

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI:

**NGHIÊN CỨU HỆ THỐNG ĐỀ XUẤT NỘI DUNG
ÁP DỤNG KĨ THUẬT CONTENT-BASED FILTERING TRONG
ỨNG DỤNG NGHE NHẠC TRỰC TUYẾN INCOGNITO MUSIC**

Giảng viên hướng dẫn: ThS. TRẦN PHONG NHÃ

Sinh viên thực hiện: HOÀNG MINH TÀI

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá : 60

Tp. Hồ Chí Minh, năm 2023

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI:

**NGHIÊN CỨU HỆ THỐNG ĐỀ XUẤT NỘI DUNG
ÁP DỤNG KĨ THUẬT CONTENT-BASED FILTERING TRONG
ỨNG DỤNG NGHE NHẠC TRỰC TUYẾN INCOGNITO MUSIC**

Giảng viên hướng dẫn: ThS. TRẦN PHONG NHÃ

Sinh viên thực hiện: HOÀNG MINH TÀI

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá : 60

Tp. Hồ Chí Minh, năm 2023

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

BỘ MÔN: CÔNG NGHỆ THÔNG TIN

-----***-----

Mã sinh viên: 6051071102

Họ tên SV: HOÀNG MINH TÀI

Khóa: 60

Lớp: CÔNG NGHỆ THÔNG TIN

1. Tên đề tài

Nghiên cứu hệ thống đề xuất nội dung áp dụng kĩ thuật Content-based Filtering trong ứng dụng nghe nhạc trực tuyến Incognito Music

2. Mục đích, yêu cầu

a. Mục đích

- Xây dựng ứng dụng nghe nhạc trực tuyến hoàn toàn miễn phí với đầy đủ tính năng nhằm mang lại trải nghiệm nghe nhạc tốt hơn cho người dùng.
- Nghiên cứu hệ thống đề xuất nội dung giúp người dùng tìm kiếm thông tin một cách nhanh chóng và hiệu quả hơn.

b. Yêu cầu

- Có bộ dữ liệu đủ lớn về các thể loại và nghệ sĩ
- Phát nhạc ở chất lượng cao nhất và có thể tùy chọn chất lượng

- Giao diện dễ nhìn và thân thiện cho các tác vụ như tìm kiếm và trải nghiệm các bài hát
- Cá nhân hóa nội dung hiển thị dựa vào sở thích nghe nhạc của mỗi người dùng.
- Cung cấp chức năng tạo các danh sách phát và lưu danh sách yêu thích.
- Chia sẻ các bài hát yêu thích cho mọi người.
- Chuyển đổi linh hoạt giữa các ngôn ngữ.

3. Nội dung và phạm vi đề tài

a. Nội dung

i. Về ứng dụng

- Xây dựng giao diện ứng dụng dựa trên Spotify.
- Sử dụng Spotify và YouTube API để tận dụng kho dữ liệu có sẵn làm tiền đề cho hệ thống đề xuất nội dung.
- Thực hiện các chức năng được đặt ra ở phần yêu cầu.
- Nghiên cứu sử dụng Firebase Realtime Database để lưu trữ dữ liệu theo thời gian thực.

ii. Về hệ thống đề xuất nội dung

- Sử dụng bộ dữ liệu từ Firebase làm dữ liệu đầu vào cho hệ thống.
- Nghiên cứu kỹ thuật Content-based Filtering bao gồm quá trình Tiền xử lý dữ liệu, xây dựng ma trận và sử dụng phép toán Cosine trong tính toán mức độ tương đồng.
- Thực hiện các đề xuất: những ca khúc người dùng có thể thích, bài hát tương tự với bài hát đang nghe, danh sách nhạc thịnh hành.

b. Phạm vi

- Nghiên cứu và thu thập dữ liệu: Sinh viên lớp CNTT-K60 và nhân viên văn phòng từ 20 đến 30 tuổi.
- Kỹ thuật: Dart(Flutter), Python (Flask, Jupyter Notebook), RESTful API.

4. Công nghệ, công cụ và ngôn ngữ lập trình

- Công nghệ: Flutter, Flask, RESTful API, Jupyter Notebook.
- Công cụ: Visual Studio Code, Android Studio.
- Ngôn ngữ lập trình: Dart, Python.
- Cơ sở dữ liệu: Firebase Realtime Database

5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng

- Ứng dụng Incognito Music đáp ứng được tất cả các yêu cầu đề ra.
- Hệ thống đề xuất nội dung hoạt động với tỉ lệ ổn định.
- Bài báo cáo đề tài (Word, Powerpoint).

6. Giáo viên và cán bộ hướng dẫn

Họ tên: ThS. Trần Phong Nhã

Đơn vị công tác: Trường Đại học Giao thông Vận tải phân hiệu tại Thành phố Hồ Chí Minh

Điện thoại: 0906 761 014 Email: tpnha@utc2.edu.vn

Ngày tháng 03 năm 2023

Đã giao nhiệm vụ TKTN

Trưởng BM Công nghệ Thông tin

Giáo viên hướng dẫn

ThS. Trần Phong Nhã

ThS. Trần Phong Nhã

Đã nhận nhiệm vụ TKTN

Sinh viên: Hoàng Minh Tài

Ký tên:

Điện thoại: 0852 976 038

Email: 6051071102@st.utc2.edu.vn

LỜI CẢM ƠN

Thông qua bài báo cáo lần này, em xin gửi lời cảm ơn chân thành đến tất cả những cá nhân đã giúp đỡ em trong quá trình hoàn thành Đồ án tốt nghiệp lần này. Dự án sẽ không thể diễn ra suôn sẻ nếu không có sự giúp đỡ của những người này.

Lời đầu tiên, em xin bày tỏ lòng biết ơn tới người hướng dẫn của mình, Thạc sĩ Trần Phong Nhã. Thầy là người đã cho em rất nhiều lời khuyên hữu ích cũng như những gợi ý mới lạ cho đề tài của mình. Đặc biệt là về lĩnh vực máy học, kĩ thuật mà em chưa từng nghĩ rằng sẽ có trong đồ án của mình. Nhờ thầy mà em có đủ tự tin để thử thách bản thân cho một xu hướng mới và mang đầy tính tương lai.

Dự án này là thành quả đúc kết sau 4 năm học tập và rèn luyện tại Trường Đại học Giao thông Vận tải phân hiệu tại Thành phố Hồ Chí Minh. Em xin gửi lời cảm ơn sâu sắc đến tất cả các giảng viên đã trực tiếp giảng dạy và giúp đỡ em trong chặng hành trình dài này. Lời cảm ơn này đi kèm với lời chúc sức khỏe dành cho thầy cô, mong sẽ có thêm nhiều thế hệ sinh viên Giao thông nữa nhận được sự kèm cặp và chỉ dạy của thầy cô.

Mặc dù đã cố gắng hết sức để hoàn thành đồ án nhưng cũng không thể tránh khỏi những sai sót trong ứng dụng. Em rất cảm kích và rất mong nhận được những đánh giá và đóng góp từ quý thầy cô Bộ môn Công nghệ Thông tin, những sự góp ý của thầy cô sẽ là hành trang quý báu mà em cần có trong chặng đường sắp tới.

Cuối cùng, em muốn nói lời cảm ơn đến cả gia đình, đặc biệt là bố mẹ và chị gái vì họ đã an ủi, hỗ trợ và động viên để em tiếp tục tiến về phía trước khi đối mặt với trở ngại trong cuộc sống đại học của mình.

Một lần nữa, em xin chân thành cảm ơn!

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Tp. Hồ Chí Minh, ngày tháng năm

Giáo viên hướng dẫn

ThS. Trần Phong Nhã

MỤC LỤC

| | |
|--|------|
| NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP | i |
| LỜI CẢM ƠN | iv |
| NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN | v |
| MỤC LỤC | vi |
| DANH MỤC CHỮ VIẾT TẮT | xii |
| DANH MỤC BẢNG BIỂU | xiii |
| DANH MỤC HÌNH ẢNH | xiv |
| TỔNG QUAN | 1 |
| Lý do chọn đề tài | 1 |
| Mục tiêu nghiên cứu | 1 |
| Phạm vi chọn đề tài | 1 |
| Cấu trúc báo cáo đồ án tốt nghiệp | 2 |
| CHƯƠNG 1. CƠ SỞ LÝ THUYẾT | 3 |
| 1.1 Công nghệ sử dụng | 3 |
| 1.1.1 Giới thiệu về Dart và Flutter Framework | 3 |
| 1.1.2 Giới thiệu về Python | 4 |
| 1.1.3 Giới thiệu về Firebase Realtime Database | 4 |

| | |
|---|----|
| 1.2 Tổng quan về Flutter | 5 |
| 1.2.1 Tiện ích con - Widget | 5 |
| 1.2.2 Hỗ trợ tương tác - Gesture | 5 |
| 1.2.3 Quản lý trạng thái - State | 5 |
| 1.2.4 Các lớp con trong ứng dụng Flutter | 6 |
| 1.2.5 Ưu/nhược điểm của Flutter | 7 |
| 1.3 Tổng quan về Python, Flask và Jupyter Notebook | 8 |
| 1.3.1 Lịch sử của Python | 9 |
| 1.3.2 Các tính năng của Python | 10 |
| 1.3.3 Flask | 11 |
| 1.3.4 Jupyter Notebook | 11 |
| 1.4 Tổng quan về Firebase Realtime Database | 12 |
| 1.4.1 Khái niệm | 12 |
| 1.4.2 Các tính năng tốt và nhược điểm của Realtime Database | 12 |
| 1.4.3 Các dịch vụ khác của Firebase | 13 |
| 1.5 Tổng quan về Machine Learning | 15 |
| 1.5.1 Giới thiệu | 15 |
| 1.5.2 Lý do chọn Machine Learning | 16 |
| 1.5.3 Các vấn đề mà Machine Learning có thể giải quyết | 17 |

| | |
|--|----|
| CHƯƠNG 2. HỆ THỐNG ĐỀ XUẤT NỘI DUNG | 20 |
| 2.1 Giới thiệu | 20 |
| 2.2 Các khái niệm chính | 21 |
| 2.2.1 User - Item | 21 |
| 2.2.2 Thông tin phản hồi và hai dạng bài toán chính trong RS | 22 |
| 2.3 Các kĩ thuật chính trong RS | 23 |
| 2.3.1 Lọc cộng tác | 23 |
| 2.3.2 Lọc dựa trên nội dung | 26 |
| 2.3.3 Hệ thống gợi ý lai (Hybrid recommender systems) | 27 |
| 2.3.4 Các kĩ thuật không cá nhân hóa | 30 |
| 2.4 Chi tiết về kĩ thuật lọc dựa trên nội dung (Content-based Filtering) | 31 |
| 2.4.1 Tiền xử lý dữ liệu | 31 |
| 2.4.2 Tính độ tương đồng của thông tin | 35 |
| 2.4.3 Các dạng đề xuất bài hát trong ứng dụng | 39 |
| CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG | 44 |
| 3.1 Phân tích hệ thống | 44 |
| 3.2 Thiết kế hệ thống | 47 |
| 3.2.1 Sơ đồ phân rã chức năng BFD | 47 |
| 3.2.2 Sơ đồ Use Case | 49 |

| | |
|--|-----------|
| 3.2.3 Sơ đồ hoạt động | 50 |
| 3.3 Cơ sở dữ liệu | 54 |
| CHƯƠNG 4. XÂY DỰNG ỨNG DỤNG INCOGNITO MUSIC | 56 |
| 4.1 Giới thiệu | 56 |
| 4.1.1 Trang chủ | 56 |
| 4.1.2 Màn hình YouTube | 58 |
| 4.1.3 Thư viện | 59 |
| 4.1.4 Cài đặt và Bảng xếp hạng | 62 |
| 4.2 Các chức năng quan trọng | 63 |
| 4.2.1 Nghe nhạc | 63 |
| 4.2.2 Tìm kiếm | 69 |
| 4.2.3 Tùy chỉnh cài đặt | 70 |
| KẾT LUẬN VÀ KIẾN NGHỊ | 72 |
| Kết quả đạt được | 72 |
| Hạn chế | 72 |
| Kiến nghị | 72 |
| PHỤ LỤC | 73 |
| Phụ lục 1: Hướng dẫn cài đặt | 73 |
| Phụ lục 2: Đường dẫn Github : IncognitoMusic | 73 |

TÀI LIỆU THAM KHẢO 74

DANH MỤC CHỮ VIẾT TẮT

| STT | Tù viết tắt | Tù đầy đủ | Ý nghĩa |
|-----|-------------|-----------------------------------|--|
| 1 | UI | User Interface | Giao diện người dùng |
| 2 | API | Application Programming Interface | Một phương thức cho phép các ứng dụng khác nhau có thể giao tiếp, trao đổi dữ liệu qua lại |
| 3 | AI | Artificial Intelligence | Trí tuệ nhân tạo |
| 4 | ML | Machine Learning | Máy học |
| 5 | RS | Recommendation System | Hệ thống đề xuất |
| 6 | FRD | Firebase Realtime Database | Cơ sở dữ liệu theo thời gian thực |
| 7 | SDK | Software Development Kit | Tập hợp các công cụ cần thiết để phát triển phần mềm cho một nền tảng cụ thể |
| 8 | REST | REpresentational State Tranfer | Một tiêu chuẩn dùng trong việc thiết kế API |
| 9 | HTML | HyperText Markup Language | Một ngôn ngữ tạo nên các thành phần trong trang web hoặc ứng dụng di động |

| | | | |
|-----------|------|----------------------------|---|
| 10 | VM | Virtual Machine | Máy ảo |
| 11 | JSON | JavaScript Object Notation | Một kiểu định dạng dữ liệu, là tiêu chuẩn mở để trao đổi dữ liệu. |

DANH MỤC BẢNG BIỂU

| | |
|---|----|
| Bảng 3.1 : Phân tích yêu cầu hệ thống | 46 |
| Bảng 3.2 : Thực thể Bài hát | 54 |
| Bảng 3.3 : Thực thể Người dùng | 55 |
| Bảng 3.4 : Thực thể Thông số | 55 |

DANH MỤC HÌNH ẢNH

| | |
|--|----|
| Hình 1.1: Sự khác biệt giữa Stateful và Stateless Widget | 6 |
| Hình 1.2: Tổng quan chung về một lớp trong Flutter | 7 |
| Hình 1.3: Những dịch vụ mà Firebase đang cung cấp | 14 |
| Hình 2.1: Ví dụ về hệ thống đề xuất nội dung | 21 |
| Hình 2.2: Ma trận biểu diễn dữ liệu trong RS (user-item-rating matrix) | 22 |
| Hình 2.3: Lấy ra danh sách bài hát từ FRB | 31 |
| Hình 2.4: Chuyển đổi dữ liệu bằng pandas DataFrame | 32 |
| Hình 2.5: Chuyển tất cả từ về dạng chữ thường | 33 |
| Hình 2.6: Dưa các từ trong danh sách về dạng nguyên bản bằng PorterStemmer | 34 |
| Hình 2.7: Số hóa dữ liệu sử dụng CountVectorizer | 35 |
| Hình 2.8: Số hóa dữ liệu sử dụng TfidfVectorizer | 36 |
| Hình 2.9: Tính độ tương đồng bằng Cosine_similarity | 38 |
| Hình 2.10: Đề xuất những bài hát phổ biến | 39 |
| Hình 2.11: Danh sách các bài hát thịnh hành | 40 |
| Hình 2.12: Lấy ra những bài hát mà người dùng đã nghe | 41 |
| Hình 2.13: Đề xuất những bài hát có thể phù hợp với người dùng | 42 |
| Hình 2.14: Tính cá nhân hóa đã được mô hình thể hiện | 43 |
| Hình 3.1: Sơ đồ phân rã chức năng tổng quát | 47 |

| | |
|---|----|
| Hình 3.2: Sơ đồ phân rã chức năng Quản lý bài hát..... | 48 |
| Hình 3.3: Sơ đồ Use Case tổng quát | 49 |
| Hình 3.4: Sơ đồ hoạt động chức năng Nghe nhạc | 50 |
| Hình 3.5: Sơ đồ hoạt động chức năng Tải nhạc | 51 |
| Hình 3.6: Sơ đồ hoạt động chức năng Quản lý danh sách phát..... | 52 |
| Hình 3.7: Sơ đồ hoạt động chức năng Tìm kiếm | 53 |
| Hình 4.1: Màn hình trang chủ | 56 |
| Hình 4.2: Thanh sidetab | 56 |
| Hình 4.3: Màn hình YouTube | 58 |
| Hình 4.4: Thông tin Album của YouTube | 58 |
| Hình 4.5: Màn hình Thư viện | 59 |
| Hình 4.5: Danh sách đang phát | 60 |
| Hình 4.6: Danh sách nghe gần đây | 60 |
| Hình 4.7: Danh sách bài hát yêu thích | 61 |
| Hình 4.8: Quản lý các danh sách phát | 61 |
| Hình 4.10: Màn hình Cài đặt | 62 |
| Hình 4.11: Màn hình Bảng xếp hạng | 62 |
| Hình 4.12: Danh sách bài hát tương tự | 63 |
| Hình 4.13: Xem lời bài hát | 64 |

| | |
|--|----|
| Hình 4.14: Chia sẻ bài hát..... | 65 |
| Hình 4.15: Hẹn giờ tắt nhạc | 66 |
| Hình 4.16: Lựa chọn phương thức hẹn giờ..... | 67 |
| Hình 4.17: Tùy chỉnh tốc độ phát..... | 68 |
| Hình 4.18: Bộ điều khiển cường độ | 68 |
| Hình 4.19: Quá trình tìm kiếm bài hát | 69 |
| Hình 4.20: Tùy chọn thay đổi nền | 70 |
| Hình 4.21: Tùy chọn thay đổi ngôn ngữ | 71 |

TỔNG QUAN

Lý do chọn đề tài

Ở thời điểm hiện tại, việc các ứng dụng bị chi phối quá nhiều bởi quảng cáo là một vấn đề nhức nhối đối với người dùng phổ thông. Bản thân em đã từng gặp trường hợp đang nghe một bản nhạc yêu thích trên ứng dụng Spotify và bị ngắt ngang với các quảng cáo yêu cầu nâng cấp gói tài khoản hoặc bán hàng và nhiều dẫn đến trải nghiệm không tốt. Ngoài ra, đối với các ứng dụng nghe nhạc nổi tiếng như Spotify, iTune,...thì quyền hạn dành cho tài khoản miễn phí rất hạn chế như: só lần bỏ qua quảng cáo bị giới hạn, không thể tải nhạc, thậm chí còn giới hạn cả việc chuyển bài...Điều đó đưa người dùng vào quyết định lựa chọn mua những gói có quyền hạn cao hơn hoặc nghe từ các ứng dụng không chính thống và có khả năng bị theo dõi lợi dụng bởi những mã độc được cài trong các ứng dụng đó.

Đặc biệt các đối tượng như học sinh, sinh viên với mức thu nhập hạn chế thì việc lựa chọn mua các gói tài khoản trả phí sẽ trở nên rất khó khăn. Với cương vị một sinh viên Công nghệ thông tin có sự yêu thích với âm nhạc thì mục tiêu của em đối với đồ án lần này là tạo ra ứng dụng nghe nhạc di động có thể giải quyết được các vấn đề trên.

Mục tiêu nghiên cứu

Xây dựng ứng dụng nghe nhạc trực tuyến hoàn toàn miễn phí với đầy đủ tính năng nhằm mang lại trải nghiệm nghe nhạc tốt hơn cho người dùng.

Nghiên cứu hệ thống đề xuất nội dung giúp người dùng tìm kiếm thông tin một cách nhanh chóng và hiệu quả hơn.

Phạm vi chọn đề tài

Ứng dụng phát triển đặt nền tảng cho một dự án với kỳ vọng sẽ hoạt động trên nhiều nền tảng khác nhau cả ứng dụng web và di động. Để đảm bảo trải nghiệm người dùng nên em quyết định phát triển ở phía người dùng trước khi mở rộng ra ứng dụng

web cho quản lý dữ liệu bài hát, nghệ sĩ cũng như thông tin người dùng. Bên cạnh đó nghiên cứu các phương pháp tổ chức hệ thống, đảm bảo hệ thống luôn trong trạng thái dễ phát triển và bảo trì về sau.

Cấu trúc báo cáo đồ án tốt nghiệp

Chương 1: Cơ sở lý thuyết

Giới thiệu khái quát về những công cụ và công nghệ đã được sử dụng cho đồ án tốt nghiệp lần này. Bao gồm Flutter Framework, Python, Firebase Realtime Database,...

Chương 2: Hệ thống đề xuất nội dung

Làm rõ khái niệm đề xuất nội dung, những hình thức đề xuất nội dung phổ biến và cụ thể về kĩ thuật lọc theo nội dung (content-based filtering).

Chương 3: Phân tích thiết kế hệ thống

Mô hình hóa hệ thống thông qua các sơ đồ, bảng biểu và hình ảnh.

Chương 4: Xây dựng ứng dụng

Giới thiệu về tổng quan hệ thống và các chức năng quan trọng đối với ứng dụng.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1 Công nghệ sử dụng

Những công nghệ chính được sử dụng để xây dựng hệ thống lần này lần lượt là ngôn ngữ lập trình Dart (Flutter Framework) cho ứng dụng, ngôn ngữ lập trình Python cho hệ thống đề xuất nội dung và FRD cho việc lưu trữ dữ liệu.

1.1.1 Giới thiệu về Dart và Flutter Framework

Trước tiên, đến với ngôn ngữ lập trình Dart - nền tảng cho ứng dụng.

Dart là một ngôn ngữ lập trình đa năng mã nguồn mở ban đầu được phát triển bởi Google và sau đó được ECMA phê duyệt như một tiêu chuẩn.^[3]

Dart là một ngôn ngữ lập trình mới dành cho máy chủ cũng như trình duyệt. Được phát triển và giới thiệu bởi Google, Dart SDK đi kèm với trình biên dịch riêng - Dart VM. SDK cũng bao gồm tiện ích - dart2js, một trình chuyển tiếp tạo JavaScript tương đương với Dart Script.^[4]

Tiếp theo là Flutter Framework, công nghệ chính cho việc thiết kế ứng dụng.

Flutter là framework mã nguồn mở dùng để tạo ra các ứng dụng di động với chất lượng và hiệu suất cao trên các hệ điều hành di động - Android và iOS. Flutter cung cấp một bộ SDK đơn giản, mạnh mẽ, hiệu quả và dễ hiểu để viết ứng dụng di động bằng ngôn ngữ lập trình Dart.^[5]

Ngoài ra, ứng dụng Flutter còn có thể hoạt động trên nền tảng web và các hệ điều hành dành cho máy tính như Windows và MacOS. Nhờ vậy, framework này luôn là lựa chọn hàng đầu cho các hệ thống đa nền tảng.

1.1.2 Giới thiệu về Python

Python là một ngôn ngữ lập trình thông dịch, đa năng và dễ học, được phát triển vào cuối những năm 1980 bởi Guido van Rossum tại Hà Lan. Python giúp ích cho chúng ta khi làm việc với nhiều loại dữ liệu khác nhau nhờ có bộ thư viện rộng lớn và cộng đồng ngày càng đông đảo. [3]

Ngoài khả năng vượt trội trong việc xử lý dữ liệu, ngôn ngữ này cũng có thể được sử dụng để phát triển các ứng dụng web và phần mềm máy tính. Với đặc điểm là dễ tiếp cận nên Python ngày càng được sử dụng phổ biến và dần trở nên thịnh hành.

1.1.3 Giới thiệu về Firebase Realtime Database

Firebase Realtime Database là một cơ sở dữ liệu thời gian thực được lưu trữ trên đám mây của Google. Nó cho phép bạn lưu trữ và đồng bộ hóa dữ liệu giữa các ứng dụng và các thiết bị khác nhau. Firebase Realtime Database sử dụng mô hình dữ liệu JSON (dạng cấu trúc key : value) để lưu trữ và truy cập dữ liệu. Ngoài ra, cơ sở dữ liệu này cũng cung cấp các tính năng như xác thực người dùng, quản lý lỗi và phân tích thống kê.[6]

Firebase Realtime Database được sử dụng trong các ứng dụng đa nền tảng như Android, iOS và Web. Tất cả dữ liệu được lưu trữ nhanh và với bất kỳ một sự thay đổi dữ liệu nào đều có sự phản hồi ngay lập tức.

1.2 Tổng quan về Flutter

Với nhu cầu tạo ra một công nghệ có thể tương thích trên nhiều hệ điều hành thì Flutter - một framework đơn giản và hiệu suất cao dựa trên ngôn ngữ lập trình Dart, cung cấp đầy đủ mọi yêu cầu bằng cách hiển thị giao diện người dùng trực tiếp trong khuôn mẫu của hệ điều hành thay vì thông qua các công nghệ gốc như Java với Android hay Objective-C với iOS.

Flutter cung cấp nhiều tiện ích con sẵn sàng sử dụng để tạo một ứng dụng hiện đại mới. Các widget này được tối ưu hóa cho môi trường di động và thiết kế ứng dụng bằng widget cũng đơn giản như thiết kế HTML.

1.2.1 Tiện ích con - Widget

Ứng dụng Flutter tự nó là một widget. Các widget Flutter cũng hỗ trợ các hình ảnh động và cử chỉ. Logic ứng dụng dựa trên lập trình phản ứng, Widget có thể xử lý các thay đổi theo thời gian thực kể cả đối với nội dung tĩnh.[5] Bằng cách thay đổi trạng thái của tiện ích, Flutter sẽ tự động so sánh trạng thái của widget (cũ và mới) và chỉ hiển thị widget với những thay đổi cần thiết thay vì hiển thị lại toàn bộ.

1.2.2 Hỗ trợ tương tác - Gesture

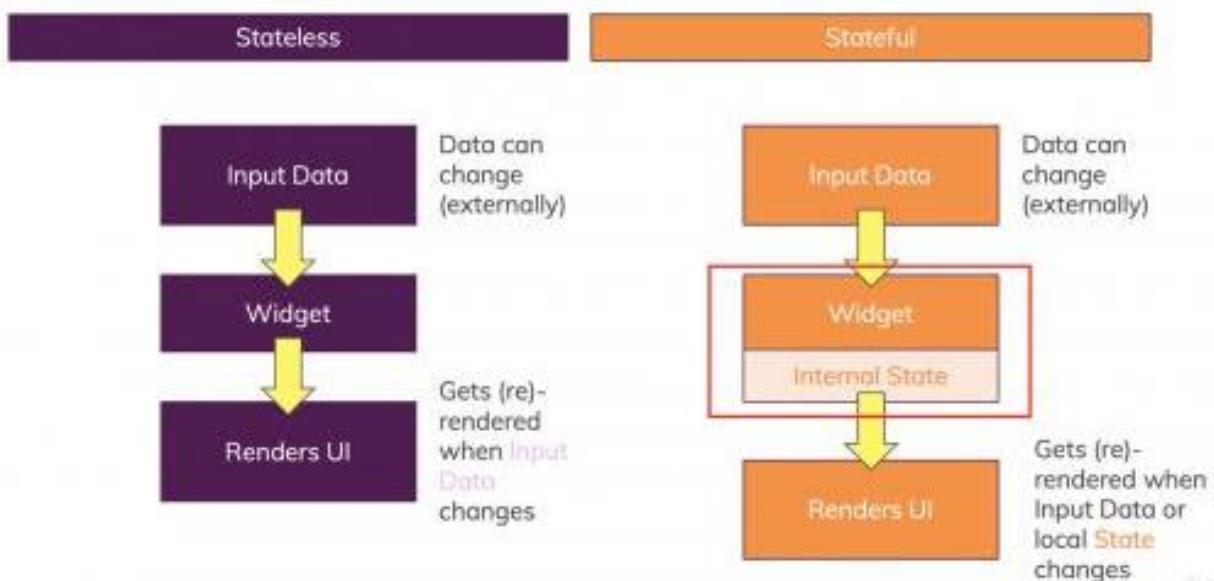
Các widget trong Flutter hỗ trợ tương tác thông qua một widget đặc biệt có tên là GestureDetector - một tiện ích con vô hình có khả năng kiểm soát các tương tác của người dùng như chạm, kéo, v.v. của tiện ích con chứa trong nó. Nhiều widget gốc của Flutter hỗ trợ tương tác thông qua việc sử dụng GestureDetector.[5] Lập trình viên cũng có thể kết hợp các tính năng tương tác vào tiện ích hiện có bằng cách đóng gói nó với GestureDetector.

1.2.3 Quản lý trạng thái - State

Các widget Flutter hỗ trợ quản lý trạng thái bằng cách cung cấp một widget đặc biệt mang tên StatefulWidget. Tiện ích con cần được bắt nguồn từ StatefulWidget để hỗ trợ duy trì trạng thái và tất cả các widget khác phải bắt nguồn từ StatelessWidget.[5] Điều

này tương tự như ReactJs và StatefulWidget sẽ được tự động hiển thị bất cứ khi nào trạng thái bên trong bị thay đổi. Kết xuất được tối ưu hóa bằng cách tìm ra sự khác biệt giữa giao diện người dùng widget cũ và mới và chỉ hiển thị những thay đổi cần thiết.

Stateless vs Stateful



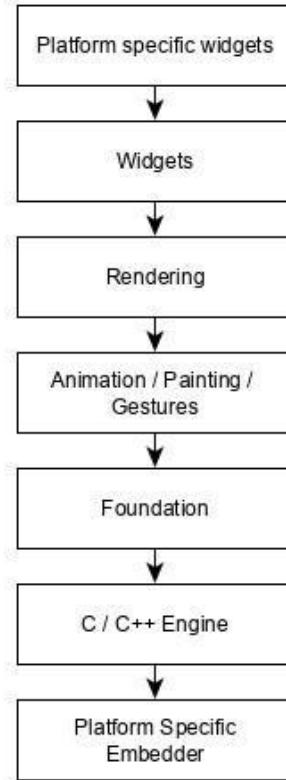
Hình 1.1: Sự khác biệt giữa Stateful và Stateless Widget

1.2.4 Các lớp con trong ứng dụng Flutter

Khái niệm quan trọng nhất của khung Flutter là khung được nhóm thành nhiều loại về mức độ phức tạp và được sắp xếp rõ ràng theo các lớp giảm dần độ phức tạp. Một lớp được xây dựng bằng cách sử dụng lớp cấp tiếp theo ngay sau nó. Lớp trên cùng là các widget dành riêng cho Android và iOS. Lớp tiếp theo có tất cả các widget gốc. Lớp tiếp theo là lớp Rendering, đây là một thành phần trình kết xuất cấp thấp và hiển thị mọi thứ trong ứng dụng Flutter. Các lớp dưới dành riêng cho việc phát triển ứng dụng.

Trong Flutter, mọi thứ đều là một widget và một widget phức tạp bao gồm các widget đã có sẵn. Các tính năng tương tác có thể được kết hợp bất cứ khi nào cần thiết bằng widget GestureDetector. Trạng thái của một widget có thể được duy trì bất cứ khi nào

cần thiết bằng cách sử dụng widget StatelessWidget. Flutter cung cấp thiết kế phân lớp để bất kỳ lớp nào cũng có thể được lập trình tùy thuộc vào mức độ phức tạp của nhiệm vụ.



Hình 1.2: Tổng quan chung về một lớp trong Flutter

1.2.5 Ưu/nhược điểm của Flutter

Flutter đi kèm với các widget đẹp và có thể tùy chỉnh để ứng dụng có hiệu suất cao và tính năng nổi bật. Nó đáp ứng tất cả các yêu cầu của khách hàng. Bên cạnh những điều này, Flutter còn có nhiều ưu điểm khác được đề cập dưới đây:

- Dart có một kho phần mềm lớn cho phép bạn mở rộng khả năng của ứng dụng của mình.
- Các nhà phát triển chỉ cần viết một mã duy nhất cho cả hai ứng dụng (cả nền tảng Android và iOS). Flutter cũng có thể được mở rộng sang một nền tảng khác trong tương lai.

- Flutter ít kiểm thử hơn. Do cơ sở mã duy nhất của nó là đủ nếu chúng ta viết các test case một lần cho cả hai nền tảng.
- Sự đơn giản của Flutter khiến nó trở thành một ứng cử viên sáng giá để phát triển ứng dụng. Khả năng tùy chỉnh và mở rộng làm cho framework này trở nên mạnh mẽ hơn.
- Với Flutter, các nhà phát triển có toàn quyền kiểm soát các widget và bộ cục của chúng.
- Flutter cung cấp các công cụ tuyệt vời dành cho nhà phát triển, với khả năng hot reload đáng kinh ngạc.

Mặc dù có nhiều ưu điểm, nhưng Flutter cũng có những nhược điểm sau:

- Vì được phát triển trên ngôn ngữ Dart nên khi bắt đầu sử dụng các nhà phát triển phải học ngôn ngữ mới (mặc dù rất dễ học).
- Framework hiện đại cố gắng tách logic và giao diện càng nhiều càng tốt, nhưng trong Flutter, giao diện và logic được trộn lẫn với nhau. Chúng ta có thể khắc phục điều này bằng cách sử dụng mã hóa thông minh và sử dụng mô-đun cấp cao để phân tách UI và phần mã logic.
- Flutter là framework mới để tạo ứng dụng di động. Các nhà phát triển đang gặp khó khăn trong việc lựa chọn công cụ phát triển phù hợp.

1.3 Tổng quan về Python, Flask và Jupyter Notebook

Ngôn ngữ lập trình Python được dùng trong hệ thống đề xuất nội dung, hệ thống sẽ trả về dữ liệu bằng việc sử dụng API lấy nền tảng từ Flask framework. Ngoài ra, em sử dụng Jupyter Notebook cho việc phát triển và thử nghiệm cho kĩ thuật.

Trong mục này, chúng ta cùng lần lượt tìm hiểu về 3 công nghệ kể trên.

1.3.1 Lịch sử của Python

Python được phát triển bởi Guido van Rossum vào cuối thập niên 80 và đầu thập niên 90 tại CWI (Centrum voor Wiskunde en Informatica – Trung tâm Toán-Tin học) ở Hà Lan và được phát hành lần đầu tiên vào tháng 2 năm 1991.

Python có nguồn gốc từ nhiều ngôn ngữ khác, bao gồm ABC, Modula-3, C, C++, Algol-68, SmallTalk và Unix shell và các ngôn ngữ script khác.

Python hiện được duy trì bởi một nhóm các nhà phát triển tại CWI và Guido van Rossum vẫn giữ một vai trò quan trọng trong việc chỉ đạo.

Sự phát triển Python đến nay có thể chia làm các giai đoạn:

Python 1: bao gồm các bản phát hành 1.x. Giai đoạn này kéo dài từ đầu đến cuối thập niên 1990. Từ năm 1990 đến 1995, Guido làm việc tại CWI tại Amsterdam.

Python 2: bao gồm các bản phát hành từ 2.0 đến 2.7. Giai đoạn này kéo dài từ năm 2000 đến năm 2010.

Python 3: bao gồm các bản phát hành từ 3.0 trở đi. Giai đoạn này kéo dài từ năm 2008 đến nay.

1.3.2 Các tính năng của Python

Những đặc điểm nổi bật của Python bao gồm :

Là ngôn ngữ lập trình bậc cao: Python được xem là một ngôn ngữ lập trình bậc cao do có cấu trúc rõ ràng và có khả năng giải quyết các vấn đề phức tạp. Python có cú pháp rõ ràng, đơn giản và quá trình sử dụng cơ chế cấp phát bộ nhớ hoàn toàn tự động.

Có thể xử lý được nhiều tác vụ nhờ việc hỗ trợ nhiều mô hình lập trình khác nhau như lập trình hướng đối tượng, lập trình hàm và lập trình hướng thủ tục.

Giúp đơn giản hóa việc lập trình khi có nhiều thư viện và framework hỗ trợ cho lập trình viên giải quyết các vấn đề phức tạp một cách dễ dàng và còn có thể chạy trên nhiều nền tảng, nhiều hệ điều hành khác nhau.

Các thư viện nổi bật[2]:

NumPy: Thư viện hỗ trợ tính toán khoa học và toán học. Là thư viện Python được sử dụng để xử lý các mảng và ma trận số. NumPy cung cấp các đối tượng mảng đa chiều hiệu suất cao và các công cụ để làm việc với các mảng này. Thư viện này được sử dụng rộng rãi trong khoa học dữ liệu và tính toán khoa học.

Pandas: Thư viện hỗ trợ xử lý dữ liệu. Pandas được thiết kế để làm việc dễ dàng và trực quan với dữ liệu có cấu trúc như các bảng dữ liệu (dataframe) và chuỗi thời gian (time series). Pandas cung cấp các công cụ để đọc và ghi dữ liệu từ nhiều định dạng tệp khác nhau, bao gồm CSV, Excel, SQL và nhiều định dạng khác.

Matplotlib: Thư viện hỗ trợ vẽ đồ thị. Là một thư viện trực quan hóa dữ liệu phổ biến trong ngôn ngữ lập trình Python, một công cụ mạnh mẽ để tạo ra các biểu đồ và đồ thị chất lượng cao, giúp người dùng hiển thị dữ liệu một cách trực quan và dễ hiểu. Matplotlib có thể được sử dụng để tạo ra nhiều loại biểu đồ khác nhau, bao gồm biểu đồ đường, biểu đồ cột, biểu đồ phân tán và nhiều hơn nữa.

Scikit-learn: Thư viện hỗ trợ machine learning. Cung cấp các công cụ để xử lý dữ liệu và thực hiện các thuật toán học máy khác nhau, bao gồm phân loại, hồi quy và phân cụm. Scikit-learn còn có thể đánh giá và tinh chỉnh các tham số của mô hình.

TensorFlow: Thư viện hỗ trợ deep learning. Là một thư viện học sâu mã nguồn mở được phát triển bởi Google. Với các công cụ để xây dựng và huấn luyện các mô hình học sâu khác nhau, bao gồm mạng nơ-ron sâu và mạng nơ-ron tích chập. Tương tự như Scikit-learn, TensorFlow cũng cung cấp các công cụ để đánh giá mô hình và tinh chỉnh các tham số của mô hình.

1.3.3 Flask

Flask là một framework web của Python, được phát triển bởi Armin Ronacher và được coi là một microframework bởi vì nó không yêu cầu các công cụ hoặc thư viện cụ thể. Flask phát triển dựa trên bộ công cụ Werkzeug WSGI và công cụ mẫu Jinja2. Cả hai đều là các dự án của Pocco.

Flask giúp tạo các RESTful API một cách đơn giản và nhanh chóng, rất phù hợp cho người mới bắt đầu lập trình Python. Flask cung cấp các tính năng không khác gì các framework web khác như server được tích hợp sẵn, tính năng debug, hỗ trợ unit testing, nhiều extension đa năng được phát triển bởi cộng đồng.

Trong đồ án này, Flask được sử dụng chủ yếu cho việc tạo API trao đổi dữ liệu từ hệ thống để xuất nội dung đến giao diện ứng dụng di động.

1.3.4 Jupyter Notebook

Jupyter Notebook là một ứng dụng web mã nguồn mở cho phép bạn tạo và chia sẻ các tài liệu có chứa mã, chương trình, hình ảnh và văn bản giải thích.[2] Ứng dụng này được sử dụng rộng rãi trong cộng đồng khoa học dữ liệu, AI cũng như ML.

Một số tính năng nổi bật của Jupyter Notebook bao gồm:

- Cho phép tạo các tài liệu có chứa mã, phương trình, hình ảnh và cả văn bản giải thích, giúp cho việc trình bày thuật toán trở nên trực quan và dễ hiểu hơn.
- Tính tiện dụng của Jupyter Notebook thể hiện ở chỗ nó hỗ trợ nhiều ngôn ngữ lập trình như Python, R, Julia... và có thể chạy trực tiếp các đoạn mã trong đó.
- Các tính năng hữu ích khác như: cho phép tạo các widget để tương tác với dữ liệu. Có thể xuất ra các tài liệu dưới nhiều định dạng như HTML, PDF...

1.4 Tổng quan về Firebase Realtime Database

Mục này chứa một số thông tin cơ bản về Firebase nói chung và FRD - cơ sở dữ liệu chính và duy nhất trong hệ thống. Gồm có khái niệm, các tính năng nổi bật và những dịch vụ khác mà Firebase cung cấp.

1.4.1 Khái niệm

Firebase là một nền tảng giúp phát triển các ứng dụng di động và web. Firebase cung cấp cho lập trình viên các công cụ để tạo ra các ứng dụng chất lượng cao mà không cần phải lo lắng về việc quản lý máy chủ hoặc cơ sở dữ liệu. Firebase được Google sở hữu và quản lý.

Realtime Database là cơ sở dữ liệu lưu trữ trên mây. Dữ liệu được lưu trữ và đồng bộ hóa theo thời gian thực với mỗi kết nối. Khi xây dựng ứng dụng đa nền tảng với iOS, Android, và javascript SDK, tất cả các người dùng chia sẻ một phiên bản Realtime Database và tự động tiếp nhận các thay đổi với dữ liệu mới nhất.

1.4.2 Các tính năng tốt và nhược điểm của Realtime Database

Như khái niệm đã nêu ở trên thì lưu trữ dữ liệu dưới dạng JSON và đồng bộ hóa theo thời gian thực với tất cả các máy khách kết nối là chức năng quan trọng và cần thiết nhất khi một lập trình viên lựa chọn sử dụng nó cho ứng dụng của họ.

Với việc sử dụng secret key để truy cập vào cơ sở dữ liệu từ mã phía máy khách một cách an toàn và nhanh chóng. Mặt khác, để cai thiện trải nghiệm người dùng thì dữ liệu được lưu trữ cục bộ ngay cả khi không có kết nối mạng.

FRD cho phép nhập/xuất dữ liệu ra file JSON giúp cho việc mở rộng dữ liệu trở nên dễ dàng. Việc sử dụng định dạng JSON làm cơ sở dữ liệu này ngày càng được sử dụng rộng rãi.

Tuy nhiên, FRD chưa thực sự là một lựa chọn hoàn hảo bởi những lý do sau:

Cơ sở dữ liệu của Firebase được tổ chức theo kiểu cây, không phải là kiểu bảng, nên những ai quen với SQL có thể gặp khó khăn trong việc truy vấn và thiết kế dữ liệu.

Cơ sở dữ liệu của Firebase có dung lượng giới hạn 10GB cho gói miễn phí và 10TB cho gói trả phí. Nếu cần lưu trữ dữ liệu lớn hơn, bạn có thể phải tìm giải pháp khác.

Cơ sở dữ liệu của Firebase có độ bảo mật còn yếu, do tất cả dữ liệu được lưu trữ dưới dạng JSON. JSON là một định dạng nhẹ, không phụ thuộc vào ngôn ngữ và dễ sử dụng, nhưng cũng dễ bị tấn công và đánh cắp.

Cơ sở dữ liệu của Firebase có thể gặp vấn đề về hiệu năng khi có nhiều kết nối đồng thời hoặc khi có nhiều thao tác đọc/ghi trên cùng một node.

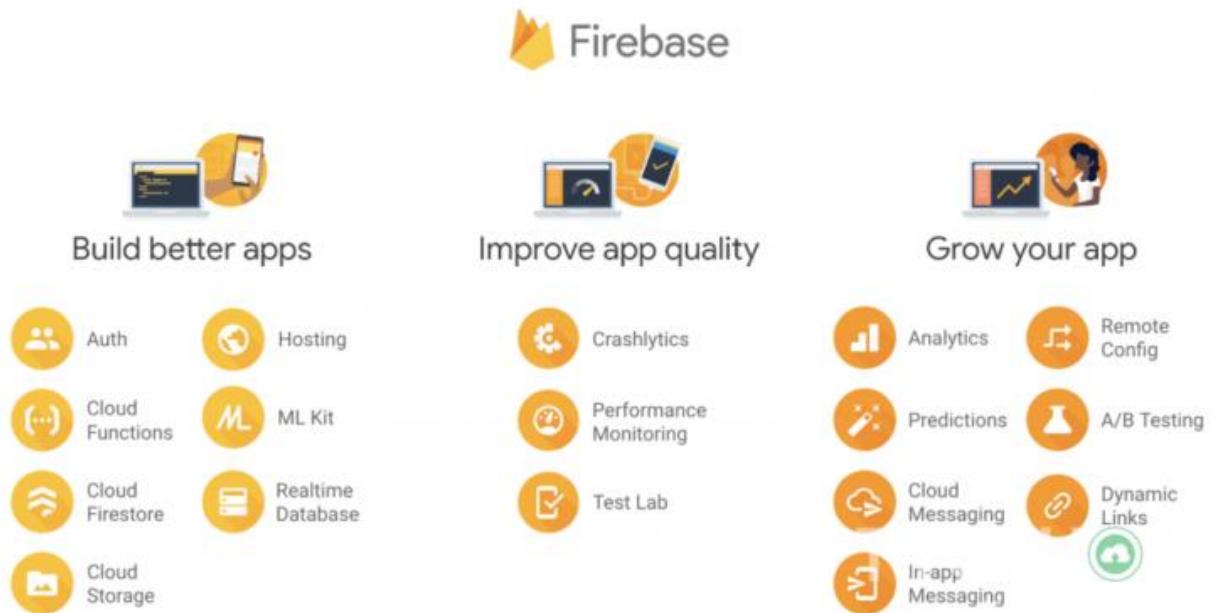
1.4.3 Các dịch vụ khác của Firebase

Ngoài Realtime Database thì Firebase còn có rất nhiều dịch vụ hữu ích cho việc phát triển phần mềm nói chung như:

- Cloud Firestore là một cơ sở dữ liệu linh hoạt, có thể mở rộng để phát triển thiết bị di động, web và máy chủ từ Firebase và Google Cloud Platform.
- Firebase Remote Config lưu trữ các cặp key-value do nhà phát triển chỉ định để thay đổi hành vi và giao diện ứng dụng của bạn mà không yêu cầu người dùng tải xuống bản cập nhật.

- Firebase Hosting lưu trữ HTML, CSS và JavaScript cho trang web của bạn cũng như các nội dung khác do nhà phát triển cung cấp như đồ họa, phông chữ và biểu tượng.
- Cloud Storage lưu trữ các tệp như hình ảnh, video và âm thanh cũng như nội dung khác do người dùng tạo.
- ML Kit chứa các mẫu máy học căn bản có thể được dùng trong việc phát triển ứng dụng. Ngoài ra, ML Kit còn cung cấp tính năng Hosting mẫu máy học do các lập trình viên sáng tạo ra.[6]

Ngoài ra còn rất nhiều các dịch vụ cực kì hữu ích khác được liệt kê ở hình bên dưới.



Hình 1.3: Những dịch vụ mà Firebase đang cung cấp

1.5 Tổng quan về Machine Learning

Ngày nay, học máy là một phần không thể thiếu trong nhiều ứng dụng thương mại và dự án nghiên cứu đối với các lĩnh vực khác nhau trong xã hội. Học máy hiện nay không chỉ được áp dụng bởi các công ty lớn với đội ngũ nghiên cứu sâu rộng mà đang dần thâm nhập vào các ứng dụng hàng ngày của mọi người.

1.5.1 Giới thiệu

Học máy là trích xuất kiến thức từ dữ liệu. Đây là một lĩnh vực nghiên cứu kết hợp giữa thống kê, trí tuệ nhân tạo và khoa học máy tính, nó còn được gọi là phân tích dự đoán hoặc học thống kê. Việc áp dụng các phương pháp học máy trong những năm gần đây đã trở nên phổ biến trong cuộc sống hàng ngày. Từ các đề xuất tự động về phim nên xem, thức ăn nên đặt hoặc mua sản phẩm nào, đến đài phát thanh trực tuyến được cá nhân hóa và nhận dạng bạn bè trong ảnh của bạn, nhiều trang web và thiết bị hiện đại có thuật toán máy học làm cốt lõi. Khi bạn xem một trang web phức tạp như Facebook, Amazon hoặc Netflix, rất có thể mọi phần của trang web đều chứa nhiều mô hình máy học.[2]

Ngoài các ứng dụng thương mại, học máy đã có ảnh hưởng to lớn đến cách thức nghiên cứu dựa trên dữ liệu được thực hiện ngày nay. Các công cụ được giới thiệu trong cuốn sách này đã được áp dụng cho các vấn đề khoa học đa dạng như tìm hiểu các ngôi sao, tìm kiếm các hành tinh xa xôi, khám phá các hạt mới, phân tích trình tự DNA và cung cấp các phương pháp điều trị ung thư được cá nhân hóa.

Tuy nhiên, ứng dụng của bạn không cần phải có quy mô lớn hoặc thay đổi thế giới như những ví dụ này để hưởng lợi từ học máy. Trong chương này, chúng tôi sẽ giải thích lý do tại sao học máy trở nên phổ biến và thảo luận về những loại vấn đề nào có thể được giải quyết bằng học máy.[2] Sau đó, chúng tôi sẽ chỉ cho bạn cách xây dựng mô hình máy học đầu tiên của bạn, đồng thời giới thiệu các khái niệm quan trọng trong quá trình thực hiện.

1.5.2 Lý do chọn Machine Learning

Trong những ngày đầu của các ứng dụng “thông minh”, nhiều hệ thống đã sử dụng các quy tắc được mã hóa thủ công với các khối lệnh If-else để xử lý dữ liệu hoặc điều chỉnh theo đầu vào của người dùng. Hãy nghĩ về một bộ lọc thư rác có nhiệm vụ di chuyển các email đến thích hợp vào một thư mục thư rác. Bạn có thể tạo một danh sách đen các từ sẽ dẫn đến việc email bị đánh dấu là thư rác. Đây sẽ là một ví dụ về việc sử dụng hệ thống quy tắc do chuyên gia thiết kế để thiết kế một ứng dụng “thông minh”. Các quy tắc quyết định được tạo thủ công là khả thi đối với một số ứng dụng, đặc biệt là những ứng dụng mà con người hiểu rõ về quy trình để lập mô hình.[2]

Tuy nhiên, sử dụng các quy tắc được mã hóa thủ công để đưa ra quyết định có hai nhược điểm chính:

- Logic cần thiết để đưa ra quyết định cụ thể cho một lĩnh vực và nhiệm vụ. Thay đổi tác vụ dù chỉ một chút có thể yêu cầu viết lại toàn bộ hệ thống.
- Việc thiết kế các quy tắc đòi hỏi một chuyên gia về con người phải có hiểu biết sâu sắc về cách một quyết định nên được đưa ra.

Một ví dụ về việc phương pháp mã hóa thủ công này sẽ thất bại trong việc phát hiện khuôn mặt trong hình ảnh. Ngày nay, mọi điện thoại thông minh đều có thể nhận diện khuôn mặt trong ảnh. Tuy nhiên, nhận diện khuôn mặt là một vấn đề chưa được giải quyết cho đến tận năm 2001. Vấn đề chính là cách mà các pixel (tạo nên hình ảnh trong máy tính) được máy tính “nhận thức” rất khác so với cách con người nhận thức. một khuôn mặt. Sự khác biệt về cách thể hiện này khiến con người về cơ bản không thể đưa ra một bộ quy tắc tốt để mô tả những gì cấu thành một khuôn mặt trong hình ảnh kỹ thuật số.

Tuy nhiên, khi sử dụng máy học, chỉ cần trình bày một chương trình với một bộ sưu tập lớn các hình ảnh khuôn mặt là đủ để thuật toán xác định những đặc điểm cần thiết để nhận dạng khuôn mặt.

1.5.3 Các vấn đề mà Machine Learning có thể giải quyết

Các loại thuật toán học máy thành công nhất là những thuật toán tự động hóa quy trình ra quyết định bằng cách khái quát hóa từ các ví dụ đã biết. Trong cài đặt này, được gọi là học có giám sát, người dùng cung cấp cho thuật toán các cặp đầu vào và đầu ra mong muốn, và thuật toán sẽ tìm cách tạo ra đầu ra mong muốn với một đầu vào. Đặc biệt, thuật toán có thể tạo đầu ra cho đầu vào mà nó chưa từng thấy trước đây mà không cần bất kỳ sự trợ giúp nào từ con người. Quay trở lại ví dụ về phân loại thư rác, sử dụng máy học, người dùng cung cấp cho thuật toán một số lượng lớn email (là đầu vào), cùng với thông tin về việc liệu có bất kỳ email nào trong số này là thư rác hay không (là đầu ra mong muốn). Đưa ra một email mới, thuật toán sau đó sẽ đưa ra dự đoán liệu email mới có phải là thư rác hay không.[2]

Các thuật toán học máy học từ các cặp đầu vào/đầu ra được gọi là thuật toán học có giám sát vì một “giáo viên” cung cấp sự giám sát cho các thuật toán dưới dạng đầu ra mong muốn cho mỗi ví dụ mà chúng học được. Mặc dù việc tạo tập dữ liệu đầu vào và đầu ra thường là một quy trình thủ công tốn nhiều công sức, nhưng các thuật toán học có giám sát được hiểu rõ và hiệu suất của chúng rất dễ đo lường. Nếu ứng dụng của bạn có thể được xây dựng như một bài toán học có giám sát, và bạn có thể

Ví dụ về các nhiệm vụ học máy được giám sát bao gồm:

Xác định mã zip từ các chữ số viết tay trên phong bì

Ở đây, đầu vào là bản quét chữ viết tay và đầu ra mong muốn là các chữ số thực trong mã zip. Để tạo tập dữ liệu để xây dựng mô hình học máy, bạn cần thu thập nhiều phong bì. Sau đó, bạn có thể tự đọc mã zip và lưu trữ các chữ số theo kết quả mong muốn của mình.

Xác định xem một khối u lành tính dựa trên hình ảnh y tế

Ở đây, đầu vào là hình ảnh và đầu ra là liệu khối u có lành tính hay không. Để tạo tập dữ liệu để xây dựng mô hình, bạn cần có cơ sở dữ liệu về hình ảnh y tế. Bạn cũng

cần ý kiến chuyên gia, vì vậy bác sĩ cần xem xét tất cả các hình ảnh và quyết định khối u nào lành tính và khối u nào không. Thậm chí có thể cần phải thực hiện chẩn đoán bổ sung ngoài nội dung của hình ảnh để xác định xem khối u trong hình ảnh có phải là ung thư hay không.

Phát hiện hoạt động gian lận trong giao dịch thẻ tín dụng

Ở đây, đầu vào là bản ghi giao dịch thẻ tín dụng và đầu ra là liệu giao dịch đó có khả năng gian lận hay không. Giả sử rằng bạn là thực thể phân phối thẻ tín dụng, việc thu thập tập dữ liệu có nghĩa là lưu trữ tất cả các giao dịch và ghi lại nếu người dùng báo cáo bất kỳ giao dịch nào là gian lận.

Một điều thú vị cần lưu ý về những ví dụ này là mặc dù đầu vào và đầu ra trông khá đơn giản, nhưng quy trình thu thập dữ liệu cho ba nhiệm vụ này lại khác nhau rất nhiều. Trong khi đọc phong bì tốn nhiều công sức, nó rất dễ dàng và rẻ tiền. Mặt khác, việc chụp ảnh và chẩn đoán y tế không chỉ đòi hỏi máy móc đắt tiền mà còn cần kiến thức chuyên môn hiếm và đắt tiền, chưa kể đến những lo ngại về đạo đức và các vấn đề về quyền riêng tư. Trong ví dụ về phát hiện gian lận thẻ tín dụng, việc thu thập dữ liệu đơn giản hơn nhiều. Khách hàng của bạn sẽ cung cấp cho bạn đầu ra mong muốn, vì họ sẽ báo cáo gian lận. Tất cả những gì bạn phải làm để có được các cặp đầu vào/đầu ra của hoạt động gian lận và không gian lận là chờ đợi. Thuật toán không giám sát là loại thuật toán khác mà chúng tôi sẽ trình bày trong cuốn sách này. Trong học tập không giám sát, chỉ dữ liệu đầu vào được biết và không có dữ liệu đầu ra đã biết nào được cung cấp cho thuật toán. Mặc dù có nhiều ứng dụng thành công của các phương pháp này, nhưng chúng thường khó hiểu và khó đánh giá hơn.[2]

Ví dụ về học tập không giám sát bao gồm:

Phân khúc khách hàng thành các nhóm có cùng sở thích

Với một tập hợp các bản ghi khách hàng, bạn có thể muốn xác định những khách hàng nào giống nhau và liệu có những nhóm khách hàng có sở thích tương tự hay

không. Đối với một trang web mua sắm, đây có thể là “cha mẹ”, “một sách” hoặc “game thủ”. Bởi vì bạn không biết trước những nhóm này có thể là gì, hoặc thậm chí có bao nhiêu nhóm, nên bạn không biết đầu ra.

Phát hiện các kiểu truy cập bất thường vào một trang web

Để xác định tình trạng lạm dụng hoặc lỗi, việc tìm các mẫu truy cập khác với tiêu chuẩn thường rất hữu ích. Mỗi mẫu bất thường có thể rất khác nhau và bạn có thể không có bất kỳ trường hợp hành vi bất thường nào được ghi lại. Bởi vì trong ví dụ này, bạn chỉ quan sát lưu lượng truy cập và bạn không biết điều gì cấu thành hành vi bình thường và bất thường, nên đây là sự cố không được giám sát.

Đối với cả nhiệm vụ học tập có giám sát và không giám sát, điều quan trọng là phải có một bản trình bày dữ liệu đầu vào mà máy tính có thể hiểu được. Thường thì sẽ rất hữu ích nếu coi dữ liệu của bạn là một bảng. Mỗi điểm dữ liệu mà bạn muốn lý giải (từng email, từng khách hàng, từng giao dịch) là một hàng và mỗi thuộc tính mô tả điểm dữ liệu đó (ví dụ: tuổi của khách hàng hoặc số tiền hoặc địa điểm của giao dịch) là một cột. Bạn có thể mô tả người dùng theo độ tuổi, giới tính, thời điểm họ tạo tài khoản và tần suất họ mua hàng từ cửa hàng trực tuyến của bạn. Bạn có thể mô tả hình ảnh của một khối u bằng các giá trị thang độ xám của mỗi pixel hoặc có thể bằng cách sử dụng kích thước, hình dạng và màu sắc của khối u.

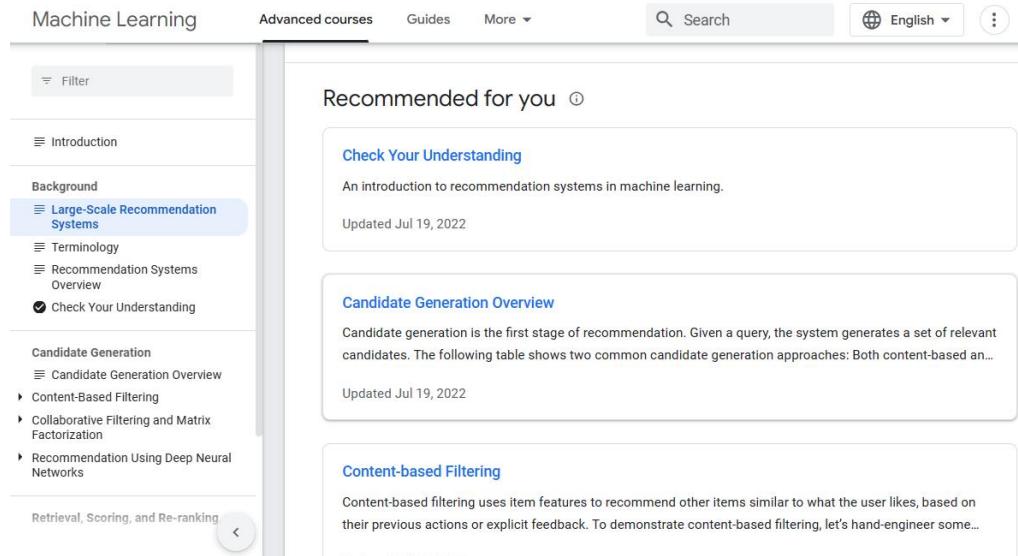
CHƯƠNG 2. HỆ THỐNG ĐỀ XUẤT NỘI DUNG

2.1 Giới thiệu

Hệ thống đề xuất nội dung - RS là một dạng con của hệ thống lọc thông tin (information filtering), nó được sử dụng để dự đoán theo sở thích (preferences) hay xếp hạng (rating) mà người dùng có thể dành cho một mục thông tin (item) nào đó mà họ chưa xem xét tới trong quá khứ (item có thể là bài hát, bộ phim, đoạn video clip, sách, bài báo...).

Ví dụ, trong các ứng dụng nghe nhạc như Spotify, iTunes, nhằm tối ưu hóa trải nghiệm nghe nhạc của người dùng (user), người ta quan tâm đến việc những user nào đã ‘yêu thích’ những bài hát (item) nào bằng cách dựa vào dữ liệu quá khứ của họ (dữ liệu này có thể là xếp hạng mà người dùng đã bình chọn cho bài hát, số lượt nghe bài hát, thời lượng nghe...) [7] từ đó hệ thống sẽ dự đoán được người dùng có thể thích bài hát nào và đưa ra những gợi ý phù hợp cho họ.

Hiện tại RS được ứng dụng khá thành công trong nhiều lĩnh vực khác như: gợi ý phim ảnh trên hệ thống của Netflix, gợi ý các video clip trên hệ thống của YouTube; trong giáo dục và đào tạo RS được dùng để gợi ý nguồn tài nguyên học tập như sách, bài báo, địa chỉ web,... cho người học. Dưới đây là hình ảnh minh họa về hệ thống đề xuất của Firebase đã gợi ý những bài viết tương tự với bài viết mà user đang đọc.



Hình 2.1: Ví dụ về hệ thống đề xuất nội dung

Hệ thống đề xuất nội dung không chỉ đơn thuần là một dạng Hệ thống thông tin mà nó còn là cả một lĩnh vực nghiên cứu hiện đang rất được các nhà khoa học quan tâm. Kể từ năm 2007 đến nay, hàng năm đều có hội thảo chuyên về hệ thống gợi ý của ACM (ACM RecSys) cũng như các tiểu bang dành riêng cho RS trong các hội nghị lớn khác như ACM KDD, ACM CIKM,..

2.2 Các khái niệm chính

Trước khi đi sâu vào chi tiết kỹ thuật hãy cùng tìm hiểu các khái niệm chính như các đối tượng user - item, thông tin phản hồi (feedback) và các dạng chính của hệ thống.

2.2.1 User - Item

Trong RS, thông thường người ta quan tâm đến ba thông tin chính là người dùng (user), mục tin (item, item có thể là sản phẩm, bộ phim, bài hát, bài báo,... tùy hệ thống), và phản hồi (feedback) của người dùng trên mục tin đó (thường là các xếp hạng/đánh giá – rating biểu diễn mức độ thích/quan tâm của họ). Các thông tin này được biểu diễn thông qua một ma trận như trong Hình 2.2. Ở đó, mỗi dòng là một user, mỗi cột là một item, và mỗi ô là một giá trị phản hồi biểu diễn “mức độ thích” của user trên item tương ứng.[8] Các ô có giá trị là những item mà các user đã xếp hạng trong quá khứ.

Mặt khác, có thể thấy là mỗi user chỉ xếp hạng cho một vài item trong quá khứ, do vậy có rất nhiều ô trống trong ma trận này – còn gọi là ma trận thưa(sparse matrix).

| | | Items | | | | | |
|-------|-----|-------|---|-----|-----|-----|-----|
| | | 1 | 2 | ... | i | ... | m |
| Users | 1 | 5 | 3 | | 1 | 2 | |
| | 2 | | 2 | | | | 4 |
| | : | | | 5 | | | |
| | u | 3 | 4 | ? | 2 | 1 | |
| | : | | | | | 4 | |
| | n | | | 3 | 2 | | |

Hình 2.2: Ma trận biểu diễn dữ liệu trong RS (user-item-rating matrix)

Nhiệm vụ chính của RS là dựa vào các ô đã có giá trị trong ma trận trên tức là dữ liệu thu được từ quá khứ, thông qua mô hình đã được xây dựng và dự đoán các ô còn trống của user hiện hành, sau đó sắp xếp kết quả dự đoán từ cao xuống thấp và chọn ra Top-N items trong danh sách, từ đó gợi ý chúng cho người dùng.

2.2.2 Thông tin phản hồi và hai dạng bài toán chính trong RS

Trong RS, giá trị phản hồi (feedback) r_{ui} của mỗi người dùng trên mục tin sẽ được ghi nhận lại để làm cơ sở cho việc dự đoán các giá trị kế tiếp. Tùy theo hệ thống mà giá trị này sẽ có ý nghĩa khác nhau, ví dụ nó có thể dùng để đo độ “phù hợp” hay “mức độ thích” thường là các đánh giá trên các sản phẩm trong các hệ thống thương mại điện tử hay “năng lực/kết quả thực hiện” của người dùng trong các hệ thống e-learning. Trong hệ thống lần này thông được lựa chọn để ghi nhận đó chính là số lượt nghe bài hát.

Giá trị r_{ui} có thể được xác định một cách tường minh như thông qua việc đánh giá/xếp hạng từ ★ đến ★★★★★; hay like (1) và dislike (0),... mà người dùng u đã bình chọn cho item i ; hoặc r_{ui} có thể được xác định một cách không tường minh thông qua số lần click chuột, thời gian mà u đã duyệt/xem i ,... những cách này hiện tại đang

gây nghi ngại về quyền riêng tư khi các hệ thống gần như đã theo dõi người dùng mà chưa được sự đồng ý của họ.

Có 2 dạng bài toán chính trong RS là *dự đoán xếp hạng* (*rating prediction*) của các hệ thống có phản hồi tường minh như đã trình bày ở trên và *dự đoán mục thông tin* (*item prediction/recommendation*) là việc xác định xác suất mà người dùng thích mục tin tương ứng.

2.3 Các kĩ thuật chính trong RS

Hiện tại, trong RS có rất nhiều giải thuật được đề xuất, tuy nhiên có thể gom lại vào trong các nhóm chính: nhóm giải thuật lọc theo nội dung (*content-based filtering*), nhóm giải thuật lọc cộng tác (*collaborative filtering*), nhóm giải thuật lai ghép (*hybrid filtering*) và nhóm giải thuật không cá nhân hóa (*non-personalization*).

2.3.1 Lọc cộng tác

Một cách tiếp cận để thiết kế các hệ thống recommender được sử dụng rộng rãi là lọc cộng tác. Các phương pháp lọc cộng tác dựa trên việc thu thập và phân tích một lượng lớn thông tin về hành vi, hoạt động hoặc sở thích của người dùng và dự đoán những gì người dùng sẽ thích dựa trên sự tương đồng của họ với người dùng khác. Một lợi thế quan trọng của phương pháp lọc cộng tác là nó không dựa vào nội dung phân tích máy và do đó nó có khả năng đề xuất chính xác các mục phức tạp như phim mà không yêu cầu “hiểu biết” về mục đó.^[3] Nhiều thuật toán đã được sử dụng để đo lường sự giống nhau của người dùng hoặc sự tương đồng về mặt hàng trong các hệ thống giới thiệu [8]. Ví dụ, cách tiếp cận hàng xóm gần nhất (*k-nearest neighbor*) và Pearson Correlation được Allen triển khai lần đầu tiên.

Lọc cộng tác dựa trên giả định rằng những người đã đồng ý trong quá khứ sẽ đồng ý trong tương lai và rằng họ sẽ thích các loại mặt hàng tương tự như họ thích trong quá khứ.

Khi xây dựng mô hình từ hành vi của người dùng, sự phân biệt thường được thực hiện giữa các hình thức thu thập dữ liệu rõ ràng và tiềm ẩn.

Ví dụ về thu thập dữ liệu rõ ràng bao gồm:

- Yêu cầu người dùng xếp hạng một mục trên thang trượt.
- Yêu cầu người dùng tìm kiếm.
- Yêu cầu người dùng xếp hạng một bộ sưu tập các mục từ yêu thích đến ít yêu thích nhất.
- Trình bày hai mục cho một người dùng và yêu cầu anh ta / cô ấy chọn một trong số chúng tốt hơn.
- Yêu cầu người dùng tạo danh sách các mục mà anh / cô ấy thích.

Ví dụ về thu thập dữ liệu ngầm bao gồm:

- Quan sát các mục mà người dùng xem trong cửa hàng trực tuyến.
- Phân tích thời gian xem mục / người dùng.
- Lưu giữ một bản ghi các mục mà người dùng mua trực tuyến.
- Lấy danh sách các mục mà người dùng đã nghe hoặc xem trên máy tính của họ.
- Phân tích mạng xã hội của người dùng và khám phá những lượt thích và không thích tương tự.

Hệ thống đề xuất so sánh dữ liệu đã thu thập với dữ liệu tương tự và khác nhau được thu thập từ những người khác và tính toán danh sách các mục được đề xuất cho người dùng. Một số ví dụ thương mại và phi thương mại được liệt kê trong bài viết về các hệ thống lọc cộng tác .

Một trong những ví dụ nổi tiếng nhất về lọc cộng tác là lọc cộng tác theo từng mục (những người mua x cũng mua y), một thuật toán được phổ biến rộng rãi bởi hệ thống gợi ý của Amazon.com. Các ví dụ khác bao gồm:

Như đã đề cập chi tiết ở trên, Last.fm đề xuất âm nhạc dựa trên sở thích quen nghe của những người dùng tương tự, trong khi Readgeek so sánh xếp hạng sách cho các đề xuất.

Facebook , MySpace , LinkedIn và các mạng xã hội khác sử dụng tính năng lọc cộng tác để giới thiệu bạn bè, nhóm và các kết nối xã hội khác (bằng cách kiểm tra mạng kết nối giữa người dùng và bạn bè của họ). Twitter sử dụng nhiều tín hiệu và tính toán trong bộ nhớ để giới thiệu cho người dùng của họ rằng họ nên “theo dõi”.

Các phương pháp lọc cộng tác thường gặp phải ba vấn đề: Cold Start, khả năng mở rộng và sự thưa thớt (sparsity).

- Cold Start: Các hệ thống này thường yêu cầu một lượng lớn dữ liệu hiện có của người dùng để đưa ra các đề xuất chính xác.
- Khả năng mở rộng: Trong nhiều môi trường mà các hệ thống này đưa ra các khuyến nghị, có hàng triệu người dùng và sản phẩm. Do đó, một lượng lớn công suất tính toán thường là cần thiết để tính toán các gợi ý.
- Sparsity: Số lượng các mặt hàng được bán trên các trang web thương mại điện tử lớn là cực kỳ lớn. Những người dùng tích cực nhất sẽ chỉ đánh giá một tập con nhỏ của cơ sở dữ liệu tổng thể. Do đó, ngay cả những mặt hàng phổ biến nhất cũng có rất ít xếp hạng.

Một loại thuật toán lọc cộng tác cụ thể sử dụng hệ số ma trận hóa (matrix factorization), kỹ thuật xấp xỉ ma trận cấp thấp (low-rank matrix approximation)[8].

Các phương pháp lọc cộng tác được phân loại là bộ lọc cộng tác dựa trên bộ nhớ và dựa trên mô hình. Một ví dụ nổi tiếng về các phương pháp dựa trên bộ nhớ là thuật toán

dựa trên người dùng và các phương pháp dựa trên mô hình là Kernel-Mapping Recommender.

2.3.2 Lọc dựa trên nội dung

Một cách tiếp cận phổ biến khác khi thiết kế hệ thống đề xuất là lọc nội dung . Phương pháp lọc dựa trên nội dung dựa trên mô tả về mặt hàng và hồ sơ về các tùy chọn của người dùng.

Trong hệ thống gợi ý dựa trên nội dung, từ khóa được sử dụng để mô tả các mục và hồ sơ người dùng được xây dựng để chỉ ra loại mục mà người dùng này thích. Nói cách khác, các thuật toán này cố gắng đề xuất các mục tương tự với các mục mà người dùng đã thích trong quá khứ (hoặc đang kiểm tra trong hiện tại). Cụ thể, các mục đề cử khác nhau được so sánh với các mục được đánh giá trước đây bởi người dùng và các mục phù hợp nhất được đề xuất. Cách tiếp cận này có nguồn gốc từ việc thu thập thông tin và nghiên cứu lọc thông tin .

Để tóm tắt các tính năng của các mục trong hệ thống, một thuật toán trình bày mục được áp dụng. Một thuật toán được sử dụng rộng rãi là biểu diễn tf – idf (còn được gọi là biểu diễn không gian vecto).

Để tạo hồ sơ người dùng, hệ thống chủ yếu tập trung vào hai loại thông tin:

1. Một mô hình ưu tiên của người dùng.
2. Lịch sử tương tác của người dùng với hệ thống gợi ý.

Về cơ bản, các phương thức này sử dụng một hồ sơ mặt hàng mô tả mục trong hệ thống. Hệ thống tạo hồ sơ dựa trên nội dung của người dùng dựa trên vectơ trọng số của các đối tượng địa lý. Trọng số biểu thị tầm quan trọng của từng tính năng đối với người dùng và có thể được tính từ các vectơ nội dung được xếp hạng riêng lẻ bằng nhiều kỹ thuật. Các phương pháp đơn giản sử dụng các giá trị trung bình của vector hạng mục trong khi các phương pháp phức tạp khác sử dụng các kỹ thuật máy học

như Bayesian Classifiers , phân tích cụm , cây quyết định và mạng thần kinh nhân tạo (artificial neural networks) để ước tính xác suất người dùng sẽ thích mục đó.

Phản hồi trực tiếp từ người dùng, thường dưới dạng nút thích hoặc không thích , có thể được sử dụng để gán trọng số cao hơn hoặc thấp hơn về tầm quan trọng của các thuộc tính nhất định (sử dụng phân loại Rocchio hoặc các kỹ thuật tương tự khác).

Một vấn đề quan trọng với lọc dựa trên nội dung là liệu hệ thống có thể tìm hiểu các tùy chọn của người dùng từ hành động của người dùng liên quan đến một nguồn nội dung hay không và sử dụng chúng trên các loại nội dung khác. Khi hệ thống bị hạn chế để xuất nội dung cùng loại với người dùng đang sử dụng, giá trị từ hệ thống để xuất thấp hơn đáng kể so với các loại nội dung khác từ các dịch vụ khác có thể được đề xuất. Ví dụ: giới thiệu các bài viết tin tức dựa trên việc duyệt tin tức hữu ích nhưng sẽ hữu ích hơn nhiều khi bạn có thể đề xuất âm nhạc, video, sản phẩm, cuộc thảo luận, v.v. từ các dịch vụ khác nhau dựa trên duyệt tin tức.

Pandora Radio là một ví dụ về hệ thống giới thiệu dựa trên nội dung phát nhạc có các đặc điểm tương tự như một bài hát do người dùng cung cấp làm hạt giống ban đầu. Ngoài ra còn có một số lượng lớn các hệ thống gợi ý dựa trên nội dung nhằm cung cấp các đề xuất phim, một vài ví dụ như Rotten Tomatoes , Internet Movie Database , Jinni , Rovi Corporation và Jaman[8]. Các hệ thống gợi ý giới thiệu tài liệu liên quan nhằm mục đích cung cấp các đề xuất tài liệu cho các nhà nghiên cứu. Các chuyên gia y tế công cộng đã nghiên cứu các hệ thống gợi ý để cá nhân hóa giáo dục sức khỏe và các chiến lược phòng ngừa.

2.3.3 Hệ thống gợi ý lai (Hybrid recommender systems)

Nghiên cứu gần đây đã chứng minh rằng một phương pháp lai, kết hợp lọc cộng tác và lọc dựa trên nội dung có thể hiệu quả hơn trong một số trường hợp. Các phương pháp lai có thể được thực hiện theo nhiều cách:

- Bằng cách đưa ra các dự đoán dựa trên nội dung và dựa trên lọc cộng tác riêng biệt và sau đó kết hợp chúng.
- Bằng cách thêm các khả năng dựa trên nội dung vào phương pháp cộng tác (và ngược lại).
- Bằng cách thống nhất các phương pháp tiếp cận thành một mô hình.

Một số nghiên cứu thực nghiệm so sánh hiệu suất của phương pháp lai với các phương pháp cộng tác thuần túy và chứng minh rằng các phương pháp lai có thể cung cấp các khuyến nghị chính xác hơn các phương pháp thuần túy. Những phương pháp này cũng có thể được sử dụng để khắc phục một số vấn đề thường gặp trong hệ thống gợi ý như Cold Start và vấn đề thưa thớt.

Netflix là một ví dụ tốt về việc sử dụng các hệ thống đề xuất lai. Trang web đưa ra các đề xuất bằng cách so sánh thói quen xem và tìm kiếm của những người dùng tương tự (ví dụ: lọc cộng tác) cũng như bằng cách cung cấp những bộ phim có chung đặc điểm với những bộ phim mà người dùng đánh giá cao (lọc dựa trên nội dung).

Một loạt các kỹ thuật đã được đề xuất làm cơ sở cho các hệ thống gợi ý: các kỹ thuật hợp tác (collaborative), dựa trên nội dung (content-based), dựa trên kiến thức (knowledge-based) và nhân khẩu học (demographic techniques). Mỗi kỹ thuật này đều có những thiếu sót, như vấn đề Cold Start cho các hệ thống cộng tác và dựa trên nội dung (phải làm gì với người dùng mới với ít xếp hạng) và tắc nghẽn kỹ thuật tri thức (knowledge engineering bottleneck) trong các phương pháp dựa trên tri thức. Một hệ thống gợi ý lai là một hệ thống trong đó kết hợp nhiều kỹ thuật với nhau để đạt được một số sức mạnh tổng hợp giữa chúng.

Cộng tác – Collaborative: Hệ thống tạo đề xuất chỉ sử dụng thông tin về hồ sơ xếp hạng cho những người dùng hoặc mục khác nhau. Các hệ thống cộng tác định vị “người dùng/mục” ngang hàng với lịch sử xếp hạng tương tự như người dùng hoặc mục hiện tại và tạo đề xuất sử dụng vùng lân cận này. Các thuật toán dựa trên người dùng và

dựa trên hàng gần nhất có thể được kết hợp để giải quyết vấn đề Cold Start và cải thiện kết quả đề xuất.

Dựa trên nội dung – Content-based: Hệ thống tạo đề xuất từ hai nguồn: các tính năng liên quan đến sản phẩm và xếp hạng mà người dùng đã cung cấp cho họ. Đề xuất dựa trên nội dung coi đề xuất là sự cố phân loại người dùng cụ thể và tìm hiểu trình phân loại cho lượt thích và không thích của người dùng dựa trên các tính năng của sản phẩm.

Nhân khẩu học – demographic techniques: Trình giới thiệu nhân khẩu học cung cấp các đề xuất dựa trên hồ sơ nhân khẩu học của người dùng. Sản phẩm được đề xuất có thể được sản xuất cho các mục nhân khẩu học khác nhau, bằng cách kết hợp xếp hạng của người dùng trong các mục đó.

Dựa trên tri thức – knowledge-based: Trình giới thiệu dựa trên kiến thức gợi ý các sản phẩm dựa trên các suy luận về nhu cầu và sở thích của người dùng. Kiến thức này đôi khi sẽ chứa kiến thức chức năng rõ ràng về cách các tính năng sản phẩm nhất định đáp ứng nhu cầu của người dùng.

Thuật ngữ Hybrid recommender systems được sử dụng ở đây để mô tả bất kỳ hệ thống recommender nào kết hợp nhiều kỹ thuật đề xuất với nhau để tạo dữ liệu đầu ra của nó.

Có bảy kỹ thuật lai cơ bản (hybridization techniques):

- Có trọng số (Weighted): Điểm số của các thành phần đề xuất khác nhau được kết hợp theo số lượng.
- Chuyển đổi (Switching): Hệ thống chọn giữa các thành phần đề xuất và áp dụng hệ thống được chọn.
- Hỗn hợp (Mixed): Các khuyến nghị từ những người giới thiệu khác nhau được trình bày cùng nhau để đưa ra đề xuất.

- Kết hợp tính năng (Feature Combination): Các tính năng được lấy từ các nguồn tri thức khác nhau được kết hợp với nhau và được đưa ra cho một thuật toán gợi ý duy nhất.
- Tính năng tăng cường (Feature Augmentation): Một kỹ thuật gợi ý được sử dụng để tính toán một tính năng hoặc tập hợp các tính năng, sau đó là một phần của đầu vào cho kỹ thuật tiếp theo.
- Cascade: Các khuyến nghị được ưu tiên nghiêm ngặt, với những ưu tiên thấp hơn phá vỡ các mối quan hệ trong việc tính điểm của những người cao hơn.
- Cấp độ meta (Meta-level): Một kỹ thuật đề xuất được áp dụng và tạo ra một số loại mô hình, sau đó là đầu vào được sử dụng bởi kỹ thuật tiếp theo.

2.3.4 Các kỹ thuật không cá nhân hóa

Trong nhóm kỹ thuật này, do chúng khá đơn giản, dễ cài đặt nên thường được các website/hệ thống tích hợp vào, gồm cả các website thương mại, website tin tức, hay giải trí. Chẳng hạn như trong các hệ thống bán hàng trực tuyến, người ta thường gợi ý các sản phẩm được xem/mua/bình luận/.. nhiều nhất; gợi ý các sản phẩm mới nhất; gợi ý các sản phẩm cùng loại/ cùng nhà sản xuất/..; gợi ý các sản phẩm được mua/chọn cùng nhau. Một ví dụ khá điển hình là thông qua luật kết hợp (như Apriori), Amazon đã áp dụng khá thành công để tìm ra các sản phẩm hay được mua cùng nhau.

Tuy vậy, bất lợi của các phương pháp này là không cá nhân hóa cho từng người dùng, nghĩa là tất cả các user đều được gợi ý giống nhau khi chọn cùng sản phẩm[8].

2.4 Chi tiết về kĩ thuật lọc dựa trên nội dung (Content-based Filtering)

Trong phần này sẽ là cách em vận dụng kĩ thuật đan xen giữa lý thuyết và cách áp dụng lý thuyết đó vào các bước như Tiền xử lý dữ liệu, Tính độ tương đồng của thông tin và đưa ra gợi ý trong bài toán đề xuất bài hát.

Ví dụ dưới đây sử dụng ít thuộc tính hơn so với trong ứng dụng nhằm mục đích khái quát về phương pháp áp dụng.

2.4.1 Tiền xử lý dữ liệu

Trước tiên, chúng ta cần kết nối mã Python với FRD chứa dữ liệu về bài hát và các số liệu thống kê cần thiết để sử dụng cho kĩ thuật lọc theo nội dung.

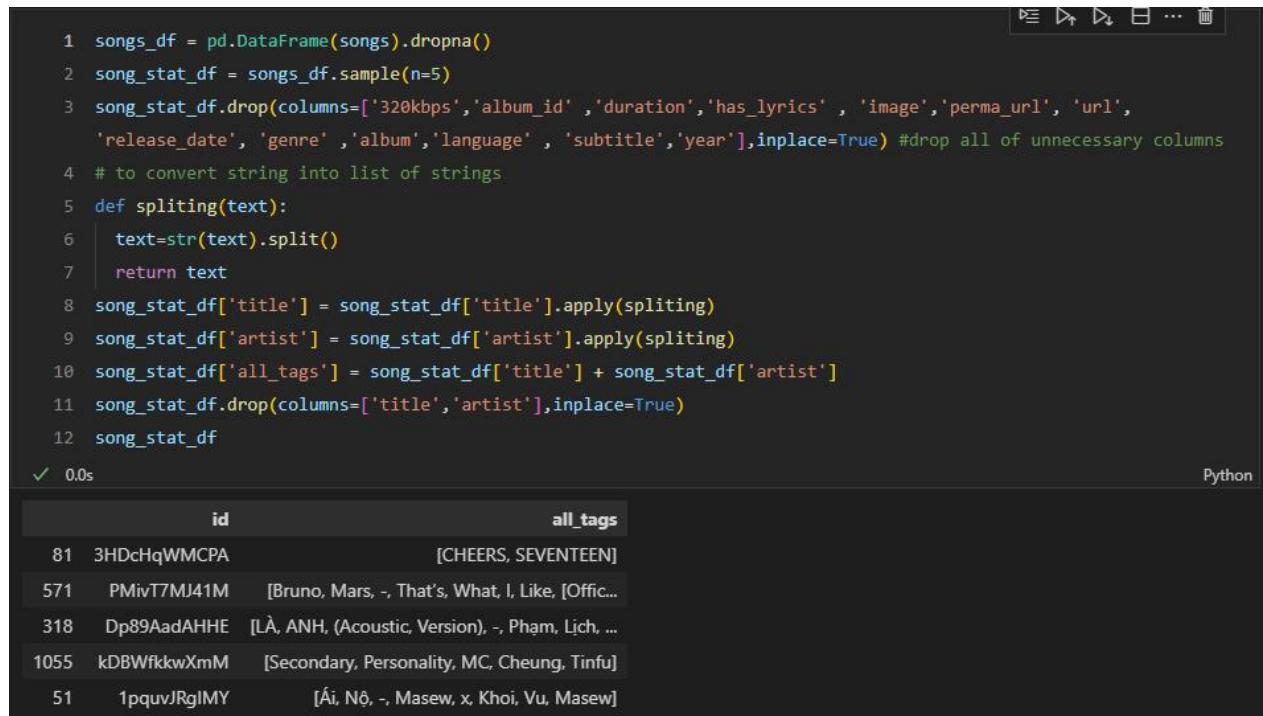
Vì dữ liệu được lưu dưới dạng cặp key: value với key được sinh ra từ động nên key sẽ không có ý nghĩa gì cho việc đề xuất của hệ thống. Trong Hình 2.3 ta có thể thấy rằng giá trị trả về chỉ là danh sách các bài hát.

```
ref = db.reference('/')
data = ref.child('songs').order_by_child('id').get()
values = []
for key in data:
    values.append(data[key])
values

[{'320kbps': 'false',
 'album': 'PHÚC DU',
 'album_id': 'UCPQM-wqVHzULADG9zJir5HQ',
 'artist': 'PHÚC DU',
 'duration': '263',
 'genre': 'YouTube',
 'has_lyrics': 'false',
 'id': '-7sISWuTdj0',
```

Hình 2.3: Lấy ra danh sách bài hát từ FRB

Sử dụng thư viện pandas để trích xuất và xử lý dữ liệu dạng danh sách. Pandas cung cấp các đối tượng dữ liệu để chứa và xử lý dữ liệu bao gồm các đối tượng Series (dữ liệu một chiều) và DataFrame (dữ liệu hai chiều).



```

1 songs_df = pd.DataFrame(songs).dropna()
2 song_stat_df = songs_df.sample(n=5)
3 song_stat_df.drop(columns=['320kbps','album_id','duration','has_lyrics','image','perma_url','url',
4 'release_date','genre','album','language','subtitle','year'],inplace=True) #drop all of unnecessary columns
5 # to convert string into list of strings
6 def splitting(text):
7     text=str(text).split()
8     return text
9 song_stat_df['title'] = song_stat_df['title'].apply(splitting)
10 song_stat_df['artist'] = song_stat_df['artist'].apply(splitting)
11 song_stat_df['all_tags'] = song_stat_df['title'] + song_stat_df['artist']
12 song_stat_df.drop(columns=['title','artist'],inplace=True)
13 song_stat_df

```

0.0s

| | id | all_tags |
|------|-------------|--|
| 81 | 3HDcHqWMCPA | [CHEERS, SEVENTEEN] |
| 571 | PMivT7MJ41M | [Bruno, Mars, ~, That's, What, I, Like, [Offic...] |
| 318 | Dp89AadAHHE | [LÀ, ANH, (Acoustic, Version), ~, Phạm, Lịch, ...] |
| 1055 | kDBWfkkwXmM | [Secondary, Personality, MC, Cheung, Tinfu] |
| 51 | 1pquvJRgIMY | [Ái, Nô, ~, Masew, x, Khoi, Vu, Masew] |

Hình 2.4: Chuyển đổi dữ liệu bằng pandas DataFrame

Trong Hình 2.4 ta bỏ đi những trường hợp dữ liệu không hợp lệ và chọn ra 5 bài hát ngẫu nhiên để sử dụng trong ví dụ lần này. Tiếp theo, ta bỏ đi những thuộc tính không cần thiết và tách tất cả các thuộc tính hữu ích còn lại (tên bài hát và nghệ sĩ) thành các danh sách. Cuối cùng, ghép hai danh sách đó lại tạo thành một danh sách các từ đại diện cho thông tin của bài hát và xóa đi các thuộc tính nay đã không cần dùng đến.

Kết quả cho ta thấy một DataFrame với chỉ hai cột: id - dùng để định danh bài hát và all_tags - là mảng gộp hai thuộc tính đại diện cho thông tin mà hệ thống sẽ sử dụng để tính độ tương đồng giữa các bài hát.

```

1 def convert_lower(text):
2     l=[]
3     for item in text:
4         l.append(item.lower())
5     return l
6 song_stat_df['all_tags']=song_stat_df['all_tags'].apply(convert_lower)
7 song_stat_df

```

✓ 0.0s

| | id | all_tags |
|------|-------------|--|
| 81 | 3HDcHqWMCPA | [cheers, seventeen] |
| 571 | PMivT7MJ41M | [bruno, mars, -, that's, what, i, like, [offic... |
| 318 | Dp89AadAHHE | [là, anh, (acoustic, version), -, phạm, lịch, ...] |
| 1055 | kDBWfkkwXmM | [secondary, personality, mc, cheung, tinfu] |
| 51 | 1pquvJRgIMY | [ái, nô, -, masew, x, khoi, vu, masew] |

Hình 2.5: Chuyển tất cả từ về dạng chữ thường

DataFrame trong Hình 2.5 là kết quả sau khi chuyển tất cả từ trong all_tags về dạng chữ thường để đảm bảo tính đồng bộ và tăng độ chính xác của việc tính toán.

Lúc này, dữ liệu đã không còn khác biệt về chữ hoa-chữ thường. Nhưng vẫn còn một vấn đề đó chính là các từ trong danh sách chưa được đưa về dạng nguyên bản. Ví dụ như từ “personality” thực ra chỉ là trạng từ của từ nguyên bản “person”. Mặc dù việc này chỉ ảnh hưởng đến các từ tiếng Anh nhưng vẫn cần có thêm bước các từ về dạng nguyên bản, ở đây em sử dụng phương pháp PorterStemmer.

```

1  from nltk.stem.porter import PorterStemmer
2  ps=PorterStemmer()
3  def stemming(text):
4      l=[]
5      for i in text:
6          l.append(ps.stem(i))
7      return l
8
9  song_stat_df['all_tags'] = song_stat_df['all_tags'].apply(stemming)
10 song_stat_df['all_tags'] = song_stat_df['all_tags'].apply(lambda x: " ".join(x))
11 song_stat_df

```

✓ 0.0s

| | id | all_tags |
|------|--------------|---|
| 81 | 3HDcHqWMCPA | cheer seventeen |
| 571 | PMivT7MJ41M | bruno mar - that' what i like [offici music vi... |
| 318 | Dp89AadAHHE | là anh (acoust version) - phạm lịch phạm lịch 🎵 |
| 1055 | kDBWfkkwXmM | secondari person mc cheung tinfu |
| 51 | 1pquivJRgIMY | ái nô - masew x khoi vu masew |

Hình 2.6: Đưa các từ trong danh sách về dạng nguyên bản bằng PorterStemmer

PorterStemmer là một trong những phương pháp tạo gốc phổ biến được đề xuất vào năm 1980. Nó dựa trên ý tưởng rằng các hậu tố trong tiếng Anh được tạo thành từ sự kết hợp của các hậu tố nhỏ hơn và đơn giản hơn. Thuật toán này sử dụng một loạt các quy tắc để loại bỏ các hậu tố không cần thiết khỏi từ. Ví dụ: “running” sẽ được chuyển thành “run”. PorterStemmer có thể được sử dụng trong xử lý ngôn ngữ tự nhiên để giảm kích thước của từ vựng và giúp cho việc so khớp từ khóa trở nên dễ dàng hơn.

Chúng ta chuyển tất cả các từ về dạng gốc và sử dụng hàm join() để ghép danh sách đó lại thành một chuỗi để chuẩn bị cho bước tính toán tiếp theo.

2.4.2 Tính độ tương đồng của thông tin

Chúng ta tiến hành chuyển đổi dữ liệu văn bản thành dữ liệu số có thể được sử dụng bởi các mô hình học máy bằng mô phỏng chúng dưới dạng các véc-tơ. Có hai cách khác nhau đó chính là sử dụng CountVectorizer và TfidfVectorizer.

Cả hai phương pháp có điểm chung là đều dựa trên các từ riêng biệt trong tập mẫu.

```
1 from sklearn.feature_extraction.text import CountVectorizer
2 vectorizer = CountVectorizer()
3 all_tags = song_stat_df['all_tags']
4 data = vectorizer.fit_transform(all_tags)
5 print("Unique vocabulary: ", len(vectorizer.vocabulary_))
6 data.toarray()
✓ 0.0s
Unique vocabulary: 26
array([[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0],
       [0, 0, 2, 0, 0, 0, 1, 0, 0, 2, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0,
       1, 0, 1, 0],
       [1, 1, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 1,
       0, 0, 0, 0],
       [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0,
       0, 0, 0, 0],
       [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 1]]], dtype=int64)
```

Hình 2.7: Số hóa dữ liệu sử dụng CountVectorizer

CountVectorizer là một phương thức của lớp *sklearn.feature_extraction.text* trong thư viện *scikit-learn*. Nó được sử dụng để chuyển đổi một văn bản cho trước thành một vector dựa trên tần suất của mỗi từ xuất hiện trong toàn bộ văn bản. Tức là, phương thức này coi mọi từ đều quan trọng như nhau.

Như hình 2.7 ta có thể thấy rằng ma trận trả ra với 5 hàng và 26 cột, tương ứng cho 5 bài hát mà ta lựa chọn kèm theo 26 từ độc nhất trong cả 5 trường thông tin. Các số trong hàng là biểu diễn cho số lần từ đó xuất hiện trong chuỗi thông tin của bài hát.

```
1 from sklearn.feature_extraction.text import TfidfVectorizer
2 vectorizer = TfidfVectorizer()
3 all_tags = song_stat_df['all_tags']
4 data = vectorizer.fit_transform(all_tags)
5 print("Unique vocabulary: ", len(vectorizer.vocabulary_))
6 data.toarray()
✓ 0.0s
```

Unique vocabulary: 26

```
array([[0.          , 0.          , 0.          , 0.          , 0.70710678, 0.          ,
       0.          , 0.          , 0.          , 0.          , 0.          , 0.          ,
       0.          , 0.          , 0.          , 0.          , 0.          , 0.          ,
       0.          , 0.          , 0.          , 0.          , 0.70710678, 0.          ,
       0.          , 0.          , 0.          , 0.          , 0.          , 0.          ,
       0.          , 1.        ]])
```

Hình 2.8: Số hóa dữ liệu sử dụng TfidfVectorizer

Khác với *CountVectorizer*, *TfidfVectorizer* không những đếm số lần xuất hiện của mỗi từ trong một tài liệu mà còn tỉ lệ hóa số lần đếm bằng nghịch đảo tần suất tài liệu của mỗi từ. Nói cách khác, *CountVectorizer* cho tất cả các từ trong một tài liệu cùng mức quan trọng trong khi *TfidfVectorizer* cho các từ hiếm gặp trên tất cả các tài liệu là quan trọng hơn.

Cụ thể cách tính TF-IDF (Term Frequency - Inverse Document Frequency) như sau:

$$\text{Tfidf}_{(t,d,D)} = \text{Tf}_{(t,d)} * \text{Idf}_{(t,D)}$$

Trong đó: Tf là tần suất xuất hiện của từ t trong tài liệu d và Idf là tần suất nghịch đảo của từ t trong tập các tài liệu D.

Cách tính TF được thực hiện bằng cách đếm số lần xuất hiện của thuật ngữ trong một tài liệu và chia cho tổng số từ trong tài liệu đó. Công thức như sau:

$$TF(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

Trong đó: $f_{t,d}$ là số lần xuất hiện của thuật ngữ t trong tài liệu d .

$\sum_{t' \in d} f_{t',d}$ là tổng số từ trong tài liệu d .

Cách tính IDF được thực hiện bằng cách lấy tổng số tài liệu trong tập văn bản và chia cho số lượng tài liệu chứa thuật ngữ đó. Kết quả là một giá trị dương và giá trị này càng cao thì thuật ngữ đó càng hiếm trong tập văn bản. Công thức như sau:

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

Trong đó: N là tổng số tài liệu trong tập văn bản D .

$|\{d \in D : t \in d\}|$ là số văn bản chứa từ nhất định, với điều kiện t xuất hiện trong văn bản d ($TF(t, d) \neq 0$). Nếu từ đó không xuất hiện ở bất cứ văn bản nào trong tập thì mẫu số sẽ bằng 0 do phép chia cho không hợp lệ, vì thế người ta thường thay bằng mẫu thức $1 + |\{d \in D : t \in d\}|$.

Những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó).

Nhận thấy được ưu điểm của TF-IDF trong hệ thống đề xuất nội dung nên em quyết định chọn phương pháp này cho hệ thống của mình.

Sau khi chuyển đổi văn bản thành dạng số thì việc tiếp theo là tính độ tương tự của các véc-tơ đã được chuyển đổi ở trên với phương thức *Cosine_similarity* thuộc lớp *sklearn.metrics.pairwise*.

Phương pháp này đo độ tương đồng giữa hai văn bản bằng cách tính cosin của góc giữa hai véc-tơ biểu diễn cho hai văn bản đó. Giá trị cosine similarity nằm trong khoảng [-1, 1], trong đó giá trị 1 thể hiện hai văn bản hoàn toàn giống nhau và giá trị -1 thể hiện hai văn bản hoàn toàn khác nhau.

```
1 from sklearn.metrics.pairwise import cosine_similarity
2 similarity = cosine_similarity(data)
3 similarity
✓ 0.0s
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.]])
```

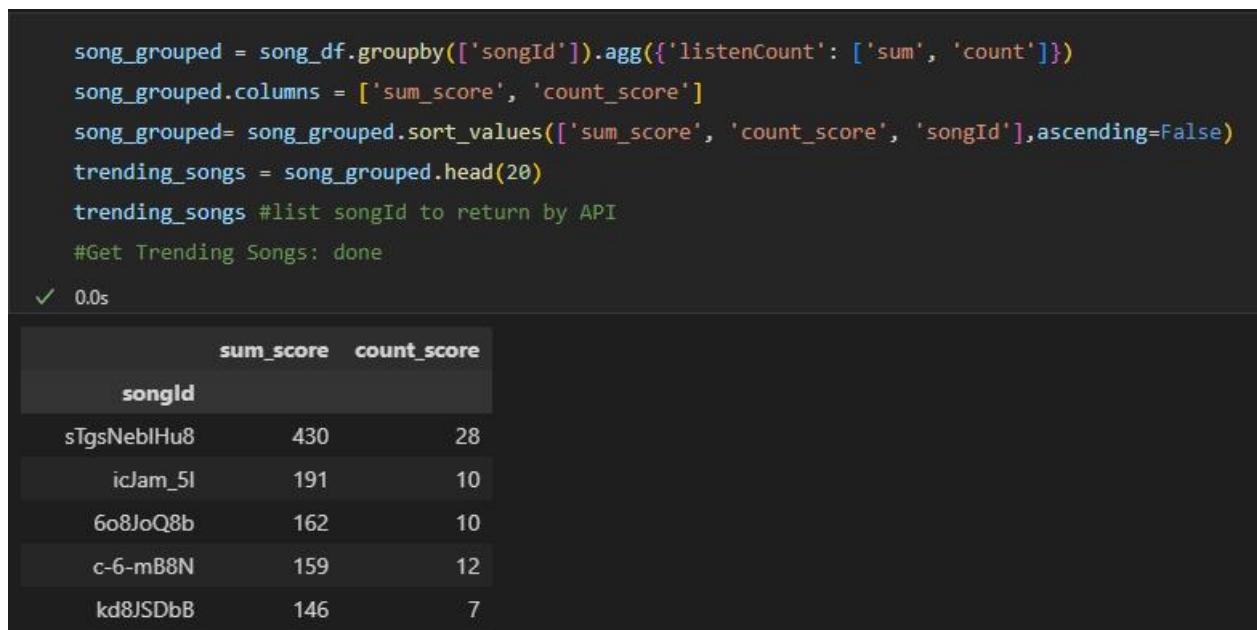
Hình 2.9: Tính độ tương đồng bằng Cosine_similarity

Trong hình 2.9, *similarity* là ma trận thể hiện độ tương đồng giữa các véc-tơ trong tập dữ liệu đang được xét đến. Dựa trên ma trận này, có thể tìm ra được những bài hát tương tự với bài hát mà người dùng đang nghe.

2.4.3 Các dạng đề xuất bài hát trong ứng dụng

a) Đề xuất các bài hát thịnh hành

“Trendings” là một cụm từ không còn xa lạ đối với các ứng dụng đa phương tiện hiện nay. Các bài hát, video, bộ phim...càng nhận được nhiều sự chú ý sẽ càng có vị trí cao trên bảng xếp hạng này. “Sự chú ý” theo từng hệ thống sẽ dựa trên những thông số không giống nhau có thể là các đánh giá, lượt like/dislike..., hệ thống lần này sử dụng số lượt nghe để biểu hiện khái niệm này.



```
song_grouped = song_df.groupby(['songId']).agg({'listenCount': ['sum', 'count']})
song_grouped.columns = ['sum_score', 'count_score']
song_grouped= song_grouped.sort_values(['sum_score', 'count_score', 'songId'],ascending=False)
trending_songs = song_grouped.head(20)
trending_songs #list songId to return by API
#Get Trending Songs: done
```

✓ 0.0s

| songId | sum_score | count_score |
|-------------|-----------|-------------|
| sTgsNebIHu8 | 430 | 28 |
| icJam_5I | 191 | 10 |
| 6o8JoQ8b | 162 | 10 |
| c-6-mB8N | 159 | 12 |
| kd8JSDbB | 146 | 7 |

Hình 2.10: Đề xuất những bài hát phổ biến

Như được thể hiện trên Hình 2.10, dữ liệu trong DataFrame mẫu được nhóm lại thông qua songId, trong mỗi nhóm sẽ thực hiện hai phép toán là tính tổng số lượt nghe và đếm số user mà bài hát đó tiếp cận tới. Hai thông số này quyết định những bài hát nào nổi bật theo thứ tự giảm dần.

Kết quả các bài hát thịnh hành được thể hiện ở hình 2.11 bên dưới. Đây chỉ là một kĩ thuật thống kê nhưng nó sẽ rất hữu dụng khi ứng dụng được nhiều người biết tới và tạo ra một cơ sở dữ liệu đủ đa dạng. Chức năng này gợi ý cho người dùng biết rằng bài hát nào đang được nhiều người tiếp cận và gián tiếp gia tăng trải nghiệm người dùng.

```

top_trending = song.loc[song['songId'].isin(trending_songs.index)]
top_trending.drop_duplicates(['songId'])

✓ 0.1s

```

| | 320kbps | album | album_id | artist | duration | genre | has_lyrics | songId |
|-----|---------|---------------------------------|---------------------------|---------------------------|----------|---------|------------|-------------|
| 78 | NaN | Equals | NaN | Ed Sheeran | 214 | NaN | NaN | 496697 |
| 139 | true | + | 10436096 | Ed Sheeran | 263 | English | false | 6o8JoQ8b |
| 238 | false | RayeVEVO | UC1aQVIVnCGa4hDJtvNxf11g | RayeVEVO | 273 | YouTube | false | DII6VJ2C7wo |
| 362 | false | Evolve | UC0aXrjVxG5pZr99v77wZdPQ | Imagine Dragons | 204 | YouTube | false | Kx7B-XvmFtE |
| 425 | true | Dandelions (slowed + reverb) | 29160455 | Ruth B., sped up + slowed | 268 | English | false | POhlVMwj |
| 454 | false | FIFTY FIFTY Official | UCJEER74X9kBenzMT_x9ik9Mw | FIFTY FIFTY Official | 203 | YouTube | false | Qc7_zRjH808 |

Hình 2.11: Danh sách các bài hát thịnh hành

b) Đề xuất các bài hát phù hợp với từng người dùng

Cá nhân hóa là mục đích lớn nhất của RS, thay vì chỉ đơn thuần hiển thị dữ liệu từ cơ sở dữ liệu lên thì mô hình dựa trên những bài hát mà người dùng đã nghe trước đó để đưa ra những bài hát mà có thể người dùng đó sẽ thích.

Chức năng này dựa trên những bài hát mà người dùng đã nghe so sánh với lịch sử của những người dùng khác, tìm ra những bài hát có tỉ lệ người nghe cao nhất mà người dùng hiện tại chưa được nghe và đề xuất danh sách đó.

```

    self_sv.get_user_similar_songs('b74d04a0-f5e4-11ed-81a7-85dc82d623d5')
✓ 0.8s

Vì Anh Đâu Có Biết - Madihu (Feat. Vũ.) | Official MV - Madihu
Ed Sheeran - Perfect (Official Music Video) - Ed Sheeran
Vai Sentando - Ludmilla, King Doudou
Grupo Frontera x Bad Bunny - un x100to (Video Oficial) - Grupo Frontera
Nắng Ấm Xa Dàn - Sơn Tùng M-TP - Sơn Tùng M-TP Official
That's Hilarious - Charlie Puth
Hold My Hand - Lady Gaga
Attention - The Weeknd
Toosii - Favorite Song (Official Video) - Toosii2xVEVO
"Can You Feel The Love Tonight" - English/Chinese/Korean/Japanese/Thai Cover (Lion King) - MusicNeverSleeps
Alec Benjamin - Let Me Down Slowly [Official Music Video] - Alec Benjamin
Iris (Acoustic) - Adam Christopher
Sơn Tùng MTP - Remember Me (SlimV 2017 Mix) - SlimV
Because I Love You (EarthBound) Cover w/ lyrics | Michelle Heafy - michelleheafy
Chuyện Đôi Ta - Emcee L (Da LAB) ft Muôii (Official MV) - Da LAB Official
Deep Sea - Binz x Thanh Nguyễn x Triple D - Binz Da Poet
It's You - Ali Gatie
Jis Din Teri Meri Baat Nahin Hoti - Udit Narayan, Anuradha Paudwal
Lonely (feat. Ronie, Dominika Mirgova) - dvatri tridva
Christina Perri ft. Steve Kazee - A Thousand Years, Pt. 2 - Christina Perri

```

Hình 2.12: Lấy ra những bài hát mà người dùng đã nghe

Cụ thể, mô hình dựa trên mã người dùng được cung cấp sau đó lấy ra danh sách tất cả các ca khúc họ đã nghe mục đích ngoài việc dùng làm cơ sở để tìm ra những bài hát phù hợp thì còn là để tránh việc đề xuất lại những bài hát mà người dùng đã nghe từ trước đó. Chúng ta có thể thấy trong Hình 2.12 ở trên xuất hiện khá nhiều bài hát của Sơn Tùng MTP và Ed Sheeran.

```
No. of unique songs for the user: 89
no. of unique songs in the training set: 661
Non zero values in cooccurrence_matrix :20067
```

| | user_id | song | score | rank |
|----|--------------------------------------|---|----------|------|
| 0 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | Cô Chắc Yêu Là Đây - Sơn Tùng M-TP | 0.150153 | 1 |
| 1 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | Perfect - Ed Sheeran | 0.129191 | 2 |
| 2 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | Shape of You - Ed Sheeran | 0.127693 | 3 |
| 3 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | Flower - Jisoo, BLACKPINK | 0.122057 | 4 |
| 4 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | Boy's a liar, Pt. 2 - PinkPantheress, Ice Spice | 0.120782 | 5 |
| 5 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | Six Feet Under - The Weeknd | 0.119872 | 6 |
| 6 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | SAD BITCH - Anne-Marie | 0.116390 | 7 |
| 7 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | The Chainsmokers - Closer (Official Video) ft.... | 0.114451 | 8 |
| 8 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | I Will Remember You - Ed Sheeran | 0.114241 | 9 |
| 9 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | ĐAU ĐỄ TRƯỞNG THÀNH ONLYC OFFICIAL MV - On... | 0.113019 | 10 |
| 10 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | Matmos - Amplifier | 0.112832 | 11 |
| 11 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | RAYE, 070 Shake - Escapism. (Official Music Vi... | 0.112520 | 12 |
| 12 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | You Only Love Me - Rita Ora | 0.112114 | 13 |
| 13 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | Happier - Marshmello | 0.111954 | 14 |
| 14 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | Shawn Mendes, Camila Cabello - Señorita - Shaw... | 0.111584 | 15 |
| 15 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | The Chainsmokers - Don't Let Me Down (Official... | 0.111428 | 16 |
| 16 | b74d04a0-f5e4-11ed-81a7-85dc82d623d5 | STAY - Justin Bieber, The Kid Laro | 0.110719 | 17 |

Hình 2.13: Đề xuất những bài hát có thể phù hợp với người dùng

Đây là kết quả của mô hình khi đề xuất bài hát, Hình 2.11 cho thấy mô hình đề xuất được những bài hát đã từng được nghe của người dùng, cụ thể ở đây là các bài hát khác của Sơn Tùng MTP và Ed Sheeran.

```

self_sv.get_user_similar_songs('0c2983d0-f16d-11ed-9c88-514c38a48ebe')
✓ 1.0s

No. of unique songs for the user: 164
no. of unique songs in the training set: 661
Non zero values in cooccurrence_matrix :38781


```

| | user_id | song | score | rank |
|----|--------------------------------------|--|----------|------|
| 0 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | Dandelions (slowed + reverb) - Ruth B., sped u... | 0.084370 | 1 |
| 1 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | Flowers - Miley Cyrus | 0.083022 | 2 |
| 2 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | 土岐麻子 / 「HOME」Music Video - 土岐麻子 | 0.082536 | 3 |
| 3 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | Matmos - Amplifier | 0.076858 | 4 |
| 4 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | KHÔNG THỂ SAY - HIEUTHUHAI | 0.075734 | 5 |
| 5 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | Rod Wave - Fight The Feeling (Official Video) ... | 0.073693 | 6 |
| 6 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | Dù Cho Mai Về Sau (Acoustic Version) - buitruo... | 0.073316 | 7 |
| 7 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | MONO - Waiting For You (Album 22 - Track No.10...) | 0.072777 | 8 |
| 8 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | Iris (Acoustic) - Adam Christopher | 0.072254 | 9 |
| 9 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | DJ Snake - Let Me Love You ft. Justin Bieber ... | 0.072141 | 10 |
| 10 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | Doja Cat - Woman (Official Video) - dojacatVEVO | 0.069527 | 11 |
| 11 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | SIMPLE LOVE - Obito x Seachains x Davis x Lena... | 0.069527 | 12 |
| 12 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | Beyoncé - Halo - BeyoncéVEVO | 0.069527 | 13 |
| 13 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | In The Stars - Benson Boone | 0.069527 | 14 |
| 14 | 0c2983d0-f16d-11ed-9c88-514c38a48ebe | Angel Pt. 1 - Jimin | 0.069527 | 15 |

Hình 2.14: Tính cá nhân hóa đã được mô hình thể hiện

Kết quả khi yêu cầu mô hình đề xuất các bài hát cho những người dùng khác nhau là khác nhau, từ đó có thể khẳng định rằng mô hình này đảm bảo tính cá nhân hóa. Đây là API do hệ thống trả về khi ứng dụng được khởi động và gọi đến nó. Danh sách này sẽ được thể hiện ở màn hình chính để gợi ý cho người dùng những bài nhạc nên nghe.

CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1 Phân tích hệ thống

| STT | Tên yêu cầu | Mô tả |
|-----|------------------------|---|
| 1 | Đăng nhập tài khoản | Người dùng chỉ cần nhập tên để gợi nhớ mà không nhất thiết phải cung cấp quá nhiều thông tin cá nhân. |
| 2 | Nghe nhạc | Khi lựa chọn được bài nhạc ưa thích người dùng nhấn vào item đó, ứng dụng sẽ chuyển tới trang phát nhạc và nhạc sẽ tự động phát (trang này dùng cho cả nhạc online và offline). Ở đây người dùng có thể chuyển bài, tạm dừng, thay đổi tốc độ phát. |
| 3 | Tải nhạc | Chức năng này dùng để lưu lại những bài hát mà người dùng tâm đắc, nó được tích hợp dưới dạng nút bấm trong trang phát nhạc cũng như trong danh sách. |
| 4 | Thông báo | Khi phát nhạc, sẽ hiển thị thanh thông báo với các thông tin cơ bản của item đang được phát cùng với những tác vụ căn bản như chuyển bài, tạm dừng, yêu thích...(các tác vụ sẽ do người dùng tự điều chỉnh) |
| 5 | Bộ điều khiển âm thanh | Tích hợp trong màn hình phát nhạc giúp người dùng điều chỉnh mức độ to nhỏ của |

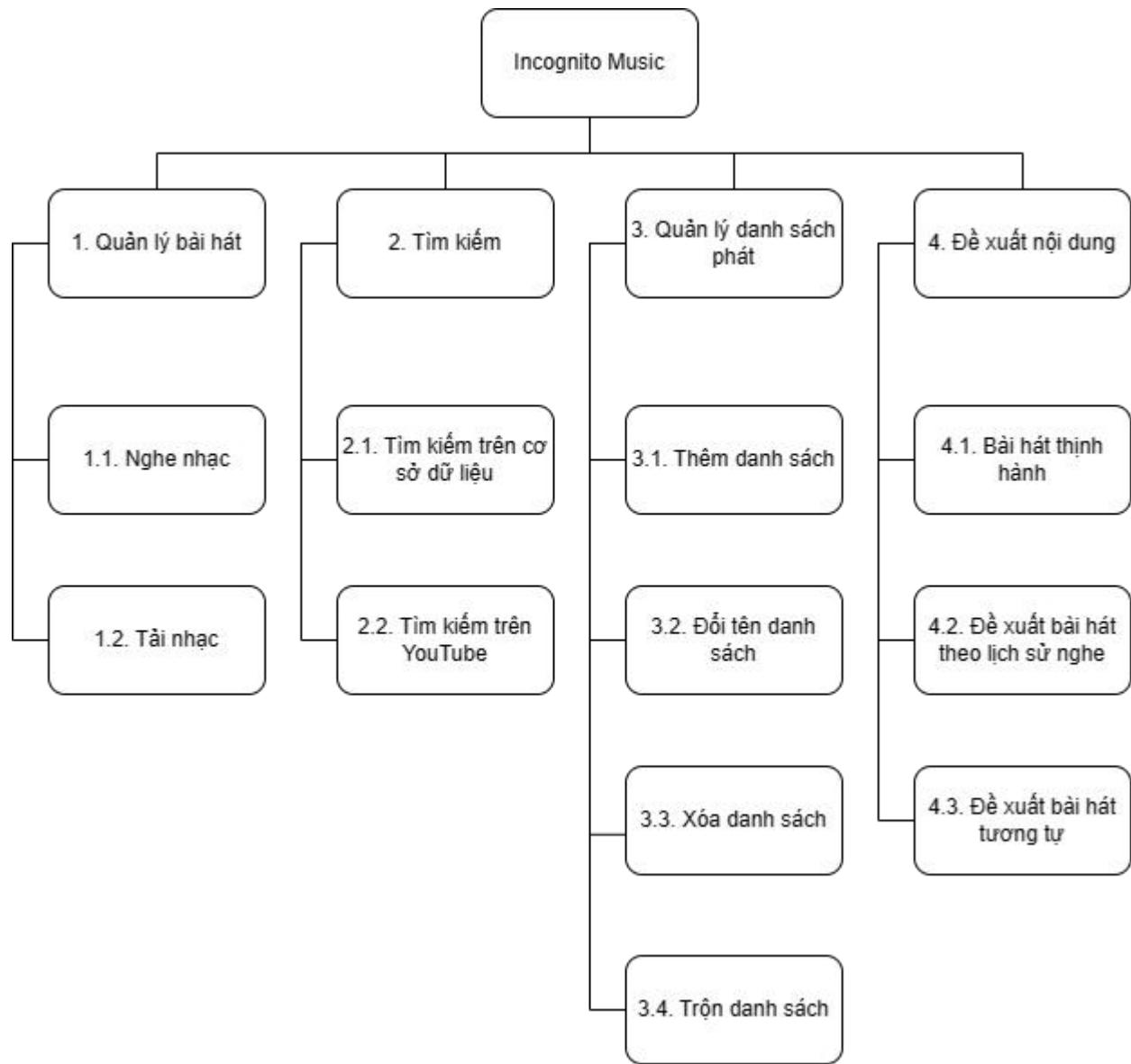
| | | |
|----|------------------------|--|
| | | các dải tần số âm thanh. Tùy theo từng thể loại nhạc mà điều chỉnh cho phù hợp. |
| 6 | Quản lý danh sách phát | Các danh sách phát (playlist) được tạo mới, đổi tên, xóa và trộn lại với nhau. Chức năng này giúp cho người dùng phân loại các ca khúc theo chủ đề mà mình mong muốn. |
| 7 | Tùy chỉnh cài đặt | Người dùng có thể thay đổi các cài đặt của ứng dụng như giao diện, ngôn ngữ, chất lượng phát nhạc, cursive... |
| 8 | Nghe nhạc từ YouTube | Để tăng thêm trải nghiệm cho người dùng, ứng dụng có sẵn tính năng phát các video trên YouTube dưới dạng audio. Người dùng có thể tìm kiếm, nghe và tải nhạc từ nguồn này. |
| 9 | Xem BXH từ Spotify | Sử dụng Spotify API lấy dữ liệu về bảng xếp hạng theo khu vực, người dùng có thể nhấn chọn bài hát mình muốn nghe và ứng dụng sẽ tìm kiếm dữ liệu của bài hát đó trên YouTube và chuyển tới trang phát nhạc. |
| 10 | Chia sẻ bài hát | Chức năng được tích hợp thông qua một nút ở màn hình phát nhạc. Khi nhấn vào, ứng dụng sẽ chia sẻ đường dẫn của bài hát đến tài khoản mạng xã hội mà bạn chỉ định. |

| | | |
|----|----------------------|--|
| 11 | Quản lý hàng đợi | Danh sách đang phát sẽ được hiển thị dưới dạng hàng đợi mà ở đó người dùng có thể thay đổi thứ tự phát, xóa bài hát. |
| 12 | Cá nhân hóa nội dung | Thông qua hệ thống để xuất nội dung được tích hợp, ứng dụng sẽ dựa vào lịch sử nghe nhạc của người dùng và hiển thị những nội dung tương tự nhằm tăng trải nghiệm. |

Bảng 3.1: Phân tích yêu cầu hệ thống

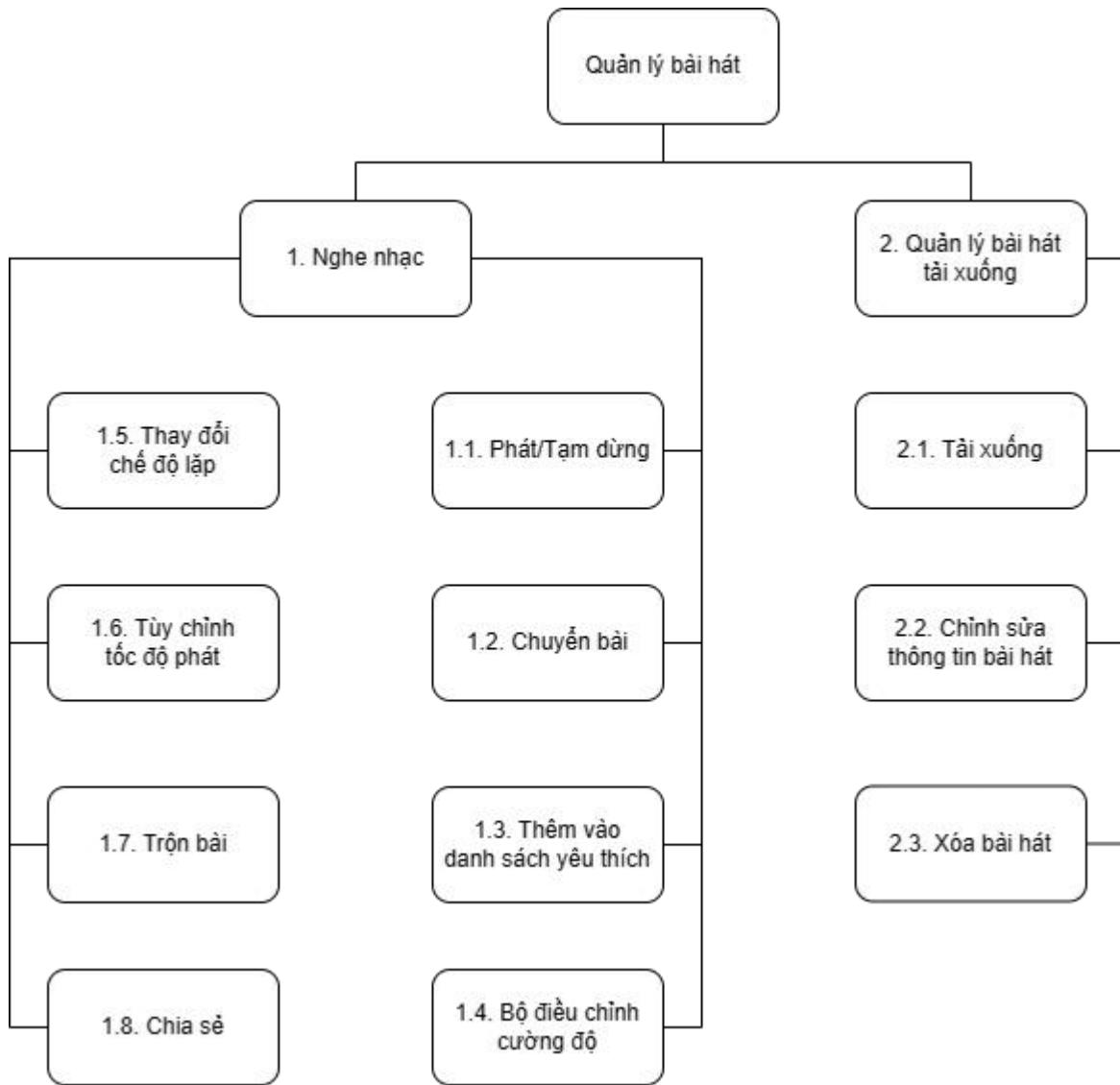
3.2 Thiết kế hệ thống

3.2.1 Sơ đồ phân rã chức năng BFD



Hình 3.1: Sơ đồ phân rã chức năng tổng quát

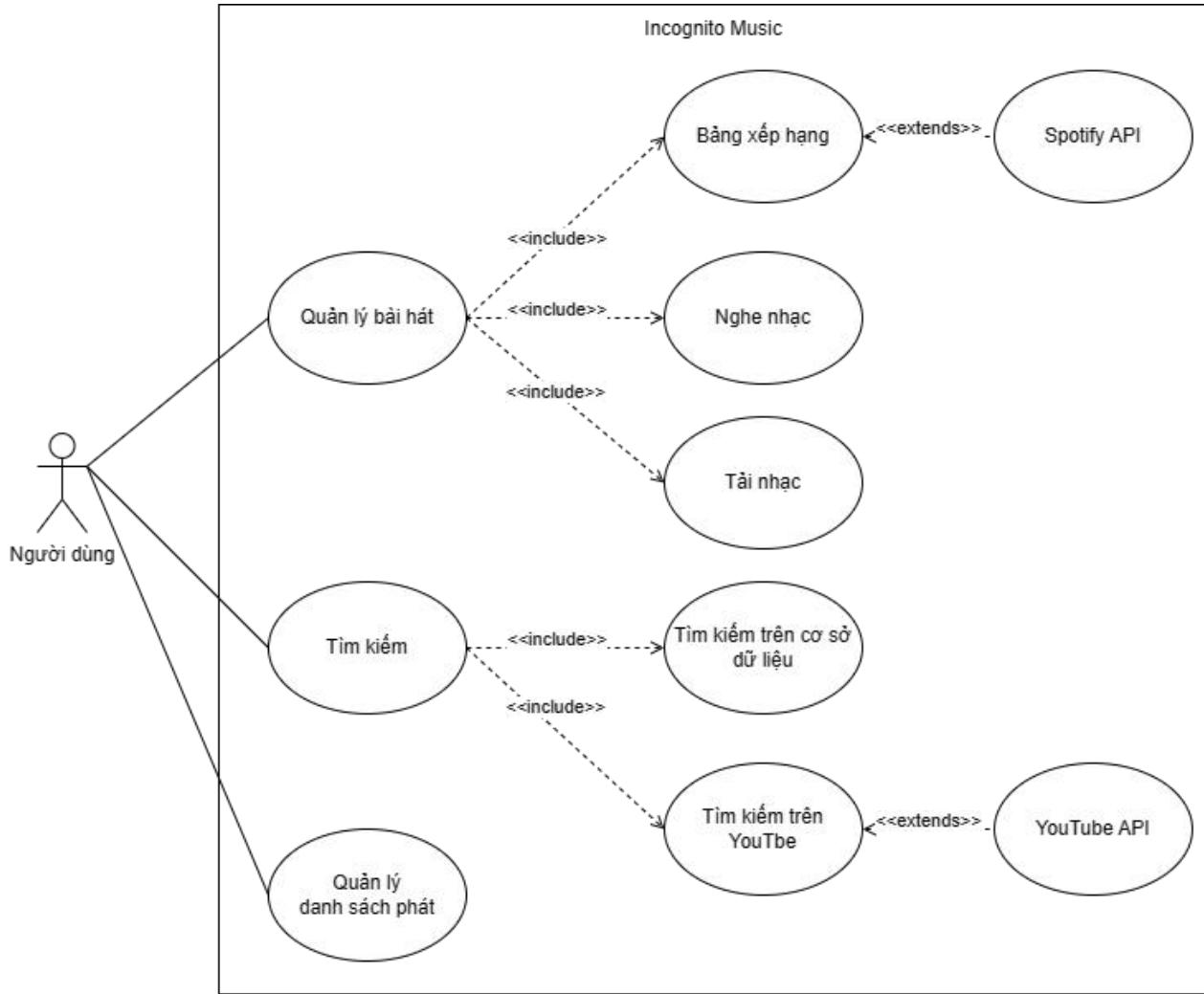
Ứng dụng có bốn chức năng chính bao gồm quản lý bài hát, tìm kiếm bài hát theo nhiều nguồn, quản lý danh sách phát và đề xuất nội dung.



Hình 3.2: Sơ đồ phân rã chức năng Quản lý bài hát

Tiếp tục phân rã chức năng Quản lý bài hát ta lại có chức năng nghe nhạc và quản lý nhạc tải xuống.

3.2.2 Sơ đồ Use Case

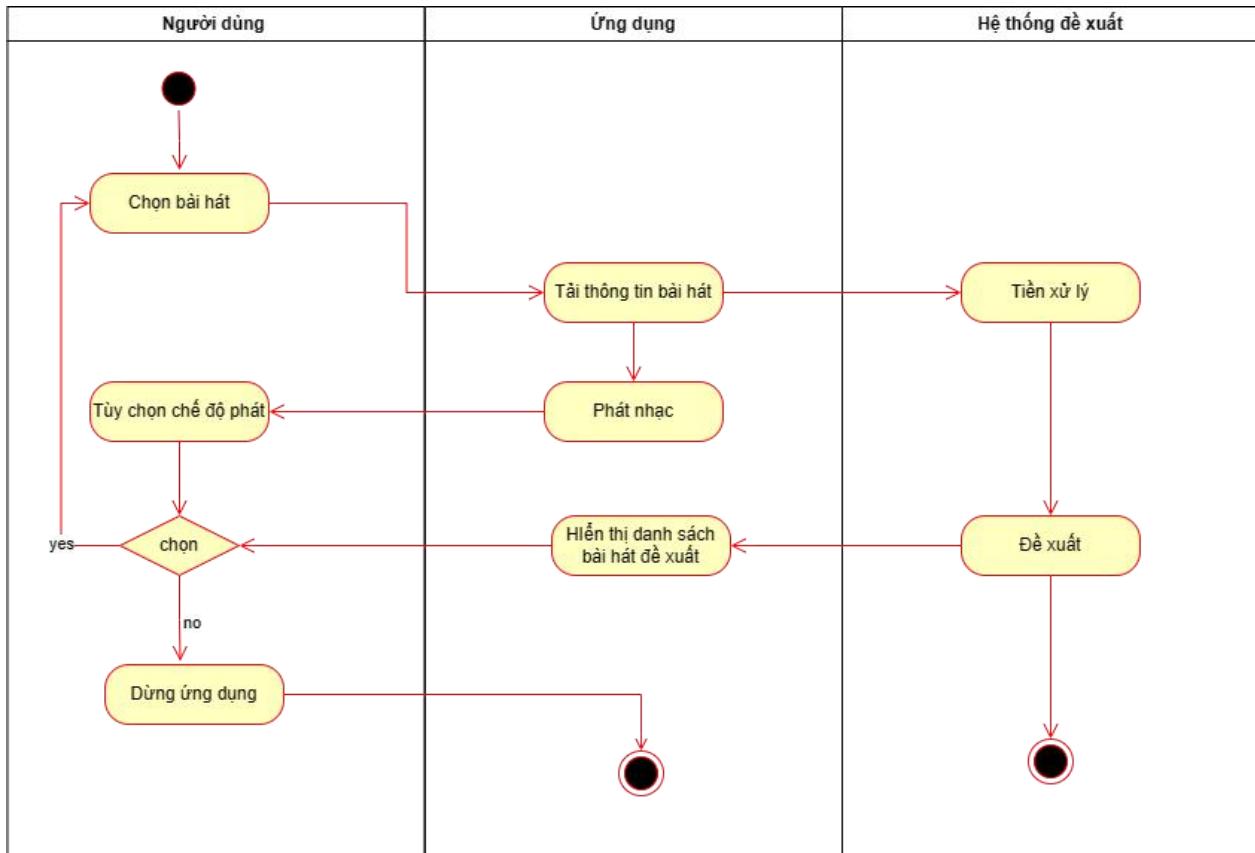


Hình 3.3: Sơ đồ Use Case tổng quát

Ứng dụng dành cho người dùng và tất cả các chức năng đều được cung cấp đầy đủ ngoại trừ hệ thống đề xuất nội dung.

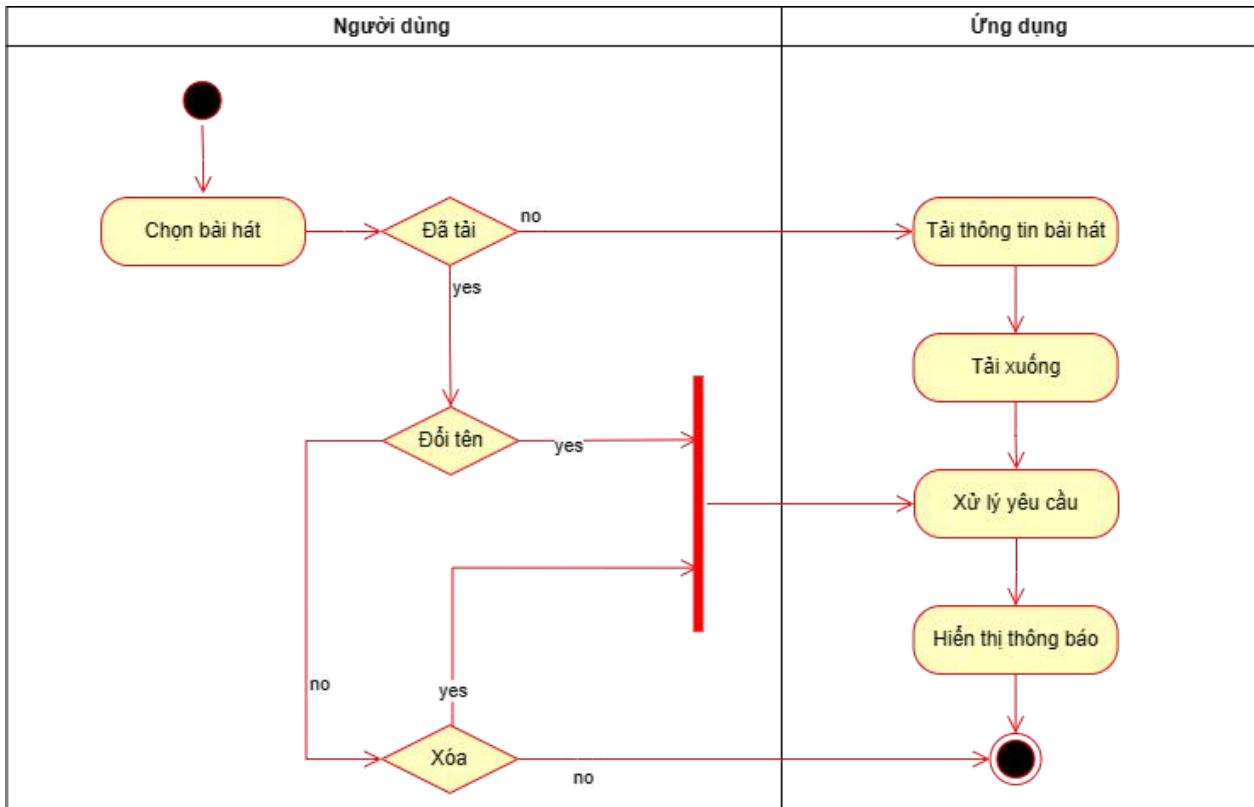
3.2.3 Sơ đồ hoạt động

a) Nghe nhạc



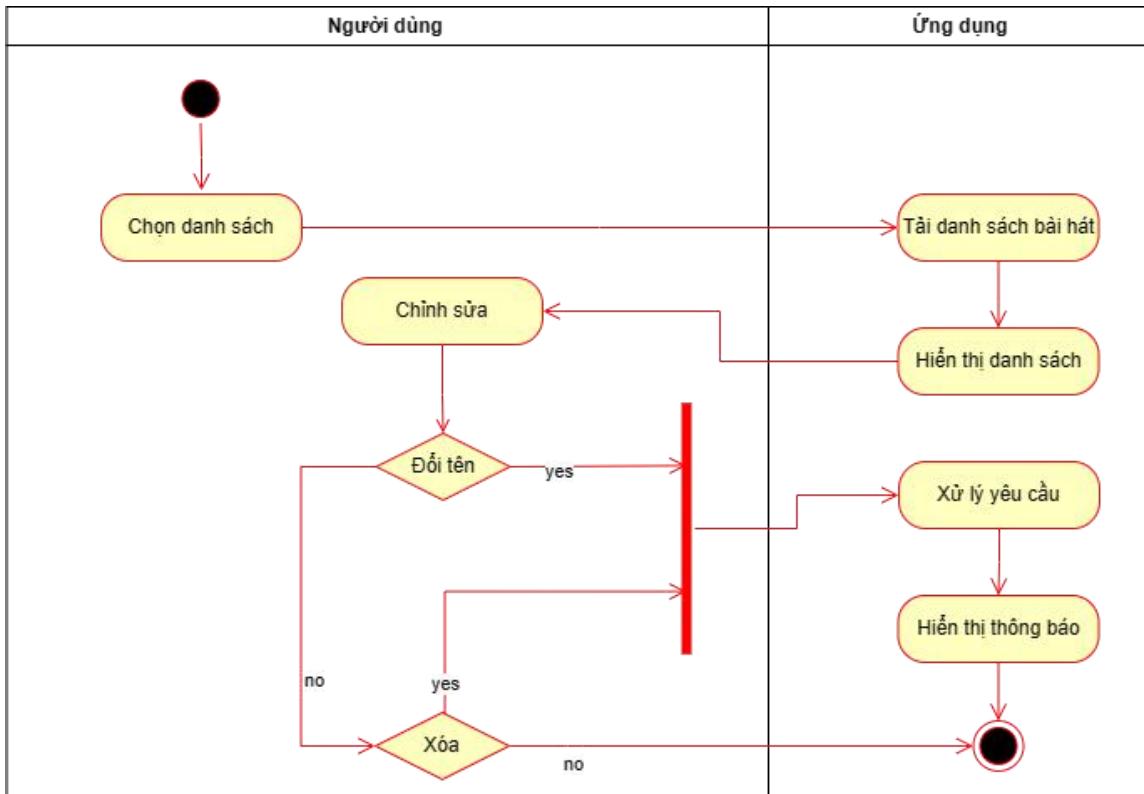
Hình 3.4: Sơ đồ hoạt động chức năng Nghe nhạc

b) Tải nhạc



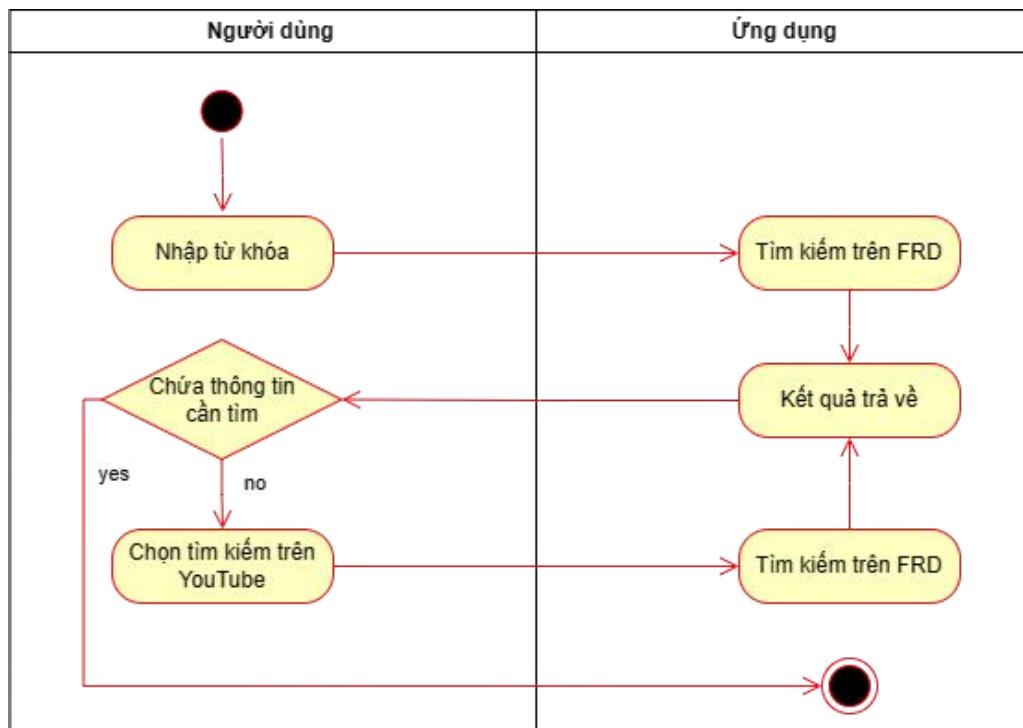
Hình 3.5: Sơ đồ hoạt động chức năng Tải nhạc

c) Quản lý danh sách phát



Hình 3.6: Sơ đồ hoạt động chức năng Quản lý danh sách phát

d) Tìm kiếm



Hình 3.7: Sơ đồ hoạt động chức năng Tìm kiếm

3.3 Cơ sở dữ liệu

Vì sử dụng NoSQL nên các thực thể dưới đây chỉ mô tả tên trường và vai trò của trường đó đối với ứng dụng.

a) Thực thể Bài hát

| Tên trường | Chú thích |
|--------------|---|
| 320kbps | Phát nhạc với chất lượng cao nhất |
| album | Tên Album, dùng trong RS |
| album_id | Mã Album, dùng để truy xuất dữ liệu |
| artist | Tên ca sĩ, dùng để hiển thị và xử lý dữ liệu trong RS |
| duration | Thời lượng bài hát |
| genre | Thể loại, dùng trong RS |
| Id | Mã bài hát, dùng để truy xuất dữ liệu |
| Image | Đường dẫn của ảnh, hiển thị trên giao diện |
| Language | Ngôn ngữ, dùng trong RS |
| Perma_id | Đường dẫn đến bài hát, dùng để chia sẻ bài hát |
| Release_date | Ngày ra mắt bài hát |
| Subtitle | Tiêu đề phụ, dùng để xử lý dữ liệu trong RS |
| Title | Tên bài hát, hiển thị và xử lý dữ liệu trong RS |
| Url | Đường dẫn bài hát, dùng để phát nhạc |
| Year | Năm phát hành |

Bảng 3.2: Thực thể Bài hát

b) Thực thể Người dùng

| Tên trường | Chú thích |
|-----------------|--|
| id | Mã tài khoản, dùng để truy vấn dữ liệu |
| name | Tên tài khoản, dùng để hiển thị |
| timeZone | Múi giờ |
| accountCreateOn | Thời điểm tạo tài khoản |

Bảng 3.3: Thực thể Người dùng

c) Thực thể Thông số

| Tên trường | Chú thích |
|-------------|---|
| songId | Mã bài hát, dùng để ánh xạ dữ liệu |
| userId | Mã người dùng, dùng để ánh xạ dữ liệu |
| listenCount | Số lượt nghe của người dùng với bài hát tương ứng, dùng để tính toán và đánh giá mức độ tương tự trong RS |

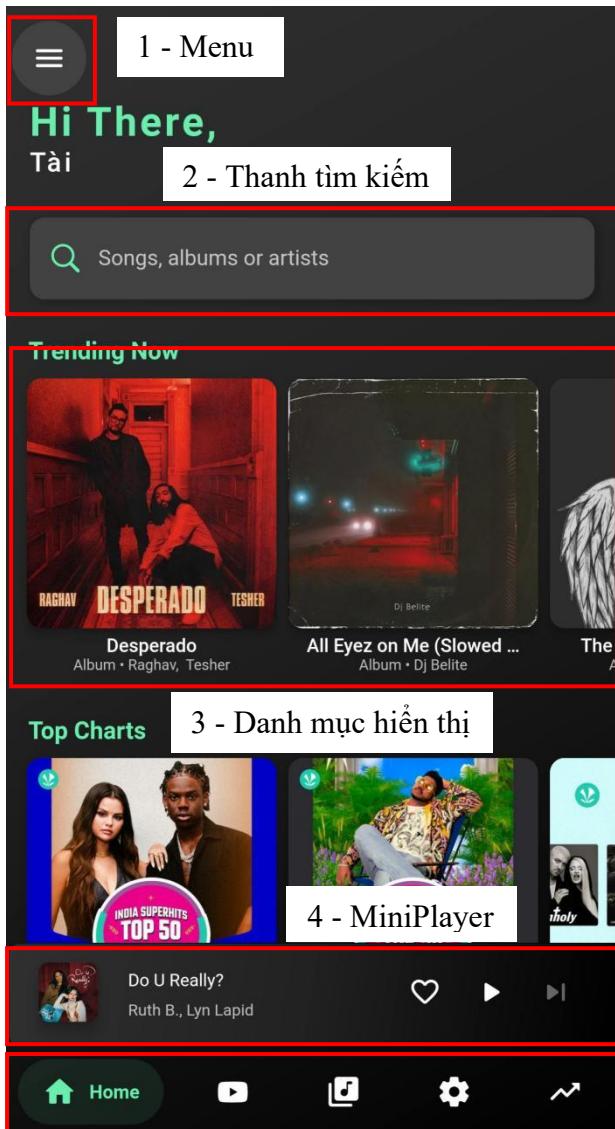
Bảng 3.4: Thực thể Thông số

CHƯƠNG 4. XÂY DỰNG ỨNG DỤNG INCOGNITO MUSIC

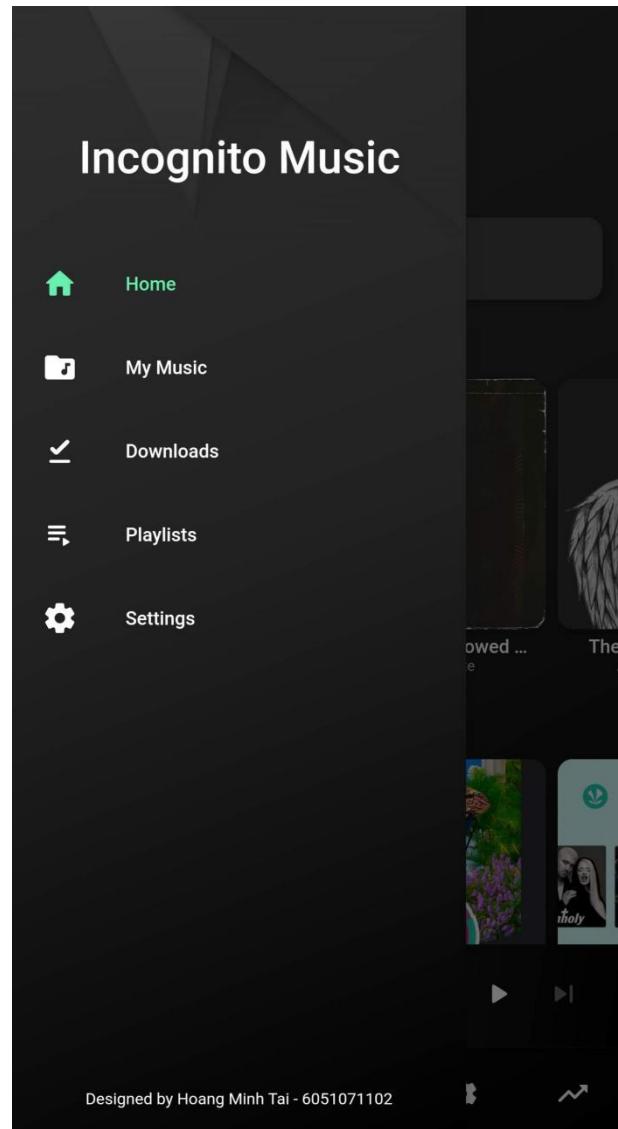
4.1 Giới thiệu

Các màn hình tương tác chính của người dùng khi sử dụng ứng dụng như Trang chủ, màn hình YouTube, màn hình Thư viện, tùy chọn cài đặt hệ thống và bảng xếp hạng.

4.1.1 Trang chủ



Hình 4.1: Màn hình trang chủ



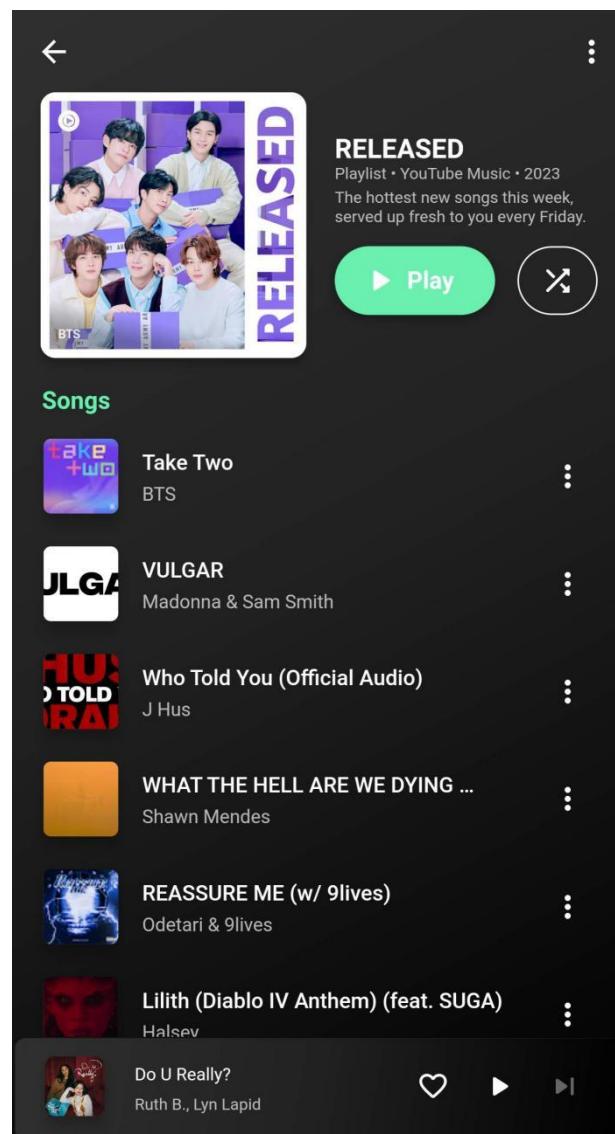
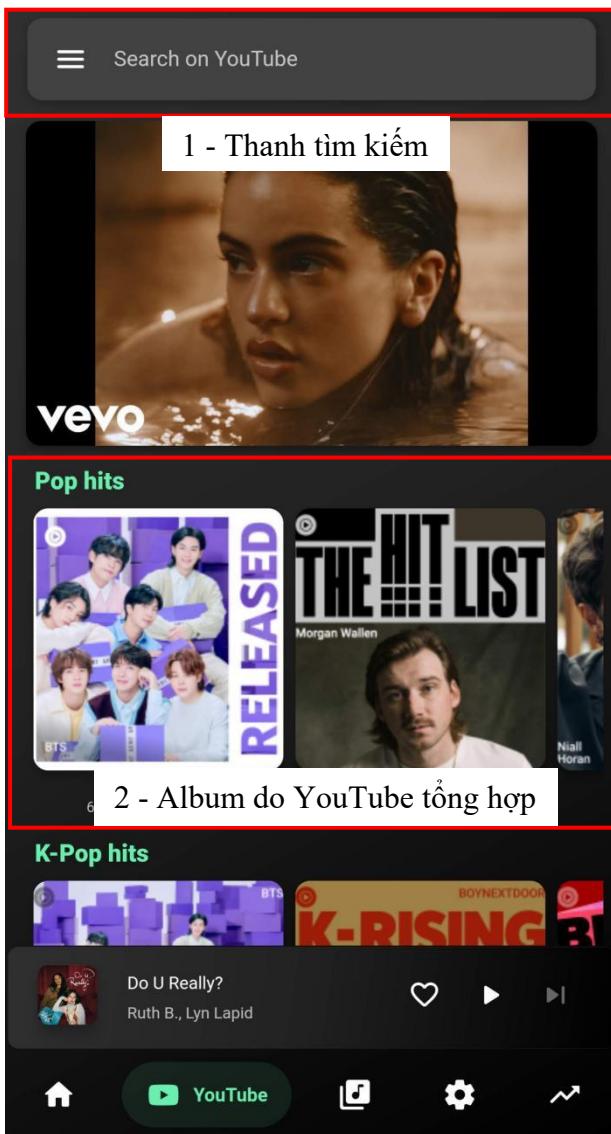
Hình 4.2: Thanh sidetab

Màn hình chính ứng dụng bao gồm những thông tin quan trọng sau:

1. Nút Menu: Khi nhấn vào sẽ hiển thị thanh SideTab như hình 4.2, ở đó người dùng có thể chuyển hướng đến các màn hình khác nhau mà mình muốn.
2. Thanh tìm kiếm: Sử dụng để tìm kiếm trên cơ sở dữ liệu, sẽ chuyển hướng đến trang tìm kiếm khi nhấn vào.
3. Danh mục hiển thị: Sẽ xuất hiện nhiều danh mục khác nhau trên trang chủ, mục đích là để hiển thị thông tin về những bài hát theo từng chủ đề khác nhau. Hệ thống để xuất nội dung sẽ hiển thị danh sách các bài hát thịnh hành ở đây.
4. Miniplayer: Một bản thể nhỏ hơn của trang phát nhạc với các thông tin cơ bản về bài hát cùng với những cử chỉ tác động đến bài hát hiện đang phát như Tạm dừng, chuyển bài...
5. Các Tab truy cập nhanh: Có vai trò tương tự như thanh SideTab nhưng ở đây chuyển hướng tới những chức năng mà người dùng sẽ sử dụng nhiều.

Các mục 1, 4 và 5 là các widget riêng biệt và được xuất hiện trong nhiều trang với chức năng tương tự.

4.1.2 Màn hình YouTube



Hình 4.3: Màn hình YouTube

Hình 4.4: Thông tin Album của YouTube

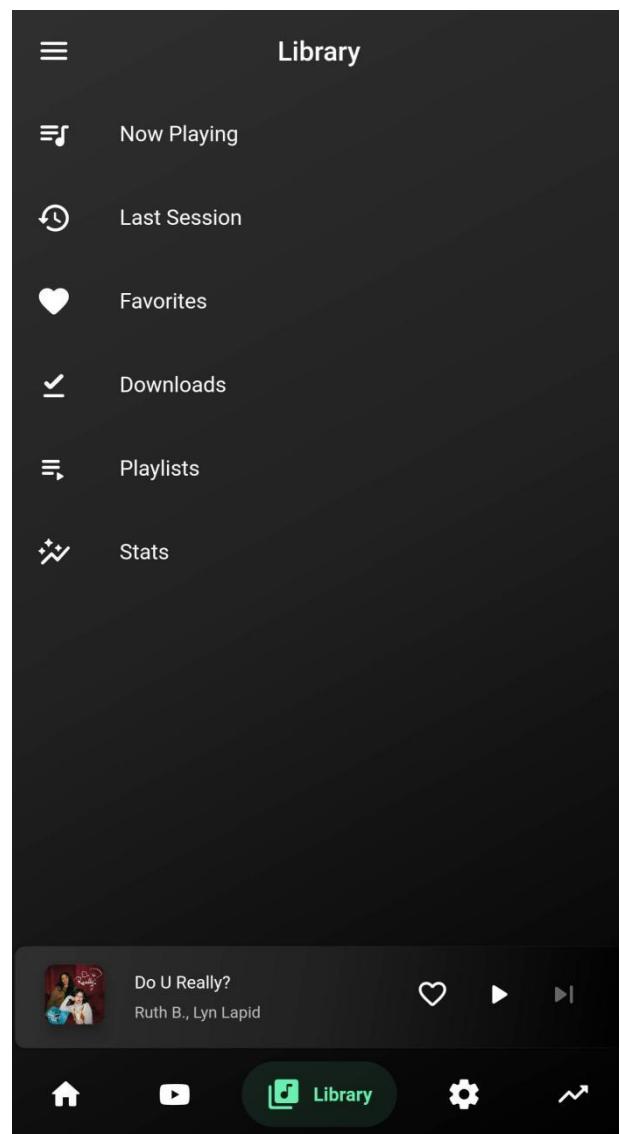
Màn hình này bao gồm:

- Thanh tìm kiếm: Sử dụng để tìm kiếm thông qua YouTube API, sẽ chuyển hướng đến trang tìm kiếm khi nhấn vào.
- Album do YouTube tổng hợp: Sẽ xuất hiện nhiều album khác nhau được phân loại theo những nền âm nhạc nổi tiếng như K-Pop, US-UK... Chi tiết thông tin và các bài hát trong đó sẽ xuất hiện như hình 4.4 sau khi người dùng nhấn vào xem album.

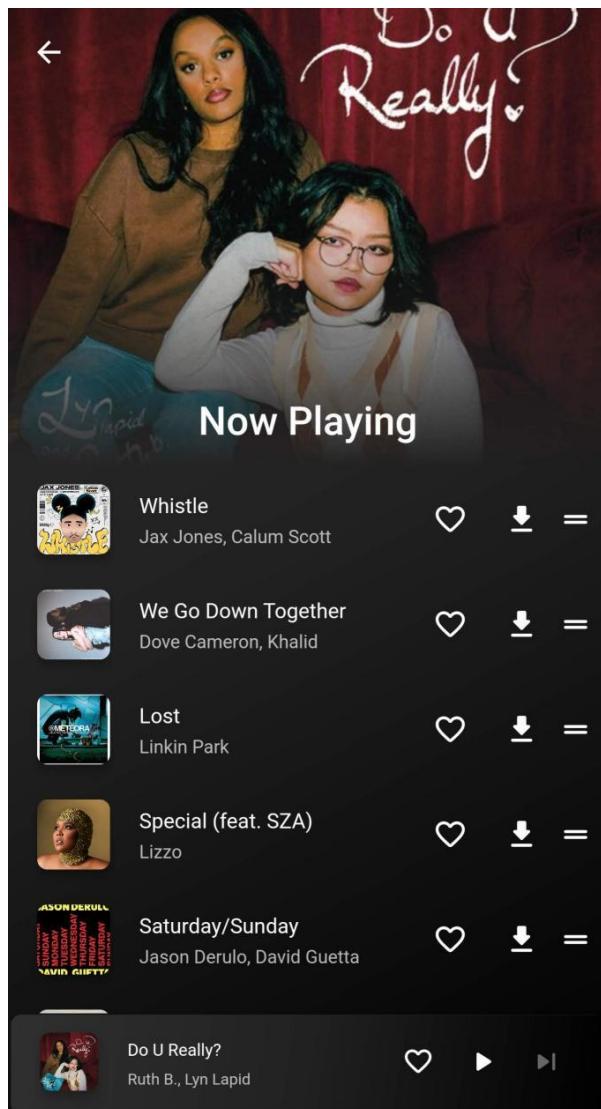
4.1.3 Thư viện

Màn hình này được dùng để quản lý các danh sách bài hát. Cụ thể:

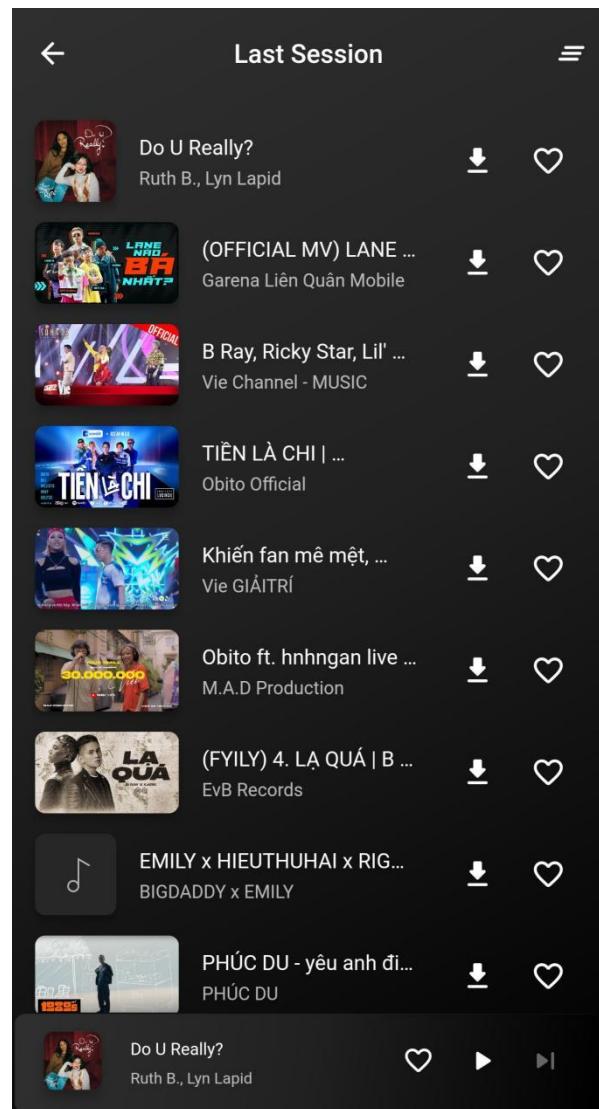
1. Danh sách đang phát: Khi ứng dụng đang phát 1 bài nhạc bất kì thì danh sách này sẽ hiển thị danh sách các bài hát liên quan dưới dạng hàng đợi chi tiết ở hình 4.6.
2. Danh sách nghe gần đây: Lưu lại lịch sử các bài hát được nghe gần nhất, chi tiết ở hình 4.7.
3. Danh sách bài hát yêu thích: Các bài hát được người dùng “thả tim”, chi tiết ở hình 4.8.
4. Danh sách các bài hát đã tải xuống
5. Quản lý danh sách phát: Tùy chỉnh các danh sách phát của người dùng, chi tiết ở hình 4.9.
6. Thống kê: Nêu ra số lượt nghe và bài hát được nghe nhiều nhất.



Hình 4.5: Màn hình Thư viện

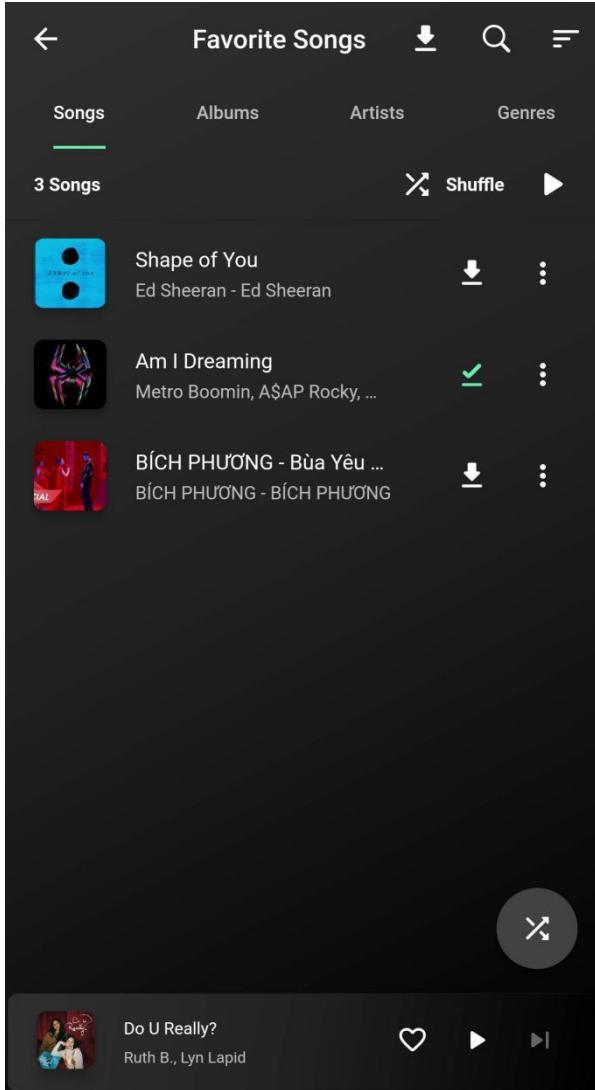


Hình 4.5: Danh sách đang phát

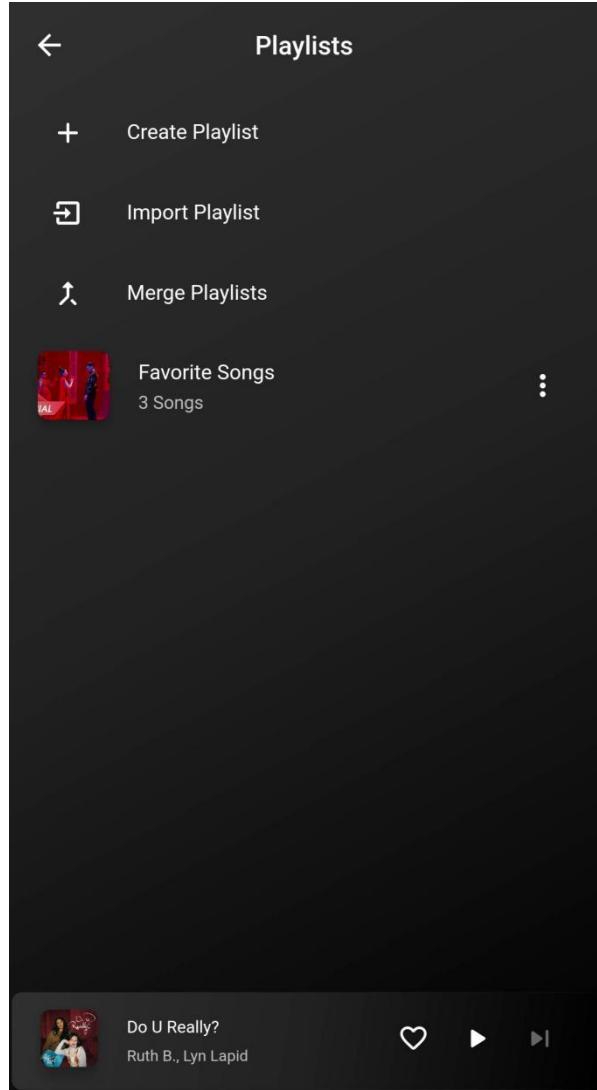


Hình 4.6: Danh sách gần đây

Mỗi màn hình đều có các tùy chọn để tải xuống và thêm vào danh sách yêu thích.



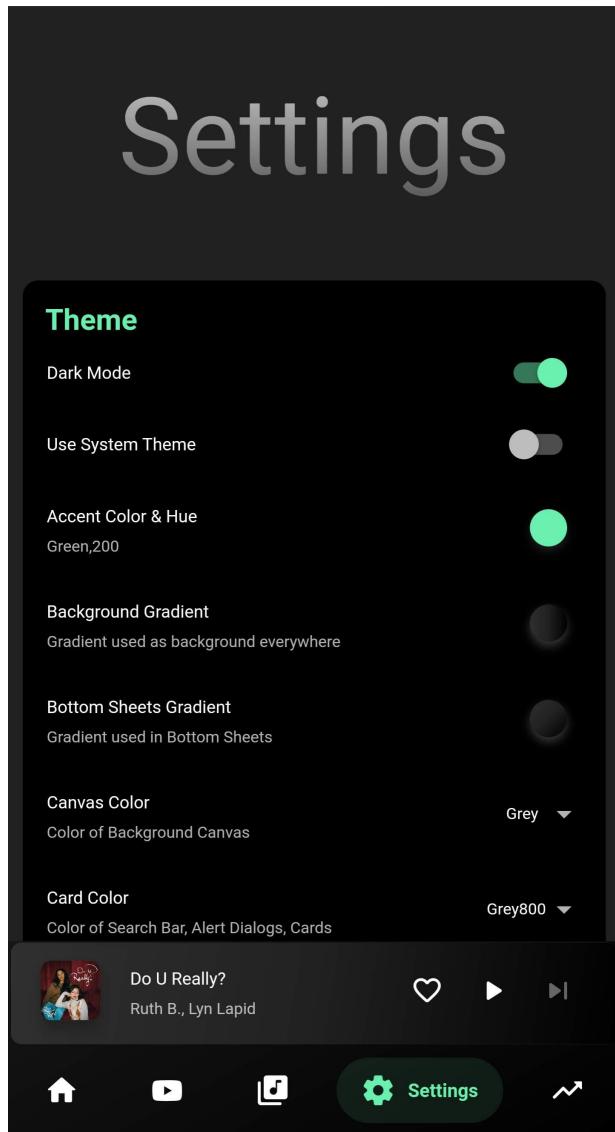
Hình 4.7: Danh sách bài hát yêu thích



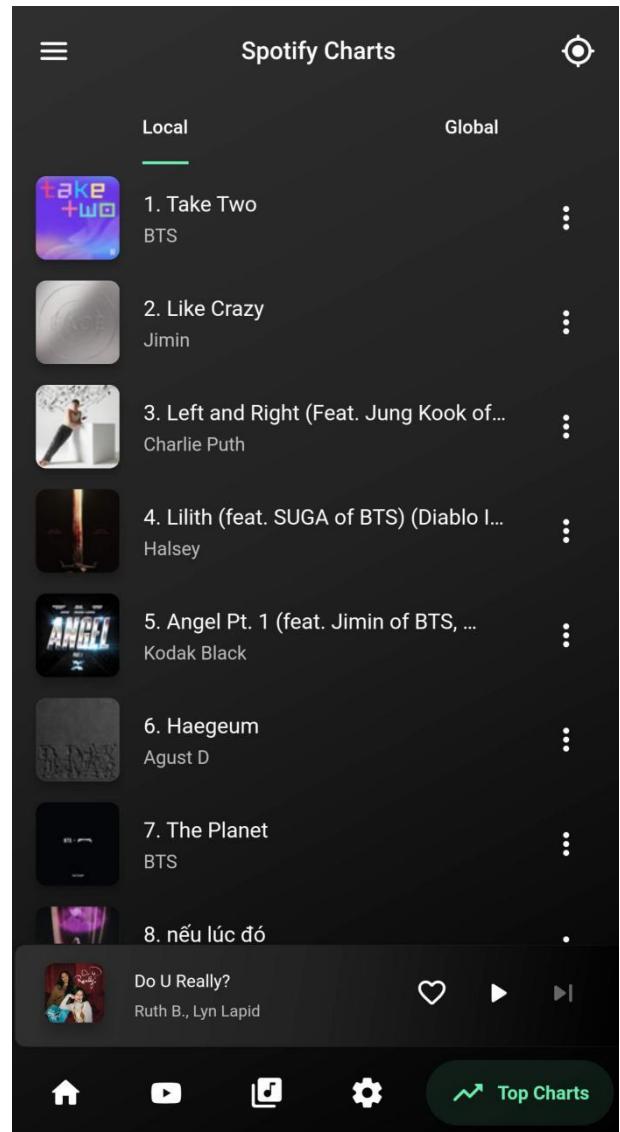
Hình 4.8: Quản lý các danh sách phát

Danh sách yêu thích cũng được coi là một playlist nhưng không thể bị xóa và đổi tên.

4.1.4 Cài đặt và Bảng xếp hạng



Hình 4.10: Màn hình Cài đặt



Hình 4.11: Màn hình Bảng xếp hạng

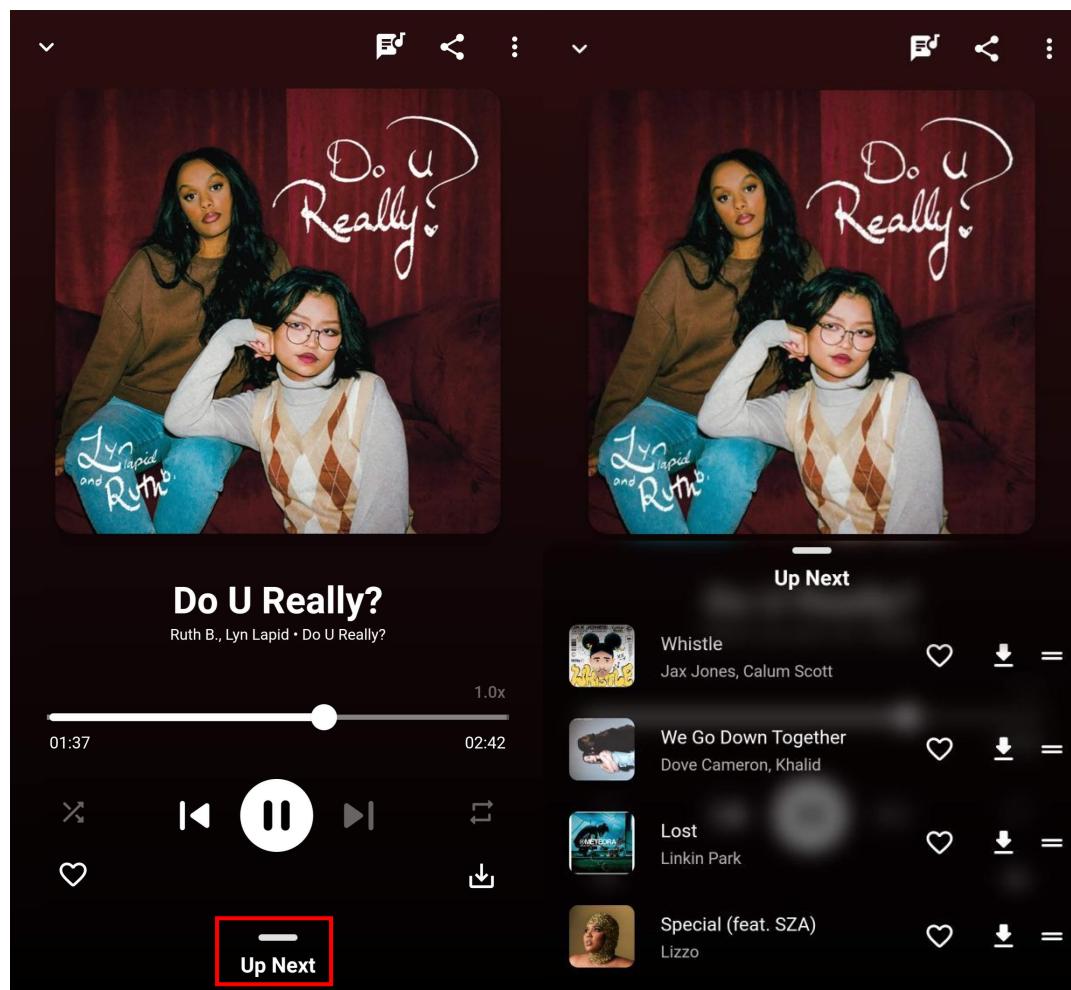
1. Cài đặt: Thay đổi các cài đặt của hệ thống về hiển thị, các tác vụ liên quan đến việc nghe và tải nhạc và một vài tùy chọn bổ trợ.
2. Bảng xếp hạng: Hiển thị danh sách các bài hát thịnh hành theo dữ liệu của Spotify .

4.2 Các chức năng quan trọng

Các chức năng không thể thiếu của một ứng dụng phát nhạc trực tuyến đó chính là nghe nhạc, tìm kiếm bài hát.

4.2.1 Nghe nhạc

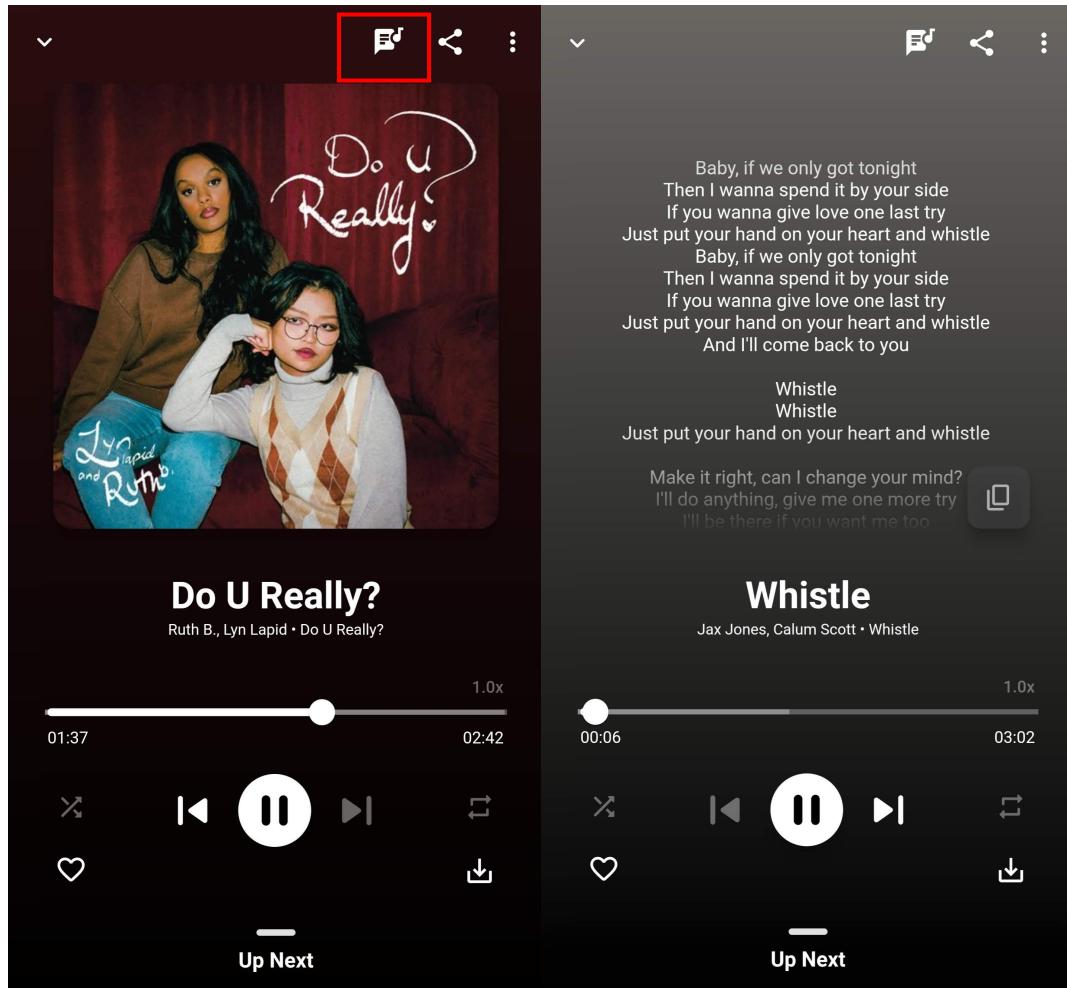
- a) Các bài hát sẽ được phát tiếp theo



Hình 4.12: Danh sách bài hát tương tự

Trong giao diện phát nhạc, khi nhấn vào nút được khoanh vùng để xem những ca khúc gợi ý sẽ hiển thị ra danh sách như hình 4.12 ở danh sách này người dùng có thể tùy chọn thứ tự phát và loại bỏ những bài hát mình không thích ra khỏi danh sách.

b) Xem lời bài hát

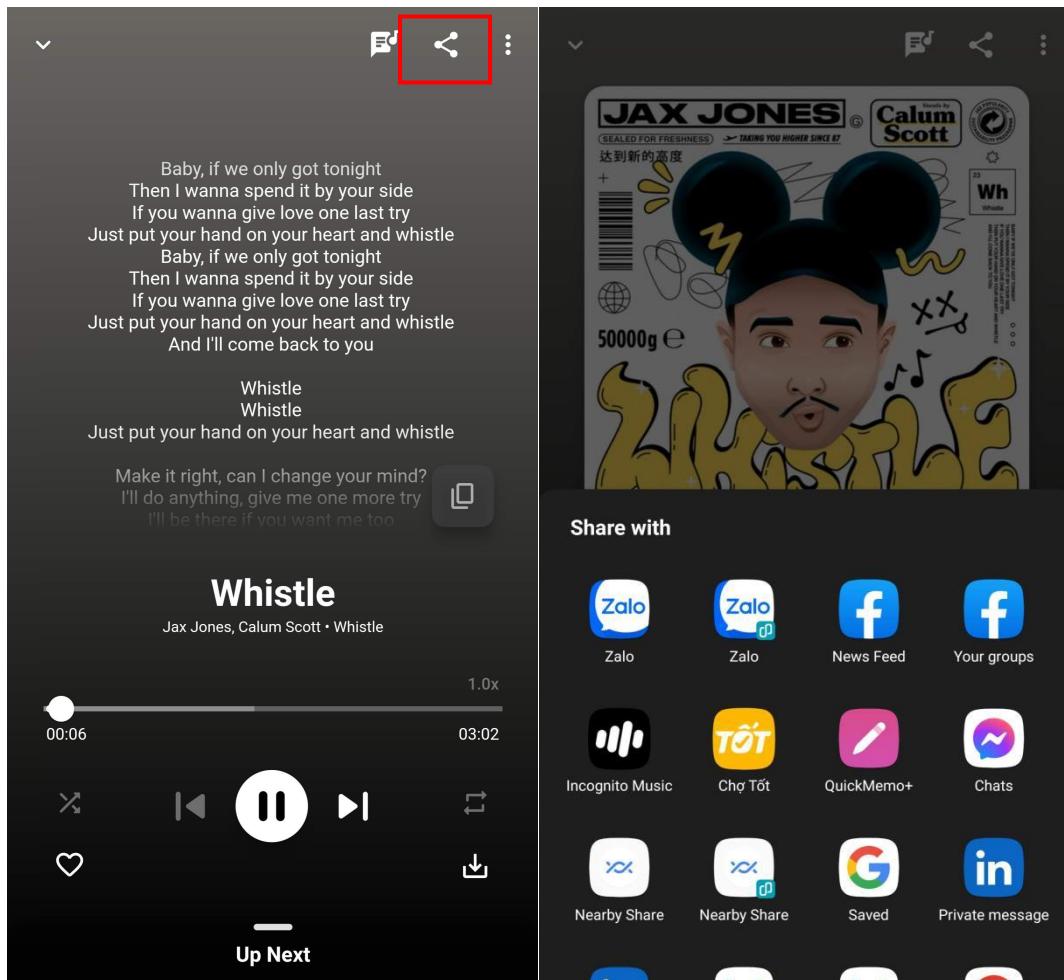


Hình 4.13: Xem lời bài hát

Để xem lời bài hát người dùng nhấn vào nút được khoanh hoặc nhấn đúp vào chính giữa hình ảnh. Lúc này hệ thống sẽ truy vấn lời bài hát và hiển thị lên giao diện.

Lưu ý sẽ có trường hợp không tìm thấy lời.

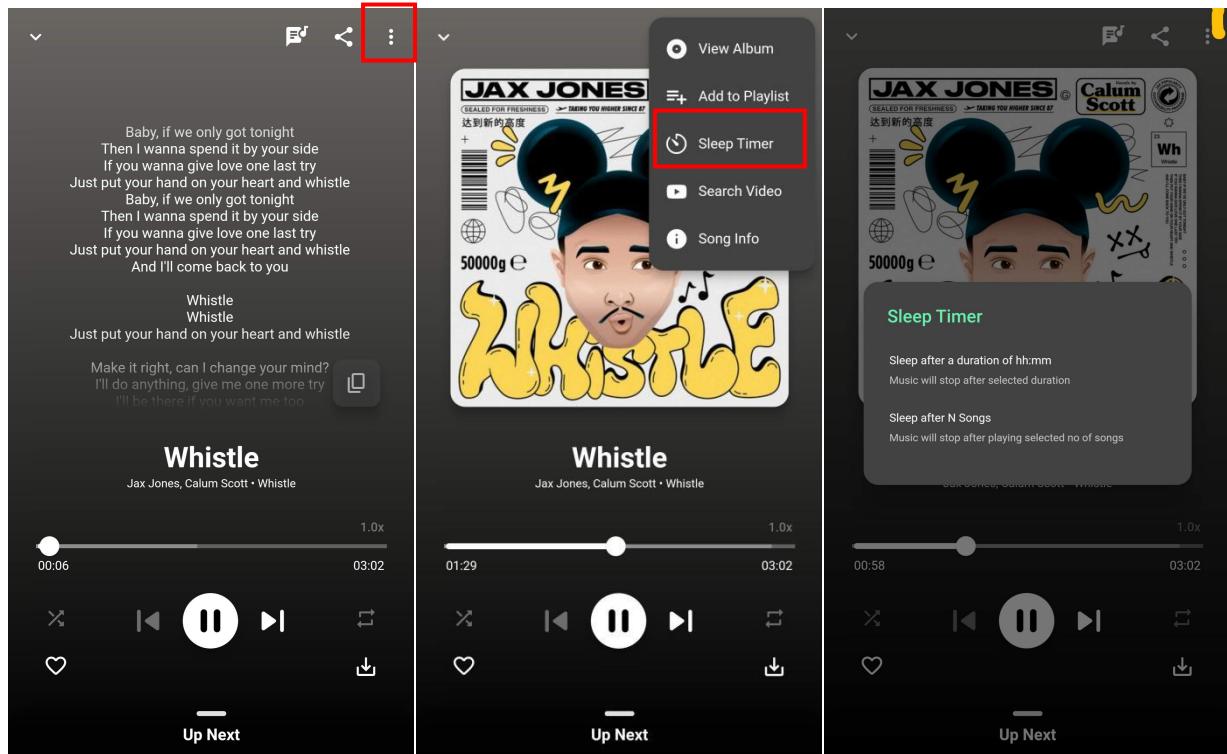
c) Chia sẻ bài hát



Hình 4.14: Chia sẻ bài hát

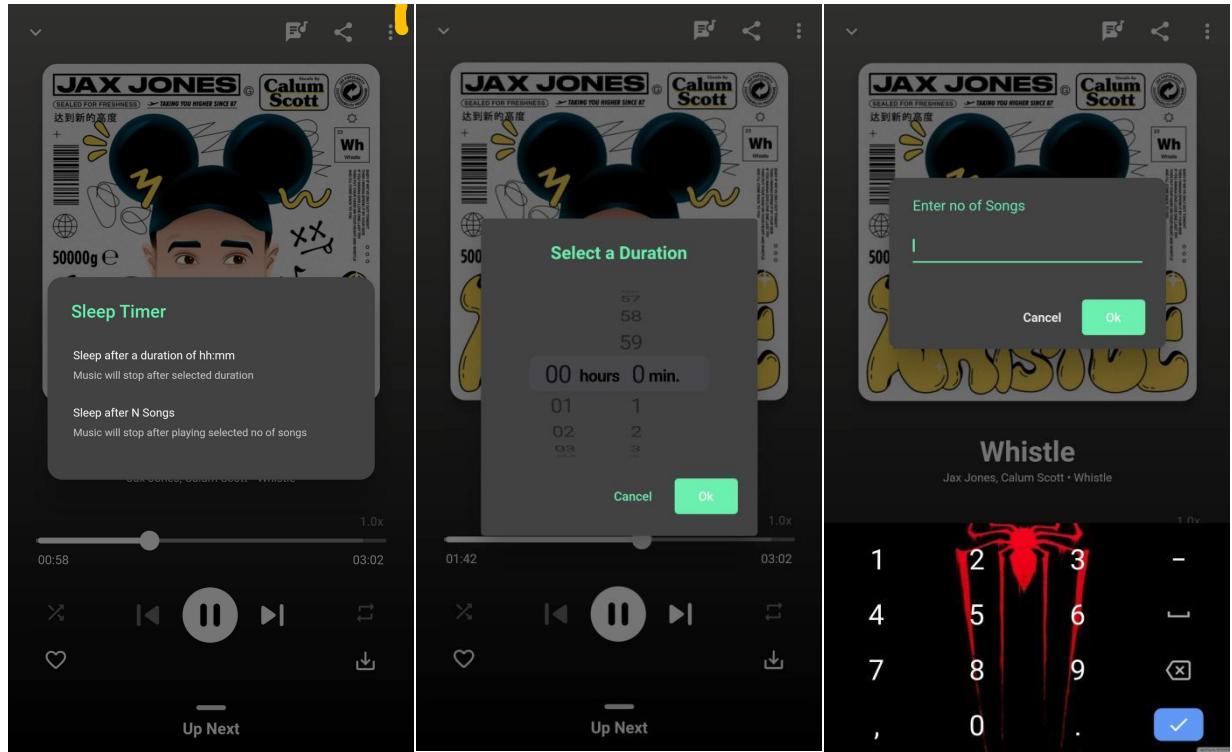
Chức năng chia sẻ bài hát lên mạng xã hội sẽ được kích hoạt khi người dùng nhấn vào nút được khoanh và lựa chọn người muốn chia sẻ. Lúc này hệ thống sẽ gửi cho người nhận đường dẫn phát ca khúc đó.

d) Hẹn giờ tắt nhạc



Hình 4.15: Hẹn giờ tắt nhạc

Thực hiện theo các nút được khoanh trên hình 4.15 từ trái sang phải ta được kích hoạt được tính năng hẹn giờ tắt nhạc. Tính năng này giúp người dùng kiểm soát được mức độ hao tốn tài nguyên của điện thoại đi kèm với việc nghe nhạc vừa đủ. Có hai phương thức hẹn giờ đó là hẹn theo một khoảng thời gian cụ thể và hẹn theo số lượng bài hát cụ thể.

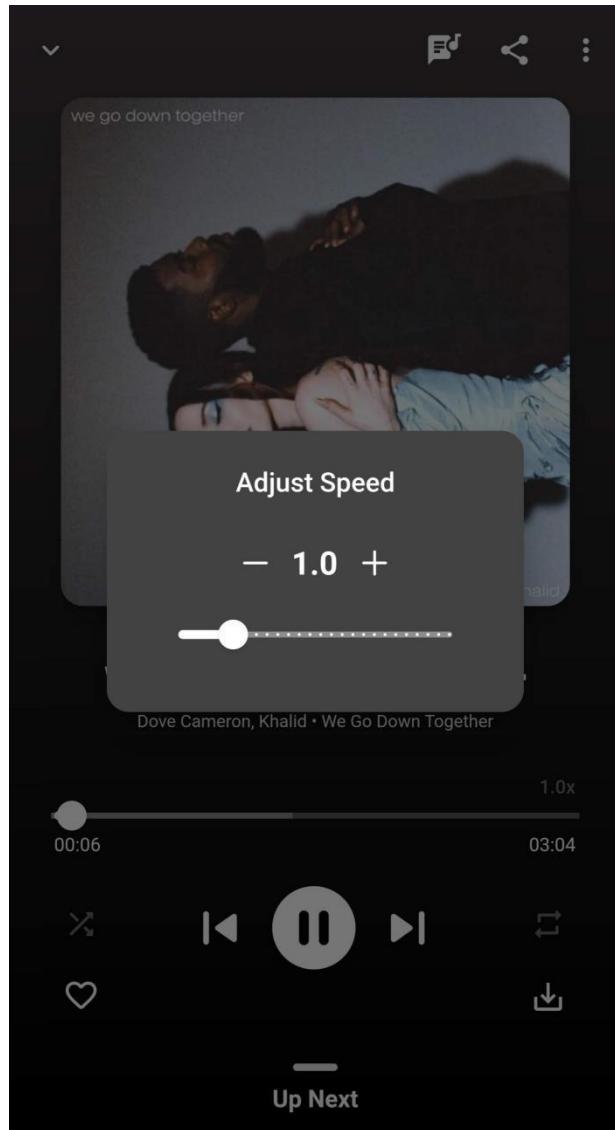


Hình 4.16: Lựa chọn phương thức hẹn giờ

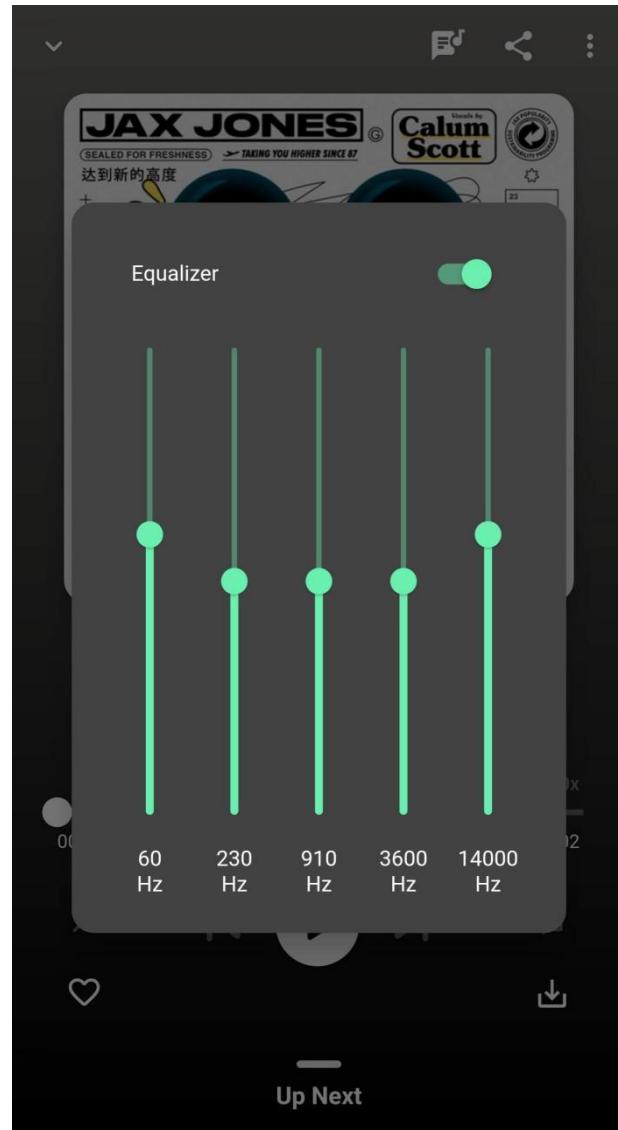
Chọn một trong hai phương thức:

1. Chọn hẹn giờ với một khoảng thời gian cụ thể và điều chỉnh mức thời gian.
2. Chọn hẹn giờ theo số lượng bài hát và nhập số lượng bài hát.

e) Tùy chỉnh tốc độ phát và cường độ âm thanh



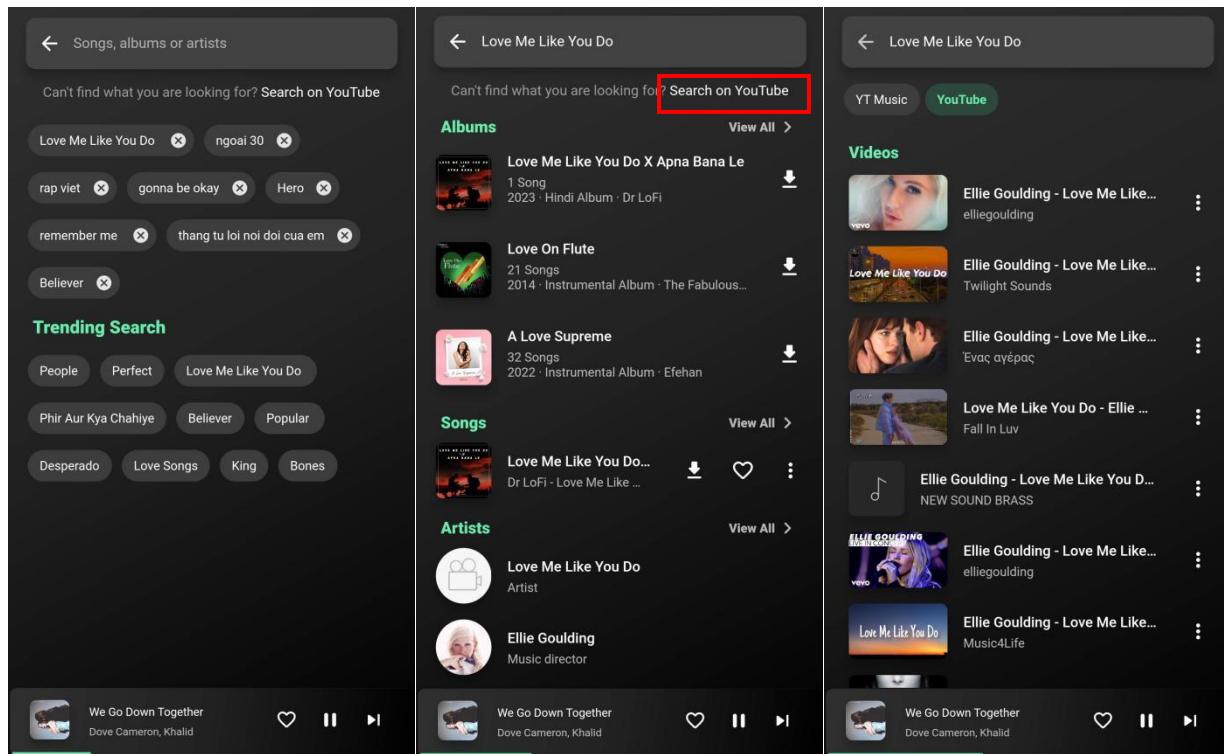
Hình 4.17: Tùy chỉnh tốc độ phát



Hình 4.18: Bộ điều khiển cường độ

Hai chức năng phục vụ chủ yếu theo nhu cầu nghe nhạc của mỗi cá nhân.

4.2.2 Tìm kiếm



Hình 4.19: Quá trình tìm kiếm bài hát

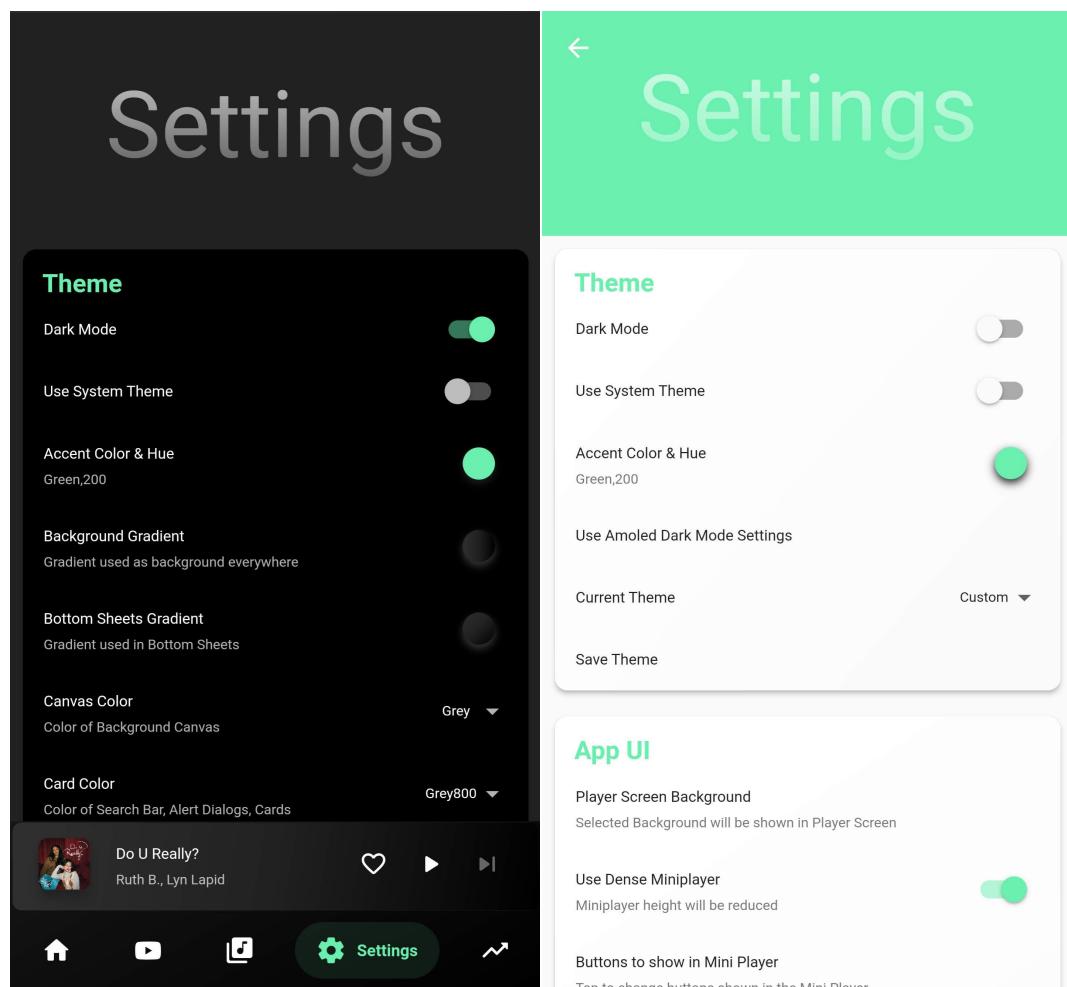
Ngoài trừ thanh tìm kiếm ở màn hình YouTube sẽ trả về thắng kết quả từ dữ liệu của YouTube, tất cả tính năng tìm kiếm đều thực hiện tuần tự như hình 4.19. Cụ thể:

1. Nhập từ khóa vào thanh tìm kiếm
2. Hệ thống trả về kết quả từ cơ sở dữ liệu trên FRD
3. Nếu người dùng chưa hài lòng với kết quả (do dữ liệu của hệ thống chưa đủ nhiều) có thể nhấn vào nút được khoanh ở hình 4.19.
4. Hệ thống sẽ dùng API và trả về kết quả từ dữ liệu của YouTube - một nguồn dữ liệu vô cùng phong phú không chỉ về nhạc mà là về đa phương tiện nói chung.

4.2.3 Tùy chỉnh cài đặt

Có hai tính năng tiêu biểu sẽ có nhiều người dùng tới đó là đổi màu nền và đổi ngôn ngữ cho ứng dụng của mình.

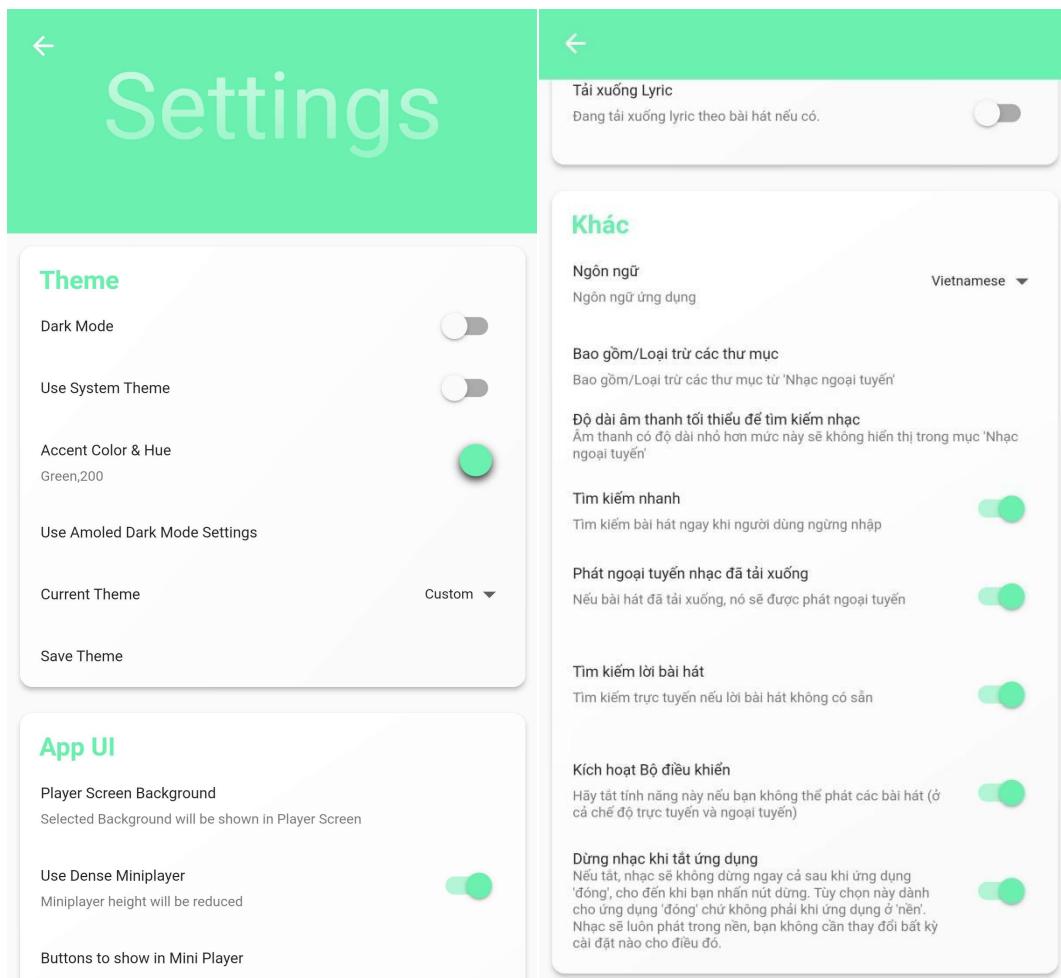
a) Đổi nền



Hình 4.20: Tùy chọn thay đổi nền

Ví dụ ở hình 4.20, một công tắc dùng để chuyển đổi giữa chế độ sáng và tối khác nhau, ngay lập tức nền của ứng dụng đã thay đổi phụ thuộc vào lựa chọn của người dùng - mặc định sẽ là màu tối.

b) Đổi ngôn ngữ



Hình 4.21: Tùy chọn thay đổi ngôn ngữ

Chúng ta có thể thấy khi đổi ngôn ngữ sang Tiếng Việt thì văn bản trong ứng dụng đã chuyển đổi theo.

KẾT LUẬN VÀ KIẾN NGHỊ

Kết quả đạt được

Trong quá trình làm đồ án, em đã học được rất nhiều kiến thức mới cũng như củng cố được những kiến thức cũ. Kết quả em đạt được cụ thể như sau:

Về chuyên môn, em đã làm quen với việc sử dụng Flutter để thiết kế và phát triển được một ứng dụng đáp ứng các chức năng yêu cầu đề ra. Biết cách sử dụng Python xây dựng hệ thống để xuất nội dung, tích hợp các thuật toán để xử lý dữ liệu và tạo API. Học và vận dụng thành công NoSQL vào một bài toán thực tế.

Về thái độ, đồ án đã giúp em nâng cao thêm khả năng xử lý logic và giải quyết vấn đề trong lập trình. Cải thiện khả năng đọc tài liệu và học những công nghệ mới trong thời gian ngắn. Nâng cao tính tập trung trong công việc.

Hạn chế

Vì thời gian hạn hẹp và bản thân chưa được trau dồi nhiều nên hệ thống của em vẫn có một vài tồn tại, cụ thể:

Về hệ thống để xuất nội dung, dữ liệu chưa thực sự nhiều và đa dạng dẫn đến việc hệ thống hoạt động với mức độ chính xác chưa ổn định.

Về ứng dụng, hiện tại chỉ chú trọng tối ưu hóa trên Android.

Kiến nghị

Trong tương lai, em sẽ cố gắng cải thiện những hạn chế kể trên. Ngoài ra, em cũng có một vài ý tưởng để cải tiến ứng dụng của mình. Cụ thể:

Vận dụng thêm kỹ thuật collaborative filtering vào hệ thống để xuất nội dung để tăng thêm trải nghiệm cho người dùng.

Phát triển chức năng cho người dùng đăng tải bài hát của chính mình và hướng đến một hệ thống giải trí đa nền tảng.

PHỤ LỤC

Phụ lục 1: Hướng dẫn cài đặt

- Nhấn vào [link](#) này để cài đặt
- Quét mã QR để cài đặt



Phụ lục 2: Đường dẫn Github : [IncognitoMusic](#)



TÀI LIỆU THAM KHẢO

- [1]. Marcus D. Bloice and Andreas Holzinger, *A Tutorial on Machine Learning and Data Science Tools with Python*, Holzinger Group HCI-KDD, Institute for Medical Informatics, Statistics and Documentation, Medical University of Graz, Austria, 2016.
- [2]. Andreas C.Müller & Sarah Guido, *Introduction to Machine Learning with Python*, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, 2017.
- [3]. Marine Chemeque-Rabe, *Content-based music recommendation system: A comparison of supervised Machine Learning models and music features*, STOCKHOLM, SWEDEN, 2020.
- [4]. Macro L.Napoli, *Beginning Flutter®: A Hands On Guide To App Development*, Join Weily & Sons, Inc., 2020.
- [5]. [Online], “Flutter Tutorial - Tutorialsport”, Available: [“https://www.tutorialspoint.com/flutter/flutter_tutorial.pdf”](https://www.tutorialspoint.com/flutter/flutter_tutorial.pdf) [Accessed 18/02/2023].
- [6]. [Online], “Firebase Realtime Database (google.com)”, Available: [“https://firebase.google.com/docs/database/”](https://firebase.google.com/docs/database/), [Accessed 04/04/2023].
- [7]. [Online], “Machine Learning cơ bản”, Available: [“https://machinelearningcoban.com/2017/05/17/contentbasedrecordersys/”](https://machinelearningcoban.com/2017/05/17/contentbasedrecordersys/), [Accessed 08/05/2023].
- [8]. [Online], “Recommendation Overview”, Available: [“https://developers.google.com/machine-learning/recommendation/overview”](https://developers.google.com/machine-learning/recommendation/overview), [Accessed 20/04/2023].
- [9]. [Online], “Dart Tutorial - Tutorialsport”, Available: [“https://www.tutorialspoint.com/dart_programming/dart_programming_tutorial.pdf”](https://www.tutorialspoint.com/dart_programming/dart_programming_tutorial.pdf) [Accessed 18/02/2023].