Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ciencias de la Computación

Proyecto 2
Resultados Iniciales

Daniela Villamar 19086
Diego Crespo 19541
Rene Ventura 19554
Andres Paiz 191142

Algoritmos a utilizar

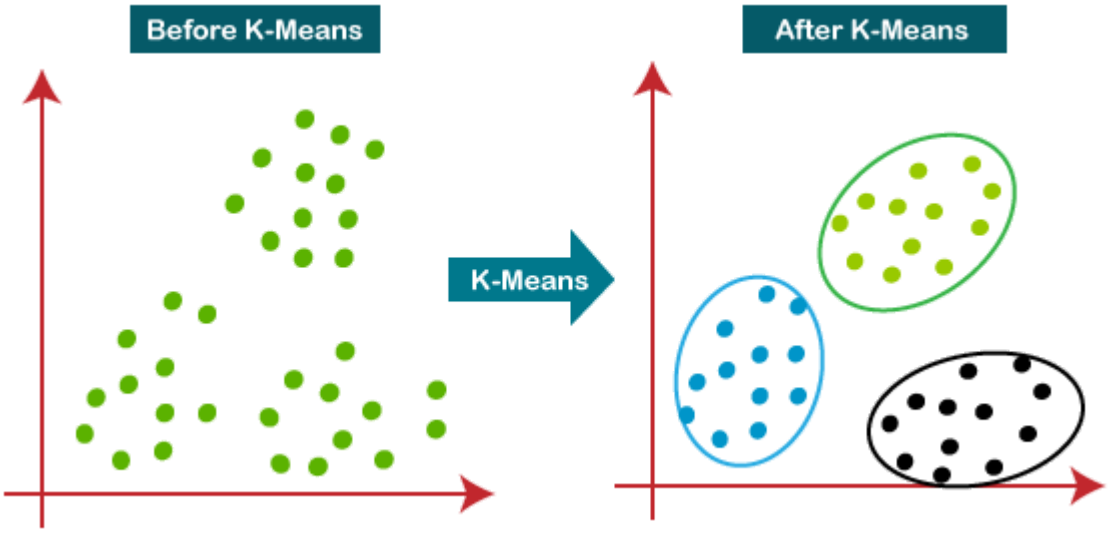
K-Means Clustering

El agrupamiento de K-means es un tipo de aprendizaje no supervisado, que se usa cuando tiene datos sin etiquetar (es decir, datos sin categorías o grupos definidos). El objetivo de este algoritmo es encontrar grupos en los datos, con la cantidad de grupos representados por la variable K. El algoritmo funciona de forma iterativa para asignar cada punto de datos a uno de los grupos K en función de las características proporcionadas. Los puntos de datos se agrupan en función de la similitud de características. Los resultados del algoritmo de agrupamiento de K-medias son:

Los centroides de los grupos K, que se pueden usar para etiquetar nuevos datos
Etiquetas para los datos de entrenamiento (cada punto de datos se asigna a un solo grupo)

En lugar de definir grupos antes de ver los datos, la agrupación le permite encontrar y analizar los grupos que se han formado orgánicamente. La sección "Elegir K" a continuación describe cómo se puede determinar el número de grupos.

Cada centroide de un clúster es una colección de valores de características que definen los grupos resultantes. El examen de los pesos de las características del centroide se puede utilizar para interpretar cualitativamente qué tipo de grupo representa cada grupo.



El algoritmo consta de tres pasos:

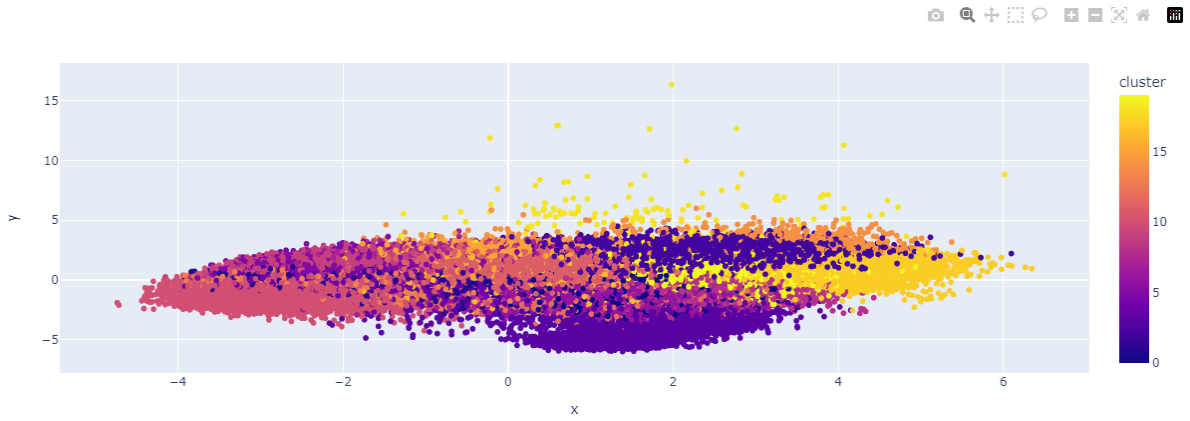
- Inicialización: una vez escogido el número de grupos, k, se establecen k centroides en el espacio de los datos.
- Asignación objetos a los centroides: cada objeto de los datos es asignado a su centroide más cercano.

Actualización centroides: se actualiza la posición del centroide de cada grupo tomando como nuevo centroide la posición del promedio de los objetos pertenecientes a dicho grupo.

El algoritmo de agrupamiento de K-means se usa para encontrar grupos que no se han etiquetado explícitamente en los datos. Esto se puede usar para confirmar suposiciones comerciales sobre qué tipos de grupos existen o para identificar grupos desconocidos en conjuntos de datos complejos. Una vez que se ha ejecutado el algoritmo y se han definido los grupos, cualquier dato nuevo se puede asignar fácilmente al grupo correcto.

Se consideró la utilización de K means ya que es un dataset muy grande y es una buena herramienta para iniciar nuestro proceso de recomendación. Igualmente es muy fácil de utilizar e implementar. Igualmente nos puede ayudar a identificar posibles datos o conjuntos de datos que pueden no estar identificados por los labels previamente establecidos en el dataset.

Clustering de Canciones:



Clustering de Géneros:

Min Max Normalization

Min-Max Normalization (conocida también por escalado de características) es un método que realiza una transformación lineal en los datos originales. Esta técnica obtiene todos los datos escalados en el rango (0,1). La fórmula que se utiliza para conseguir lo mencionado anteriormente es lo siguiente:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

La normalización min-max conserva las relaciones entre los valores de datos originales. El costo de tener este rango acotado es que terminaremos con desviaciones estándar más pequeñas, lo que puede suprimir el efecto de los valores atípicos.

Se normaliza lla data por medio del algoritmo min max con las columnas a utilizar, para hacer un fit

```
feature_cols=['acousticness', 'danceability', 'duration_ms', 'energy',
              'instrumentalness', 'key', 'liveness', 'loudness', 'mode',
              'speechiness', 'tempo',  'valence',]

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
normalized_df =scaler.fit_transform(df[feature_cols])

print(normalized_df[:2])
```

0s ✓ 0.3s

```
[[0.01024843 0.82482599 0.19073524 0.4263629  0.02243852 0.18181818
  0.15386234 0.74114059 1.         0.51444066 0.59603317 0.26243209]
 [0.19999772 0.72041763 0.3144808  0.35008137 0.00626025 0.09090909
  0.12439486 0.69216224 1.         0.07100517 0.6544742  0.57793565]]
```

Una cosa importante a tener en cuenta cuando se usa MinMax Scaling es que está muy influenciado por los valores máximos y mínimos en nuestros datos, por lo que si nuestros datos contienen valores atípicos, estarán sesgados. También la escala de Min Max podría comprimir todos los valores internos en un rango estrecho.

Cosine similarity

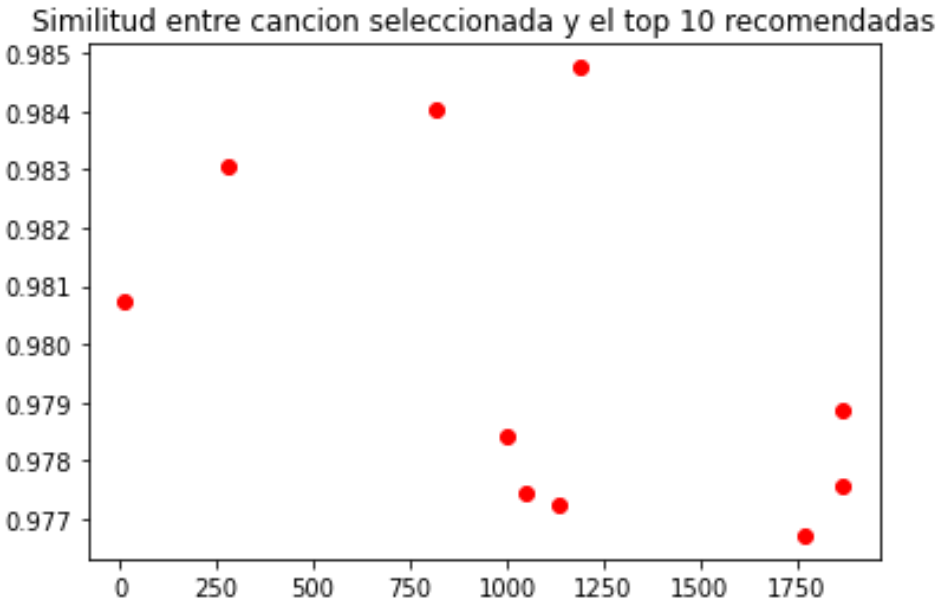
En términos de análisis de datos la similitud del coseno es una medida de similitud entre dos secuencias de números. Para definirlo, las sucesiones se ven como vectores en un espacio de producto interior, y la similitud de coseno se define como el coseno del ángulo entre ellos, es decir, el producto escalar de los vectores dividido por el producto de sus longitudes.

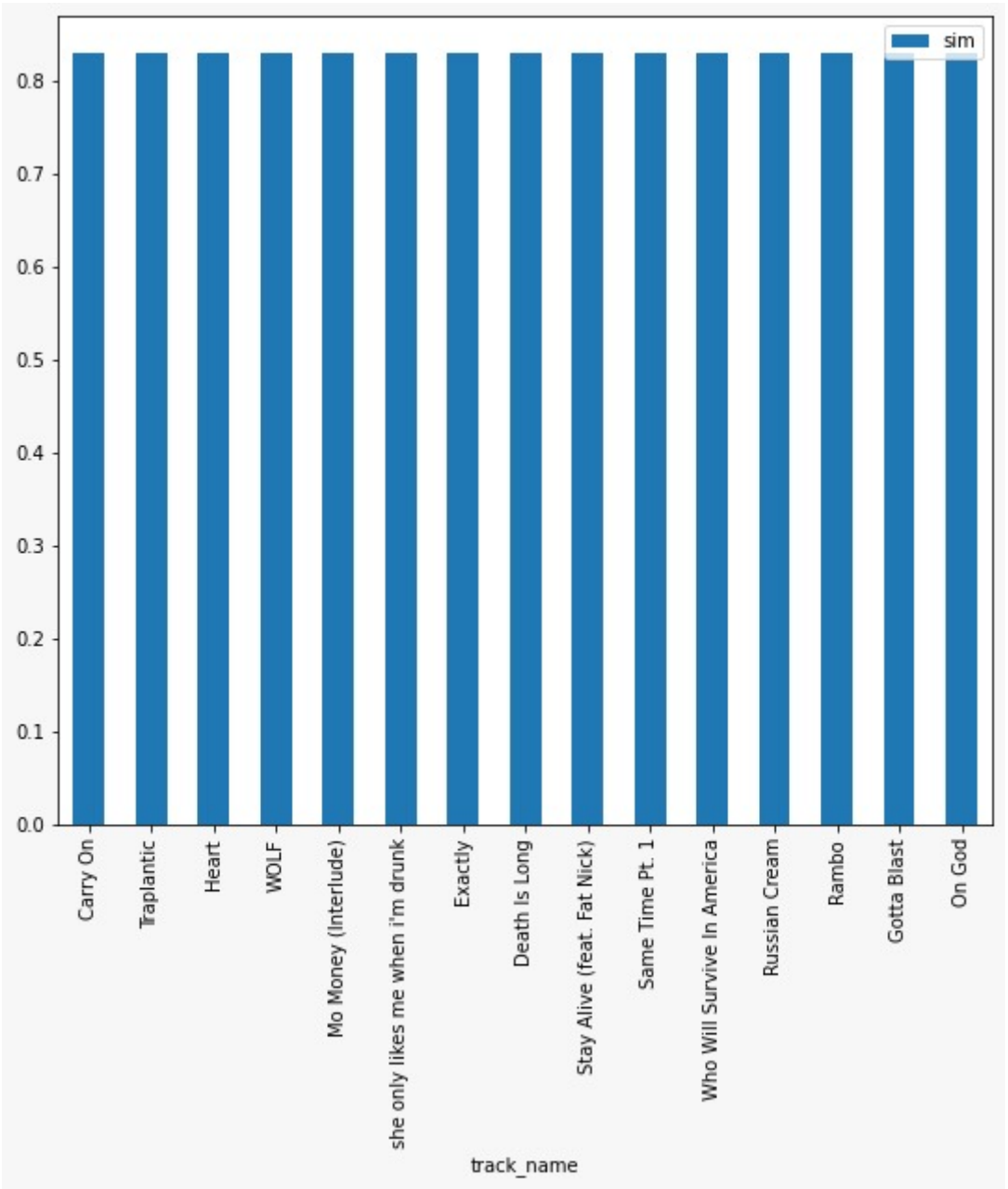
De ello se deduce que la similitud del coseno no depende de las magnitudes de los vectores, sino sólo de su ángulo. La similitud del coseno siempre pertenece al intervalo $[-1, 1]$. Por ejemplo, en la recuperación de información y la minería de texto, a cada palabra se le asigna una coordenada diferente y un documento se representa mediante el vector del número de ocurrencias de cada palabra en el documento.

La similitud del coseno luego brinda una medida útil de qué tan similares pueden ser dos documentos, en términos de su tema e independientemente de la longitud de los documentos. La técnica también se utiliza para medir la cohesión dentro de los clústeres en el campo de la minería de datos. Una ventaja de la similitud del coseno es su baja complejidad, especialmente para vectores dispersos, sólo se deben considerar las coordenadas distintas de cero.

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

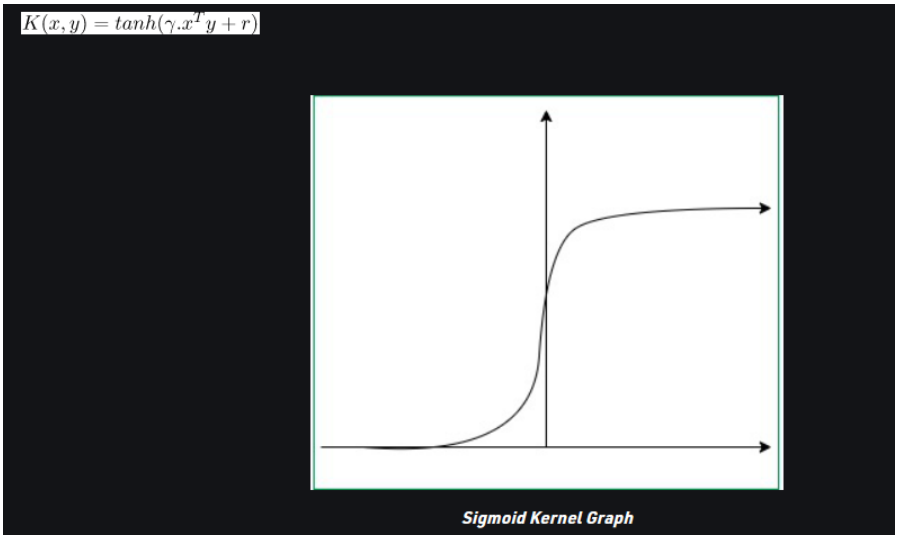
```
Similarity entre cancion escogida y recomendadas:  [(1187, 0.9847539845102656), (818, 0.9840309506844511), (281, 0.9830586579546059), (12, 0.9807487637970956), (1867, 0.9788710374641363), (1000, 0.9784134753225019), (1869,
0.9775609145362028), (1047, 0.9774532204955416), (1136, 0.9772488658365086), (1769, 0.9767138824482666)]
```



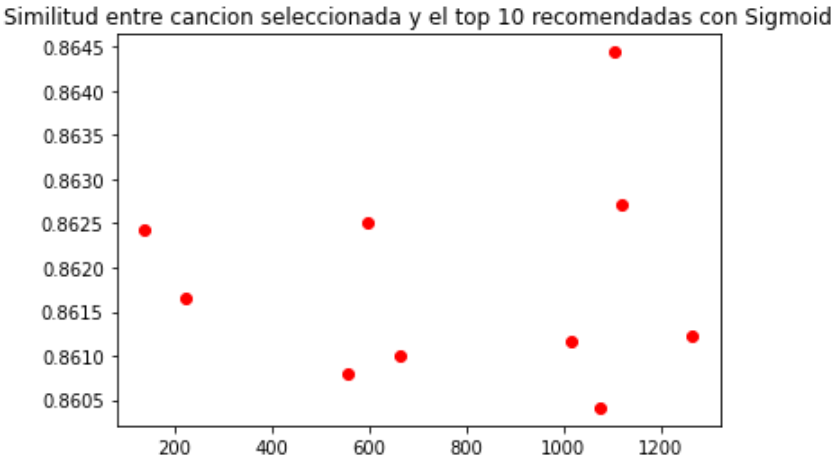

sim
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Carry On
Traplantic
Heart
WOLF
Mo Money (Interlude)
she only likes me when i'm drunk
Exactly
Death Is Long
Stay Alive (feat. Fat Nick)
Same Time Pt. 1
Who Will Survive In America
Russian Cream
Rambo
Gotta Blast
On God
track_name

Sigmoid SVM Kernel Functions

El kernel sigmoide se aplica ampliamente en redes neuronales para tareas de clasificación. El clasificador SVM, que se aplica con el kernel sigmoide, tiene una excelente precisión de clasificación. Sin embargo, como el núcleo sigmoide tiene una estructura complicada, generalmente es difícil para los expertos humanos interpretar y comprender cómo el núcleo sigmoide toma su decisión de clasificación.



Esta función es equivalente a un modelo de perceptrón de dos capas de la red neuronal, que se utiliza como función de activación para neuronas artificiales.




```
sig_kernel = sigmoid_kernel(normalized_df)
a=generate_recommendation('Redbone',sig_kernel).values
```

```
Similarity entre cancion escogida y recomendadas:  [(1106, 0.8644406383322815), (1120, 0.8627221030363934), (598, 0.8625160346873847), (138, 0.8624361909857228), (222, 0.8616521369907595), (1266, 0.8612202470818735), (1015,
0.8611638718189624), (665, 0.8609898626256612), (557, 0.8608045723131829), (1075, 0.8604115445719285)]
```

Discusión

El algoritmo de agrupamiento de K-means esta garantizado para converger a un resultado. El resultado puede ser un óptimo local (es decir, no necesariamente el mejor resultado posible), lo que significa que evaluar más de una ejecución del algoritmo con centroides iniciales aleatorios puede dar un mejor resultado. Para el clustering de canciones y generos nos ayuda a tener un modelo visual del conjunto de datos, mas que nada por la cantidad de datos que se obtuvo en el dat set. Definitivamente es un modelo que debemos de incluir en el proyecto para poder no solo tener conclusiones reales y confiables si no terminar de entender el dataset. La idea principal detrás de la normalización/estandarización es siempre la misma. Las variables que se miden en diferentes escalas no contribuyen por igual a la función de ajuste del modelo y aprendizaje del modelo y pueden terminar creando un sesgo. Por lo tanto, para lidiar con este problema potencial, la normalización de funciones, como Min-Max Scaling, generalmente se usa antes del ajuste del modelo. Esto puede ser muy útil para algunos modelos ML como los perceptrones multicapa (MLP), donde la propagación hacia atrás puede ser más estable e incluso más rápida cuando las características de entrada tienen una escala mínima-máxima (o en general escala) en comparación con el uso del original. La similitud del coseno es una medida que cuantifica la similitud entre dos o más vectores. La similitud del coseno es el coseno del ángulo entre vectores. Los vectores suelen ser distintos de cero y están dentro de un espacio de producto interno. Este tambien se describe matemáticamente como la división entre el producto escalar de los vectores y el producto de las normas euclidianas o la magnitud de cada vector. La similitud del coseno es una técnica de medición de similitud de uso común que se puede encontrar en bibliotecas y herramientas ampliamente utilizadas como Matlab, SciKit-Learn, TensorFlow, etc. Los algoritmos SVM utilizan un conjunto de funciones matemáticas que se definen como el kernel. La función del kernel es tomar datos como entrada y transformarlos en la forma requerida. Diferentes algoritmos SVM usan diferentes tipos de funciones del núcleo. Estas funciones pueden ser de diferentes tipos. Por ejemplo, lineal, no lineal, polinomial, función de base radial (RBF) y sigmoide.
Introduzca las funciones de Kernel para datos de secuencia, gráficos, texto, imágenes y vectores. El tipo de función kernel más utilizado es RBF. Porque tiene una respuesta localizada y finita a lo largo de todo el eje x. Las funciones del núcleo devuelven el producto interno entre dos puntos en un espacio de características adecuado. Definiendo así una noción de similitud, con poco coste computacional incluso en espacios de muy alta dimensión. En base a estos 4 modelos vamos a trabajar para poder obtener visualizaciones estáticas reales que puedan ser faciles de comprender para cualquier usuario. Lo unico dificil de este reto sera el manejo del dataset ya que si son demasiados los datos que hay que manejar. Por esta misma razón, consideramos que estos 4 modelos podran brindarnos estadisticas confiables,

Conclusiones

La generación de clusters no tiene fundamento, por ende, debemos utilizar un método estadístico para obtener la cantidad de clusters más óptima. Analizando nuestros resultados concluimos que el método del codo o el método de la silueta serían óptimos para obtener esta cantidad. Por otra parte, se utilizaron datasets no del mismo tamaño , se debe obtener una muestra del dataset de spotify que cese con el dataset Kagle en términos de tamaño.

Los algoritmos utilizados brindaron un sistema de recomendación bastante acertado, el algoritmo de cosine similarity fue el que más precisión tuvo ya que el valor fue muy cercano a 1 , indicando alta precisión. Todos los algoritmos van a ser utilizados en el progreso del proyecto, Cosine similarity fue el mas preciso en esta iteración , sin embargo, cuando se haga la modificación a los clústeres puede que alguno de los otros algoritmos utilizados sea más preciso.

Referencias

IBM Cloud Education. (2020). Natural Language Processing (NLP). IBM Cloud Learn Hub. Recuperado de: https://www.ibm.com/cloud/learn/natural-language-processin

(2022). Retrieved 25 September 2022, from https://www.hindawi.com/journals/cin/2022/7157075/

Music Recommendation System using Spotify Dataset. (2022). Retrieved 26 September 2022, from https://www.kaggle.com/code/vatsalmavani/music-recommendation-system-using-spotify-dataset/notebook

Music Recommender System Based on Genre using Convolutional Recurrent Neural Networks . (2022). Retrieved 26 September 2022, from https://www.sciencedirect.com/science/article/pii/S1877050919310646