen que realizar minería constante y muy rápida para manejar los ingresos de información. (Lingras y Akerkar, 2008)

**Métodos bayesianos:** Estos métodos permiten construir modelos con los cuales se pueden realizar inferencias probabilísticas, de esta forma se puede calcular la probabilidad de pertenencia de un elemento a un grupo de distintos elementos, de esta forma separando la información. (Gutiérrez *et. al.*, 2011)

**Aprendizaje de máquina:** El aprendizaje de máquina, como su nombre lo dice, se refiere a la simulación de que un sistema aprenda ciertas cosas. La idea principal es que un sistema aprenda automáticamente a reconocer patrones complejos y luego realizar decisiones a partir de los patrones ya aprendidos y llegada de información nueva.

El aprendizaje de máquina suele ser solucionado con algoritmos de redes neuronales, cuyo principal objetivo es el reconocimiento de patrones en grandes cantidades de información. (Lingras y Akerkar, 2008)

El aprendizaje de máquina se divide principalmente en dos partes: aprendizaje supervisado y aprendizaje no supervisado.

**1. Aprendizaje supervisado:** este se refiere a los sistemas que requieren de una supervisión para realizar su aprendizaje. Es decir, el sistema recibe un conjunto inicial de información que usará para partir, este conjunto es llamado el conjunto de entrenamiento, ya que esta información será utilizada como ejemplo para que el sistema posteriormente pueda reconocer información similar a la ya analizada.

Por ejemplo: un sistema que reconoce letras escritas a mano puede recibir la letra “a” n veces escrita por *n* diferentes personas, al reconocer el patrón de la letra el sistema podrá reconocer una letra “a” de cualquier otro tipo de escritura que no haya sido analizado antes.

En el ejemplo las n-veces que el sistema analizó las diferentes maneras de escribir la letra “a” fueron el conjunto de datos de entrenamiento ya que al analizar constantemente ese patrón eventualmente puede reconocer patrones similares a la letra “a”.

Un ejemplo muy utilizado de aprendizaje supervisado son los perceptrones, que son un tipo de red neuronal utilizada para el reconocimiento de patrones con conjuntos de datos de entrenamiento. (Lingras y Akerkar, 2008)

**2. Aprendizaje no supervisado:** este se refiere a los sistemas que al recibir un conjunto de información tienen la tarea de encontrar patrones sobre la misma conforme

se va procesando, estos métodos tienden a representar la información en formas numéricas y utilizar algoritmos de comparaciones y de vecindades para poder agrupar la información dependiendo de los patrones encontrados. (Lingras y Akerkar, 2008)

### 2.2.3 Agrupamiento

Un ejemplo muy común del aprendizaje no supervisado es el problema de agrupamiento.

Este consiste en tratar de agrupar grandes cantidades de información en diferentes grupos que tengan alguna característica en común.

Ejemplos de este tipo de métodos son los algoritmos que resuelven el problema de agrupamiento(clustering) como los algoritmos K-Means y K vecinos más cercanos o las redes neuronales como el mapa auto organizado SOM: self-organizing map.

En la figura 2.7 se puede observar una gráfica que representa 3 grupos con una marcada separación. Cada pequeño círculo representa un dato específico en la colección de información y se puede observar que se identifican 3 grupos donde cada elemento de grupo tendrá características en común con los elementos de ese mismo grupo.

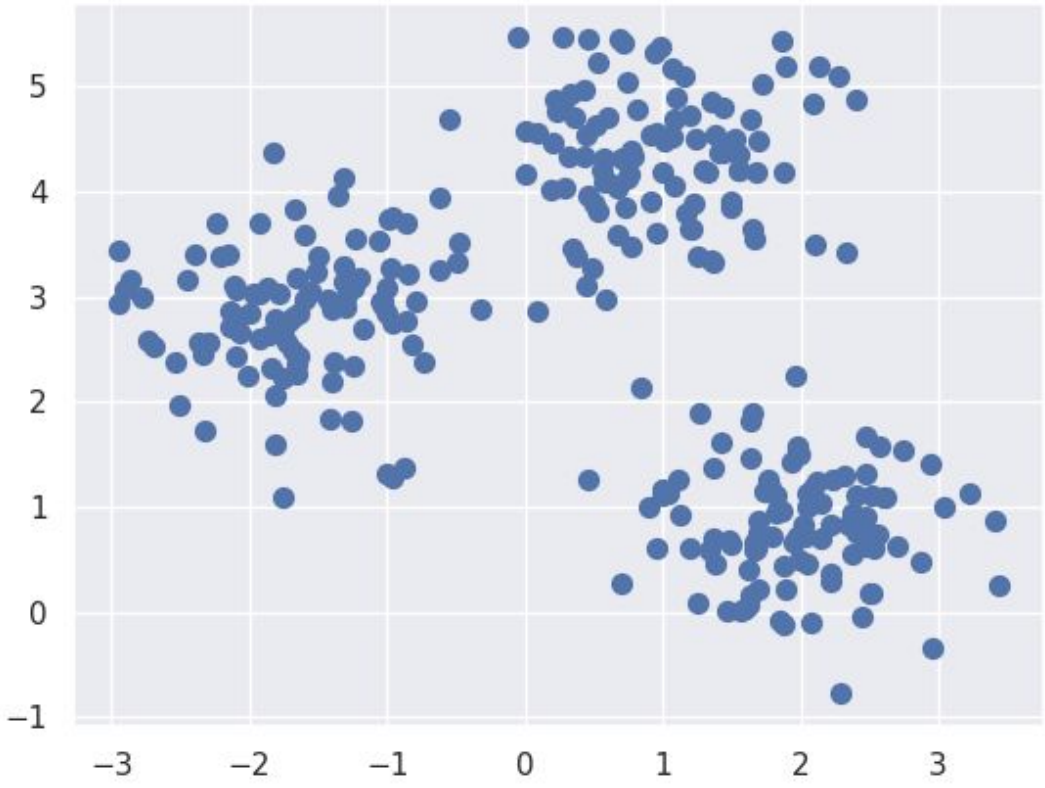


**Figura 2.7**: Representación de grupos separados.

Para solucionar el problema de agrupamientos existen diferentes modelos que buscan dar una respuesta óptima, estos son:

**Modelos de conectividad:** que se basan en distancias de conectividad.
**Modelos de distribución:** por medio de distribuciones estadísticas.
**Modelos de centroides:** modelos que se basan en distancias entre vectores.

Existen muchos otros modelos que presentan soluciones al problema de agrupamiento, estos solo son algunos de ellos. El último modelo, de centroides, es el que se tratará durante este trabajo de donde se obtienen varios algoritmos que se basan en distancias entre vectores para encontrar las distancias entre elementos.

**Algoritmo K - Means:**

Cuando la información es demasiada, una persona no puede procesar toda la información y encontrar patrones de similitudes entre bloques de datos para poder agruparlos en un mismo

sitio basado en estas similitudes, para eso existen diversos algoritmos que permiten procesar grandes cantidades de datos y agruparlos en diferentes clases.

Uno de los algoritmos más conocidos y utilizados para resolver el problema de agrupamiento es el algoritmo de K - Means. El nombre del algoritmo viene de las distancias medias que serán calculadas para asignar cada elemento a un solo grupo de elementos.

El algoritmo de K - Means tiene una estructura sencilla que primero se explicará a grandes rasgos, es básicamente seleccionar la cantidad de grupos, posteriormente se eligen k-puntos aleatoriamente, estos serán los centros de las agrupaciones (centroides), en seguida se asigna cada elemento de información al centroide más cercano y de esta forma cada colección de puntos de información asociada a un centroide, es un grupo. Una vez que todos los puntos fueron asociados se vuelve a recalcular la posición de los centroides de acuerdo a los puntos en su colección para que queden al centro de cada uno de los otros puntos y se repite también el cálculo para reasignar los puntos a cada centroide. Estos cálculos se repetirán hasta que los centroides no cambien de posición o los grupos no cambien de elementos. (Tan *et. al.,* 2005)

El algoritmo se basa, como ya fue mencionado en cálculos de distancias, es necesario que los datos se encuentren en un formato que permita dicho cálculo, ya que no hay manera de calcular similitudes entre información con formato de texto. Es por eso que la información antes de llegar a este punto, debió haber sido representada con algún método expuesto anteriormente. De esta forma cada texto de una colección será representado por medio de un vector de pesos numéricos y utilizando vectores, el cálculo de distancias es una operación sencilla.

Este algoritmo recibe como parámetro principal el número de grupos en los que será dividida la información, es posible que en algunas ocasiones reciba como un parámetro también el número de iteraciones que se harán en caso de que se considere necesario, pero regularmente el algoritmo actúa hasta encontrar convergencia.

El algoritmo primero debe asignar un centroide a cada uno de los grupos, este mismo actuará como el elemento representativo para comparar los demás elementos. Este centroide puede generarse de diversas maneras dependiendo de la aplicación del algoritmo que se esté realizando. Puede utilizarse aleatoriamente algún elemento de los datos a procesar como centroide o puede asignarse de manera aleatoria, este último es el método más común.

Una vez inicializados los centroides, el algoritmo alterna entre 2 pasos: el paso de asignación y el paso de actualización. Estos pasos se iteran constantemente hasta que finaliza el algoritmo, que como se mencionó anteriormente es hasta alcanzar convergencia o si se definió un número específico de iteraciones.

Para poder trabajar con este algoritmo la información debe ser presentada en un formato en el que se puedan realizar cálculos, es decir, no se puede trabajar el algoritmo con la información

en formato de texto, se requiere un método vectorial de representación del dato. Más adelante se describen los formatos vectoriales y algunos métodos de obtención de estos formatos de representación. Por el momento basta con suponer que la información ya viene presentada en el formato requerido que son vectores numéricos, y cabe mencionar que todos los vectores deben tener la misma longitud ya que los cálculos se realizan entrada por entrada. A continuación se enumeran los pasos del algoritmo:

**1. Asignación:** este paso consiste iterar sobre todos los datos, compararlos contra cada centroide y asignarlo al grupo donde la comparación con su centroide haya sido la más óptima. Esto se logra calculando la distancia euclidiana entre el dato que se encuentre procesando y los centroides, este dato se asignará al grupo que corresponda a la menor de las distancias calculadas. A continuación se puede ver la fórmula de asignación de un elemento Xp a un grupo: si en el tiempo t calculando la distancia entre el punto Xp y en este caso $m_i$ será la media del grupo i. Es decir su centroide y $m_j$ sería el centroide del grupo j. De esta forma se observa que Xp será asignado al grupo i, si es menor o igual. Dado que un dato sólo puede ser asignado a un grupo se aplica el menor o igual, de esta forma si llega a ser igual se asignará automáticamente a uno de los grupos al cumplir la regla quedando fuera del grupo anterior.

$$St_i = \{Xp : Xp - mt_i||2 \leq ||Xp - mt_j||2 \,\forall j, 1 \leq j \leq k\}$$

**2. Actualización:** que consiste en calcular el nuevo centroide para cada grupo. El centroide se intenta que sea el punto medio de todo el grupo se busca la distancia más corta entre este centroide y todos los elementos del grupo. Este cálculo es muy sencillo ya que es la media general entre todos los vectores y se obtiene mediante la siguiente operación.

$$m(ti) + 1 = \frac{1}{|S(ti)| \sum_{xj=S(i,t)}^{n} (xj)}$$

La ecuación se puede definir como la media en el tiempo t + 1 que es para la siguiente iteración, esto es igual a la suma de todos los vectores divididos entre la cantidad de vectores existentes, con esto se obtiene la media aritmética que será el nuevo punto que se usará como el centroide del grupo, esto se realiza para cada grupo. Entre más se realice este proceso combinado con el paso anterior, se garantiza que el grupo va minimizando la distancia entre sus elementos ya que el centroide cada vez es más cercano a todos y comienzan a agruparse los elementos del grupo mientras que los que no pertenezcan son atraídos a otro grupo donde pasará lo mismo.

**3. Revisión:** consiste en verificar si el algoritmo debe continuar o finalizar. Como ya se mencionó, estos dos pasos serán iterados hasta alcanzar algunas de las condiciones de término o la convergencia, esta convergencia puede ser obtenida cuando los centroides ya no cambien con respecto a la iteración anterior y por lo tanto los grupos ya no cambian de

contenido y al no cambiar de contenido, la media aritmética permanece idéntica, dejando el mismo centroide.

Otra de las condiciones ya mencionada es asignar un número finito de iteraciones, el algoritmo regularmente busca ejecutarse hasta converger en una solución, pero hay variantes del algoritmo que tienen un tope de iteraciones ya sea por cuidar el tiempo de ejecución o la memoria del sistema que se encuentra ejecutando el algoritmo. De igual manera esta condición detendrá la ejecución aunque no se tenga una solución óptima. Cabe mencionar que si el algoritmo alcanza la convergencia antes de alcanzar este límite, el algoritmo terminará y no será necesario llegar a la condición de iteraciones.

Finalmente, hay que mencionar que este algoritmo es considerado muy efectivo. Sin embargo, tiene el inconveniente de que es necesario conocer de antemano el número de grupos en los que se desea dividir la información, esto produce que las soluciones del algoritmo varíen y no siempre se entregue un óptimo ya que tal vez la información sería mejor clasificada en más o en menos grupos de los que se había pensado en un inicio. Otro inconveniente es la primera asignación de centroides a los grupos, ya que hay dos maneras de hacerlo, tomar elementos aleatorios de la lista de datos existentes y la otra es generarlo de manera aleatoria para cada grupo. Estos dos métodos causan que cada que se ejecuta el algoritmo pueda haber una diferencia a la versión anterior, ya que los centroides iniciales siempre son diferentes.

Al ser un algoritmo de heurística, no se garantiza que siempre se vaya a encontrar el óptimo global, pero encontrará un óptimo local que en ocasiones puede llegar a ser el global.

En la figura 2.8 se puede observar el ejemplo de la figura 2.7 una vez que el algoritmo K-Means finalizó las iteraciones y se generaron los grupos. En la imagen se pueden observar marcados con un círculo negro los 3 centroides de los 3 grupos y a su vez cada grupo, se ilustra de diferente color.

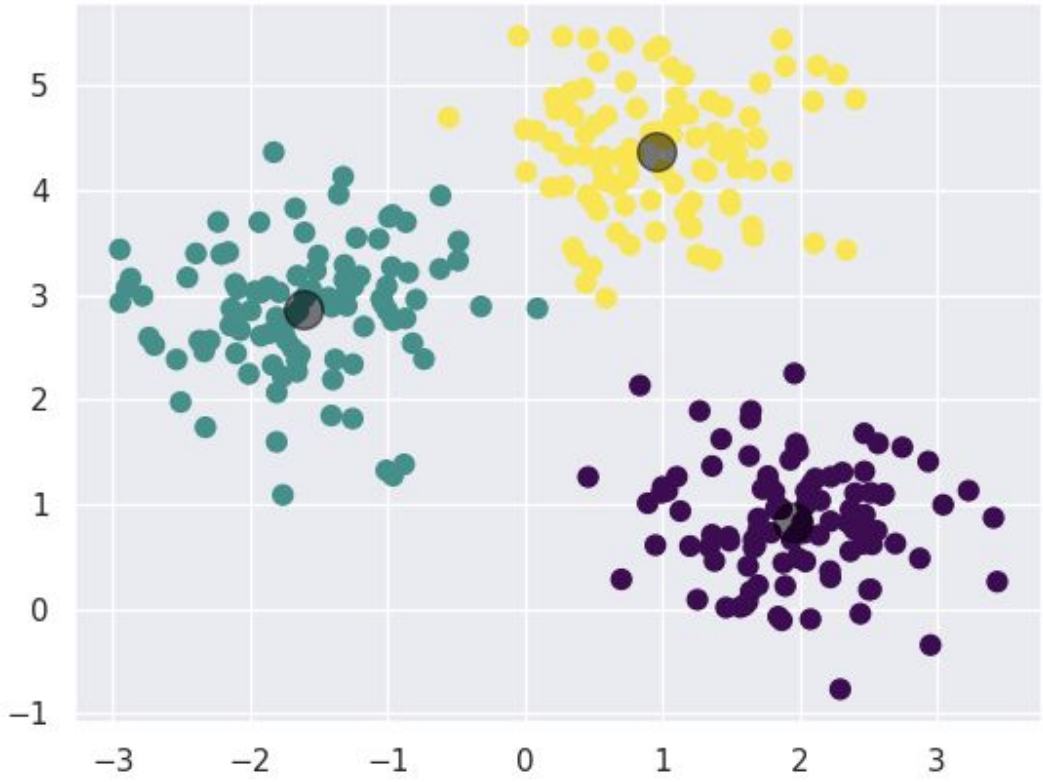


**Figura 2.8**: Ilustración de evolución del algoritmo K-means a través de las iteraciones.

## 2.3 Sistemas automáticos de recomendaciones

El objetivo de los sistemas automáticos de recomendaciones es buscar y predecir la preferencia que un usuario podría tener por un elemento específico en el sistema en el que se esté navegando. Estos sistemas han ido creciendo en popularidad en los últimos años debido a la creciente cantidad de sitios web que buscan facilitar a los usuarios las búsquedas de los elementos que necesiten. Los sitios que incluyen recomendaciones varían en géneros, pueden ser recomendaciones de películas, series, música, noticias o productos en tiendas virtuales.
Algunos autores proponen principalmente 3 métodos de sistemas de recomendaciones (Mobasher 2007):

**1. Sistemas basados en reglas:** Los sistemas que se basan en reglas funcionan realizando las decisiones a través de reglas generadas ya sea de forma automática o de forma manual. Las reglas que pueden incluirse, son basadas en la información demográfica, geográfica u otro tipo de información de los usuarios del sistema. Usualmente las reglas son insertadas de manera manual por los administradores de los sistemas o por los mismos usuarios por interacciones explícitas con el sistema por medio de preguntas que el sistema genera al

usuario. De esta forma el sistema inserta automáticamente reglas para un usuario específico usando las respuestas a los cuestionarios realizados.

Desventajas de este tipo de recomendaciones es que las reglas tienden a ser insertadas en el sistema de forma manual por lo que estas reglas suelen ser una descripción subjetiva de los intereses mismos del usuario por lo que pueden causar parcialidad. (Mobasher 2007). Otro inconveniente es que al insertar estas reglas, los perfiles son usualmente estáticos por lo que los algoritmos comienzan a fallar al paso del tiempo conforme el perfil envejece. (Mobasher 2007). Ejemplos de este tipo de sistemas son muchos sitios de comercio en línea, algunas de estas reglas pueden ser descuentos o productos que el usuario desee en cierto rango de precios, en estos casos las reglas fueron puestas por los mismos usuarios para filtrar los productos existentes dependiendo de promociones y precios.

**2. Sistemas basados en filtrado de contenido:** Estos sistemas utilizan perfiles de usuarios en los cuales se va almacenando el interés que el usuario ha mostrado en ciertos elementos, se toman algunas características de dichos elementos y luego las recomendaciones se realizan con objetos que el usuario aún no conozca pero compartan características en común con los ya vistos por el usuario.

Un ejemplo de este tipo de sistemas son recomendaciones en sitios de videos como en el sitio youtube, en este sitio al ver un video automáticamente despliega videos similares por contenido, ya sea videos del mismo canal, de los mismos protagonistas o de géneros similares, posteriormente el mismo sitio recomendará videos relacionados no solamente con el video que se está observando en ese momento si no también del historial general del perfil. Aunque este sistema también incluye híbridos de algoritmos también tomando en cuenta que han visto otros usuarios con gustos similares. Algunos otros ejemplos concretos de este tipo de sistemas de recomendaciones son: Letizia, NewsWeeder, Personal WebWatcher, InfoFinder entre otros.

La principal desventaja de este tipo de sistemas es que al utilizar los intereses mostrados por los usuarios el algoritmo tiende a recomendar cosas demasiado especializadas en los mismos temas, lo cual puede representar monotonía al usuario, ya que, al recomendar constantemente del mismo tema sin agregar nuevos elementos el usuario también puede perder el interés y algunos estudios han mostrado que los usuarios responden mejor a las recomendaciones cuando se llegan a agregar elementos menos esperados dentro del bloque de las recomendaciones que ya se tenían en cuenta para el usuario (Mobasher 2007). Es por esto que los sitios como youtube anteriormente mencionado insertan algoritmos híbridos, que son combinaciones de más de un algoritmo, para poder tener un elemento de sorpresa y que el usuario no pierda el interés.

**3. Sistemas basados en filtrado colaborativo:** Los sistemas de filtrado colaborativo tratan de cubrir problemas que en ocasiones presentan los dos anteriores tipos de sistemas. Las recomendaciones se realizan por medio de calificaciones de los usuarios a los elementos, de esta forma un usuario recibirá recomendaciones basadas en los usuarios con más afinidad al

usuario entregando un elemento que los otros usuarios hayan calificado alto y el usuario en cuestión no ha visto aún (Mobasher, 2007).

Un ejemplo de estos sistemas son sitios de películas y sitios de series como por ejemplo Netflix. Este sistema tiene también algoritmos híbridos, pero parte de su sistema de recomendaciones es por medio de las calificaciones que de un usuario a películas o series que ya ha visto en alguna ocasión, el sistema entonces toma a otros usuarios diferentes que hayan dado calificaciones positivas a estos mismos elementos y procede a recomendar otros elementos con calificaciones altas que el primer usuario no haya visto.

## 2.4 Formato RSS

Una forma muy común de difusión de información en la web, es por medio de sitios en formato RSS. El formato RSS es utilizado para proveer a un sitio web un documento con un formato tipo XML que contiene diversos registros, los cuales contienen la información del registro, como: autor, fecha, copyright y otros datos relevantes, un resumen y una liga al contenido completo de dicho registro (Wusteman, 2004).

Este tipo de formato es ampliamente utilizado en sitios como periódicos y revistas particulares, así como en sitios o herramientas personalizados que se alimentan de noticias de diferentes fuentes. Esto permite a un usuario poder leer un resumen e información relevante antes de acceder a la noticia completa.

## 2.5 Resumen

En este capítulo se presentó la base teórica de diversas técnicas que participan en la creación de sistemas automáticos de recomendaciones.

Primero se observó la definición y la importancia de la recuperación de información de algún medio para poder usarla posteriormente.

Después se abordó la necesidad de representar los documentos en formatos específicos para poder procesarlos, se presentaron diferentes métodos de representación, que ventajas y desventajas conllevan y ejemplos de ellos.

Finalmente se mencionó la minería de datos, su utilidad en este tipo de desarrollos haciendo énfasis en la minería de datos para la web y algunos tipos de procesamiento.

El análisis realizado de las ventajas y desventajas permitió seleccionar algunas técnicas que se usarán en el siguiente capítulo.

# Capítulo 3

# Sistema automático de recomendación de noticias

## 3.1 Descripción del problema

El objetivo principal de este trabajo es realizar un sistema automático de recomendaciones de noticias. El sistema desarrollado será capaz de obtener las noticias para procesarlas y agruparlas de forma automática de acuerdo a los temas que se discuten en su contenido. A partir de estas agrupaciones, se identificarán las noticias que pueden ser de interés para un usuario en específico, de acuerdo a la información de su perfil, y se generará una recomendación final.

## 3.2 Sistema automático de recomendación de noticias basado en agrupamiento

A continuación se presenta el funcionamiento detallado del sistema automático de recomendación de noticias propuesto.

### 3.2.1 Funcionamiento general

El primer paso para poder lograr la construcción del sistema automático de recomendación de noticias es la obtención de dichas noticias de algunos sitios web u otra fuente de información. Para este trabajo se utilizaron como fuente de información sitios web de diferentes periódicos, como el periódico Reforma, que ofrece sus noticias organizadas en RSS. En la figura 3.1 se presenta un ejemplo de una noticia en este formato. Los detalles de la estructura de estas fuentes RSS se presentan en los anexos.

```
<item>
  <title>Se debilita la creación de empleos en la industria</title>
  <link>
    http://www.jornada.unam.mx/2016/10/03/economia/025n1eco?partner=rss
  </link>
  <description>
    La Confederación de Cámaras Industriales (Concamin) advirtió este domingo
    que la generación de empleos en la industria comenzó a debilitarse, y que el
    avance observado durante el primer semestre del año &#8220;difícilmente se
    extenderá por mucho tiempo más&#8221;.
  </description>
  <pubDate>Mon, 03 Oct 2016 06:01:09 GMT</pubDate>
</item>
```

**Figura 3.1**: Estructura general de una noticia en formato RSS

Un cliente RSS es un software capaz de obtener contenido en este formato, para este caso, las noticias de cada sección pueden ser obtenidas y almacenadas para ser tratadas como colecciones de información, donde cada colección contiene la siguiente información: *el periódico, el autor, el encabezado, una descripción general de la noticia completa, una URL a la noticia completa en el sitio web* y más información que puede ser de utilidad para las aplicaciones que leen los formatos RSS.

El procedimiento completo del sistema se presenta a continuación:

**Obtención de información:** se obtienen las noticias mediante un cliente RSS y se generan los objetos que contienen la información necesaria para el procesamiento de cada noticia. Para cada noticia se considera únicamente el título, la descripción y la URL a la noticia completa.

**Representación de documentos:** el título y la descripción son convertidos a letras minúsculas y se eliminan todos los caracteres que no sean letras, las letras con acentos son reemplazadas por la misma letra sin acento y finalmente, son procesados a través del algoritmo de Porter para ser almacenados en la colección final de textos. La URL se coloca por separado (en un HashMap) para que al finalizar el proceso de agrupamientos se realice la recomendación entregando dicha URL.

**Cálculo del total de términos distintos:** la colección de textos ya procesados, se recorre para generar una nueva colección que contendrá todos los términos diferentes que existan entre todos los textos.

**Conteo de términos y normalización de vectores:** Cada texto dentro de la colección será recorrido para calcular el valor TF*IDF de cada término dentro de cada texto de la colección y se almacenará en un HashMap de vectores.

**Adaptación de vectores:** como se menciona más adelante en la sección 3.2.2, se utilizaron bibliotecas de Java de terceros para el manejo de información, que requieren la información en un formato específico, entonce es necesario convertir a esta representación, por lo que el HashMap de vectores numéricos es recorrido nuevamente para convertir cada vector en un vector que sea válido.

**Agrupamiento:** posteriormente se ejecuta el algoritmo K-Means con los parámetros seleccionados y este devuelve las agrupaciones correspondientes.

**Generación de recomendaciones:** una vez realizadas las agrupaciones

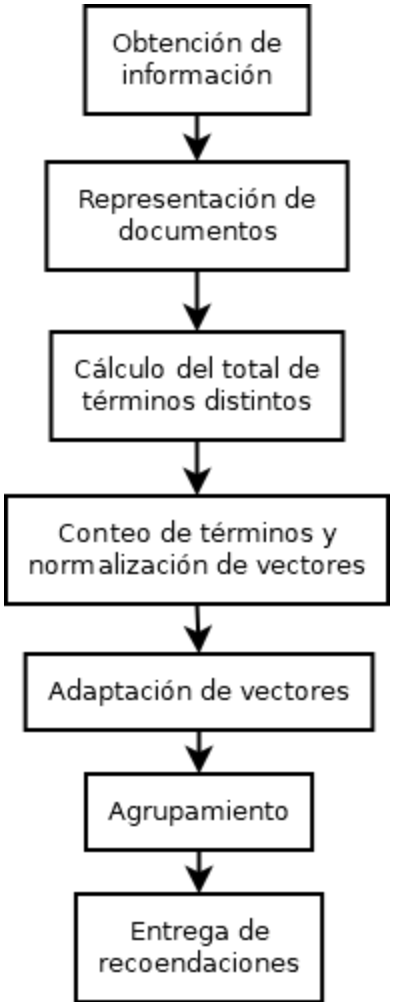


**Figura 3.2**: Diagrama general del flujo de ejecución de la aplicación

### 3.2.2 Desarrollo

El sistema automático de recomendación de noticias fue desarrollado por 2 caminos diferentes para cubrir diferentes necesidades durante la creación y las pruebas ejecutadas del sistema.

> El primer desarrollo fue hecho por medio de Java con el apoyo de las bibliotecas nativas de Java: java.util para las estructuras de datos, java.net para las conexiones necesarias a los sitios web y javax.xml para el manejo de las estructuras en formato XML También se ocupó la biblioteca de terceros javaml para el manejo del algoritmo de agrupamiento. Este desarrollo se utilizó principalmente para poder observar paso a paso el algoritmo y los resultados que se iban obteniendo y al hacer un desarrollo particular se pueden obtener los resultados deseados en cualquier punto de la ejecución.
>
> El segundo desarrollo fue realizado también por medio de Java pero utilizando la biblioteca de Weka, software desarrollado por la Universidad de Waikato en Nueva Zelanda especializado en herramientas de aprendizaje de máquina y de minería de datos. (Witten *et. al.*, 2016). Este desarrollo se ocupó ya que al ser una herramienta especializada tiene un manejo estable para grandes cantidades de información. En adición, para las pruebas finales que están orientadas a resultados finales ya no es necesario realizar revisiones de cómo va luciendo la información durante las partes intermedias del proceso. Por lo que esta herramienta resulta más conveniente para el manejo de grandes volúmenes de información como un periódico completo.

Todos los ejemplos descritos en esta sección, fueron tomados de una ejecución realizada con noticias del día 18 de octubre de 2016 del periódico “Reforma”, que contaba con 10 noticias de la sección de “gadgets” lo cual permitió analizar fácilmente los resultados del sistema.

Este capítulo cubre principalmente el desarrollo particular realizado en Java ya que aborda la creación de un sistema de recomendación de noticias por medio del desarrollo de las herramientas vistas en el capítulo anterior para observar su construcción en un sistema real.

#### 3.2.2.1 Obtención de información

Como ya se mencionó anteriormente el primer paso es la recuperación de las noticias que se utilizarán como la fuente de información.

Para este desarrollo específico se utilizó un periódico que cuenta con una edición web de consulta que es: *El Reforma*.

Dichos periódicos se seleccionaron ya que su versión de consulta web se encuentra en formato *RSS.* Recordemos que solo se utiliza el título, la descripción y la URL completa de la noticia" por las razones ya mencionadas.

Utilizando las siguientes bibliotecas de java nativas mencionadas se realiza la conexión a los sitios de los periódico y se obtienen los documentos en formato XML.

A continuación se presenta un ejemplo del contenido XML de una noticia del periódico Reforma.

```
<item>
  <idelementosubcategoria>965461</idelementosubcategoria>
  <title>
    <![CDATA[ Apuesta Walmart por streaming gratuito ]]>
  </title>
  <link>
    http://www.reforma.com/aplicaciones/articulo/default.aspx?id=965461
  </link>
  <guid isPermaLink="true">
    http://www.reforma.com/aplicaciones/articulo/default.aspx?id=965461
  </guid>
  <pubDate>Tue, 18 Oct 2016 19:04:07 GMT</pubDate>
  <description>
    <![CDATA[
        Anuncia Walmart la plataforma Vudu Movies On Us, que ofrece streaming
gratuito de             películas soportado por anuncios.
     ]]>
  </description>
  <author>Gadgets / Staff</author>
  <textofacebook>
    <![CDATA[ Apuesta Walmart por streaming gratuito ]]>
  </textofacebook>
  <textotwitter>
    <![CDATA[
        El servicio de streaming #VuduMoviesOnUs de @Walmart ofrece películas
gratis  soportadas por anuncios.
    ]]>
  </textotwitter>
  <IdCategoria>90</IdCategoria>
  <IdSubCategoria>1562</IdSubCategoria>
</item>
```

**Figura 3.3**: Ejemplo de noticia en formato RSS

Luego del procesamiento de la noticia, el sistema considerará solamente lo siguiente:
*Apuesta Walmart por streaming gratuito Anuncia Walmart la plataforma Vudu Movies On Us, que ofrece streaming gratuito de películas soportado por anuncios.*

### 3.2.2.2 Representación de documentos

Como se mencionó en el capítulo anterior, primero se le eliminan todos los caracteres especiales y las letras que estén acentuadas son reemplazadas por la misma letra sin acento. Posteriormente el texto es sometido a la ejecución del algoritmo de Porter, adaptado al español, para reducir el texto como ya se explicó en el capítulo anterior. (Panessi y Bordignon, 2001)

A continuación se presenta la forma final del ejemplo presentado anteriormente:
*apuest walmart por streaming gratuit anunci walmart la plataform vudu movi on us que ofrec streaming gratuit de pelicul soport por anunci*

todos los textos son almacenados en un arreglo de cadenas y dentro de un *HashMap* se almacenarán las URLs a las noticias completas usando el índice de la noticia en el arreglo como la llave en el HashMap, para que, cuando se termine el proceso completo, se puedan acceder a las noticias recomendadas.

### 3.2.2.3 Cálculo del total de términos distintos

Siguiendo el proceso mencionado, es necesario obtener una representación vectorial de las noticias.

Teniendo los textos almacenados en las colecciones mencionadas, es necesario conocer todos los términos diferentes que aparecen en todos los textos disponibles, para esto, es necesario recorrer nuevamente todos los textos en el arreglo e ir almacenando en otro arreglo los términos diferentes. Al finalizar este proceso, se tendrá el arreglo general de textos y un arreglo de todos los términos diferentes.

A continuación se muestra la matriz de los primeros términos del ejemplo realizado, cada vector cuenta con 141 términos. En el ejemplo se muestran sólo 6 términos de 2 textos de los 10 que se tienen.

| Texto / Término | apuest | walmart | por | streaming | arrest | a | ... |
|---|---|---|---|---|---|---|---|
| Texto 1 | 1 | 1 | 1 | 1 | 0 | 0 | ... |
| Texto 2 | 0 | 0 | 0 | 0 | 1 | 1 | ... |

**Figura 3.4**: Matriz de apariciones de términos

### 3.2.2.4 Conteo de términos y normalización de vectores

En este paso se calcula el vector de valores *TF*IDF* para cada término en cada texto de la colección. Dentro de este proyecto esto se logra de la siguiente manera: se va a iterar sobre la colección de elementos, por cada texto se va a iterar sobre el vector de términos.

Para cada término, se calcula su frecuencia dentro de la noticia actual y su frecuencia inversa, para esto último, se deberá considerar la aparición de este término dentro de toda la colección.

Finalmente, en el vector de términos se almacena el producto de la frecuencia y la frecuencia inversa antes calculadas

A continuación se muestra la matriz de términos para el primer texto ya con sus valores *TF*IDF.*

| Texto / Término | apuest | walmart | por | streaming | arrest | a | ... |
|---|---|---|---|---|---|---|---|
| Texto 1 | 0.10 | 0.21 | 0.08 | 0.21 | 0.00 | 0.00 | ... |
| Texto 2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.09 | 0.00 | ... |

**Figura 3.5**: Matriz de valores TF*IDF

El vector generado para esta noticia se almacena en un HashMap con la llave usando la misma llave antes mencionada.

### 3.2.2.5 Adaptación de vectores

Para realizar el agrupamiento, fue necesario adaptar la representación vectorial anterior al formato de vectores (Instance) de javaml, también son almacenados en un nuevo HashMap usando la misma llave y un objeto DataSet de javaml para que puedan ser procesados a continuación.

### 3.2.2.6 Agrupamiento

Finalmente el sistema está listo para la ejecución del algoritmo *KMeans* para encontrar los agrupamientos. La biblioteca recibe como parámetro del algoritmo el número de grupos que se desean obtener y un número máximo de iteraciones en caso de no encontrar convergencia, que como ya se había mencionado anteriormente, son las dos posibles formas de terminar la ejecución del algoritmo.

En la figura 3.6 se puede observar un ejemplo ejecutado con la misma sección que los ejemplos anteriores, pero en otra fecha, a pesar de que fue ejecutado con pocas noticias de

una sección de periódico se puede ver que en el grupo 2, 3 de las 5 noticias hablan explícitamente de apps y mencionan el término, a pesar de que no hay una separación marcada si se puede observar cierta similitud en al menos algunas de las noticias del grupo.

```
--------------------GRUPO 1--------------------
present microsoft competidor de slack microsoft team es un chat corpor
que permit compart archiv y notif de distint apps
lanz facebook su plataform de jueg gameroom tendr jueg movil tant de ios
com andro asi com titul par 'hardcor gamers'
da cook mensaj tras victori de trump el director de apple envi un corre a
sus emple en el que les pid unirs e ir haci adel
lleg spectacl de snapchat a eu las gaf conectad se vend en maquin despach
a un preci de 129 dolar
'trollean' a trump en su siti web una fall en la pagin del candidat
permiti que usuari aprovechar par cambi el encabez y dej mensaj de burl
--------------------GRUPO 2--------------------
liber uber su nuev app en mexic la nuev version incluy acces rap a destin
favorit y dar indic al pasajer sobr dond abord su uber
renuev uber su app pront ser posibl ingres el nombr de un contact en lug
de un destin y conect el calendari con la aplicacion
lanz instagram funcion par compr los usuari de la app en estad unid podr
obten preci de distint product y ser redirig a las pagin de los comerci
present xperi con sup cam los nuev sony xperi xz y xperi x compact tien
cam de 23 megapixel par situacion de poc luz
super macbook pro a competidor un report indic que en los primer dias de
vent la laptop super a competidor com microsoft surfac book y la
chromebook de asus
```

**Figura 3.6**: Ejemplo de ejecución con una sección de periódico

En el capítulo anterior, se explicó que un problema de este algoritmo es saber cuál es el número de grupos que se desean generar, para este trabajo se optó por realizar diferentes ejecuciones variando los posibles parámetros (número de grupos e iteraciones). Los resultados se discuten en el siguiente capítulo.

### 3.2.2.6.1 Integración del perfil de usuario

La etapa anterior genera una agrupación de los vectores, en donde cada uno de los grupos contiene las noticias que los cálculos realizados calificaron que tenían mayor similitud.

Para realizar una recomendación será preciso tomar en cuenta ciertas características del usuario en cuestión, para saber cuál de los diversos grupos le causaría más interés, para lograr esto es necesario realizar un perfil del usuario que represente sus intereses específicos.

Supondremos que se tiene un usuario con intereses principalmente en noticias acerca de telefonía celular, computación y otros temas de tecnología, se tiene que expresar su perfil como una colección de esos intereses, en este caso se eligió usar el mismo esquema vectorial: *[celular computadora algoritmo internet drone drones red base de datos app]*

Para poder determinar la relación entre el perfil del usuario y alguno de los grupos identificados anteriormente, se decidió insertar este perfil de usuario como un elemento más de la colección y volver a ejecutar el agrupamiento, de esta forma el procedimiento tratará al perfil del usuario igual que a una noticia. Al finalizar el procedimiento, el perfil del usuario estará insertado dentro de alguno de los grupos generados por el algoritmo, de esta forma se espera que las noticias de este grupo es de potencial interés para este usuario como se puede observar en la figura 3.7.

```
--------------------GRUPO 1--------------------
liber uber su nuev app en mexic la nuev version incluy acces rap a destin
favorit y dar indic al pasajer sobr dond abord su uber
renuev uber su app pront ser posibl ingres el nombr de un contact en lug
de un destin y conect el calendari con la aplicacion
super macbook pro a competidor un report indic que en los primer dias de
vent la laptop super a competidor com microsoft surfac book y la
chromebook de asus
celul comput algoritm internet dron red bas de dat app
--------------------GRUPO 2--------------------
present microsoft competidor de slack microsoft team es un chat corpor
que permit compart archiv y notif de distint apps
lanz facebook su plataform de jueg gameroom tendr jueg movil tant de ios
com andro asi com titul par 'hardcor gamers'
lanz instagram funcion par compr los usuari de la app en estad unid podr
obten preci de distint product y ser redirig a las pagin de los comerci
da cook mensaj tras victori de trump el director de apple envi un corre a
sus emple en el que les pid unirs e ir haci adel
lleg spectacl de snapchat a eu las gaf conectad se vend en maquin despach
a un preci de 129 dolar
present xperi con sup cam los nuev sony xperi xz y xperi x compact tien
cam de 23 megapixel par situacion de poc luz
'trollean' a trump en su siti web una fall en la pagin del candidat
permiti que usuari aprovechar par cambi el encabez y dej mensaj de burl
```

**Figura. 3.7**: Resultado de la ejecución con perfil de usuario integrado

### 3.2.2.7 Generación de recomendaciones

Como se mencionó en la sección 3.2.2.6.1 del presente trabajo, una vez ejecutado el algoritmo con la integración del perfil de usuario, se tendrá dicho perfil en un grupo el cual se estima las noticias sean del interés del usuario.

Finalmente el sistema entregará la lista de las noticias incluidas generando un archivo RSS similar a los archivos que se utilizaron de entrada. Dichos archivos de recomendaciones son entregados al usuario como se puede observar en la figura 3.8.


**Figura. 3.8**: Muestra de estructura de recomendaciones utilizando el resultado en un archivo XML con una hoja de estilo CSS.

## 3.3 Resumen

En este capítulo se ejemplificó el funcionamiento del sistema automático de recomendación de noticias propuesto en este trabajo con una cantidad reducida de noticias para facilitar la explicación.

Primero se abordó la descripción de su funcionamiento general y posteriormente se realizó la explicación detallada de cuáles de las técnicas que se observaron en el capítulo anterior fueron utilizadas.

Se mostró un ejemplo de ejecución del agrupamiento de noticias para explicar la propuesta. Finalmente, se habló de la integración de un perfil de usuario como parte del sistema para ser procesado y se realizó un nuevo ejemplo de ejecución contemplando dicho perfil para analizar las posibles recomendaciones del sistema.

En el siguiente capítulo se observarán los resultados de diversas ejecuciones del procedimiento para analizar su funcionamiento con mayor detalle.

## Capítulo 4

# Análisis de resultados

### 4.1 Introducción

En este capítulo se analizarán los resultados de varias ejecuciones del sistema dividiéndolos en 2 pruebas.

La primera prueba consistió en observar el desempeño del algoritmo de agrupamiento con pocas noticias para analizar cuáles grupos se formaban y compararlos con la categoría de noticias de la fuente original (periódico “Reforma”).

La segunda prueba consistió en analizar el funcionamiento del sistema desarrollado evaluando la relación de las noticias recomendadas con respecto a un perfil ficticio, es decir, observar qué noticias se agrupan junto con el perfil ya que éstas serían las noticias recomendadas.

Como se mencionó al inicio de este trabajo, los algoritmos de agrupamiento suelen manejar cantidades muy grandes de información, llegando a cientos de miles o incluso millones de textos, mientras que este trabajo se enfocó en las noticias que tenía un periódico regular de un día. Esto se hizo para explorar el comportamiento de dichos algoritmos con una cantidad baja de información y debido a que un sistema de recomendaciones de cualquier servicio como un periódico la información que tiene que procesar al día sería exclusivamente la de las noticias diarias.

### 4.2 Pruebas de rendimiento de la propuesta

#### 4.2.1 Descripción de la prueba: aplicación del algoritmo a 2 grupos de noticias con 2 temáticas diferentes

El objetivo de esta prueba es analizar el comportamiento del algoritmo de agrupamiento utilizando muy pocas noticias de un periódico, considerando noticias de exclusivamente 2 temáticas diferentes. De esta forma, el resultado esperado sería obtener 2 grupos con noticias de cada una de las temáticas previamente seleccionadas.

Con la información obtenida de una prueba bajo un ambiente muy controlado donde se pueden detectar los patrones que existen manualmente y comparar contra los resultados obtenidos de la aplicación del algoritmo, podemos extrapolar que el comportamiento debería ser similar en una ejecución con más información.

Para el desarrollo de la prueba se revisó el periódico “Reforma” en diferentes fechas entre el 7 y 14 de marzo de 2017. A lo largo de estas fechas se hicieron 3 ejecuciones.

En las primeras pruebas se consideraron 9 noticias que hacían referencia a “Trump” y 10 de la sección “Gadgets” del periódico, que incluyen diferentes temáticas. El resultado esperado, sería obtener una agrupación de forma tal que uno de los grupos tuviera principalmente las noticias referentes a “Trump” y el otro grupo todas las demás. Con el fin de analizar el impacto de las palabras vacías en el proceso de agrupamiento, se realizarán pruebas con textos completos y sin palabras vacías.

La siguiente parte de esta prueba se realizó con 6 noticias que se recopilaron referentes a “López Obrador” y 7 de las noticias referentes a “Trump”. El resultado esperado de esta prueba, al igual que en la anterior, sería obtener dos grupos, cada uno principalmente con las noticias de cada una de las temáticas.

### 4.2.2 Resultados primera prueba

#### 4.2.2.1 Resultados de la primera parte

Durante la primera ejecución se consideraron los textos completos (sin eliminar palabras vacías) y se obtuvieron grupos mezclados, un grupo obtuvo 5 noticias de cada una de las temáticas mientras que el otro grupo obtuvo 5 noticias de la sección de “Gadgets” y 4 noticias referentes a “Trump”.

A continuación se manejarán matrices de confusión para ilustrar la comparación entre lo que el sistema identificó y lo que realmente se sabía previamente de la división de las noticias.

Dicha matriz es cuadrada, cada fila representa el resultado de clasificación en cada uno de los grupos y cada columna representa el total de las noticias de esa temática, de esta forma se puede observar gráficamente cuáles noticias fueron incorrectamente clasificadas.

| | Trump | Gadgets |
|---|---|---|
| Trump | 5 | 5 |
| Gadgets | 4 | 5 |

**Figura 4.1**: Matriz de confusión de la primera ejecución.

Durante la segunda ejecución, también con textos completos, se obtuvieron resultados más acercados a lo esperado. Un grupo tuvo 1 noticia referente a “Trump” y 5 noticias de la sección “Gadgets”, mientras que el otro grupo obtuvo 5 noticias de la sección “Gadgets” y 8 noticias referentes a “Trump” (véase matriz de confusión. Figura 4.2).

| | Trump | Gadgets |
|---|---|---|
| Trump | 8 | 5 |
| Gadgets | 1 | 5 |

**Figura 4.2**: Matriz de confusión de la segunda ejecución.

Como ya se mencionó anteriormente, para las siguientes ejecuciones se eliminaron las palabras vacías para tener textos con términos mucho más específicos. El resultado de esta ejecución fue similar al primero, el primer grupo con 4 noticias referentes a “Trump” y 4 noticias de la sección “Gadgets”, el segundo grupo con 5 noticias referentes a “Trump” y 6 de la sección “Gadgets” (véase matriz de confusión. Figura 4.3).

| | Trump | Gadgets |
|---|---|---|
| Trump | 4 | 4 |
| Gadgets | 5 | 6 |

**Figura 4.3**: Matriz de confusión de la tercera ejecución.

#### 4.2.2.2 Resultados de la segunda parte

En la segunda prueba se obtuvieron resultados acercados al resultado esperado. Uno de los grupos tuvo como resultado 5 noticias referentes a “Trump” y 2 noticias referentes a “López Obrador”, mientras que el otro grupo obtuvo 3 noticias referentes a “Trump” y 4 noticias referentes a “López Obrador” (véase matriz de confusión. Figura 4.4).

| | Trump | López Obrador |
|---|---|---|
| Trump | 5 | 2 |
| López Obrador | 3 | 4 |

**Figura 4.4**: Matriz de confusión de la cuarta ejecución.

### 4.2.3 Observaciones de las pruebas

En las primeras dos ejecuciones que cuentan con todos los términos existentes de las noticias se obtuvieron dos resultados distintos, uno con grupos muy mezclados y otro con grupos que se acercaron más al resultado que se tenía esperado. Esto significa que las noticias contenían demasiados términos diferentes como para encontrar suficientes términos en común, esto se debe parcialmente a la aparición de las palabras vacías dentro de las noticias.

Para tratar de corregir este desperfecto se decidió eliminar las palabras vacías de todas las noticias para las siguientes pruebas, de esta forma el vector queda con los términos más relevantes y se omiten todos los términos que son irrelevantes.

En la siguiente ejecución se puede observar que, si bien el algoritmo ya discrimina mejor al eliminar las palabras vacías, siguen mostrándose grupos mezclados. Al ver los textos de las noticias que se están procesando, se puede deducir que este comportamiento se debe a que logra identificar un pequeño grupo de noticias con algunos términos en común, y en otro grupo se incluyen el resto de las noticias. En particular, la mayoría de las noticias son de la sección “gadgets” que al ser de una temática más variada usa términos más diversos.

Para la última ejecución se abordó el problema anterior y se tomaron noticias con dos temáticas más específicas, con esto el resultado esperado sería obtener los dos grupos originales.

El resultado fue cercano al resultado esperado, obteniendo dos grupos con una mayoría de noticias de cada uno de los temas: “Trump” y “López Obrador”. Con esto se puede ver que a lo largo de las iteraciones del algoritmo las noticias que tienen términos en común van quedando juntos gradualmente. Debido a la naturaleza de los textos y del funcionamiento del algoritmo, es natural que en ocasiones no exista una agrupación perfecta.

## 4.3 Ejecución del algoritmo de recomendación

### 4.3.1 Descripción de la ejecución

Esta sección de pruebas tiene como objetivo describir el análisis del comportamiento del algoritmo con una colección de noticias (título y resumen de la noticia), variando algunos parámetros, que son descritos en los diferentes escenarios de las pruebas realizadas.

Para todos los escenarios a continuación se ocuparon las 184 noticias, de 21 secciones del periódico Reforma en adición al perfil generado manualmente, dando un total de 185 textos para procesar.

Las pruebas se realizaron de la siguiente forma: se ejecutó el algoritmo variando, en cada ejecución, la semilla de generación de números aleatorios, que causa una variación en la selección de los centroides iniciales y el número de grupos, entre 3 y 5 para los primeros 2 escenarios de prueba y entre 3 y 6 para los siguientes 2 escenarios.

El perfil de usuario fue generado manualmente a partir de diversas noticias de fechas anteriores en las cuales se seleccionaron noticias personalmente que generaron interés. Las palabras incluídas en el perfil son las siguientes:

*atentan, parlamentario, kabul, ataque, miembro, afganistan, coche, bomba, varios, heridos, jefe, seguridad, congreso, mark, zuckerberg, conisgui, crear, asistencia, inteligencia, artificial, casa, propuso, principios, año, proponen, transportar, frutas, telarañas, alumnos, up, proceso, transportacion, almacenamiento, conservacion, venta, directa, frutas, emulando, telaraña, china, pasos, agigantados, stem, competitivo, vislumbra, mercado, laboral, miles, chinos, pagan, cursos, robotica, programacion, hijos, niños, acuerdan, alto, fuego, siria, turquia, rusia, acordaron, cese, fuego, toda, siria, iniciara, medianoche, podrian, negociar, kazajistan, estancaran, dolar, recorte, becas, conacyt, dudan, conacyt, 2017, pueda, aumentar, padron, becas, debido, recorte, devaluacion, peso, frente, dolar, revelan, secretos, cerebro, investigadores, encuentran, cerebro, decodificar, mensajes, distorsionados, escucha, segunda, alista, segunda, edicion, exomars, agencia, espacial, europea, firmo, consorcio, thales, alenia, space, desarrollar, visitas, marte, llega, futuro, suv, tesla, model, ofrece, conduccion, autonoma, conexion, internet, defensa, armas, biologicas*

#### 4.3.1.1 Escenario 1

Para la limpieza, se utilizó una lista estándar de palabras vacías para el idioma español, al igual que en la sección anterior.

En este escenario se tenían las siguientes caracteristicas:

Número total de términos (columnas): 1620
Tamaño del perfil (número de términos): 113

Tras diversas ejecuciones variando los parámetros ya mencionados se observó que en todas ellas se obtiene un grupo con una concentración más alta de noticias que todos los demás grupos, independientemente de la cantidad de grupos seleccionados o la cantidad de noticias con las que se ejecutó el algoritmo. Los demás grupos se reparten el resto de las noticias teniendo concentraciones mucho más bajas. En la siguiente tabla (Figura 4.5), pueden observarse algunos de los ejemplos de las ejecuciones realizadas, en ella se muestran la saturación de noticias en un solo grupo.

| | Grupo 0 | Grupo 1 | Grupo 2 | Grupo 3 | Grupo 4 |
|---|---|---|---|---|---|
| 3 grupos | 14 | 10 | 160 + perfil | NA | NA |
| 4 grupos | 7 | 10 | 12 | 155 + perffil | NA |
| 5 grupos | 7 | 10 | 12 | 150 + perfil | 5 |

**Figura 4.5**: Tabla de distribución de noticias en grupos, escenario 1

En la mayoría de las ejecuciones, el grupo con la concentración más alta de ellas tenía entre 130 y 160 noticias, el resto de las noticias se repartían en los grupos restantes, en los grupos pequeños se obtuvieron cantidades de menos de 10 noticias, en pocas ocasiones se mostró un grupo de entre 10 y 15 noticias.

Con respecto a las palabras vacías, una lista estándar puede no ser suficiente para una colección de este tipo, pues se observan los siguientes 2 casos: hay palabras que son demasiado utilizadas y aunque no sean palabras vacías estándar podríamos quitarlas para ver analizar su impacto en el agrupamiento; o al contrario, hay términos que solo se ocupan en una de las noticias, por lo que no aportan mucho al análisis y se podrían quitar.

Al colocar el perfil del usuario como parte de los datos de entrada del algoritmo, siempre lo colocó en el grupo más grande, es decir, el algoritmo no fue capaz de encontrar un grupo de noticias de interés claramente.

En la tabla 4.6 se puede observar la frecuencia de las 15 palabras más utilizadas dentro del grupo en el que fue asignado el perfil de usuario. Se pueden observar 2 palabras que se encuentran en el perfil: *casa y edicion*.

| **Palabra** | **Cantidad** |
|---|---|
| años | 15 |
| mexico | 12 |
| presidente | 8 |
| peña | 6 |
| ifa | 6 |
| **casa** | 6 |
| tlc | 5 |
| primera | 5 |
| semana | 5 |
| ciudad | 5 |
| **edicion** | 5 |
| musica | 5 |
| eleccion | 5 |
| cuerpo | 5 |

**Figura 4.6**: Frecuencia de palabras dentro del grupo al que pertenece el perfil de usuario para ejecución con 5 grupos.

Bajo este escenario, se puede observar que el algoritmo no está arrojando un resultado adecuado, ya que al usuario se le recomendarían noticias del grupo más numeroso, lo cual puede ser que no sea de su interés, pues como se dijo anteriormente estas noticias tienen

pocos términos en común con el usuario, entonces las recomendaciones serían poco precisas ya que, el sistema le recomendaría casi todas las noticias.

#### 4.3.1.2 Escenario 2

Se identificó que algunos de los términos más utilizados en las noticias eran: *años, México, presidente, Trump, elección, TLC, EU, etc*. Estos términos se repetían muchas veces en una cantidad alta de noticias, por lo que pudieron haber tenido un efecto similar al de las palabras vacías y calificarse como términos poco importantes en la ejecución del escenario 1. En este escenario se exploró eliminarlos para disminuir la dimensión de la matriz de términos, para ello se agregaron a la lista de las palabras vacías, para que fueran eliminadas en las ejecuciones de este escenario.

En este escenario se tenían las siguientes caracteristicas:

Número total de términos (columnas): 1605
Tamaño del perfil (número de términos): 109

Al realizar estas ejecuciones variando los parámetros se muestran resultados similares a los presentados durante el escenario anterior; un grupo con una concentración más alta de noticias que los demás. Nuevamente el perfil del usuario quedó agrupado, en cada ocasión, dentro del grupo de la concentración más alta.

Con estas ejecuciones se observó que cuando los términos más frecuentes se eliminaron se obtenían resultados ligeramente menos satisfactorios que en el escenario anterior. De igual forma se obtuvo un grupo con una concentración más alta de noticias y los demás con las demás noticias distribuídas. La diferencia entre este y el escenario anterior es que los grupos que con pocas noticias tuvieron menos que en el escenario pasado haciendo el grupo con más elementos aún más saturado.

| | Grupo 0 | Grupo 1 | Grupo 2 | Grupo 3 | Grupo 4 |
|---|---|---|---|---|---|
| 3 grupos palabras más repetidas eliminadas | 170 + perfil | 7 | 7 | NA | NA |
| 4 grupos palabras más repetidas eliminadas | 168 + perfil | 7 | 7 | 2 | NA |
| 5 grupos palabras | 6 | 6 | 7 | 2 | 163 + perfil |

| | | | | | |
|---|---|---|---|---|---|
| más repetidas eliminadas | | | | | |

**Figura 4.7**: Tabla de distribución de noticias en grupos, escenario 2, palabras más repetidas eliminadas

Dados estos resultados podemos observar que eliminar los términos más frecuentes causa que no existan términos en común importantes para realizar una agrupación.

A continuación se puede observar en la tabla 4.8 la frecuencia de las 15 palabras más utilizadas dentro del grupo en donde fue asignado el perfil de usuario. En esta tabla se puede observar que solo una pertenece al perfil de usuario.

| **Palabra** | **Cantidad** |
|---|---|
| ciudad | 7 |
| nacional | 6 |
| 25 | 5 |
| lluvias | 5 |
| politica | 5 |
| presento | 5 |
| dan | 4 |
| zona | 4 |
| tres | 4 |
| elecciones | 4 |
| ciencia | 4 |
| muertos | 4 |
| guerrero | 4 |
| caer | 4 |
| **año** | 4 |

**Figura 4.8**: Frecuencia de palabras dentro del grupo al que pertenece el perfil de usuario para ejecución con 3 grupos.

En este escenario se vuelve a observar que no hay una recomendación adecuada ya que el perfil de usuario fue colocado nuevamente en el grupo con mayor saturación y esto causa que no haya realmente un grupo de recomendaciones particular para el perfil.

#### 4.3.1.3 Escenario 3

Durante los experimentos se detectó que algunos de los términos menos utilizados durante el procedimiento eran: *acaba, acabo, afectar, afectado, alta, alto, academia, accesorios, etc.*

Estos términos tenían solo una y dos apariciones entre todas las noticias comparado con otros términos que iban desde 3 hasta 15 apariciones. En este escenario se exploró eliminar estos términos poco repetidos para reducir el volumen de términos para lo cual fueron agregados junto con la lista de palabras vacías para ser eliminados junto con ellos.

En este escenario se tenían las siguientes caracteristicas:

Número total de términos (columnas): 1493
Tamaño del perfil (número de términos): 101

Se realizaron nuevamente las ejecuciones variando los mismos parámetros ya mencionados y los resultados obtenidos son, nuevamente similares a los obtenidos en los 2 anteriores escenarios de ejecución, un grupo con una concentración más alta de noticias que contiene al perfil de usuario y los demás grupos dividiendo las noticias restantes.

| | Grupo 0 | Grupo 1 | Grupo 2 | Grupo 3 | Grupo 4 | Grupo 5 |
|---|---|---|---|---|---|---|
| 3 grupos palabras menos repetidas eliminadas | 3 | 7 | 174 + perfil | NA | NA | NA |
| 4 grupos palabras menos repetidas eliminadas | 2 | 1 | 176 + perfil | 5 | NA | NA |
| 5 grupos palabras menos repetidas eliminadas | 6 | 7 | 4 | 2 | 165 + perfil | NA |
| 6 grupos palabras menos repetidas | 3 | 5 | 3 | 1 | 165 + perfil | 7 |

**Figura 4.9**: Tabla de distribución de noticias en grupos, escenario 3, palabras menos repetidas eliminadas

A continuación se puede observar en la tabla 4.10 la frecuencia de las 15 palabras más utilizadas dentro del grupo en donde fue asignado el perfil de usuario. De esas palabras se puede observar que solo una de las más repetidas pertenece al perfil de usuario.

| Palabra | Cantidad |
|---|---|
| años | 14 |
| mexico | 13 |
| trump | 10 |
| tlc | 7 |
| primera | 7 |
| ciudad | 7 |
| presidente | 7 |
| mil | 7 |
| eleccion | 7 |
| personas | 7 |
| peña | 6 |
| morena | 6 |
| cuerpo | 6 |
| **casa** | 6 |
| semana | 5 |

**Figura 4.10**: Frecuencia de palabras dentro del grupo al que pertenece el perfil de usuario para ejecución con 6 grupos.

En este escenario la recomendación vuelve a ser inadecuada ya que el perfil de usuario queda agrupado en el grupo con mayor concentración de noticias y eso causa nuevamente que las posibles recomendaciones sean demasiadas para el perfil.

#### 4.3.1.4 Escenario 4

Para este escenario se usó la lista estándar de palabras vacías y en adición se agregaron palabras vacías extras que se consideró podrían mejorar el rendimiento del proceso obteniéndolas del sitio web (Spanish Stop Words, s.f).

También se afinó el perfil de usuario, en los escenarios anteriores, el perfil era una colección mucho más amplia de términos, de 128 términos, mientras que las noticias promedio contienen entre 20 y 30 términos lo cual lo hacía mucho más amplio que el vector promedio.

También se consideraron los términos de interés más reciente ya que las primeras pruebas se realizaron tomando en cuenta fechas del periódico más antiguas y el perfil no mostraba una cercanía a las noticias que estaban circulando cuando se hicieron nuevas prueba, con esto se hizo un perfil de usuario mucho más corto utilizando dichos términos.

En este escenario se tenían las siguientes caracteristicas:

Número total de términos (columnas): 1550

Tamaño del perfil (número de términos): 25

El perfil de usuario generado contiene las siguientes palabras:
eu, amenaza, norcorea, pentagono, corea, norte, militar, ciencia, revolucion, museo, nacional, armado, prd, fiscal, congreso, republica, elecciones, politica, alianzas, proceso, electoral, traicion, mexico, corrupcion, impunidad.

Tras realizar la ejecución con las mismas variaciones de parámetros, los resultados obtenidos en este experimento fueron mucho más satisfactorios que los anteriores 3 escenarios. Aún se presenta un grupo con una concentración de noticias mucho más alta que los demás, sin embargo en este escenario, en varias ocasiones se pudo observar que los grupos pequeños obtuvieron una cantidad más alta de noticias, generando grupos más poblados que en cualquiera de los 3 anteriores escenarios anteriores, como se puede observar en la siguiente tabla (Figura 4.11).

| | Grupo 0 | Grupo 1 | Grupo 2 | Grupo 3 | Grupo 4 | Grupo 5 |
|---|---|---|---|---|---|---|
| 3 grupos perfil corto | 1 | 10 | 173 + perfil | NA | NA | NA |
| 4 grupos perfil corto | 154 | 14 | 3 | 13 + perfil | NA | NA |
| 5 grupos perfil corto | 149 | 14 | 3 | 13 + perfil | 5 | NA |
| 6 grupos perfil corto | 143 | 14 | 3 | 12 + perfil | 5 | 7 |

**Figura 4.11**: Tabla de distribución de noticias en grupos, escenario 4, palabras menos repetidas eliminadas en adición a perfil de usuario corto

El resultado que es más importante es que el perfil de usuario no quedó siempre en el grupo mayor, sino que quedó en alguno de los grupos más pequeños.

respondera **eu** amenazas **norcorea** jefe **pentagono** james mattis aseguro cualquier **amenaza corea norte** recibira respuesta **militar** unidos
responde surcorea ejercicios **norcorea** ejercito corea sur realizo ejercicio misiles airetierra balisticos respuesta sexta prueba nuclear **norcorea**
confirma **norcorea** sexta prueba nuclear corea **norte** confirmo realizo exito sexto ensayo nuclear trump afirmo acciones peligrosas **eu**
buscaria **eu** terminar tlc surcorea diario the washington post revelo trump tendria planeado preparar salida comercial corea sur
condena **mexico** ensayo nuclear **norcorea** gobierno **mexico** reprobo ensayo nuclear realizado corea **norte** califico acto hostil irresponsable

**Fig 4.12**: Ejemplo de noticias con términos similares al perfil.

En la tabla 4.13 se puede observar la frecuencia de las 12 palabras más utilizadas dentro del grupo en donde fue asignado el perfil de usuario, esto permite observar que hay muchas coincidencias con las palabras: *mexico, eu, corea, norcorea, norte* y *amenaza*.

| **Palabra** | **Cantidad** |
|---|---|
| trump | 10 |
| **mexico** | 6 |
| **eu** | 6 |
| **corea** | 5 |
| presidente | 5 |
| **norcorea** | 5 |
| nuclear | 5 |
| tlc | 5 |
| **norte** | 3 |
| ensayo | 3 |
| acciones | 3 |
| **amenaza** | 3 |
| aristoteles | 2 |
| terminar | 2 |
| automatico | 2 |

**Figura 4.13**: Frecuencia de palabras dentro del grupo al que pertenece el perfil de usuario para ejecución con 4 grupos.

Al observar ejemplos de las noticias que componen dichos grupos, en los que se encontró el perfil de usuario, se logra observar noticias con términos similares al perfil, es decir, de posible interés para el usuario.

Como se puede observar en este escenario, la ejecución del algoritmo con un perfil de usuario más corto ayudó a la agrupación del mismo, colocándolo más fácilmente dentro de grupos con los que puede tener más en común.

Acotar el grupo con términos que sean más concernientes a las noticias más recientes, facilita que el perfil sea asociado con noticias recientes en lugar de ser asociado con casi todas las noticias existentes. Esto indica que en un sistema de recomendaciones el perfil tiene que renovarse constantemente para mantener la actualidad con la información entregada.

Esto se observó en todas las ejecuciones realizadas para 3 grupos y en algunas ejecuciones para más grupos. Esto es importante identificarlo ya que permite observar que el algoritmo para 3 grupos no encuentra una agrupación óptima ya que son demasiados vectores para pocas agrupaciones.

## 4.4 Resumen

En este capítulo se observaron los resultados obtenidos por las ejecuciones del sistema propuesto en el capítulo anterior, dichos resultados se obtuvieron en 2 fases.

En la primera fase se realizaron ejecuciones controladas obteniendo un número específico de noticias de las cuales se sabía de antemano que se dividían en 2 grupos, posteriormente se revisó si los resultados obtenidos coincidían con las agrupaciones que se sabían existía previamente.

La segunda fase consistió en ejecutar el algoritmo completo en todas las noticias disponibles para una fecha específica de un periódico y el perfil de usuario generado a partir de noticias previas y variando los parámetros de cantidad de grupos y de semillas aleatorias de inicio de centroides y se analizaron los resultados obtenidos en diferentes tipos de ejecuciones.

En la siguiente liga se pueden encontrar todos los archivos generados durante las diferentes ejecuciones del sistema: *https://drive.google.com/drive/folders/1IR85Qg1KJ0NipdcngiMKmURUl5Rb2rou?usp=sharing*

# Capítulo 5

# Conclusiones

## 5.1 Resumen general

El presente trabajo presentó una investigación y revisión de algunos métodos de minería de datos, con el fin de diseñar un sistema de recomendaciones de noticias de manera automática.

Inicialmente se presentaron los temas relevantes para el trabajo:

Representación de documentos: se abordaron las diferentes formas en las que se puede presentar la información de los documentos de texto así como su preprocesamiento para poderla analizar con los algoritmos específicos de minería de datos.

Técnicas de minería de datos: se revisaron diferentes técnicas existentes técnicas que hay para la identificación de patrones en grandes volúmenes de información, particularmente aquellos aplicados a la web.

Diferentes sistemas automáticos de recomendaciones: en esta sección se analizaron diferentes sistemas de recomendación para observar sus características, ventajas y desventajas.

Con base en el análisis del funcionamiento de los sistemas de recomendación clásicos, se decidió construir un sistema de recomendación basado en el filtrado de contenidos pero considerando una estrategia diferente, el uso de técnicas de agrupamiento. Con este enfoque, se buscó explorar la capacidad del algoritmo k-means para determinar las relaciones implícitas en los términos de las noticias de interés del usuario y así encontrar un grupo de noticias que puedan ser de su interés.

Siguiendo esta idea, se diseñó una estrategia de recomendación de noticias la cual fue implementada en el sistema propuesto, se explicó a detalle la realización de dicho sistema en cada uno de sus pasos, algoritmos utilizados y entrega de resultados obtenidos.

Finalmente se hizo un análisis de los resultados obtenidos con dicho sistema. Se desarrollaron diferentes pruebas, inicialmente manteniendo pocas noticias y grupos para poder observar las noticias recomendadas, y posteriormente se analizaron las noticias que ofrece un periódico en RSS en todo un día y se observó el comportamiento que el algoritmo presentó en ellas.

## 5.2 Conclusiones

Los algoritmos seleccionados para la realización del presente trabajo, constituyen una propuesta diferente de abordar el problema de la recomendación de noticias. Se retomó la flexibilidad que ofrece el análisis basado en cluster y se propuso una estrategia flexible que permite actualizar el perfil de usuario de una forma ágil.

Como se mencionó en ocasiones anteriores los algoritmos de agrupamiento suelen operar con mucha información. Sin embargo este trabajo abordó la problemática de realizar agrupamientos partiendo de una fuente con un volumen más bajo de información como lo es un periódico regular y las noticias de un día. Esto se realizó de esta manera para tratar de abordar la problemática de recomendación de noticias del día o las más recientes. Esta problemática sigue siendo muy actual, ya que los sistemas de recomendación de noticias utilizados por los dispositivos móviles siguen buscando mejores estrategias, considerando diversas fuentes de información, pero aún siguen presentando el problema de recomendar la misma noticia de diferentes fuentes.

Los resultados obtenidos en el presente trabajo, como se observó en el análisis de ellos, fueron favorables bajo algunos de los escenarios propuestos y la experimentación que se realizó modificando los parámetros del algoritmo principal que se tomó para los agrupamientos.

De estos resultados se puede observar que los algoritmos de agrupamiento también pueden tener resultados positivos al utilizarse con cantidades de información con cientos de elementos y no necesariamente miles o millones.

Se puede concluir que el sistema propuesto en el presente trabajo sí puede generar una serie de recomendaciones a un perfil de usuario cuyos intereses se agrupen dentro de una colección de noticias para colocarlo en un grupo donde se encuentre mayor afinidad.

Dicho sistema, podrá ser reforzado con el tiempo afinando el perfil de usuario reemplazando intereses antiguos o aumentando nuevos, así como constantemente realizando la ejecución contra nuevas colecciones de noticias completas.

## 5.3 Trabajo a futuro

Como se mencionó previamente, el sistema funcionará mejor si se actualiza constantemente el perfil, es decir, el perfil de usuario debería sufrir cambios constantes tanto automatizados como manualmente para eliminar intereses antiguos o agregar intereses nuevos. Así como en algunos sitios de noticias, videos o series y películas, en el sistema de recomendación propuesto, podría realizarse una calificación manual del usuario para evaluar el interés que tuvo con la noticia o automáticamente eliminar términos del perfil de usuario si constantemente el usuario no lee las noticias recomendadas con dicha temática.

Otra mejora que existe para el sistema es la portabilidad y comodidad del usuario. Como se mencionó durante el desarrollo, el resultado se entrega en un formato de archivo RSS, dicho archivo puede ser leído por cualquier sistema de manejo de noticias en formatos de RSS o incluso alguna aplicación de celular para mayor comodidad del usuario.

# Anexo A

## A.1. Estructura de una noticia proveniente de una fuente RSS

RSS (RDF Site Summary) es un formato para distribución de elementos en la red en formato XML. Cada elemento en la colección contiene una URL para extraerlo de la red y una serie de datos relevantes del mismo. (Guha *et. al.*, 2004)

A continuación se muestran los diferentes elementos particulares de una noticia del periódico Reforma en formato RSS.

Item: cada noticia se encuentra englobada dentro de esta etiqueta.
idelementosubcategoria: identificador de la noticia para el periódico.
title: Título de la noticia.
enclosure: imagen asociada a la noticia.
link: enlace a la noticia completa.
guid: enlace alterno de la noticia.
grlink: enlace corto de la noticia para publicación en redes sociales.
pubdate: fecha de publicación de la noticia.
description: descripción breve de la noticia.
author: autor de la noticia.
textofacebook: texto para publicación en facebook.
textotwitter: texto para publicación en twitter.
IdCategoria: identificador del periódico para categorizar las noticias.
IdSubCategoria: identificador del periódico para categorizar más profundamente.

```
<item>
  <idelementosubcategoria>1864190</idelementosubcategoria>
  <title>
    <![CDATA[ Cayó PIB 0.1%; peor desempeño en 10 años ]]>
  </title>
  <enclosure>
    <![CDATA[
      https://img.gruporeforma.com/imagenes/960x640/4/783/3782101.jpg
    ]]>
  </enclosure>
  <link>
    http://www.reforma.com/aplicaciones/articulo/default.aspx?id=1864190
  </link>
  <guid isPermaLink="true">
    http://www.reforma.com/aplicaciones/articulo/default.aspx?id=1864190
  </guid>
  <grlink>https://refor.ma/cahY7K</grlink>
```

```
  <pubDate>Thu, 30 Jan 2020 06:20:13 CDT</pubDate>
  <description>
    <![CDATA[
      El PIB de México se contrajo 0.1% en 2019 respecto al año previo, su peor
      desempeño en 10 años, según estimación oportuna del Inegi.
    ]]>
  </description>
  <author>Juan Carlos Orozco</author>
  <textofacebook>
    <![CDATA[ Cayó PIB 0.1%; peor desempeño en 10 años ]]>
  </textofacebook>
  <textotwitter>
    <![CDATA[ Cayó PIB 0.1%; peor desempeño en 10 años ]]>
  </textotwitter>
  <IdCategoria>162</IdCategoria>
  <IdSubCategoria>1666</IdSubCategoria>
</item>
```

# Bibliografía


- Abeel Thomas, Van de Peer Yves, Saeys Yvan (2009), Java-ML: A Machine Learning Library. Recuperado 20 de abril de 2019, http://java-ml.sourceforge.net/api/0.1.7/.
- Aggarwal Charu C., Zhai ChengXiang (2012), A survey of text clustering algorithms. En: Aggarwal Charu C., Zhai C., *Mining Text Data*, Springer.
- Beged-Dov, G., Bricley, D., Dornfest, R., Davis, I., Dodds, L., Eisenzopf, J., Galbraith, D., Guha, R.V., MacLeod, M., Miller, E., Swartz, A., van der Vlist, E (2001), RDF Site Summary (RSS) 1.0. Recuperado 1 de febrero de 2020, http://web.resource.org/rss/1.0/spec.
- Billsus Daniel, Pazzani Miachael J. (2007). Adaptive News Access. En: P. Brusilovsky, A. Kobsa, W. Nejdl, *The Adaptive Web. Methods and Strategies of Web Personalization*, Springer.
- Ghorab M. Rami, Zhou Dong, O'Connor Alexander, Wade Vincent (2013). Personalised Information Retrieval: survey and classification. *User Modeling and User-Adapted Interaction*, 23, pag. 381 - 443.
- Han Jiawei, Kamber Micheline, Pei Jian (2000), Data Mining: Concepts and Techniques, Morgan Kaufmann.
- LeGutier (2006), Algoritmo de Porter en español. Recuperado 20 de abril de 2019, http://legutier.blogspot.com/2006/03/algoritmo-de-porter-enespaol.html.
- Lingras Pawan, Akerkar Rajendra (2008), Building an Intelligent Web: Theory and Practice, Jones & Bartlett Learning.
- Manning Christopher D., Raghavan Prabhakar, Schütze Hinrich (2009), Introduction to Information Retrieval, Cambridge University Press.
- Mobasher, B. (2007), Data Mining for Web Personalization. En: P. Brusilovsky, A. Kobsa, W. Nejdl, *The Adaptive Web. Methods and Strategies of Web Personalization*, Springer.
- Panessi Walter, Bordignon Fernando Raúl Alfredo (2001), Procesamiento de Variantes Morfológicas en Búsquedas de Textos en Castellano. Revista Interamericana de Bibliotecología, Volumen 24, 1, pag. 69 - 88.
- Salton G., Wong A., Yang C. S. (1975), A vector space model for automatic indexing. *Communications of the ACM*, Volume 18 Issue 11, pag: 613 - 620.
- Spanish Stop Words (s.f). Recuperado 20 de abril de 2019, https://countwordsfree.com/stopwords/spanish.
- Su Xiaoyuan, Khoshgoftaar Taghi M. (2009), A survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, DOI: 10.1155/2009/421425.
- Tan Pang-Ning, Steinbach Michael, Karpatne Anuj, Kumar Vipin, Introduction to Data Mining, Pearson 2005.
- Witten Ian H., Frank Eibe, Hall Mark A., Pal Christopher J. (2016), Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann.

- Weka Wiki (s.f). Recuperado 20 de abril de 2019, http://weka.sourceforge.net/doc.stable/.
- Wusteman Judith (2004), RSS: The latest feed. *Library Hi Tech*, Volume 2 Number 4, pag 404 - 413.