TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO BÀI TẬP LỚN MÔN

KHAI PHÁ WEB

ĐỀ TÀI: XÂY DỰNG HỆ GỢI Ý ÂM NHẠC

Giảng viên hướng dẫn	TS. Nguyễn Kiêm Hiếu	
Nhóm thực hiện	Nhóm 9	
	Võ Văn Tài	20143927
	Nguyễn Thị Dung	20140698
	Bùi Tiến Thành	20144052
	Lại Trung Kiên	20142398

Hà Nội, Ngày 24/5/2018

Mục lục

1. Giới thiệu

Ngày nay, khi chúng ta sử dụng các dịch vụ trên mạng Internet như Facebook, xem Youtube, nghe nhạc hay mua sắm... Đã bao giờ bạn để ý thấy rằng:

- ❖ Facebook gợi ý đưa ra những người dùng khác mà có thể bạn biết.
- ❖ Youtube hiển thị những video với nhiều chủ để khác nhau ở trang chủ, trong đó có nhiều video mới và bạn cảm thấy nó đúng với sở thích.
- ❖ Spotify đưa ra gợi ý một list những bài hát mới toanh bạn chưa bao giờ nghe nhưng rất có thể bạn sẽ thích.
- ❖ Hay khi bạn mua sắm trên Amazon, hệ thống đưa ra một danh sách những sản phẩm gợi ý mà bạn có thể sẽ muốn mua vì sở thích hoặc cần để sử dụng cùng với những thứ bạn vừa mua mà bạn quên mất. Thật hữu ích phải không?

Và còn rất rất nhiều những ví dụ khác nữa về những hệ thống đưa ra gợi ý cho người dùng trên mạng Internet hiện nay. Cách hoạt động này của hệ thống không những giúp ích cho bạn có thể biết thêm nhiều thông tin bạn cần mà còn giúp cho Facebook tăng lượng tương tác người dùng, Youtube tăng lượng view của các video, Amazon tăng doanh số bán hàng và Spotify có thêm nhiều người dùng vì sự thú vị mà hệ thống gợi ý của nó mang lại. Những thuật toán đứng đằng sau những ứng dụng này là những thuật toán Machine Learning có tên gọi chung là Recommender Systems hoặc Recommendation Systems (Hệ thống gợi ý).

Recommendation Systems là một mảng khá rộng của Machine Learning và có tuổi đời ít hơn so với Classification vì Internet mới chỉ thực sự bùng nổ khoảng 10-15 năm đổ lại đây. Có hai thực thể chính trong Recommendation Systems là users và items. Users là người dùng. Items là sản phẩm, ví dụ như các bộ phim, bài hát, cuốn sách, clip, hoặc cũng có thể là các users khác trong bài toán gợi ý kết bạn. Mục đích chính của các Recommender Systems là dự đoán mức độ quan tâm của một user tới một item nào đó, qua đó có chiến lược gợi ý phù hợp.

2. Một số phương pháp trong hệ gợi ý

Các hệ thống gợi ý thường được chia thành ba nhóm:

2.1 Hệ gợi ý dựa theo nội dung (Content-based systems)

Hệ gợi ý dựa theo nội dung tức là hệ thống sẽ đề xuất một mục cho người dùng dựa trên mô tả những đặc điểm của item và hồ sơ về sở thích của người dùng.
Các hệ thống gợi ý dựa theo nội dung có thể được sử dụng trong nhiều lĩnh vực khác nhau, từ các trang web giới thiệu, đăng bài, trong các nhà hàng hay trên chương trình truyền hình…
Mặc dù chi tiết bên trong của các hệ thống là khác nhau, tuy nhiên các hệ thống gợi ý dựa theo nội dung đều có chung một cách thức để mô tả các item được đề xuất, một cách thức để xây dựng hồ sơ sở thích của người dùng và cách thức để so sánh các item với hồ sơ đó để xác định xem nên đưa ra gợi ý gì.
Hồ sơ sẽ được tự động cập nhật khi có sự thay đổi và gửi lại phản hồi về phía người dùng với những gợi ý mới.

2.2 Hệ gợi ý dựa theo lọc cộng tác (Collaborative filtering)

Là phương pháp gợi ý được triển khai rộng rãi nhất và thành công nhất trong thực tế. Hệ thống theo lọc công tác phân tích và tổng hợp các điểm số đánh giá của các đối tượng, nhận ra sự tương đồng giữa những người sử dụng trên cơ sở các điểm số đánh giá của họ và tạo ra các gợi ý dựa trên sự so sánh này. Hồ sơ (profile) của người sử dụng điển hình trong hệ thống lọc cộng tác bao gồm một vector các đối tượng (item) và các điểm số đánh giá của chúng, với số chiều tăng lên liên tục khi người sử dụng tương tác với hệ thống theo thời gian.
Một số hệ thống sử dụng phương pháp chiết khấu dựa trên thời gian (time-based discounting) để tính toán cho yếu tố “trượt” đối với sự quan tâm của người sử dụng. Trong một số trường hợp điểm số đánh giá (rating) có thể là nhị phân (thích/không thích) hoặc các giá trị số thực cho thấy mức độ ưu tiên.

Thế mạnh lớn nhất của kỹ thuật gợi ý theo lọc cộng tác là chúng hoàn toàn độc lập với sự biểu diễn của các đối tượng đang được gợi ý, và do đó có thể làm việc tốt với các đối tượng phức tạp như âm thanh và phim. Schafer, Konstan & Riedl (1999) gọi lọc cộng tác là "tương quan giữa người – với – người" (people-to-people correlation).

2.3 Hệ gợi ý dựa theo cơ sở tri thức (Knowledge-based systems)

Là hệ thống gợi ý các đối tượng dựa trên các suy luận về nhu cầu và sở thích của người dùng. Theo một nghĩa nào đó, tất cả các kỹ thuật gợi ý có thể mô tả như là làm một số suy luận. Phương pháp tiếp cận dựa trên cơ sở tri thức được phân biệt ở chỗ: chúng có kiến thức làm thế nào một đối tượng cụ thể đáp ứng nhu cầu một người dùng cụ thể, và do đó có thể lập luận về mối quan hệ giữa nhu cầu và các gợi ý cụ thể.
Sử dụng miền tri thức rõ ràng, có liên quan tới mối quan hệ giữa yêu cầu của người dùng và sản phẩm cụ thể. Ban đầu người ta đưa ra 3 dạng tri thức: tri thức về danh mục (tri thức về sản phẩm được gợi ý), tri thức người sử dụng (tri thức về các yêu cầu của người sử dụng), tri thức về các chức năng (tri thức để ánh xạ các yêu cầu của người sử dụng tới các sản phẩm thoả mãn các yêu cầu đó).
Phương pháp này không dựa trên tiểu sử người sử dụng nên không gặp phải khó khăn về sản phẩm mới và người dùng mới. Gợi ý trên cơ sở tri thức có khả năng suy diễn, khả năng suy diễn phụ thuộc vào độ phù hợp của yêu cầu người sử dụng với các thuộc tính của sản phẩm.
Mọi hệ thống dựa trên cơ sở tri thức đều là mối quan hệ thu nhận tri thức. Thực tế, chất lượng của các phương án gợi ý tùy thuộc vào độ chính xác của cơ sở tri thức. Đây cũng là hạn chế lớn nhất của phương pháp này.

3. Hệ gợi ý âm nhạc sử dụng phương pháp Collaborative filtering

3.1 Giới thiệu bài toán

Với mục đích xây dựng một hệ thống gợi ý danh sách các bài hát cho người nghe nhạc. Nhóm em sử dụng một phương pháp Collaborative Filtering(CF) có tên là Neighborhood-based Collaborative Filtering(NBCF). Ý tưởng cơ bản của NBCF là

xác định mức độ quan tâm của user với một item dựa trên các users khác gần giống với user này. Việc gần giống nhau giữa các users có thể được xác định thông qua mức độ quan tâm của các user này tới items khác mà hệ thống đã biết. Vấn đề quan trọng nhất trong hệ thống Neighborhood-based Collaborative Filtering mà nhóm em muốn xây dựng đó là giải quyết được hai câu hỏi:

- ❖ Làm thế nào để xác định sự giống nhau giữa hai users?
- ❖ Khi đã xác định được các users gần giống nhau rồi, làm thế nào dự đoán được mức độ quan tâm của user đó lên item?

Việc xác định mức độ quan tâm của mỗi user tới một item dựa trên mức độ quan tâm của những user gần giống với item đó còn được gọi là User-user collaborative filtering. Có một hướng tiếp cần khác được cho là hiệu quả hơn là Item-item collaborative filtering. Trong hướng tiếp cân này thay vì xác định user gần giống nhau, hệ thống sẽ xác định những item gần giống nhau. Từ đó hệ thống gợi ý những items gần giống với những item mà user có mức độ quan tâm cao.

3.2 User-user collaborative filtering

Hàm xác định sự giống nhau giữa hai user (Similarity functions)

Công việc quan trọng nhất phải làm trước tiên trong User-user Collaborative Filtering là xác định sự giống nhau giữa hai users. Dữ liệu duy nhất chúng ta có đó là Utility matrix Y. Nói thêm về Utility matrix Y, đó là tập tất các ratings bao gồm cả những giá trị chưa biết cần dự đoán của users cho các item. Ratings thể hiện mức độ quan tâm của user cho item đó. Thông thường, có rất nhiều users và items trong hệ thống, mỗi user thường chỉ rate một số lượng nhỏ các item, thậm chí có những user không rate item nào. Vì vậy, lượng ô không có giá trị trong Utility matrix trong các bài toán đó là rất lớn và lượng ô đã điền là rất nhỏ.

Rõ ràng rằng càng nhiều ô được điền thì độ chính xác của hệ thống sẽ càng được cải thiện. Vì vậy, các hệ thống luôn luôn hỏi người dùng về sự quan tâm của họ tới sản phẩm, và muốn người dùng đánh giá càng nhiều sản phẩm càng tốt. Việc đánh giá các sản phẩm, vì thế, không những giúp các người dùng khác biết được chất lượng sản

phẩm mà còn giúp hệ thống biết được sở thích của người dùng, qua đó có chính sách quảng cáo hợp lý.

	u_0	u_1	u_2	u_3	u_4	u_5
i_0	6	6	2	3		
i_1	3	3			1	
i_2			2			2
i_3	4			5		4
i_4	3	2	4			5

Hình 1: Ví dụ về utility matrix dựa trên số sao một user rate cho một item. Một cách trực quan, hành vi của u_0 giống với u_1 hơn là u_2, u_3, u_4, u_5. Từ đó có thể dự đoán rằng u_1 sẽ quan tâm tới i_3 vì u_0 cũng quan tâm tới item này.

Câu hỏi đặt ra là: hàm số similarity nào là tốt? Để đo similarity giữa hai users, cách thường làm là xây dựng feature vector cho mỗi user rồi áp dụng một hàm có khả năng đo similarity giữa hai vectors đó. Chú ý rằng việc xây dựng feature vector này khác với việc xây dựng item profiles như trong Content-based Recommendation Systems. Các vectors này được xây dựng trực tiếp dựa trên Utility matrix chứ không dùng dữ liệu ngoài như item profiles. Với mỗi user, thông tin duy nhất chúng ta biết là các ratings mà user đó đã thực hiện, tức cột tương ứng với user đó trong Utility matrix. Tuy nhiên, khó khăn là các cột này thường có rất nhiều mising ratings vì mỗi user thường chỉ rated một số lượng rất nhỏ các items. Cách khắc phục là bằng cách nào đó, ta giúp hệ thống điền các giá trị này sao cho việc điền không làm ảnh hưởng nhiều tới sự giống nhau giữa hai vector. Việc điền này chỉ phục vụ cho việc tính similarity chứ không phải là suy luận ra giá trị cuối cùng.

Vậy mỗi dấu '?' nên được thay bởi giá trị nào để hạn chế việc sai lệch quá nhiều? Một lựa chọn bạn có thể nghĩ tới là thay các dấu '?' bằng giá trị '0'. Điều này không thực sự tốt vì giá trị '0' tương ứng với mức độ quan tâm thấp nhất. Một giá trị an toàn hơn là 2.5 vì nó là trung bình cộng của 0, mức thấp nhất, và 5, mức cao nhất. Tuy nhiên, giá trị này có hạn chế đối với những users dễ tính hoặc khó tính. Với các users dễ tính, thích tương ứng với 5 sao, không thích có thể ít sao hơn 1 chút, 3 sao chẳng hạn. Việc chọn giá trị 2.5 sẽ khiến cho các items còn lại là quá negative đối với user đó. Điều ngược lại xảy ra với những user khó tính hơn khi chỉ cho 3 sao cho các items họ thích và ít sao hơn cho những items họ không thích.

Một giá trị khả dĩ hơn cho việc này là trung bình cộng của các ratings mà user tương ứng đã thực hiện. Việc này sẽ tránh được việc users quá khó tính hoặc dễ tính, tức lúc nào cũng có những items mà một user thích hơn so với những items khác.

Chuẩn hóa dữ liệu

Sau khi tính được giá trị trung bình của ratings cho mỗi user. Giá trị cao tương ứng với các user dễ tính và ngược lại. Khi đó, nếu tiếp tục trừ từ mỗi rating đi giá trị này và thay các giá trị chưa biết bằng 0, ta sẽ được normalized utility matrix. Bước chuẩn hóa này là rất quan trọng, lí do là bởi vì:

- Việc trừ đi trung bình cộng của mỗi cột khiến trong mỗi cột có những giá trị dương và âm. Những giá trị dương tương ứng với việc user thích item, những giá trị âm tương ứng với việc user không thích item. Những giá trị bằng 0 tương ứng với việc chưa xác định được liệu user có thích item hay không.
- Về mặt kỹ thuật, số chiều của utility matrix là rất lớn với hàng triệu users và items, nếu lưu toàn bộ các giá trị này trong một ma trận thì khả năng cao là sẽ không đủ bộ nhớ. Quan sát thấy rằng vì số lượng ratings biết trước thường là một số rất nhỏ so với kích thước của utility matrix, sẽ tốt hơn nếu chúng ta lưu ma trận này dưới dạng sparse matrix, tức chỉ lưu các giá trị khác không và vị trí của chúng. Vì vậy, tốt hơn hết, các dấu '?' nên được thay bằng giá trị '0', tức chưa xác định liệu user có thích item hay không. Việc này không những tối ưu bộ nhớ mà việc tính toán similarity matrix sau này cũng hiệu quả hơn.

Sau khi đã chuẩn hoá dữ liệu như trên, một vài similiraty function thường được sử dụng là:

Cosine Similarity

Đây là hàm được sử dụng nhiều nhất, và cũng quen thuộc với các bạn nhất. Nếu các bạn không nhớ công thức tính Cosine của góc giữa hai vector u_1, u_2 trong chương trình phổ thông, thì dưới đây là công thức:

$$\text{cosine\_similarity}(u_1, u_2) = \cos(u_1, u_2) = \frac{u_1^T u_2}{||u_1||_2 . ||u_2||_2} \quad (1)$$

Trong đó u_{12} là vectors tương ứng với users 1, 2 đã được chuẩn hoá như ở trên.

Có một tin vui là python có hàm hỗ trợ tính toán hàm số này một cách hiệu quả.

Độ similarity của hai vector là 1 số trong đoạn [-1, 1]. Giá trị bằng 1 thể hiện hai vector hoàn toàn similar nhau. Hàm số Cosine của một góc bằng 1 nghĩa là góc giữa hai vector bằng 0, tức một vector bằng tích của một số dương với vector còn lại. Giá trị Cosine bằng -1 thể hiện hai vector này hoàn toàn trái ngược nhau. Điều này cũng hợp lý , tức khi hành vi của hai users là hoàn toàn ngược nhau thì similarity giữa hai vector đó là thấp nhất.

Có một chú ý quan trọng ở đây là khi số lượng users lớn, ma trận S cũng rất lớn và nhiều khả năng là không có đủ bộ nhớ để lưu trữ, ngay cả khi chỉ lưu hơn một nửa số các phần tử của ma trận đối xứng này. Với các trường hợp đó, mới mỗi user, chúng ta chỉ cần tính và lưu kết quả của một hàng của similarity matrix, tương ứng với việc độ giống nhau giữa user đó và các users còn lại.

Rating prediction

Tương tự như KNN, trong Collaborative Filtering, missing rating cũng được xác định dựa trên thông tin về k neighbor users. Tất nhiên, chúng ta chỉ quan tâm tới các users đã rated item đang xét. Predicted rating thường được xác định là trung bình có trọng

số của các ratings đã chuẩn hoá. Có một điểm cần lưu ý, trong KNN, các trọng số được xác định dựa trên distance giữa 2 điểm, và các distance này là các số không âm. Trong khi đó, trong CF, các trọng số được xác định dựa trên similarity giữa hai users, những trọng số này có thể nhỏ hơn 0.

Công thức phổ biến được sử dụng để dự đoán rating của u cho i là:

$$\hat{y}_{i,u} = \frac{\sum_{u_j \in \mathcal{N}(u,i)} \bar{y}_{i,u_j} \text{sim}(u, u_j)}{\sum_{u_j \in \mathcal{N}(u,i)} |\text{sim}(u, u_j)|} \quad (2)$$

Trong đó N(u,i) là tập hợp k users trong neighborhood (tức có similarity cao nhất) của u mà đã rated i, và sim(u, u_j) là giá trị tương đương của u và u_j .

3.3 Item-item collaborative filtering

Một số hạn chế của User-user collaborative filtering

- Trên thực tế, số lượng users luôn lớn hơn số lượng items rất nhiều. Kéo theo đó là Similarity matrixlà rất lớn với số phần tử phải lưu giữ là hơn 1 nửa của bình phương số lượng users (chú ý rằng ma trận này là đối xứng). Việc này, như đã đề cập ở trên, khiến cho việc lưu trữ ma trận này trong nhiều trường hợp là không khả thi.
- Ma trận Utility Y thường là rất thưa. Với số lượng users rất lớn so với số lượng items, rất nhiều cột của ma trận này sẽ rất thưa, tức chỉ có một vài phần tử khác 0. Lý do là users thường lười rating. Cũng chính vì việc này, một khi user đó thay đổi rating hoặc rate thêm items, trung bình cộng các ratings cũng như vector chuẩn hoá tương ứng với user này thay đổi nhiều. Kéo theo đó, việc tính toán ma trận Similarity, vốn tốn nhiều bộ nhớ và thời gian, cũng cần được thực hiện lại.

Ngược lại, nếu chúng ta tính toán similarity giữa các items rồi recommend những items gần giống với item yêu thích của một user thì sẽ có những lợi ích sau:

- Vì số lượng items thường nhỏ hơn số lượng users, Similarity matrix trong trường hợp này cũng nhỏ hơn nhiều, thuận lợi cho việc lưu trữ và tính toán ở các bước sau.
- Vì số lượng phần tử đã biết trong Utility matrix là như nhau nhưng số hàng (items) ít hơn số cột (users), nên trung bình, mỗi hàng của ma trận này sẽ có nhiều phần tử đã biết hơn số phần tử đã biết trong mỗi cột. Việc này cũng dễ hiểu vì mỗi item có thể được rated bởi nhiều users. Kéo theo đó, giá trị trung bình của mỗi hàng ít bị thay đổi hơn khi có thêm một vài ratings. Như vậy, việc cập nhật ma trận Similarity Matrix có thể được thực hiện ít thường xuyên hơn.

Cách tiếp cận thứ hai này được gọi là Item-item Collaborative Filtering. Hướng tiếp cận này được sử dụng nhiều trong thực tế hơn.

3.4 Hệ gợi ý âm nhạc

Cơ sở dữ liệu

Nguồn lấy từ http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/

Thống kê khoảng 17 triệu lượt nghe của 1000 người dùng (Sau khi đã bỏ qua các dòng bị lỗi trong file)

Bao gồm 2 file:

- File thông tin của 1000 người dùng
  - Gồm các thông tin: User id, giới tính, tuổi, quốc tịch, lần đăng nhập gần nhất
- File thông tin lượt nghe
  - Gồm các thông tin: user id, thời gian nghe, id nghệ sĩ, tên nghệ sĩ, id bài hát, tên bài hát

Xử lý dữ liệu

- Xử lý thông tin người dùng:
  - Hệ thống gợi ý nhạc của nhóm em không phân biệt giới tính, tuổi hay quốc tịch, vì vậy chỉ lấy ra id user và đặt password trùng với id user
- Xử lý dữ liệu thông tin lượt nghe

- ➢ Lấy ra những bài hát phổ biến nhất
- ➢ Được nhiều người nghe nhất, không liên quan lượt nghe bài đó của mỗi người
- ➢ Số lượng: 500
- ➢ Thông tin lấy ra: Id bài hát, tên bài hát, tên ca sĩ, số lượng người nghe

❖ Lấy ra thông tin lượt nghe của 500 bài phổ biến nhất đó

- ➢ Thông tin lấy ra: Id user, id bài hát, timestamp

Cơ sở dữ liệu:

- ❖ Bảng user
- ❖ Bảng song
- ❖ Bảng tracking

Hoạt động của ứng dụng

Đăng ký – Đăng nhập

- ❖ Đăng ký người dùng mới trong hệ thống
  - ➢ Thêm dữ liệu vào bảng user trong csdl
- ❖ Duy trì phiên đăng nhập (session)
  - ➢ Lưu thông tin người đang sử dụng hệ thống
  - ➢ Lưu danh sách bài hát đã nghe trong phiên đăng nhập hiện tại

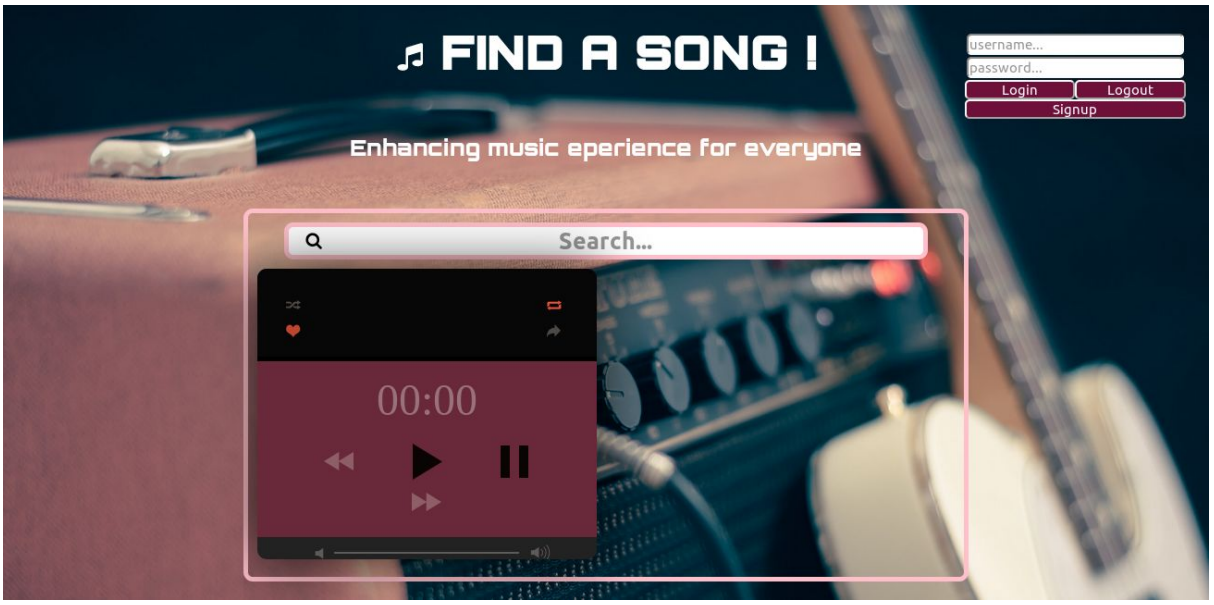
Tìm kiếm bài hát

- ❖ Theo tên bài hát và tên ca sĩ
- ❖ Sử dụng text search đơn giản

Nghe bài bát

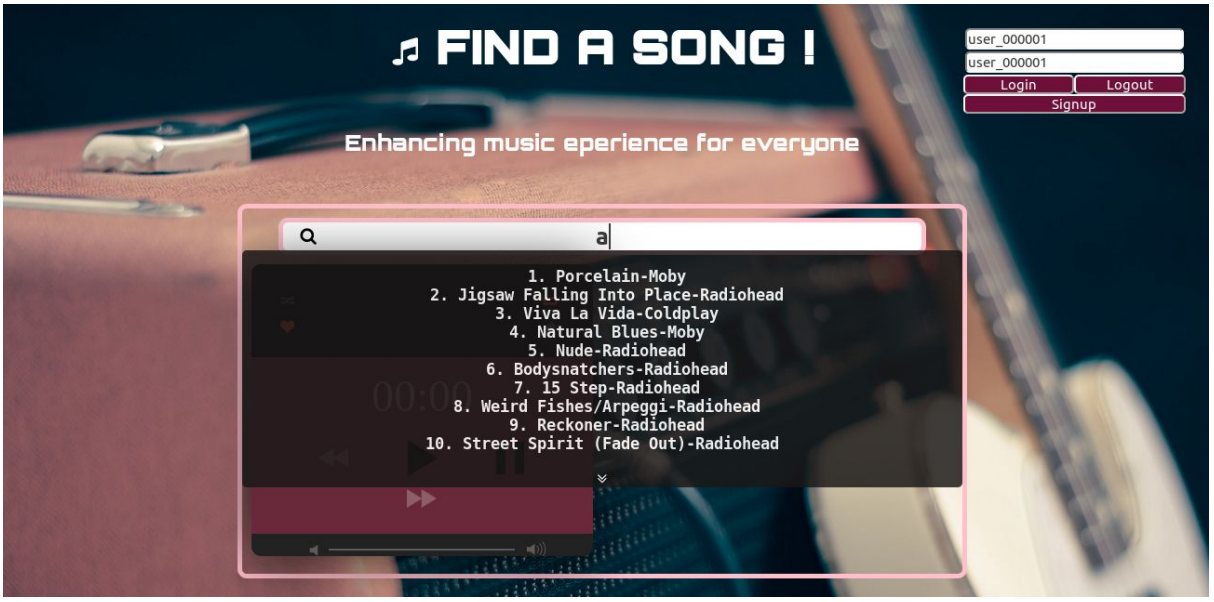
- ❖ Lấy thông tin bài hát và hiển thị
- ❖ Không có chức năng nghe vì bộ dữ liệu không cung cấp nguồn để nghe
- ❖ Thực hiện mỗi khi click vào link bài hát ở danh sách gợi ý hoặc ở danh sách tìm kiếm trả về

Một số hình ảnh của ứng dụng



Hình 2: Giao diện chính của ứng dụng

Hình 3: Phần đăng nhập



Hình 4: Phần tìm kiếm bài hát

Hình 5: Gợi ý bài hát