RECOMMENDATION SYSTEM WITH MOVIES DATASET

18520153 – Lê Phước Thành,

18520653 – Võ Đông Dương,

18520495 – Phạm Lê Chí Bảo.

Tóm tắt. Chúng em quyết định chọn đề tài "Hệ thống khuyến nghị với movies dataset - Recommender Systems with Movies Dataset" để vừa có thể nghiên cứu, tìm hiểu, học hỏi và trau dồi khả năng chuyên môn, cũng nhữ hiểu biết thêm về quá trình phân tích và xử lý dữ liệu thu thập được của các rạp phim, các kênh chiếu phim online để có thể phân tích xem khách hàng có thể sẽ thích những bộ phim nào từ đó kiến nghị các bộ phim cho khách hàng. chúng em sẽ lựa chọn hình thức Content Based Filtering (CB) của Recommender Systems, Consine Similarity và TF-IDF Vectorizer (Term Frequency – Inverse Document Frequency). Từ đó củng cố thêm khả năng phân tích, xử lý dữ liệu và ra quyết định của bản thân phục vụ cho cuộc sống và công việc sau này.

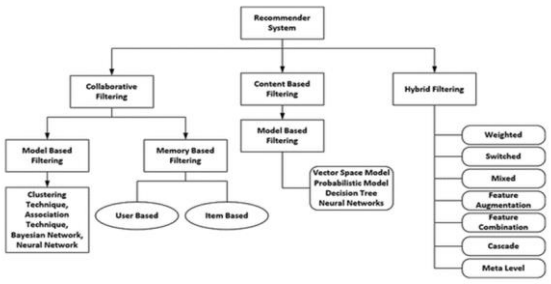
1. Giới thiệu

A. Hệ thống khuyến nghị - Recommender Systerm

Hệ thống gợi ý (Recommender systems hoặc Recommendation systems, Recommender platform, Recommender engine) là một dạng của hệ hỗ trợ ra quyết định, cung cấp giải pháp mang tính cá nhân hóa mà không phải trải qua quá trình tìm kiếm phức tạp. Hệ thống gợi ý học từ người dùng và gợi ý các sản phẩm tốt nhất trong số các sản phẩm phù hợp. Recommender System (RS) là một lớp con của filtering system với ý tưởng là dự đoán "xếp hạng" hay "sở thích" của người dùng.

B. Phương pháp/mô hình Recommender System

Phương pháp / mô hình: một hệ thống khuyên nghị thường sẽ được xây dựng dụa trên một trong ba phương pháp / mô hình sau:



Hình 1. Các loại mô hình của hệ thống Recommender System

Content Based Filtering (CB): là phương pháp phổ biến đưa ra các khuyến nghị mua bán cho người dùng dựa trên nội dung liên quan đến sản phẩm

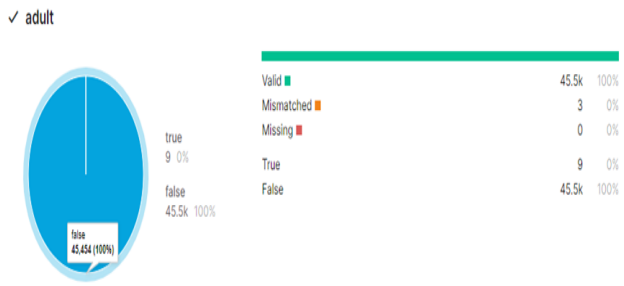
Collaborative Filtering (CF): Hay còn gọi là lọc tương tác, sử dụng sự tương tác qua lại trong hành vi mua sắm giữa các khách hàng để tìm ra sở thích của một khách hàng đối với một sản phẩm. Hầu hết các hành vi hoặc sở thích của mọi người đều có những đặc điểm chung và có thể nhóm lại thành các nhóm tương đồng.

Hybrid Methods: Ngoài ra chúng ta cũng có thể sử dụng kết hợp cả 2 phương pháp trên để tạo thành một thuật toán kết hợp. Ưu điểm của phương pháp này đó là vừa tận dụng được các thông tin từ phía sản phẩm và các thông tin về hành vi mua sắm của người dùngcase, when available, the title, the author name, and the year should be clarified in addition to the detailed address.

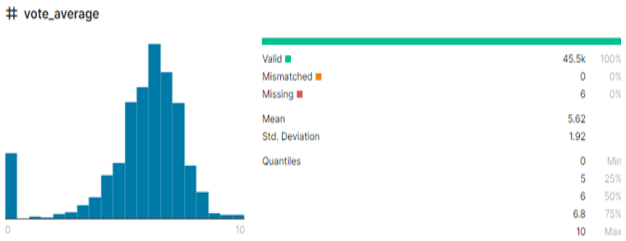
2. Mục tiêu

Biết được cách thức phân tích, xử lý dữ liệu từ tập dữ liệu (dataset) cho trước từ đó áp dụng hệ thống khuyến nghị (ở đây là CB), kết hợp cùng TF-IDF Vectorizer và Consine Similarity để có thể biết được các bộ phim có khả năng được khách hàng xem nhất . Đồng thời, biết được quá trình xử lý dữ liệu của hệ thống khuyến nghị nói chung và hình thức CB của hệ thống khuyến nghị nói riêng cũng như của cả TF-IDF Vectorizer và Consine Similarity.

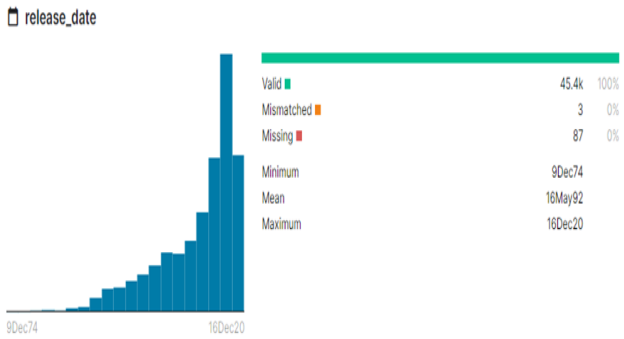
3. Phân tích dữ liệu



Hình 2. Đồ thị biểu hiện tỷ trọng cũng như số lượng khách hàng xem phim ở độ tuổi trưởng thành



Hình 3. Đồ thị thể hiện số số lượng vote trung bình của trường dữ liệu vote_average



Hình 4. Đồ thị thể hiện dữ liệu của trường dữ liệu release_date của movies_metadata.csv

4. Thuật toán

Phương pháp xây dựng dựa theo nội dung

(Content Based Filtering)

Ở phương pháp này, các nhà phát triển, xây dựng (developer) cần phải xây dựng được Item profiles và hàm mất mát (theo mô hình tuyến tính).

Item Profile: Trong các hệ thống content-based, các nội dung của mỗi item sẽ được tổng thu thập lại thành một bộ hồ sơ (proflie) được biểu diễn dưới dạng toán học là một feature vector. Trong những trường hợp đơn giản, feature vector được trực tiếp trích xuất từ item.

Hàm mất mát (Theo mô hình tuyến tính):

Giả sử rằng, ta có:

$$\begin{cases} N \text{ là số users} \\ M \text{ là số items} \\ Y \text{ là ma trận utility} \\ R \text{ là ma trận rated or not} \end{cases}$$

Trong ma trận utility Y thì y(i,j) là mức độ quan tâm (ở đây là số sao đã rate) của user thứ i với sản phẩm thứ j mà hệ thống đã thu thập được. Ma trận Y bị khuyết rất nhiều thành phần tương ứng với các giá trị mà hệ thống cần dự đoán

R thể hiện việc một user đã rated một item hay chưa. Cụ thể, rij = 1 nếu item thứ i đã được rated bởi user thứ j, ngược lại rij = 0 nếu item thứ i chưa được rated bởi user thứ j.

Mô hình tuyến tính:

Giả sử rằng ta có thể tìm được một mô hình có thể tính được mức độ

quan tâm của mỗi user với mỗi item bằng một hàm tuyến tính:

$$y_{mn} = \mathbf{x}_m \mathbf{w}_n + b_n$$

Trong đó, x(m) là vector đặc trưng của item m. Mục tiêu của chúng ta sẽ là học ra mô hình của user, tức là tìm ra w(n) và b(n).

Xét một user thứ n bất kỳ, nếu ta coi training set là tập hợp các thành phần đã được điền của yn, ta có thể xây dựng hàm mất mát tương tự như sau:

$$\mathcal{L}_n = \frac{1}{2} \sum_{m\,:\,r_{mn}=1} (\mathbf{x}_m \mathbf{w}_n + b_n - y_{mn})^2 + \frac{\lambda}{2} ||\mathbf{w}_n||_2^2$$

Trong đó, thành phần thứ hai là regularization term và λ là một tham số dương. Chú ý rằng regularization thường không được áp dụng lên bn. Trong thực hành, trung bình cộng của lỗi thường được dùng, và mất mát nên Ln được viết lại thành:

$$\mathcal{L}_n = \frac{1}{2s_n} \sum_{m\,:\,r_{mn}=1} (\mathbf{x}_m \mathbf{w}_n + b_n - y_{mn})^2 + \frac{\lambda}{s_n} ||\mathbf{w}_n||_2^2$$

Trong đó sn là số lượng các items mà user thứ n đã rated. Nói cách khác, sn là tổng các phần tử trên cột thứ n của ma trận rated or not R:

$$s_n = \sum_{m=1}^{M} r_{mn},$$

Vì hàm mục tiêu chỉ phụ thuộc vào các items đã được rated, ta có thể rút gọn nó bằng cách đặt y ̂n là sub vector của y được xây dựng bằng cách trích các thành phần khác dấu? ở cột thứ n, tức đã được rated bởi user thứ n trong ma trận Y. Đồng thời, đặt X ̂n là submatrix của ma trận feature X, được tạo bằng cách trích các hàng tương ứng với các items đã được rated bởi user thứ n. Khi đó, biểu thức hàm mất mát của mô hình cho user thứ n được viết gọn thành công thức:

Đây chính là bài toán **Ridge Regression**, đã có sẵn trong thư viện “sklearn.linear_model.Ridge” của sklearn. Chúng ta sẽ sử dụng thư viện này để tìm w(n) và b(n) cho mỗi user. Còn bây giờ chúng ta sẽ xét một ví dụ về cách xây dựng mô hình cho mỗi user.

Phương pháp TF-IDF vectorizer

TF-IDF (Term Frequency – Inverse Document Frequency) là 1 kĩ thuật sử dụng trong khai phá

dữ liệu văn bản. Trọng số này được sử dụng để đánh giá tầm quan trọng của một từ trong một văn bản. Giá trị cao thể hiện độ quan trọng cao và nó phụ thuộc vào số lần từ xuất hiện trong văn bản nhưng bù lại bởi tần suất của từ đó trong tập dữ liệu. Một vài biến thể của tf-idf thường được sử dụng trong các hệ thống tìm kiếm như một công cụ chính để đánh giá và sắp xếp văn bản dựa vào truy vấn của người dùng. Tf-idf cũng được sử dụng để lọc những từ stopwords trong các bài toán như tóm tắt văn bản và phân loại văn bản.

$$TfIDF(t,d,D) = tf(t,d) \times idf(t,D)$$

Trong đó:

TF - Term Frequency (Tần suất xuất hiện của từ): là số lần từ xuất hiện trong văn bản. Vì các văn bản có thể có độ dài ngắn khác

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

nhau nên một số từ có thể xuất hiện nhiều lần trong một văn bản dài hơn là một văn bản ngắn. Như vậy, term frequency thường được chia cho độ dài văn bản (tổng số từ trong một văn bản)

Hình công thức xác định TF

Với:

tf(t, d): tần suất xuất hiện của từ t trong văn bản d

f(t, d): Số lần xuất hiện của từ t trong văn bản d

max({f(w, d) : w ∈ d}): Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản d

IDF - Inverse Document Frequency (Nghịch đảo tần suất của văn bản): giúp đánh giá tầm

$$Cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \, \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}}$$

quan trọng của một từ. Khi tính toán TF, tất cả các từ được coi như có độ quan trọng bằng nhau. Nhưng một số từ như “is”, “of” và “that” thường xuất hiện rất nhiều lần nhưng độ quan trọng là không cao. Như thế chúng ta cần giảm độ quan trọng của những từ này xuống.

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

Hình Công thức xác định IDF

Consine similarity

Chỉ số Cosine similarity được sử dụng để xác định mức độ tương tự của các tài liệu bất kể kích thước của các đối tượng cần so sánh. Về mặt toán học, chỉ số tương tự cosine sẽ đo cosin của góc giữa hai vectơ được chiếu trong không gian đa chiều. Trong ngữ cảnh này, hai vectơ đó chính là các mảng chứa số lượng từ của hai tài liệu (các tài liệu càng gần nhau theo góc độ, thì Độ tương đồng Cosine càng cao.).

where, $\vec{a} \cdot \vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$ is the dot product of the two vectors.

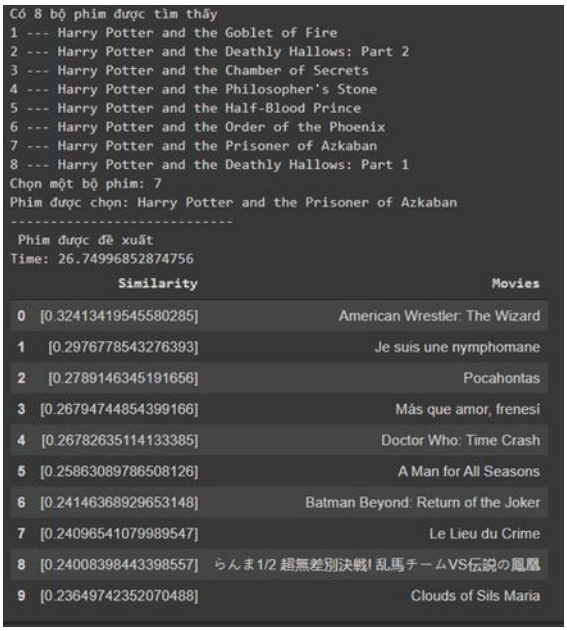
Demographic filtering

Kiểu hệ thống này sẽ gợi ý item dựa nhân khẩu học của user. Giả thuyết cho rằng những thị trường khác nhau nên được gợi ý những item khác nhau. Chẳng hạn, user sẽ được điều hướng tới những website khác nhau dựa trên ngôn ngữ và địa lý. Hoặc là việc gợi ý có thể thay đổi dựa trên tuổi của user.

5. Kết quả thực nghiệm

Tiến hành thực hiện tính toán và đưa ra kết quả là các bộ phim kiến nghị phù hợp với title đầu vào:

Và kết quả cuối cùng nhận được như sau:



```
Có 8 bộ phim được tìm thấy
1 --- Harry Potter and the Goblet of Fire
2 --- Harry Potter and the Deathly Hallows: Part 2
3 --- Harry Potter and the Chamber of Secrets
4 --- Harry Potter and the Philosopher's Stone
5 --- Harry Potter and the Half-Blood Prince
6 --- Harry Potter and the Order of the Phoenix
7 --- Harry Potter and the Prisoner of Azkaban
8 --- Harry Potter and the Deathly Hallows: Part 1
Chọn một bộ phim: 7
Phim được chọn: Harry Potter and the Prisoner of Azkaban
----------------------------
 Phim được đề xuất
Time: 26.74996852874756
```

	Similarity	Movies
0	[0.32413419545580285]	American Wrestler: The Wizard
1	[0.2976778543276393]	Je suis une nymphomane
2	[0.2789146345191656]	Pocahontas
3	[0.26794744854399166]	Más que amor, frenesí
4	[0.26782635114133385]	Doctor Who: Time Crash
5	[0.25863089786508126]	A Man for All Seasons
6	[0.24146368929653148]	Batman Beyond: Return of the Joker
7	[0.24096541079989547]	Le Lieu du Crime
8	[0.24008398443398557]	らんま1/2 超無差別決戦! 乱馬チームVS伝説の鳳凰
9	[0.2364974235207048 8]	Clouds of Sils Maria

6.Tài Liệu Tham Khảo

Tiếng Việt

https://machinelearningcoban.com/2017/05/17/contentbasedrecommendersys/

https://techmaster.vn/posts/35386/cach-xay-dung-recommender-system-rs-phan-1

https://viblo.asia/p/lam-the-nao-de-xay-dung-mot-recommender-system-rs-phan-3-E375zbeW5GW

https://nguyenvanhieu.vn/tf-idf-la-gi/

https://tailieu.vn/doc/luan-van-thac-si-ung-dung-he-thong-tu-van-recommender-systems-trong-linh-vuc-thuong-mai-dien-tu-1602259.html

https://itzone.com.vn/vi/article/tim-hieu-ve-content-based-filtering-phuong-phap-goi-y-dua-theo-noi-dung-phan-1/

Tiếng Anh

https://blog.avenuecode.com/how-to-build-a-recommender-system-in-less-than-1-hour#:~:text=Content%20Based%20Recommender%20System,similar%20preferences%20(Meteren%2C%20et