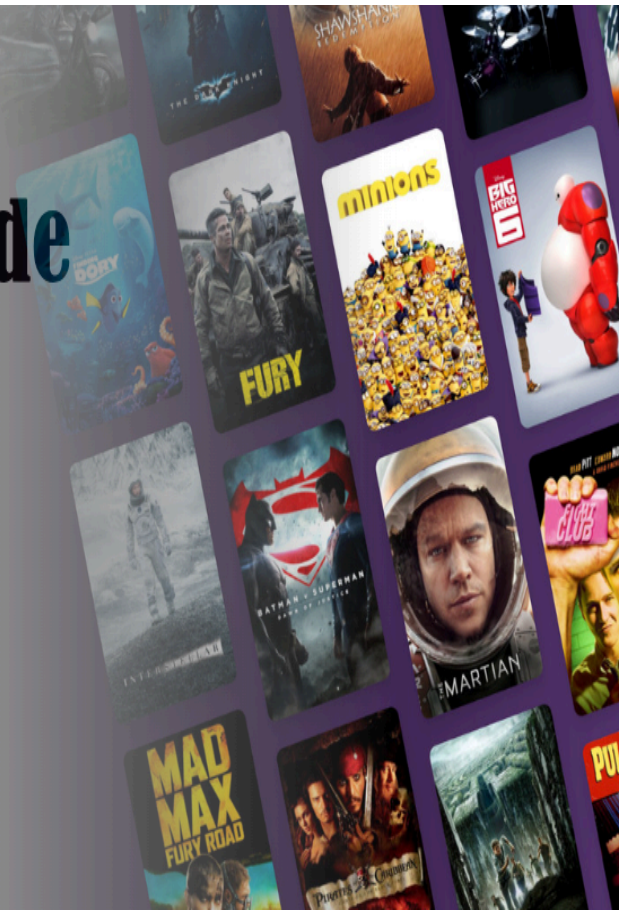
Recomendador de Películas



Grupo:
Alejandra Barbosa
Paulina Luissi
Juan Camilo Quintana
Paul Guzman
Johanna Sepulveda

Resumen:

Todos los servicios por suscripción masivos compiten por mantener y atraer a usuarios que cada vez tienen más opciones para escoger. Aún cuando creemos que este negocio ya está completamente inventado, ¿cuántas veces nos sorprendemos por las recomendaciones que recibimos de estos servicios?
El propósito de este trabajo es explorar las preferencias de los usuarios de una plataforma de películas y tratar de encontrar oportunidades que conduzcan a recomendaciones de contenido curadas y acertadas para ellos.
El trabajo presenta varios desafíos: no contamos con información sobre las características de los usuarios y debemos inferir relaciones entre ellos a través de los ratings que los mismos han dado a las películas. Existe una gran dimensionalidad en los datos lo que dificulta su manejo y no somos expertos en este negocio por lo cual se nos dificulta la evaluación de los resultados de los modelos.
Las bases de datos utilizadas corresponden a MovieLens, un grupo de investigación del Departamento de Ingeniería y Ciencias Computacionales de la Universidad de Minnesota.
Luego de tomar una muestra de los datos y limpiarlos se probaron distintos modelos para la creación de un recomendador de películas con las siguientes características:

a) Se utilizan los promedios de ratings ponderados por género para recomendar películas a usuarios nuevos.
b) De lo contrario se combina el método de filtrado colaborativo utilizando descomposición en valores singulares y el método de filtrado en base a contenidos utilizando Natural Language processing (NLP) para identificar películas similares.

Introducción:

Los sistemas de recomendación (Recommender Systems) son aplicaciones que ofrecen a los usuarios recomendaciones personalizadas de productos y servicios que aún no han adquirido, basadas en sus intereses, con el objeto de incrementar las ventas y mantener la base de clientes satisfecha. Estos sistemas son ampliamente usados en diversas áreas como en plataformas de streaming, redes sociales, planeación de viajes, recomendaciones de música, colocación laboral entre otras [1]

Estos sistemas fueron desarrollados para hacer sugerencias con base en las preferencias de los usuarios: su género favorito, actores, directores, entre otros parámetros, por ejemplo el sistema de Netflix también agrega una explicación que ayuda al usuario entender por qué esa película esta siendo recomendada, ayudando a incrementar la credibilidad de la aplicación y la lealtad del usuario. Otro enfoque es la propuesta de películas basada en emociones por ejemplo felicidad, enojo o tristeza.[2]

El problema a resolver es de gran pertinencia. Vivimos en la era del consumo digital y estamos en el medio del auge del uso de datos de los usuarios para anticiparse a sus necesidades, lo que se conoce comúnmente como análisis predictivo o personalización predictiva.

El cliente potencial de esta solución puede ser cualquier servicio de subscripción de contenidos en su búsqueda por atraer y retener a sus clientes ofreciendo buenas recomendaciones que mantengan a los usuarios involucrados con el servicio. Al mismo tiempo la solución podría aplicarse a otro contexto donde los usuarios califiquen un producto o servicio y donde se encuentre distintas categorías de los mismos.

Este tipo de servicios de recomendación puede resultar clave a la hora de adquirir ventaja competitiva frente a otras compañías y alcanzar los ingresos esperados como empresa.
En este caso puntual la pregunta es: ¿Qué películas me recomienda un sistema basado en técnicas de aprendizaje no supervisado basado en información previa como el puntaje entre 1 y 5 de los usuarios?

La pregunta es sumamente interesante puesto que al mismo tiempo es relevante evaluar de qué forma la podemos responder siendo eficientes en el uso de los recursos y siendo efectivos en cuanto a los resultados generados. También es importante considerar distintas alternativas de recomendación según si el usuario es nuevo en la plataforma o no, y poder siempre recomendarle una lista de películas.

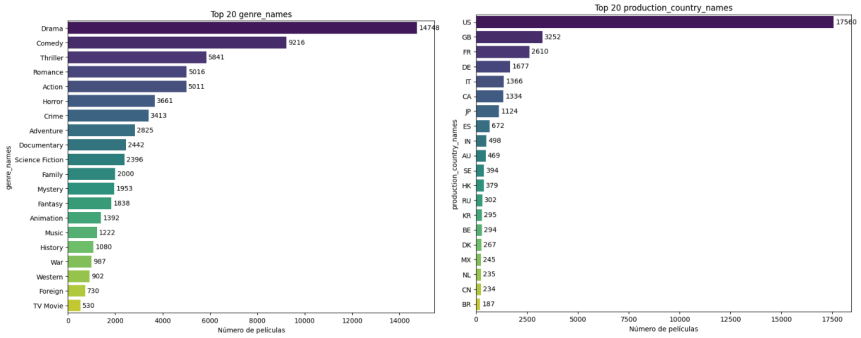
Para abordar esta pregunta decidimos probar algunos de los modelos estudiados en el curso ya que los mismos representan los enfoques tradicionales en el ámbito del aprendizaje no supervisado.
Los mismos serán explicados con mayor detalle en el siguiente punto así como los resultados principales y las limitaciones encontradas.

En la literatura encontrada “A hybrid recommender system for recommending relevant movies using an expert system” [2] se menciona el uso de SVD para hacer una unión de tres algoritmos de recomendación: Sistemas de recomendación basado en filtrado colaborativo, además del uso de sistemas de recomendación basado en contenido y un sistema experto. El uso de estos tres mejora significativamente el algoritmo de recomendación.

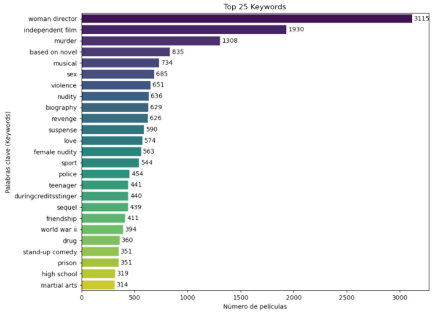
Materiales y Métodos:

Para este proyecto se usaron 3 bases:

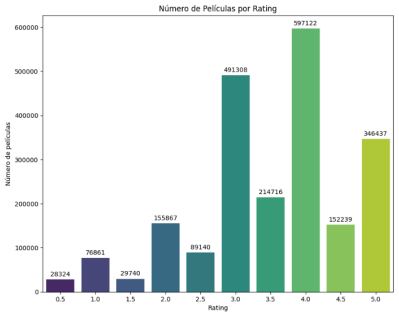
- **Metadata**: Esta base contiene la información de las películas como por ejemplo el Id, nombre de la película y el género al que pertenece. Esta base contiene en total 30.731 películas. Se resalta que de estas aproximadamente el 48% pertenecen al género de drama y el 57% son producidas en USA.

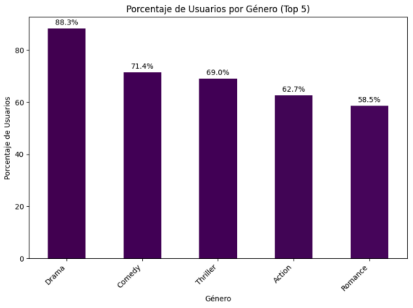


- **Keywords**: Incluye las palabras clave de la trama de las películas del conjunto de datos MovieLens. Estas son las principales 25 palabras clave:



- **Ratings**: Contiene las películas que han visto los usuarios y la calificación (rating) que le dieron a la película, esta base cuenta con 26’024.289 registros, 270.896 usuarios únicos. Teniendo en cuenta el tamaño de esta base y la limitación de recursos computacionales se decidió tomar el 20% de los datos. Una vez se limpia la base, obtenemos 226.890 usuarios únicos. En las gráficas a continuación se observa que más del 50% de los usuarios califican las películas en un rango de 3 a 5, y adicionalmente, el 88% de las películas vistas pertenecen al genero drama seguido por comedia con el 71%





Los modelos que se trabajaron son los siguientes:

1. Para usuarios nuevos se les preguntará qué género es el que más les gusta con el fin de recomendar aquellas películas con el mejor rating en base a su respuesta
2. Para usuarios que ya existen en la base se implementaran recomendadores usando filtrado por contenido basado en NLP, filtrado colaborativo usando SVD y usando clusters con similitud del coseno

Recomendador Promedio ponderado y sin ponderar

Una de las limitaciones de los sistemas de recomendación está relacionada con los usuarios nuevos ya que de estos no se tiene información previa con la cual se pueden identificar los gustos. Por lo tanto, para hacer una recomendación a estos usuarios, se les pregunta su género favorito con el fin de obtener el top 10 películas que pertenecen al mismo usando el promedio simple y ponderado. Se recomiendan aquellas películas que más usuarios ven y que al mismo tiempo han tenido los mejores ratings.

Recomendador NLP

Para hacer una recomendación con NLP se usó la base keywords y los géneros de cada película para construir un modelo de recomendación robusto y se definió finalmente una función con Count Vectorizer y la similitud de coseno. La función toma como entrada el número de usuario, busca las películas mejor calificadas por él y genera 5 recomendaciones en base a esas 3 películas. Para construir el recomendador fue necesario primero hacer una limpieza de keywords con stemming, quitar stopwords, hacer tokenización, y otra limpieza de texto como eliminar acentos, caracteres especiales, etc. Encontramos que el Count Vectorizer fue el método que mejores resultados dió gracias a que en la base tenemos géneros que se repiten y es normal que recomendemos películas del mismo género si esas fueron las que le gustaron al usuario. Una vez creado el modelo con Similitud de coseno y Count Vectorizer como insumos se definió un recomendador final que usa las siguientes funciones:

SVD

Este método busca relaciones subyacentes en los datos a través de las relaciones existentes entre las variables: películas, usuarios y sus ratings. Fue el que más problemas causó, al ser una base tan grande y a pesar de contratar memoria adicional en google collab resultó imposible realizar la descomposición con todos los valores singulares ya que la matriz a descomponer tiene dimensiones 228675 x 228675 y la librería no permite manejarlo. Por ese motivo fue necesario reducir cada vez más los valores singulares para reconstruir la matriz original disminuyendo la efectividad de los resultados.

Similitud del coseno - Clusters:

Con el fin de realizar recomendaciones con similitud del coseno, primero se agruparon los usuarios por gustos de género usando K-means a partir de la base de ratings. Usando el id de la película se extrajeron los géneros de la base metadata, y posteriormente se calculó el promedio del rating por género de cada usuario para determinar sus gustos. Usando el método silhouette se obtuvo que el número óptimo de clusters es 2, sin embargo, se definió que para los propósitos de este proyecto, esta

cantidad no era óptima pues dividía a los usuarios en aquellos que tenían un alto promedio de puntaje en los géneros y en los que tenían bajo puntaje (Anexo 1). Por esta razón se definió una cantidad de 7 clusters, para tener un set de datos más acotado y así poder aplicar la similitud del coseno ya que tampoco fue posible aplicarlo en la base de datos completa.

Resultados y Discusión:

La implementación de los modelos nos trajo algunas dificultades por la gran dimensionalidad de los datos. Descartamos el uso de PCA pues mantener la interpretabilidad en este caso resultaba esencial, ya que el objetivo es recomendar al usuario un número de películas puntual y no un componente donde no es muy claro cómo el mismo puede estar formado. Tanto en el caso de SVD cómo en el caso de las similitudes por coseno la cantidad de datos resultaba demasiado alta.
Los resultados principales a destacar son:

a) Recomendador Promedio ponderado y sin ponderar:

Este es un recomendador simple que no toma en cuenta las preferencias del usuario en particular pero permite saber rápidamente cuáles películas son las más populares y mejor calificadas.
Para los modelos basados en las preferencias de los usuarios una de las limitaciones que tenemos es que para saber si la recomendación es buena o mala, deberíamos conocer todas las películas recomendadas, lo que resulta imposible. Para efectos de este trabajo decidimos tomar algunos ejemplos de usuarios y una evaluación subjetiva comparando los resultados de los distintos modelos.

Vamos a incluir cómo ejemplo los resultados para el usuario 186200 pero evaluamos otros usuarios para llegar a las siguientes conclusiones:

User ID 186200		
Top 5	**Calificación**	**Género**
Fever Pitch	4,5	['Comedy', 'Romance']
La passion de Jeanne d'Arc	3,5	['Drama', 'History']
Crustacés et coquillages	3	['Comedy']
Hostel	3	['Horror']
Halbe Treppe	3	['Comedy', 'Drama']

b) Recomendador NLP

Al evaluar los resultados este modelo concluimos que es el más adecuado ya que las películas recomendadas estaban dentro de los géneros mejor calificados por los usuarios, como se puede ver a continuación en la tabla y en los word clouds las recomendaciones obtenidas pertenecen en su mayoría al género de comedia y en cuanto a los keywords palabras como London o torture coinciden tanto en el top 5 de películas vistas por el usuario como en las películas recomendadas por este método.

Recomendaciones NLP	Similitud Coseno	Género
FC Venus	0.48	['Comedy', 'Crime']
Unmade Beds	0.48	['Comedy', 'Drama', 'Romance']
Orphans of the Storm	0.36	['Comedy']
Joan of Arc	0.34	['Romance', 'Comedy', 'Drama']
Sommersturm	0.43	['Crime', 'Drama', 'Thriller']

c) SVD

Con este modelo los resultados fueron más variados debido a que calcula la similitud entre los usuarios en otras palabras las recomendaciones obtenidas están más relacionadas con aquellas películas que han visto otros usuarios que se considera tienen "gustos" similares por lo tanto él mismo sugería algunas películas que parecían no tener tanta relación con las favoritas de los usuarios, cómo en este ejemplo películas cuyo género es Thriller, Mystery o Accion o cuyas palabras clave son wildlife o restaurant parecieran no tener mucha similitud con los gustos del usuario.

Recomendaciones SVD	Valor SVD	Género
Madame Bovary	1.16	[Drama, Foreing, Romance]
4 luni, 3 săptămîni și 2 zile	0.69	[Drama]
Deep Blue Sea	0.59	['Fantasy', 'Science Fiction', 'Thriller']
The Next Best Thing	0.56	['Comedy']
Felidae	0.45	['Animation', 'Mystery', 'Thriller']

d) Similitud del coseno - Clusters:

Este modelo no nos resultó efectivo ya que en las recomendaciones aparecían varias películas que no parecían estar relacionadas con los gustos de los usuarios. Una de las limitantes fue que al correr el modelo con distinto número de clusters no encontramos agrupaciones realmente distintivas unas de las otras, la mayoría de los clusters mezclan varios géneros distintos dentro de sí mismo

Adicionalmente observamos que en los recomendadores Similitud del coseno & Clusters y con SVD se mezclaban películas de varios idiomas y bastante antiguas. Consideramos que el trabajo se podría mejorar ampliamente si se restringen los valores de estas variables en los modelos.

El recomendador final elegido toma en cuenta filtrado por contenido (NLP) y filtrado por contenido (SVD) para usuarios existentes en la base y el promedio simple y ponderado para usuarios nuevos que no están en la base pero que saben que genero de películas les gustaría que el algoritmo les diera de recomendación. Este recomendador final es el adecuado pues como lo menciona la literatura el unir diferentes tipos de sistemas de recomendación mejora significativamente el resultado final. Gracias a la implementación de ambos filtros se toma lo mejor de cada uno para darle al usuario. Las limitaciones de este recomendador aparte de las computacionales, usar ambos modelos y dárselo al usuario, esto al final toma recomendaciones 5 SVD y 5 NLP el usuario no sabrá realmente qué películas ver, cual de los dos modelos elegir, o solo la que le llame más la atención, otra limitación es el no tener cómo medir la utilidad de este recomendador para saber realmente su desempeño si un usuario utiliza el recomendador.

Conclusión:

Teniendo en cuenta los resultados observados decidimos descartar el recomendador basado en clustering y similitud del coseno y decidimos mantener los otros tres recomendadores: SVD, NLP y promedio simple & ponderado.De esta manera nos aseguramos que para usuarios existentes nos guiamos tanto por las películas que ha visto cómo por la potencial similitud con otros usuarios.
Consideramos que el trabajo podría mejorar ampliamente si contáramos con expertos que puedan evaluar las recomendaciones desde el punto de vista del negocio y que puedan ayudarnos a identificar otras variables significativas a ser incorporadas cómo podrían ser el año de estreno, idioma original de la película o datos demográficos de los usuarios para poder personalizar aún más las recomendaciones generadas . Así como grupos de estudio con los cuales se puedan generar métricas para poder evaluar la efectividad de nuestras recomendaciones.
Por último queremos destacar cómo aprendizaje en este tipo de proyectos que es fundamental iterar entre la evaluación de los modelos y la limpieza de los datos ya que aprendemos mucho sobre ellos una vez que los comenzamos a utilizar por lo cual contar con tiempo suficiente para realizar ese proceso nos parece fundamental.

Bibliografía:

Se citan los artículos mencionados en el texto, usando de forma correcta y consistente el estilo de referencia que se haya escogido usar. [2 puntos]

Bohorquez, Laura Natalia. (2023). Recuperado de: https://www.eltiempo.com/cultura/cine-y-tv/netflix-la-historia-del-rechazo-que-termino-en-el-exito-de-la-plataforma-763072#:~:text=La%20compa%C3%B1%C3%ADa%20ha%20invertido%20miles,190%20pa%C3%ADses%20y%2021%20idiomas.

Banik, Rounak. (2017). Recuperado de: https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset/data?select=movies_metadata.csv

DataLens. (2018). Recuperado de: https://files.grouplens.org/datasets/movielens/ml-latest-small-README.html

Bagus Murdyantoro (2023). Movie Recommender System: Building Movie Recommendations with Machine Learning. Recuperado de: https://medium.com/@bagusmurdyantoro1997/movie-recommender-system-building-smart-movie-recommendations-with-machine-learning-21bfbedb6f3d

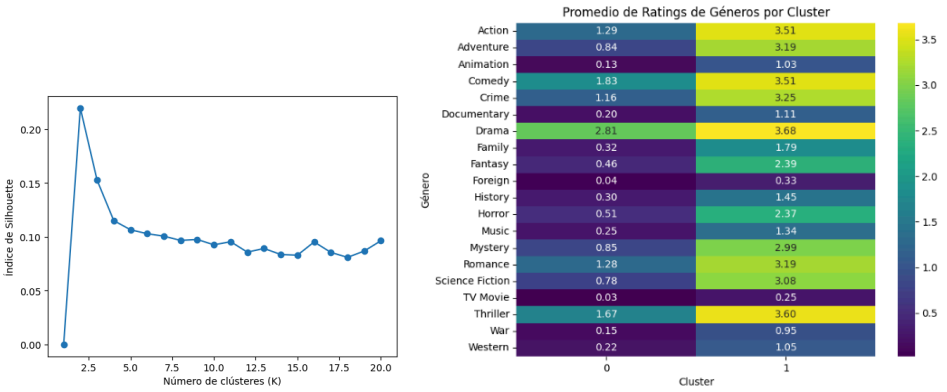
Koblin, J. (2024). The Little Streamer That Could. Recuperado de:https://www.nytimes.com/2024/08/13/business/media/tubi-movies-tv-streaming.html?unlocked_article_code=1.G04.C76S.micrXPlvHnCm&smid=url-share

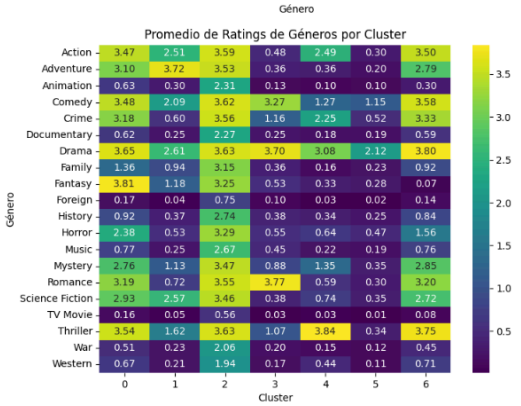
Citas:

1. *Fatih Gedikli. The Importance of Recommender Systems: A Key Technology of the World Wide Web. Recuperado de: https://frontnow.com/post/recommender-systems-key-technology-world-wide-web*
2. *Bogdan Walek, Vladimir Fojtik (2020). A hybrid recommender system for recommending relevant movies using an expert system Recuperado de: https://www.sciencedirect.com/science/article/pii/S0957417420302761*
3. *John Koblin (2024). The Little Streamer That Could Recuperado de: https://www.nytimes.com/2024/08/13/business/media/tubi-movies-tv-streaming.html?unlocked_article_code=1.G04.C76S.micrXPlvHnCm&smid=url-share*
4. *Graph Everywhere. (2024). Sistemas de recomendación ¿Que es el filtrado colaborativo? Recuperado de: https://www.grapheverywhere.com/sistemas-de-recomendacion-que-es-el-filtrado-colaborativo/*

Anexos:

1. Cluster (k=2): Como se mencionó anteriormente divide los usuarios en aquellos que en general califican a las películas con puntajes altos de los que no, por otro lado con k=7 se puede observar que aunque se obtienen una mayor diferenciación siguen existiendo similitudes tales como los gustos por los géneros de drama, comedia y thriller





2. Otros modelos intentados:

NLP (Se pueden encontrar en el GitHub en el notebook llamado NLPModels)

Limpieza de texto paso a paso:

1. **Carga del modelo de lenguaje**: Se carga un modelo en inglés que permitirá realizar tareas de procesamiento de lenguaje natural, aunque no se utiliza directamente en la función.
2. **Inicialización del stemmer**: Se crea un objeto que permite aplicar la técnica de stemming, que consiste en reducir palabras a su raíz, ayudando a simplificar el análisis del texto.
3. **Definición de la función**: Se establece una función que recibe un texto y dos opciones que indican si se debe aplicar lematización y stemming.
4. **Verificación del texto de entrada**: Se comprueba si el texto proporcionado es vacío o no está disponible. Si es así, la función devolverá una cadena vacía para evitar errores en el procesamiento posterior.
5. **Limpieza del texto**: Se realizan varias transformaciones en el texto:
    a. Se eliminan acentos y caracteres especiales para simplificar el texto.
    b. Se eliminan caracteres no alfabéticos y numéricos, dejando solo letras y espacios.
    c. Se quitan los números.
    d. Se reducen múltiples espacios a uno solo y se eliminan espacios al principio y al final del texto.
    e. Se convierte todo el texto a minúsculas para uniformidad.
6. **Tokenización**: El texto limpio se divide en palabras individuales, permitiendo un análisis más fácil y detallado.
7. **Aplicación de stemming**: Dependiendo de la opción seleccionada, se aplica la técnica de stemming a cada palabra. Si no se desea aplicar, se utilizan las palabras originales.
8. **Filtrado de palabras**: Se eliminan las palabras que son demasiado cortas, específicamente aquellas con dos o menos caracteres, para centrar el análisis en palabras más significativas.
9. **Retorno del resultado**: Finalmente, si hay palabras limpias tras el filtrado, se unen en una sola cadena y se devuelven. Si no queda ninguna palabra, se devuelve una cadena vacía.

TFIDF Vectorizer

Se intentó realizar un modelo también con TF-IDF vectorizer sin embargo, como en la base se tenían géneros que estaban repetidos en muchas películas de la base estas perdían importancia lo que no lo hacía tan útil cuando queremos que películas que tienen características similares teniendo en cuenta también el género.

Recomendador Count Vectorizer con correlación

También se intentó utilizar un recomendador con correlación pero su demora era mucha más alta que con similitud de coseno pues debía calcular en cada nueva corrida la correlación lo que no lo hacía eficiente en una base tan grande.

LDA

También se intentó usar LDA el cual se utiliza para descubrir temas latentes en un corpus de documentos.No mide directamente similitud entre documentos o términos, sino que agrupa términos y documentos bajo temas comunes y es útil para clasificación temática. Sin embargo, determinar temas latentes no es tan fácil de hacer pues se deben correr gráficas y otras funciones. Sin embargo,