Análisis de Grandes Volúmenes de Datos (Gpo 10)

David Nava Jiménez - A01168501

Edwin David Hernández Alejandre - A01794692

Jorge Fernando Bonilla Diaz - A01793935

Nombre del entregable:

Avance de proyecto 2: Sistema de Recomendación

Domingo 26 de Mayo de 2024

Contenido.

1. Objetivo

Implementar un algoritmo de recomendación avanzado, evaluarlo con métricas justificadas y documentar tanto la implementación como los resultados en el repositorio GitHub del equipo.

2. Objetivos específicos

- Implementar al menos un algoritmo de recomendación avanzado (por ejemplo, factorización matricial, enfoques basados en aprendizaje profundo).
- Identificar y justificar las métricas de evaluación utilizadas para evaluar el desempeño de los sistemas de recomendación en el contexto del proyecto elegido por el equipo.
- Enumerar al menos tres recomendaciones que muestran los resultados obtenidos de la implementación del algoritmo y documente esta evidencia en el repositorio de GitHub del equipo.

3. Desarrollo

3.1 Descripción del algoritmo de recomendación avanzado elegido.

El algoritmo propuesto con el que vamos a trabajar se basa en el índice “similaridad coseno” que puede ser usado como una medida de similaridad entre dos instancias. Este corresponde al coseno del ángulo entre dos vectores que se extiende desde el origen de cada instancia.

Analizándolo más a detalle, de acuerdo con (Kelleher et al., 2020) la “similaridad coseno” entre dos instancias se computa como el producto punto normalizado de los valores de las instancias. El producto punto se normaliza por el producto de las longitudes del valor de los vectores y se puede ejemplificar de la siguiente manera:

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{m} (\mathbf{a}[i] \times \mathbf{b}[i]) = (\mathbf{a}[1] \times \mathbf{b}[1]) + \cdots + (\mathbf{a}[m] \times \mathbf{b}[m])$$

Obtenido de: *Kelleher, J. D., Namee, B. M., & Arcy, A. D. (2020). Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies (Vol. Second Edition). The MIT Press.* *https://0-eds-p-ebscohost-com.biblioteca-ils.tec.mx/eds/ebookviewer/ebook/bmxlYmt*

fXzIzNzExODJfX0FO0?sid=20d27256-c4ff-4203-9a25-3724b9842d55@redis&vid=3&format=EB

En el modelo que vamos a desarrollar, se generará una matriz de similitud de tipo (nmovies, movies) siendo que el proyecto de recomendación se hará con base en películas. La matriz nos ayudará a extraer películas parecidas a nuestra película de interés.

Se comparte muestra de lo trabajado en Jupyter notebook:

```
cosine_sim = cosine_similarity(movie_features, movie_features)
print(f"Las dimensiones de similaridad coseno de las caracteristicas de nuestra matriz de similitud son: {cosine_sim.shape}")
```

```
Las dimensiones de similaridad coseno de las caracteristicas de nuestra matriz de similitud son: (45376, 45376)
```

Esto nos indica que al evaluar la similitud de coseno del dataframe `movie_features` en la funcion `cosine_similarity()` obtenemos una matriz de similitud del tipo $(n_{\text{movies}}, n_{\text{movies}})$.

Esta matriz contiene valores entre 0 y 1 que representan el angulo de similaridad entre las peliculas en los ejes x , y.

Crearemos un diccionario llamado `movie_idx` donde las llaves son los titulos de las peliculas y los valores son los indices de las peliculas.

```
movie_idx = dict(zip(movies['title'], list(movies.index)))
idx = movie_idx[title]
print(f"El indice de la pelicula {title} en la matriz movie_idx es: {idx}")
```

```
El indice de la pelicula X-Men en la matriz movie_idx es: 3671
```

Al usar el diccionario `movie_idx` , sabemos que Jumanji se representa por el indice 1 en nuestra matriz.

Ahora tratemos de encontrar las 10 peliculas mas similares a Jumanji.

```
numero_recomendaciones=10
sim_scores = list(enumerate(cosine_sim[idx]))
sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
sim_scores = sim_scores[1:(numero_recomendaciones+1)]
similar_movies = [i[0] for i in sim_scores]
```

`similar_movies` es un arreglo de indices que representa el top 10 de recomendaciones a partir de la pelicula elegida (en este caso X-Men). Podemos encontrar el títulos de las peliculas correspondientes al crear un mapeador invertido de `movie_idx` o usando `iloc` en la columna de título del dataframe `movies` .

```
print(f"Por que miraste la pelicula {title}, te pueden interesar las siguientes {numero_recomendaciones} peliculas:")
movies['title'].iloc[similar_movies]
```

```
Por que miraste la pelicula X-Men, te pueden interesar las siguientes 10 peliculas:
```

```
3675              The In Crowd
4570              Original Sin
4684          Don't Say a Word
6188            City of Ghosts
6406                The On the Run
6457                   Purpose
7429                The Unsaid
9307              Black Plague
9401        The Galíndez File
9649    Assault on Precinct 13
Name: title, dtype: object
```

3.2 Identificación y justificación de 3 métricas de evaluación utilizadas para evaluar el desempeño de los sistemas de recomendación:

Las métricas que ocuparemos para evaluar el desempeño de nuestro sistema de recomendación son las siguientes:

a. **Precisión.**

   Al evaluar un sistema de recomendación, el objetivo principal es determinar con qué eficacia el sistema recomienda elementos que son relevantes para el usuario. Esto se hace al comparar dos conjuntos de datos:

1. **Conjunto de elementos relevantes (ground Truth):** Son los elementos que resultan genuinamente interesantes o útiles para el usuario. Este conjunto se deriva de datos históricos de los usuarios.

2. **Conjunto de elementos recomendados:** Son los elementos que el sistema de recomendación sugiere en función de su algoritmo.

Esta métrica captura que tan frecuente un modelo hace una predicción positiva y que esta predicción sea correcta. Nos dice que tan confiados podemos estar que la instancia predicha teniendo el objetivo positivo, realmente es positivo. De acuerdo con (Kelleher et al., 2020), la precisión se calcula de la siguiente manera:

$$precision = \frac{TP}{(TP + FP)}$$

Tp: Verdaderos positivos
Fp: Falsos positivos
Fn: Falsos negativos

b. Recall.

Qué tan bien el sistema es capaz de identificar todos los elementos relevantes disponibles. Un alto recall indica que el sistema es bueno para identificar la mayoría de las preferencias del usuario, lo cual es crucial en situaciones donde es importante no perder elementos relevantes.

Es el equivalente a la tasa de verdaderos positivos. Esta métrica nos va a decir que tan confiables podemos estar en que todas las instancias con el nivel objetivo positivo se encontraron en el modelo. Se calcula como:

$$recall = \frac{TP}{(TP + FN)}$$

Una alta precisión asegura que las recomendaciones son de alta calidad, y un un alto recall asegura que se capturan todos los intereses del usuario. Idealmente, buscamos un balance entre ambas métricas.

c. F1- Score.

Se define como la media armónica entre las métricas de precisión y recall y se calcula de la siguiente manera:

$$F_1\ measure = 2 \times \frac{(precision \times recall)}{(precision + recall)}$$

Esta última métrica tiende hacia valores más pequeños en una lista de números, por lo que es menos sensible a outliers grandes que un promedio aritmético que tiende hacia valores más altos.

3.3 Justificación de las métricas seleccionadas.

La justificación para utilizar estas métricas es ya que con las mismas se puede evaluar la efectividad y desempeño del modelo desde varias perspectivas. Los ajustes que se le hagan al modelo se podrán evaluar conforme a dichas métricas.

3.4 Experimentación con al menos un algoritmo de recomendación básico

Liga del github:

https://github.com/drekkel/TC4034.10/blob/main/RecommendationSystem/Notebooks/Proyecto_Avance_2_Equipo16.ipynb

4. Conclusiones

Se ha integrado perfectamente un algoritmo de recomendación avanzado que utiliza similitud de cosenos para desarrollar un sistema de sugerencia de películas basado en la similitud de sus descripciones. El algoritmo se ha descrito y justificado meticulosamente, y su rendimiento se ha evaluado rigurosamente mediante precisión, recuperación y puntuación F1.

Estas métricas se seleccionaron intencionalmente por su capacidad de proporcionar una visión integral del desempeño del sistema de recomendación desde varias perspectivas. La precisión mide meticulosamente la exactitud de las recomendaciones para garantizar la máxima relevancia. Recall evalúa de forma sólida la capacidad del sistema para capturar todos los elementos relevantes, sin dejar lugar a pasar por alto preferencias importantes del usuario. El F1-Score, al combinar armoniosamente precisión y recuperación, ofrece una evaluación bien equilibrada del rendimiento del modelo.

El análisis detallado y los resultados se han documentado en el repositorio GitHub del equipo, estableciendo un estándar de transparencia y reproducibilidad para el proyecto.

5. Referencias

- Kelleher, J. D., Namee, B. M., & Arcy, A. D. (2020). *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies* (Vol. Second Edition). The MIT Press. https://0-eds-p-ebscohost-com.biblioteca-ils.tec.mx/eds/ebookviewer/ebook/bmxlYmtfXzIzNzExODJfX0FO0?sid=20d27256-c4ff-4203-9a25-3724b9842d55@redis&vid=3&format=EB