Sistema de Recomendación de Películas

FABIAN VLADIMIR FLOREZ AGUILAR[1] AND KELVIN PAUL PUCHO ZEVALLOS[2]

[1] fflorez@unsa.edu.pe
[2] kpucho@unsa.edu.pe

Compiled May 24, 2023

1. INTRODUCCIÓN

En el actual panorama digital, la cantidad de información disponible es abrumadora. Con la creciente cantidad de productos, servicios y contenido en línea, resulta cada vez más difícil para los usuarios encontrar lo que realmente necesitan o desean. Aquí es donde los sistemas de recomendación juegan un papel fundamental.

Un sistema de recomendación es una herramienta diseñada para analizar los datos de los usuarios, como sus preferencias, historial de compras, interacciones en línea y otros comportamientos, con el objetivo de ofrecer recomendaciones personalizadas y relevantes. Estos sistemas se han vuelto omnipresentes en plataformas digitales y se utilizan en diversos campos, como el comercio electrónico, el entretenimiento, la música, los libros y más.

El propósito principal de un sistema de recomendación es ayudar a los usuarios a descubrir productos o contenido que se adapten a sus intereses, aumentando así su satisfacción y mejorando su experiencia. Al utilizar algoritmos y técnicas de inteligencia artificial, estos sistemas pueden analizar grandes cantidades de datos y extraer patrones ocultos, permitiendo generar recomendaciones precisas y personalizadas en función de las preferencias individuales.

En este informe, exploraremos los fundamentos teóricos utilizados en la implementación. También evaluaremos los resultados de acuerdo al contenido basado en usuarios y en el contenido. Nosotros proponemos un algoritmo escrito en python que utiliza una estructura de datos como pandas y numpy en python con la ayuda de una librería llamada surprise para los procesos.

2. MOTIVACION

Los sistemas de recomendación de películas son un componente esencial que automatiza tareas como por ejemplo la plataformas de streaming como Netflix, que utiliza la inteligencia artificial y el machine learning para entender las preferencias y gustos de los usuarios y sugerirles películas o series que probablemente disfrutarán. En 2006, lanzaron un concurso de tipo crowdsourcing con el objetivo de que alguien fuera lo suficientemente eficiente para superar su algoritmo de recomendación de películas por al menos 10% con respecto a su medida actual de errores (RMSE 0.9514), donde el ganador se lleva 1 millón de dólares. [1]

3. PROBLEMA

Los sistemas de recomendación de películas se enfrentan a varios retos que pueden afectar a su precisión y eficacia, por ejemplo si un usuario no ha introducido ninguna información o la información es demasiado escasa(datos esparsos) para realizar una predicción precisa, resulta difícil para el sistema determinar si al usuario le ha gustado o no la película que ha visto.

- Datos limitados: La precisión de las recomendaciones de películas depende de la cantidad y calidad de los datos disponibles. Si los datos son limitados, es posible que el sistema no pueda hacer predicciones precisas. [2]
- Incrementación de datos: Con una gran cantidad de datos se requiere que el algoritmo del sistema de recomendación sea escalable para que los resultados sean rapidos. [3].
- Esparcidad: El problema de los datos esparsos perjudica en el rendimiento del algoritmo porque los resultados obtenidos pueden ser confusos.[3].

4. MARCO TEÓRICO

En "Combining content-based and collaborative filtering for job recommendation system" [4] utiliza un modelo hibrido para generar el sistemas de recomendacion, para lograr ello toma las principales caracteristicas del dataset para generar una recomendacion **basda en el filtrado de contenido** y el **filtrado colaborativo**, para lograr ello se utilizo el aprendizaje estadistico relacional (Statistical Relational Learning (SRL))

A. Filtrado Colaborativo

El filtrado colaborativo es el proceso de filtrado de información o modelos, que usa técnicas que implican la colaboración entre múltiples agentes, fuentes de datos, etc [5]. El Filtrado colaborativo se basa, en que si una persona A tiene la misma opinión que una persona B sobre un tema, A es más probable que tenga la misma opinión que B en otro tema diferente que la opinión que tendría una persona elegida azar. La Fig 1 muestra los tipos de filtrado que se usan.

B. Filtrado Colaborativo basado en usuarios

Para hallar si un usuario es similar al otro se utiliza las medidas de la Correlacion de Pearson y la Similitud del Coseno ya

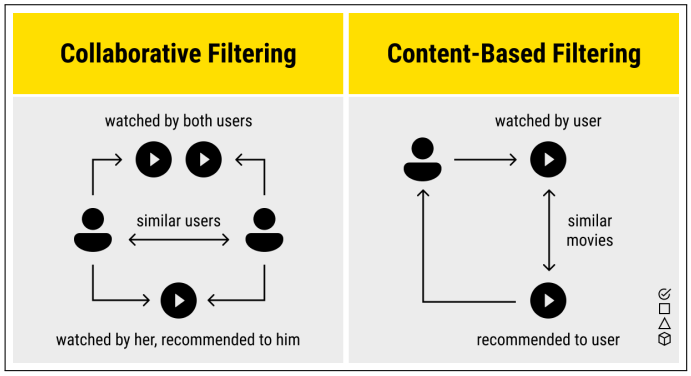

Fig. 1. Tipos de filtrado colaborativo

que las distancias manhattan y euclideana pueden no capturar adecuadamente la similitud real entre usuarios o elementos en un sistema de recomendación, ya que no consideran la relación entre los valores de las interacciones ni abordan la esparcidad y los valores faltantes. En cambio, las medidas utilizadas miden la correlacion o relacion de los usuarios o elementos.

B.1. Correlación de Pearson

El coeficiente de correlación de Pearson es una medida de dependencia lineal entre dos variables aleatorias cuantitativas.

Se puede observar su formula en la ecuacion 1

$$Rxy = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}} \quad (1)$$

donde:
n es el tamaño de la muestra
X_i, Y_i son puntos muestrales individuales indexados con i.
$\overline{X}$ es la media muestral

B.2. Similitud del Coseno

colocar concepto y formula

B.3. K-nearest neighbors

El algoritmo de k vecinos más cercanos, también conocido como KNN o k-NN, es un clasificador de aprendizaje supervisado no paramétrico, que utiliza la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual (Fig 2) . Si bien se puede usar para problemas de regresión o clasificación, generalmente se usa como un algoritmo de clasificación, partiendo de la suposición de que se pueden encontrar puntos similares cerca uno del otro. KNN también forma parte de una familia de modelos de "aprendizaje perezoso" [6], lo que significa que solo almacena un conjunto de datos de entrenamiento en lugar de pasar por una etapa de entrenamiento. Esto también significa que todo el cálculo ocurre cuando se realiza una clasificación o predicción. Dado que depende en gran medida de la memoria para almacenar todos sus datos de entrenamiento, también se lo denomina método de aprendizaje basado en instancias o basado en la memoria[**?**].

C. Filtrado Colaborativo basado en elementos(items)

El filtrado colaborativo basado en ítems es un tipo de sistema de recomendación que utiliza la similitud entre ítems calculada a partir de las valoraciones que los usuarios han dado a los ítems. [**?**]

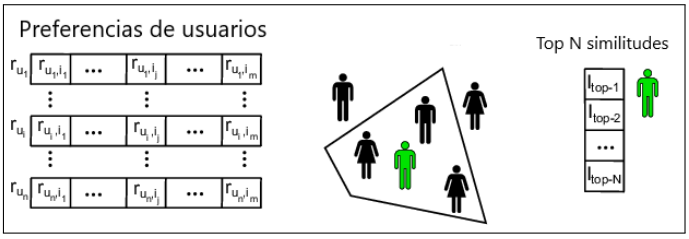

Fig. 2. Clusterizacion de usuarios

- Similitud entre elementos: El filtrado colaborativo basado en ítems busca ítems similares basándose en los ítems que ya han gustado a los usuarios o con los que han interactuado positivamente. [7]
- Emparejamiento entre artículos: en lugar de emparejar al usuario con clientes similares, el filtrado colaborativo entre artículos empareja cada uno de los artículos comprados y valorados por el usuario con artículos similares, y luego combina esos artículos similares para hacer recomendaciones. [8]

C.1. Similitud de Coseno Ajustado

La similitud se calcula a lo largo las columnas (usuarios), es decir, cada par en el conjunto de clasificación compartida corresponde a un usuario diferente. La técnica del coseno básica en un caso basado en artículos tiene un inconveniente importante: las diferencias en la escala de calificación entre diferentes evaluadores no se tienen en cuenta. La similitud de coseno ajustada compensa este inconveniente al restar el promedio de las calificaciones del usuario correspondiente de cada par clasificado conjuntamente [**?**] , la figura 4 muestra la similitud entre la relación de venta/reposición de diferentes frutas en determinados días.

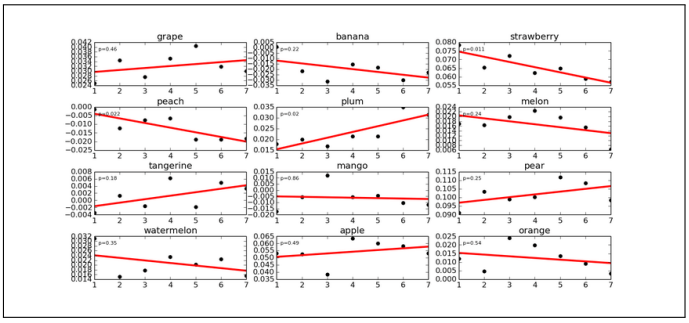

Fig. 3. Similitud de coseno para varios items

5. PROPUESTA DE RECOMENDACIÓN

A. Arquitectura

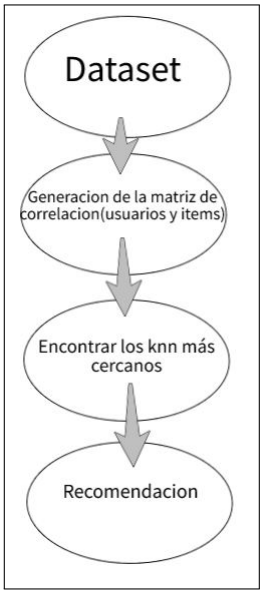

Fig. 4. Arquitectura

B. Descripción

Para tratar la esparcidas de los datos según a las calificaciones de los usuarios, en lugar de tratar de utilizar los datos agregados de comportamiento del usuario nos basamos en recomendar películas basados únicamente en los atributos de los propios películas, . Por ejemplo, se puede recomendar películas del mismo género que podría gustarle a alguien.
La posibilidad que tomamos es recomendar películas basándose en sus atributos, especialmente en el género y el año de estreno. La idea es utilizar el conjunto de datos de MovieLens, que proporciona información sobre los géneros de cada película. Si un usuario disfruta de películas de ciencia ficción, por ejemplo, se podría recomendar otras películas del mismo género. Además, al conocer el año de estreno de las películas, se puede restringir aún más las recomendaciones, sugiriendo películas de ciencia ficción que se estrenaron cerca del mismo año que las que le gustaron al usuario.
Por lo tanto la cantidad de géneros posibles que encontramos son 18 para cada película. Así que para ver la similitud que mire cuantos géneros tienen en común un par de películas usamos la similitud del coseno.

En la Figura 5 muestra un ejemplo para ver la similitud de géneros entre películas. En este gráfico 2D utilizamos valores binarios para los géneros a los que pertenece. Por ejemplo, si una película pertenece a la comedia, se asigna en el eje x, y si no pertenece, se asigna en eje y donde el ángulo entre ellos indica que son géneros totalmente diferentes. Entonces el ángulo entre estos vectores proporciona una medida de similitud entre las películas en términos de sus géneros. Por lo tanto, con esto se busca una métrica que este entre 0 y 1, donde 0 significa "nada similar" y 1 significa "totalmente lo mismo". Y si vemos que una película tiene ambos géneros esto podría colocarse a 45 grados de uno de los géneros como se ve en el gráfico.
Ahora como tenemos 18 géneros en total la solución es crear una tabla de similitud que verifique que géneros son similares por película.

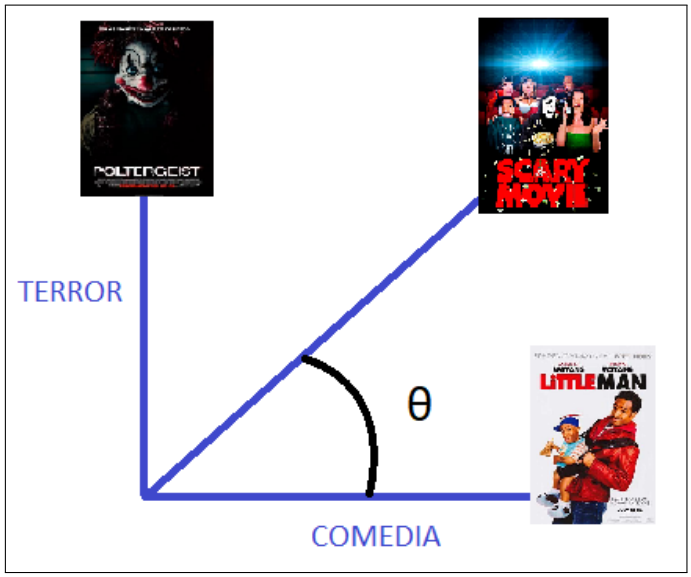

Fig. 5. Similitud del Coseno

Por otro lado podemos restringir aún más las recomendaciones, sugiriendo películas de cualquier genero que se estrenaron cerca del mismo año que las que le gustaron al usuario y para ello usamos una función exponencial de tal forma que filtre los mas cercanos a las películas de los últimos 10 años.

El problema que tuvimos fue con la escasez de datos para lo cual usamos estructuras como "matrices dispersas" que evitan almacenar espacio vacío en esta matriz. Por ello combinado con el filtro basado por usuario el cual empieza por encontrar a otros usuarios similares a un usuario basándonos en el historial de valoraciones, y luego recomendar cosas que les gustó que no se ha visto todavía. Por ejemplo, si un usuario gusta de un película el cual es común con otra pero debido a los datos esparsos no se puede inferir que ambos usuarios son similares en gusto, es por eso la combinación de un filtro por contenido puede determinar la recomendación de películas que el otro usuario pueda haberlas visto siempre y cuando su genero sea los mas similar a la película en común. Otra alternativa que vimos fue dar una umbral hacia las películas que dos usuarios que tienen en común ya que si establecemos una umbral de 10 solo se comparara con los usuarios que tengan mas de 10 películas en común, esto podría ser una solución pero es mas para analizar la similitud pero para dar una recomendación nos basamos en el filtrado por contenido, usuario e ítem.

Nosotros precalculamos una matriz de similitud para calcular rápidamente la similaridad entre usuarios. A continuación, podemos ordenar la lista por puntuaciones de similitud, y elegir la parte superior N vecinos de esa lista. Con ello tenemos las recomendaciones candidatas y saber cuales son las mas cercanas seria normalizar los valores entre 0 a 1 de tal forma que damos pesos a las películas de otros usuarios y mas el filtrado por contenido para determinar exactamente la recomendación ponderamos ambas técnicas.

Posteriormente pre calculamos la similitud de para calcular

la similitud entre items. Este filtro mira las películas que te agradaron y te recomienda películas similares pero a diferencia que nosotros mas el genero y por el año que el usuario gusta de esa película.
Se utilizo el Coseno Ajustado que se explico en la sección C.1 para el precalculo de las matrices de similaridad basándonos en la diferencia entre la valoración de un usuario para una película y su valoración media para todos las películas. Por lo que buscamos la varianza de la media de las valoraciones de cada usuario, y no sólo la valoración bruta en sí misma. Esto nos ayuda a saber a que tipos de películas el usuario se inclina cuando hace sus calificaciones.

Después procedemos a predecir una calificación para un usuario y una película determinada. Y para ello usamos la técnica de "k vecinos más cercanos" que explicamos en la sección B.3. Primero medimos la similitud basada en el usuario y el item entre un usuario y la película. Luego, se selecciona un número determinado de películas más similares, llamadas vecinos más cercanos, y se calcula una predicción de calificación ponderando sus similitudes con la película objetivo según las calificaciones dadas por el usuario. En la figura 6 se muestra un ejemplo de recomendación.

Fig. 6. Recomendaciones para el usuario Juan

6. IMPLEMENTACION

Para poder hallar la correlacion tanto entre usuarios como por items se uso la similitud de coseno y pearson con los que nos devuelve que tanta semejanza existe entre los usuarios o items como se observa a continuacion:

Matriz de correlacion entre usuarios Fig 7

```
[[1.         0.         0.         ... 1.         0.         1.        ]
 [0.         1.         0.95561425 ... 0.776114   0.89144204 0.97993672]
 [0.         0.95561425 1.         ... 0.99786069 0.94592126 0.98448284]
 ...
 [1.         0.776114   0.99786069 ... 1.         1.         0.9952275 ]
 [0.         0.89144204 0.94592126 ... 1.         1.         0.96183401]
 [1.         0.97993672 0.98448284 ... 0.9952275  0.96183401 1.        ]]
```

Fig. 7. Matriz usuarios

Matriz de correlacion entre items Fig 8

```
[[1.         0.94740842 0.9619885  ... 0.         0.         0.        ]
 [0.94740842 1.         0.92847669 ... 1.         1.         0.        ]
 [0.9619885  0.92847669 1.         ... 1.         1.         0.        ]
 ...
 [0.         1.         1.         ... 1.         1.         0.        ]
 [0.         1.         1.         ... 1.         1.         0.        ]
 [0.         0.         0.         ... 0.         0.         1.        ]]
```

Fig. 8. Matriz Items

Implementacion del KNN en funcion de las puntacione de los usuarios, para luego agregar la media de los k vecinos más cercanos. Fig 11

```
#Acumula puntuaciones de similitud entre este elemento y todo lo que el usuario ha valorado
neighbors = []
for rating in self.trainset.ur[u]:
    genreSimilarity = self.similarities[i,rating[0]]
    neighbors.append( (genreSimilarity, rating[1]) )

# Extrae las K calificaciones más similares
k_neighbors = heapq.nlargest(self.k, neighbors, key=lambda t: t[0])

# Calcular la puntuación sim media de K vecinos ponderada por las valoraciones de los usuarios
simTotal = weightedSum = 0
for (simScore, rating) in k_neighbors:
    if (simScore > 0):
        simTotal += simScore
        weightedSum += simScore * rating

if (simTotal == 0):
    raise PredictionImpossible('Sin vecinos')

predictedRating = weightedSum / simTotal

return predictedRating
```

Fig. 9. Knn

7. EXPERIMENTOS Y RESULTADOS

Matrices de Similitud por Usuarios

```
[[1.         0.         0.         ... 1.         0.         1.        ]
 [0.         1.         0.95561425 ... 0.776114   0.89144204 0.97993672]
 [0.         0.95561425 1.         ... 0.99786069 0.94592126 0.98448284]
 ...
 [1.         0.776114   0.99786069 ... 1.         1.         0.9952275 ]
 [0.         0.89144204 0.94592126 ... 1.         1.         0.96183401]
 [1.         0.97993672 0.98448284 ... 0.9952275  0.96183401 1.        ]]
```

Fig. 10. Matriz por Usuario

Matrices de Similitud por Items

```
[[1.         0.94740842 0.9619885  ... 0.         0.         0.        ]
 [0.94740842 1.         0.92847669 ... 1.         1.         0.        ]
 [0.9619885  0.92847669 1.         ... 1.         1.         0.        ]
 ...
 [0.         1.         1.         ... 1.         1.         0.        ]
 [0.         1.         1.         ... 1.         1.         0.        ]
 [0.         0.         0.         ... 0.         0.         1.        ]]
```

Fig. 11. Matriz por Items

Recomendaciones basada el filtrado colaborativo de usuarios 12

```
Lion King, The (1994)
Shrek (2001)
Speed (1994)
Dark Knight, The (2008)
Willy Wonka & the Chocolate Factory (1971)
Mrs. Doubtfire (1993)
Alien (1979)
Aliens (1986)
Rock, The (1996)
Mask, The (1994)
Clerks (1994)
```

Fig. 12. Recomendacion por usuarios

Recomendaciones basada el filtrado colaborativo en items 13

Para ambas recomendaciones se tomo un promedio de entre 30 a 50 segundos de respuesta.

La razón principal es que la correlación de Pearson se basa en la covarianza y la varianza de los datos, y en este caso, con solo una película en común, no hay suficientes puntos de datos para calcular de manera significativa estas medidas estadísticas. Además, la correlación de Pearson podría verse afectada por la diferencia en la longitud de las calificaciones de los usuarios.

Por otro lado, la similitud del coseno es una medida que evalúa la similitud entre dos vectores basándose en el ángulo formado entre ellos. Es especialmente útil cuando la longitud de los vectores varía y solo se tienen unos pocos elementos en común. La similitud del coseno no considera la magnitud absoluta de los vectores, sino que se centra en la dirección o patrones relativos entre ellos.

```
Wizard of Oz, The (1939)
Blood Simple (1984)
Witness (1985)
Thin Blue Line, The (1988)
Alien (1979)
Leaving Las Vegas (1995)
Roger & Me (1989)
Hunt for Red October, The (1990)
Bull Durham (1988)
Aliens (1986)
Dog Day Afternoon (1975)
```

Fig. 13. Recomendacion por items

8. CONCLUSIONES

En este informe verificamos diferentes formas de tratar con los datos esparsos aplicando los filtros colaborativos usando las dataset de 100k, 10m y 25m para la recomendación de películas. También precalculamos las matrices de similaridad el cual tomo tiempo entrenarlos para hallar rápidamente la similitud de los elementos. Finalmente las recomendación de películas se inclinan hacia los gustos del usuario en si y los gustos similares a los usuarios mas cercanos según su calificación y las películas mas cercanas según el tipo de genero y año usando el coseno ajustado. Sin embargo, el código no es escalable debido a la gran cantidad de datos para su procesamiento sobre todo para generar las matrices de similitud una ayuda para esto seria usando factorizacion de matrices. Incluso para los sistemas de recomendación más avanzados que se basan en el aprendizaje automático para asimilar las relaciones pueden ser mas exactos pero eso conlleva a tener que esperar el entrenamiento.

REFERENCES

1. Arkadiusz Krysik, “Everything You Need to Know About the Recommendation System of the Most Popular Streaming Portal,” https://recostream.com/blog/recommendation-system-netflix (2021).
2. Ramya Vidiyala, “How to Build a Movie Recommendation System,” https://towardsdatascience.com/how-to-build-a-movie-recommendation-system-67e321339109 (2020).
3. Yuliia Kniazieva, “What Is a Movie Recommendation System in ML?” https://labelyourdata.com/articles/movie-recommendation-with-machine-learning (2022).
4. S. Yang, M. Korayem, K. AlJadda, T. Grainger, and S. Natarajan, Knowledge-Based Syst. **136**, 37 (2017).
5. L. Terveen and W. Hill, (2001).
6. M.-L. Zhang and Z.-H. Zhou, Pattern recognition **40**, 2038 (2007).
7. Ramya Vidiyala, “Recommender systems,” https://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/algorithms.html.
8. Ramya Vidiyala, “Item-to-Item Based Collaborative Filtering,” https://www.geeksforgeeks.org/item-to-item-based-collaborative-filtering/ (2020).