Trabajo Práctico Final - Minería de Texto
Minería de Datos

Lombardo Micaela Vazquez Agustina

Julio 2025

Indice

1 Introducción

Este trabajo tiene como objetivo aplicar técnicas de minería de texto para analizar y agrupar canciones según el contenido emocional de sus letras. La idea es que las canciones puedan organizarse en distintos grupos dependiendo de si expresan amor, tristeza, enojo, entre otros sentimientos.

Para llevar a cabo este análisis, se trabajará con un dataset armado específicamente con canciones que reflejan de forma clara distintas emociones, lo que permite una mejor interpretación de los resultados. El objetivo principal es evaluar si, a través del procesamiento del lenguaje natural, es posible detectar y diferenciar estos estados emocionales en los textos.

El análisis incluirá distintas etapas de preprocesamiento del texto, como limpieza, tokenización, lematización y la aplicación de técnicas como TF-IDF, entre otras, que se explicarán más adelante. Estas herramientas permitirán representar y comparar las letras de manera adecuada para luego poder agruparlas en función de su contenido emocional.

2 Tipo de Problema

El problema a abordar es de tipo descriptivo, ya que el objetivo principal es explorar y descubrir patrones ocultos en las letras de canciones, sin que exista una variable objetivo previamente definida. A través del uso de técnicas de minería de texto y agrupamiento no supervisado, se busca identificar similitudes semánticas entre las letras que permitan clasificar las canciones según el tipo de emoción que transmiten (como amor, tristeza, enojo, entre otras).

3 Metodologías KDD

3.1 Selección de datos

El conjunto de datos utilizado en este trabajo fue construido manualmente en dos etapas. En la primera, se empleó la librería Spotipy, que permite acceder a la API oficial de Spotify, para obtener información básica sobre las canciones y los artistas seleccionados. En una segunda etapa, se incorporaron de forma manual las letras correspondientes a cada canción, completando así la información necesaria para el análisis.

El dataset resultante está compuesto por las siguientes columnas:

- Title: contiene el título de la canción.
- Artist: indica el nombre del artista o grupo musical que interpreta la canción.
- Spotify_id: especifica un identificador único proporcionado por Spotify para cada pista.
- Lyrics: incluye la letra completa de la canción, que será el principal insumo para el análisis de texto.

Este conjunto de datos fue armado cuidadosamente para asegurar que las canciones representen distintos estados emocionales, lo cual es clave para el objetivo del trabajo. Las

canciones incluidas buscan expresar emociones relacionadas al amor, felicidad, desamor y tristeza.

3.2 Preprocesamiento

Para comenzar, se llevó a cabo una limpieza básica de las letras de las canciones. Primero, se pasaron las letras a minúsculas, para poder garantizar uniformidad en las mismas; luego se eliminó cualquier carácter que no sea una letra y los espacios innecesarios en las mismas.

A partir de ahí se aplicó una tokenización a cada una de las canciones. La tokenización consiste en dividir una secuencia de texto, en este caso la letra de las canciones, en palabras o frases individuales, conocidas como tokens. De esta manera, ayuda a los modelos a poder interpretar el texto en una estructura manejable para el modelado, ya que la mayoría de los algoritmos de procesamiento de lenguaje natural (NLP) aún trabajan con tokens como entrada básica; y facilita el descubrimiento de patrones en el mismo.

Una vez tokenizadas las canciones, se procede a eliminar los stopwords o palabras vacías del texto, ya que las mismas no hacen un aporte relevante al análisis. Posteriormente, ya con la lista de palabras por canción definidas, se llevó a cabo un proceso de pasar las palabras a su forma base, eliminando variaciones gramaticales (por ejemplo, "corriendo" y "corrió" se transformaron en "correr"). Este proceso, conocido como lematización, fue elegido en lugar del stemming debido a su mayor precisión. Mientras que la lematización identifica la forma base de una palabra considerando su contexto, el stemming simplemente recorta la palabra hasta su raíz, lo que puede resultar en una forma más básica y menos precisa (por ejemplo, "corriendo" y "corrió" se convertirían en "corr").

3.3 Transformación

Una vez pre procesados los datos, se procedió a preparar los mismos para luego ser utilizados por el modelo. En consecuencia, se transformaron los textos en vectores numéricos utilizando la técnica TF-IDF. Esta técnica busca evaluar la importancia de las palabras en las canciones en relación con el resto del corpus. Combina dos componentes: la frecuencia del término (TF), que mide qué tan seguido aparece una palabra en una canción, y la frecuencia inversa de documentos (IDF), que reduce la importancia de palabras comunes y otorga mayor peso a aquellas que son más raras o distintivas.

Para ello, se utilizó la función TfidfVectorizer(), aplicada a través del método fit_transform() sobre el conjunto de canciones. De esta forma, se generó una matriz donde cada fila representa una canción y cada columna una palabra del vocabulario total. A cada palabra se le asignó un valor numérico que refleja su relevancia dentro de la canción y su especificidad con respecto al resto del corpus. Esta matriz resultante fue almacenada en una estructura que permite representar cada canción como un vector de valores ponderados, listos para ser procesados por el modelo.

3.4 Efecto del procesamiento en las distribuciones

En este caso, el objetivo del preprocesamiento fue limpiar los textos y transformarlos en una representación numérica que pueda ser interpretada por los modelos de machine learning. Las tareas realizadas incluyeron la normalización de los textos en minúsculas, eliminación de caracteres no relevantes, y la vectorización mediante TF-IDF.

Dado que los datos originales consisten en texto sin una distribución numérica predefinida, no es posible establecer una comparación directa con una distribución original. Es decir, no existe una distribución previa sobre la cual evaluar el impacto del procesamiento. Por lo tanto, no se observan cambios significativos en una distribución existente, ya que el propósito del preprocesamiento no es modificar una estructura estadística, sino preparar los datos para que puedan ser utilizados por modelos que requieren entradas numéricas.

3.5 Minería de Datos

En esta etapa del análisis se aplicó el algoritmo K-Means, una técnica de clustering no supervisado que permite identificar agrupamientos naturales dentro de un conjunto de datos sin necesidad de etiquetas predefinidas.

En este caso, el objetivo fue clasificar canciones en grupos según la similitud semántica de sus letras, representadas numéricamente mediante la técnica TF-IDF (Term Frequency - Inverse Document Frequency). Como ya se mencionó, esta representación convierte cada letra en un vector que refleja la importancia relativa de cada palabra dentro de la canción y en comparación con el resto del corpus.

K-Means funciona dividiendo los datos en k grupos (clusters), donde k debe definirse previamente. El algoritmo sigue los siguientes pasos:

- Se eligen aleatoriamente k puntos del espacio de datos como centroides iniciales, uno por cada cluster.
- Cada punto del conjunto de datos (en este caso, cada canción representada por su vector TF-IDF) se asigna al cluster cuyo centroide esté más cercano, utilizando normalmente la distancia euclídea.
- Luego, se recalculan los centroides como el promedio de todos los puntos asignados a cada grupo.
- Estos pasos se repiten iterativamente (re-asignación y re-cálculo de centroides) hasta que los centroides dejan de moverse significativamente o se alcanza un número máximo de iteraciones.

El resultado es una partición del conjunto de datos en k grupos con alta cohesión interna (las canciones dentro de un mismo grupo son similares entre sí) y baja similitud entre grupos (los distintos clusters son lo más diferentes posible).

Se seleccionó un número de clusters igual a 4, motivado por la hipótesis planteada en la introducción del trabajo: que las canciones podrían reflejar de manera predominante emociones como amor, tristeza, enojo y felicidad, y que estos estados emocionales podrían emerger naturalmente como patrones latentes al analizar las letras con técnicas de minería de texto.

El valor de $k = 4$ fue elegido de forma guiada por criterio teórico, en lugar de aplicar un método automático como el método del codo o el coeficiente de silueta, dado que el interés estaba en contrastar directamente con las emociones consideradas desde el inicio del trabajo.

3.6 Evaluación

Para evaluar la calidad del agrupamiento realizado con el algoritmo K-Means, se utilizaron dos métricas principales: la distancia entre centroides y el coeficiente de silueta.

- Distancia entre Clusters (Centroides): se calculó la matriz de distancias euclídeas entre los centroides de los 4 clusters obtenidos. Esta matriz permite observar cuán separados están los grupos entre sí en el espacio vectorial TF-IDF. Cuanto mayor sea la distancia entre centroides (color más oscuro), más diferenciados están los grupos. El valor en la diagonal principal es 0, ya que representa la distancia del cluster consigo mismo.

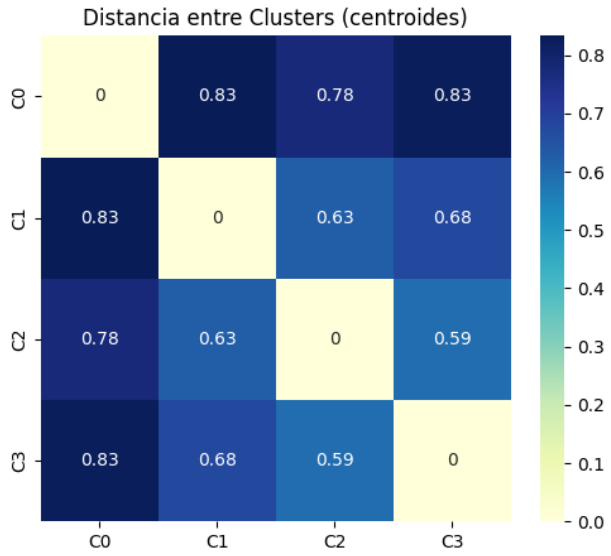

Figure 1: Distancia entre Clusters

El mapa de calor generado evidencia que los clusters más cercanos entre sí son el Cluster 2 y el Cluster 3 (distancia 0.59), lo que sugiere que sus canciones tienen características similares en términos de contenido léxico.

En cambio, el Cluster 0 es el que muestra mayor distancia con respecto a los demás (0.83), lo que sugiere que representa un grupo bien diferenciado del resto. Este análisis apoya la idea de que al menos uno de los clusters podría estar capturando una emoción claramente distinta, como por ejemplo la tristeza o el enojo, dependiendo de las palabras predominantes.

- Silhouette_score: este coeficiente mide qué tan similar es una observación a su propio cluster en comparación con otros clusters. Su valor varía entre -1 y 1:
  - Valores cercanos a 1 indican que los puntos están bien agrupados.
  - Valores cercanos a 0 sugieren que los puntos están cerca de la frontera entre clusters.

– Valores negativos indican una posible mala asignación.

Figure 2: Silhouette_score de clusters

En el gráfico se puede ver:

– La línea vertical roja punteada indica el promedio general del coeficiente, en este caso, 0.03 (explicado anteriormente). Este valor promedio es bajo, lo cual sugiere que los grupos formados por el modelo no están claramente definidos; es decir hay cierta separación entre clusters, muchos puntos están cerca de los límites o incluso podrían pertenecer mejor a otros grupos.

– Las formas de los bloques (rellenos de colores) indican la distribución de los coeficientes de silueta dentro de cada cluster:

  * El Cluster 0 es el más poblado y contiene canciones con coeficientes positivos (más altos), aunque no elevados.
  * Los Clusters 1, 2 y 3 presentan formas más estrechas y centradas en torno a cero, lo que indica que las canciones asignadas a estos clusters están muy cerca de los bordes con otros clusters.

4 Modelo Obtenido

Los resultados obtenidos por el modelo fueron los siguientes:

Cluster	Canciones
0	▪ *We Are Never Ever Getting Back Together* (Taylor Swift) ▪ *Count on Me* ▪ *When I Was Your Man*
1	▪ *Since U Been Gone* ▪ *Happy* (From *Despicable Me 2*) ▪ *Walking On Sunshine* ▪ *Best Day Of My Life* ▪ *Good Life*
2	▪ *Perfect* ▪ *Just the Way You Are* ▪ *Can't Help Falling in Love* ▪ *A Thousand Years* ▪ *All of Me* ▪ *Shout Out to My Ex* ▪ *Lost Boy*
3	▪ *Someone Like You* ▪ *Irreplaceable* ▪ *Let Her Go* ▪ *Fix You* ▪ *The Scientist*

Figure 3: Resultados del modelo

5 Interpretación de resultados

La elección de cuatro clusters se basó en una hipótesis teórica, que plantea que las letras de las canciones podrían reflejar predominantemente cuatro emociones principales: amor, tristeza, enojo y felicidad. Si bien este enfoque no partió de una validación automática del número óptimo de clusters, permitió explorar cómo el modelo distribuía las canciones a partir de esta suposición inicial.

Los resultados obtenidos tras aplicar el modelo de clustering muestran que esta segmentación teórica no se cumplió de forma estricta, aunque sí se logró una agrupación emocional razonable en varios casos.

- Cluster 0: este agrupamiento es el que muestra una mayor confusión temática. Al contener canciones de amistad, desamor y tristeza, parece carecer de un patrón emocional claro. Este resultado sugiere que el modelo, basado en TF-IDF, tuvo dificultades para capturar las sutilezas emocionales de canciones menos extremas o con mensajes mixtos. Es probable que la representación vectorial no haya sido suficiente para distinguir los matices del contexto emocional en estas letras.

- Cluster 1: agrupó correctamente canciones con un tono claramente positivo y alegre. Aunque incluyó una canción de desamor ("Since U Been Gone"), su carácter enérgico puede haber influido en que el modelo la interpretara como positiva. Esto sugiere que el modelo puede estar influido tanto por el contenido semántico como por el tono emocional general de la canción.

- Cluster 2: también fue mayormente correcto, capturando la esencia del amor y las baladas románticas. Las pocas excepciones —como "Shout Out to My Ex" o "Lost Boy"— pueden explicarse por matices en sus letras: la primera mezcla desamor con empoderamiento y la segunda tiene un tono nostálgico, lo cual podría haber confundido al modelo.

- Cluster 3: representa adecuadamente canciones tristes o de desamor profundo. La coherencia emocional de estas canciones facilitó su agrupamiento bajo un tema claro de melancolía y pérdida.

6 Justificación de la Técnica Utilizada

La elección del algoritmo K-Means como técnica de agrupamiento para este análisis se fundamenta tanto en criterios teóricos como prácticos. Dado que el objetivo del trabajo era identificar patrones latentes en las letras de canciones sin una variable objetivo predefinida, resultaba adecuado aplicar una técnica de clustering no supervisado. K-Means fue particularmente pertinente por las siguientes razones:

- Compatibilidad con representaciones vectoriales (mencionada previamente): al trabajar con letras de canciones convertidas en vectores numéricos mediante TF-IDF, se requería un algoritmo capaz de operar eficientemente sobre espacios vectoriales de alta dimensión. K-Means se adapta naturalmente a este tipo de datos, permitiendo medir la similitud entre canciones a través de distancias euclídeas en el espacio TF-IDF.

- Simplicidad interpretativa: una de las fortalezas del algoritmo es su facilidad de interpretación. Cada grupo o cluster está representado por un centroide, que puede ser analizado para detectar qué términos o temáticas predominan en él. Esto facilitó vincular cada agrupamiento con una emoción dominante.

- Eficiencia computacional: K-Means es un algoritmo rápido y escalable, ideal para conjuntos de datos como el trabajado en este proyecto, que aunque moderado en tamaño, exigía tiempos razonables de procesamiento en entornos educativos o personales.

- Correspondencia con la hipótesis del trabajo: se partió de una hipótesis teórica que proponía la existencia de cuatro emociones predominantes (amor, tristeza, enojo y felicidad). K-Means permite fijar a priori el número de clusters, lo cual fue útil para contrastar directamente la hipótesis con los resultados obtenidos.

7 Referencias Bibliográficas

1. Abid Ali Awan. *¿Qué es la tokenización?* . Disponible en: `https://www.datacamp.com/es/blog/what-is-tokenization`

2. Dan Yang ; Won-Sook Lee. *Music Emotion Identification from Lyrics.* `https://ieeexplore.ieee.org/abstract/document/5363083`