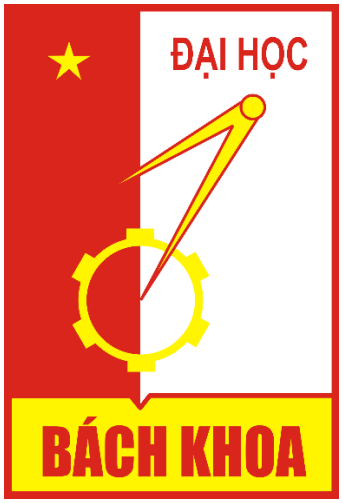
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**




# BÁO CÁO ĐỀ TÀI MÔN HỌC

# HỆ GỢI Ý

**Đề tài:**

## Xây dựng hệ gợi ý nhạc dựa trên lọc cộng tác sử dụng Multinomial Variational Auto-encoder

Giáo viên hướng dẫn: THS. Ngô Văn Linh

Nhóm sinh viên thực hiện: 1. Nguyễn Trí Quân - 20173312

2. Bùi Thị Hằng - 20173097

3. Lê Công Nguyên – 20173290

4. Mai Văn Hòa - 20173122

Hà Nội, tháng 12 năm 2020

Mục lục

**I. Tổng quan đề tài** ..... 2

**II. Dữ liệu sử dụng** ..... 3

**1. Tổng quan dữ liệu** ..... 3

**3. Tách dữ liệu cho huấn luyện, tối ưu, thử nghiệm** ..... 7

**III. Mô hình lựa chọn** ..... 8

**1. Variational Autoencoders (VAE)** ..... 9

**2. Lọc cộng tác sử dụng Mult-VAE** ..... 10

**IV. Kết quả thực nghiệm** ..... 11

**V. Tài liệu tham khảo** ..... 12

# I. Tổng quan đề tài

Hệ thống gợi ý (Recommender systems – RS) đang từng bước trở thành một lĩnh vực nghiên cứu quan trọng và được ứng dụng khá thành công trong thực tiễn, giúp người dùng đối phó với vấn đề quá tải thông tin. Hiện nay, RS đã và đang được nghiên cứu và ứng dụng trong nhiều lĩnh vực khác nhau như: thương mại điện tử (bán hàng trực tuyến), giải trí (phim ảnh, âm nhạc...), giáo dục đào tạo (gợi ý nguồn tài nguyên học tập như sách, báo...). Trên thế giới, đã có nhiều công ty, tổ chức áp dụng thành công hệ thống gợi ý, nhằm gợi ý các dịch vụ, sản phẩm và các thông tin cần thiết đến người dùng như: website mua sắm trực tuyến Amazon.com gợi ý cho mỗi khách hàng những sản phẩm mà họ có thể quan tâm, YouTube.com giới thiệu các video clip cho người xem, gợi ý phim ảnh của Netflix.com, MovieLens.org và gợi ý nhạc của Last.fm... Điều này góp phần làm tăng doanh số bán hàng hoặc số lượng truy cập, download của hệ thống, đồng thời giúp cho khách hàng có thể tìm kiếm được những thông tin thú vị hoặc những sản phẩm mà họ mong muốn dễ dàng hơn.

Cùng với sự phát triển mạnh mẽ của các loại hình truyền thông đa phương tiện thì âm nhạc là một trong những nội dung khá phổ biến và được xem như là một nhu cầu không thể thiếu trong cuộc sống, có thể chia sẻ bởi nhiều người từ nhiều quốc gia có ngôn ngữ và nền văn hóa khác nhau. Tuy nhiên, số lượng bài nhạc đang ngày càng tăng lên, đa dạng và phong phú cả về nội dung lẫn thể loại. Vì vậy, vấn đề đặt ra là khi một người sử dụng muốn tìm nghe những bài nhạc mà mình yêu thích, người sử dụng sẽ cần đến công cụ tìm kiếm Google hoặc vào một website về âm nhạc để tìm nghe. Mặc dù vậy, ở đó có nhiều bản nhạc mà người sử dụng sẽ không thể nghe thử hết để tìm ra những bài mà họ thích, điều này tốn thời gian mà lại không hiệu quả. Do đó, nhu cầu cần có một hệ thống gợi ý có khả năng dự đoán mức độ ưa thích của người sử dụng với từng bản nhạc và gợi ý cho họ các bản nhạc mới mà hệ thống cho là phù hợp.

## II. Dữ liệu sử dụng

### 1. Tổng quan dữ liệu

Đề tài chúng em sử dụng các tập dữ liệu đều được download trực tiếp từ trang chủ *http://millionsongdataset.com* , bao gồm:

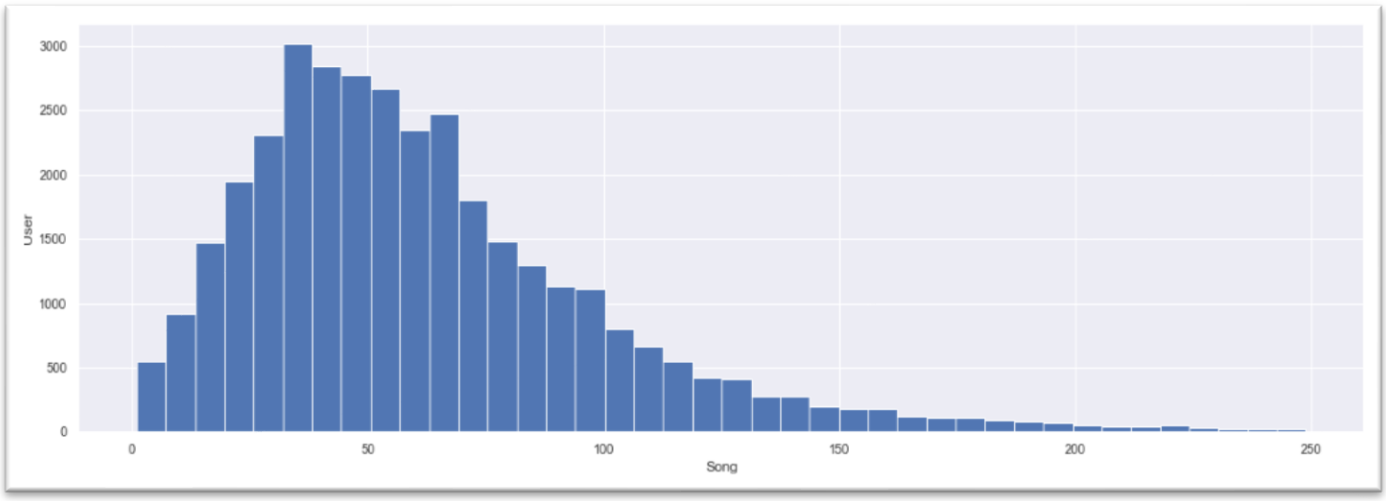
- Million Song Dataset (MSD): Tập dữ liệu này gồm 1.000.000 bài hát với các thông tin về:
  - song_id
  - title (tên bài hát)
  - release (album)
  - artist_name (tên nghệ sĩ)
  - year (năm phát hành)

- The Echo Nest Taste Profile Subset: Tập dữ liệu này là lịch sử nghe các bài hát trong tập MSD của 35123 người dùng với 3231 bài hát. Trong đó có 2214283 bản ghi, mỗi bản ghi có các trường thông tin:
  - user_id
  - song_id (id bài hát đã nghe)
  - listen_count (số lần nghe 1 bài hát của user đó)

- last.fm: Tập dữ liệu này gồm hơn 500.000 bài hát với các thông tin về:
  - song_id
  - artist (nghệ sĩ)
  - similar (độ tương đồng của bài hát này với các bài hát khác trong tập dữ liệu, chỉ thống kê các bài hát có độ tương đồng với bài hát đang xét > 0,1%)
  - tags (thể loại của bài hát)
  - title (tên bài hát)

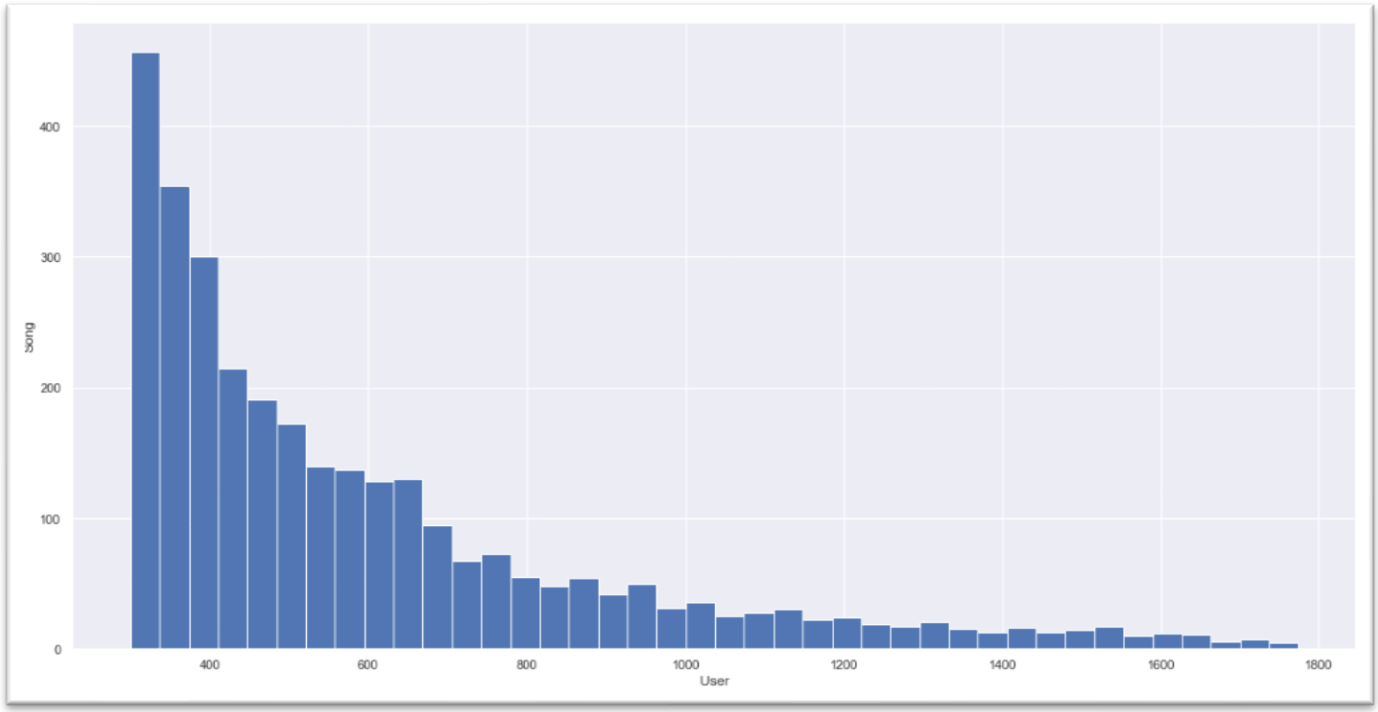
## 2. Phân tích dữ liệu

- Số lượng bài hát mà đa số mọi người nghe nhiều nhất khoảng từ 30 đến 70 bài.



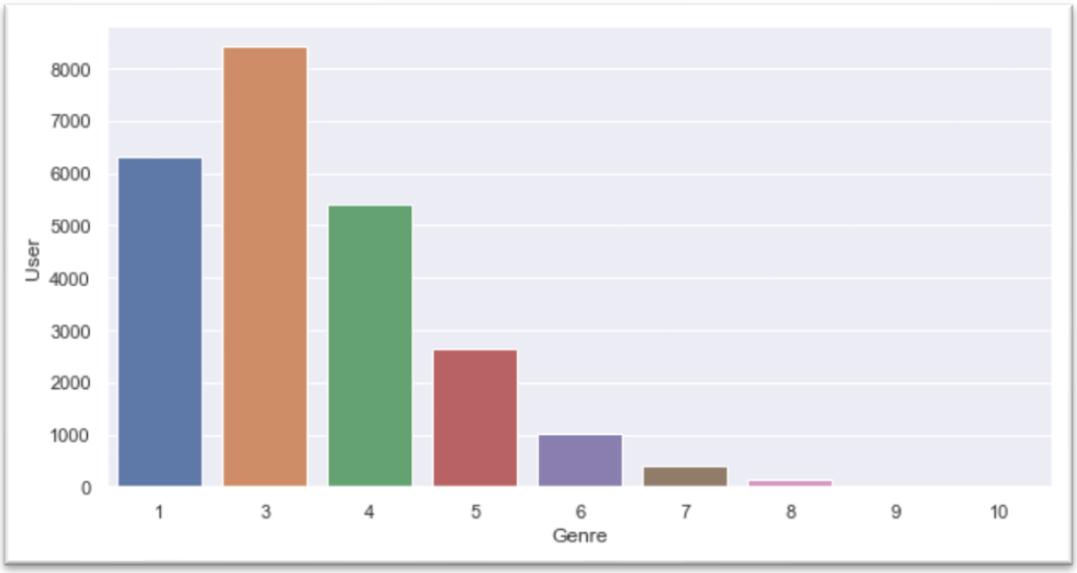

*Hình 1: Thống kê về số bài hát mà một user nghe*

- Nhiều nhất có khoảng 500 bài hát mà mỗi bài được nghe bởi khoảng 200 người.



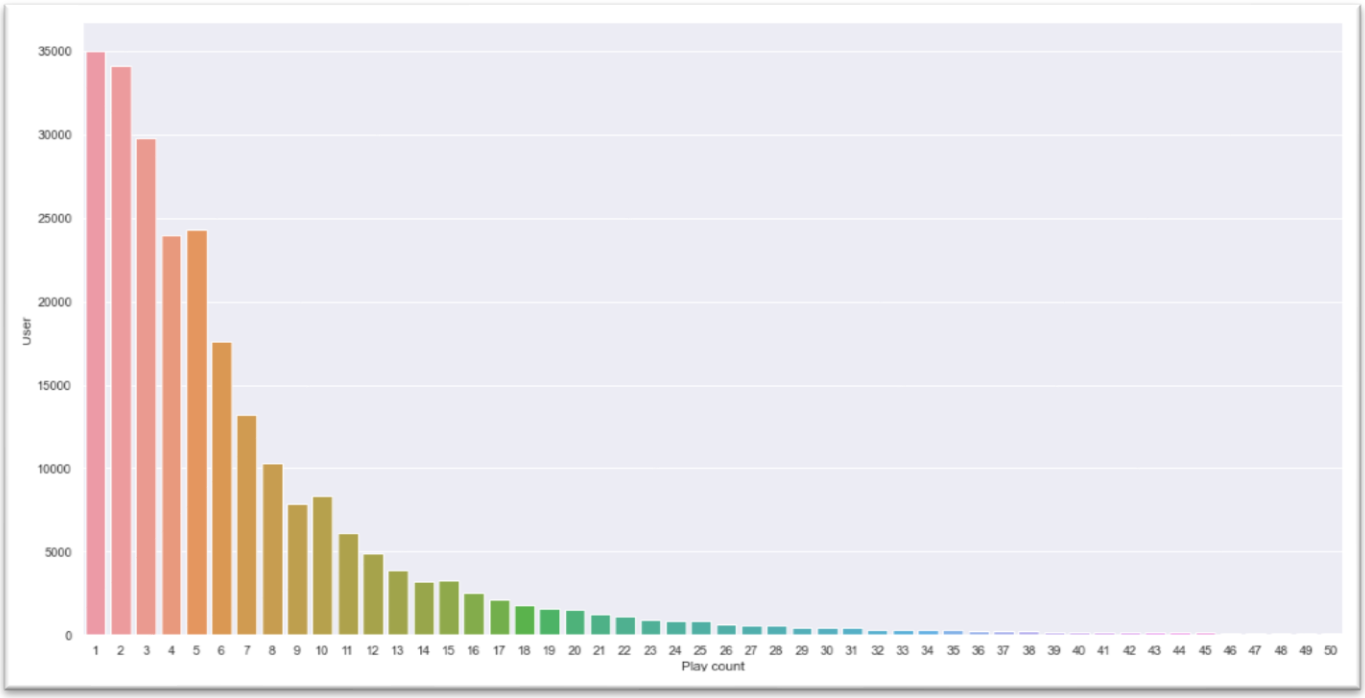

*Hình 2: Thống kê về số user nghe một bài hát*

- Mọi người nghe dưới 8 thể loại, đa số dưới 5 thể loại và nhiều nhất là 3 thể loại.



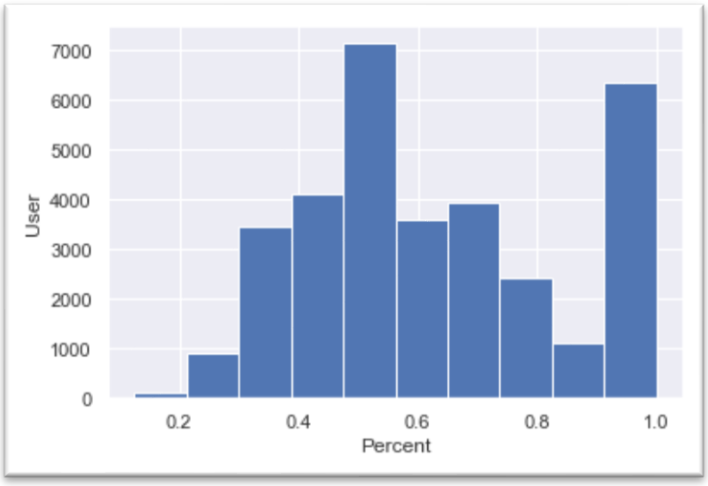

*Hình 3: Thống kê về số thể loại mà một user nghe*

- Đa phần người nghe chỉ nghe từ một đến hai lần một bài hát



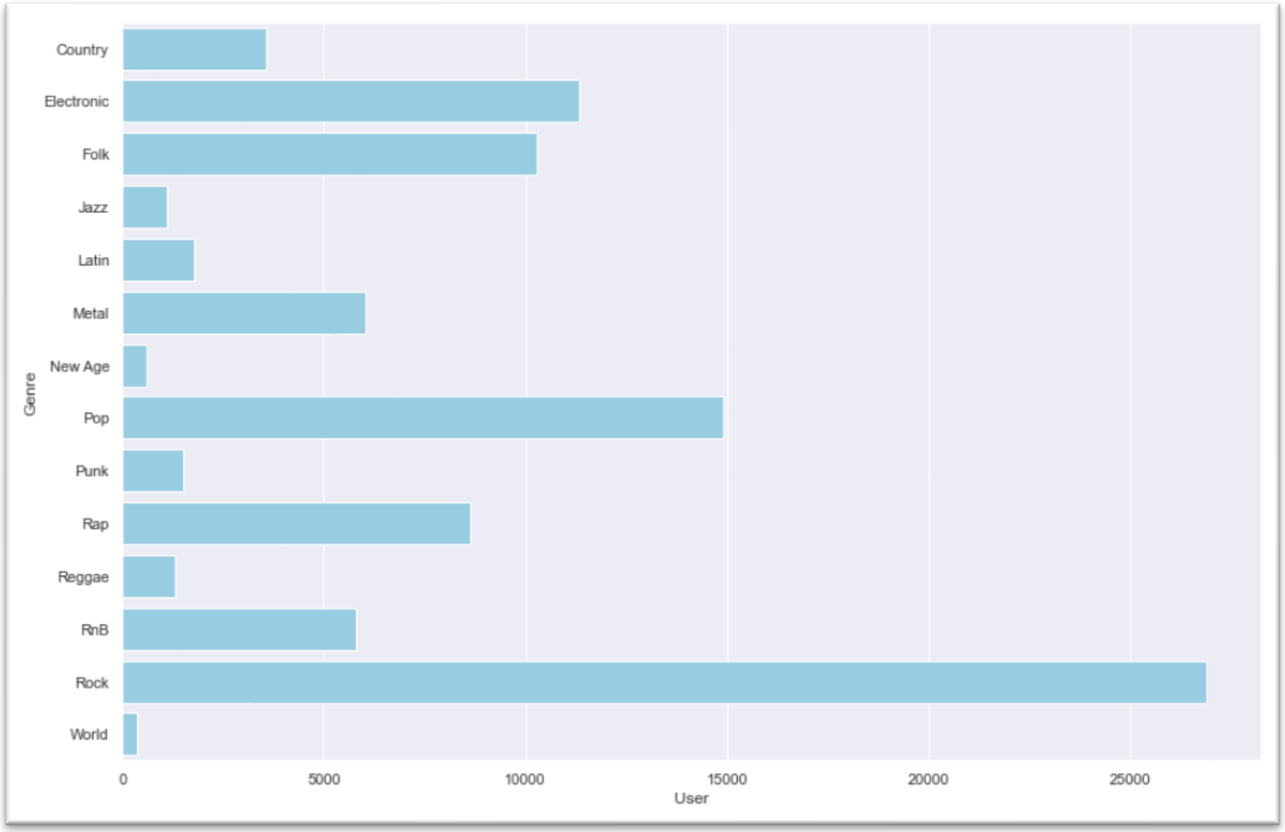

*Hình 4: Thống kê về số lần nghe một bài của user*

- Số lượng user mà 100% chỉ nghe một thể loại nhạc chiếm tỉ lệ lớn.



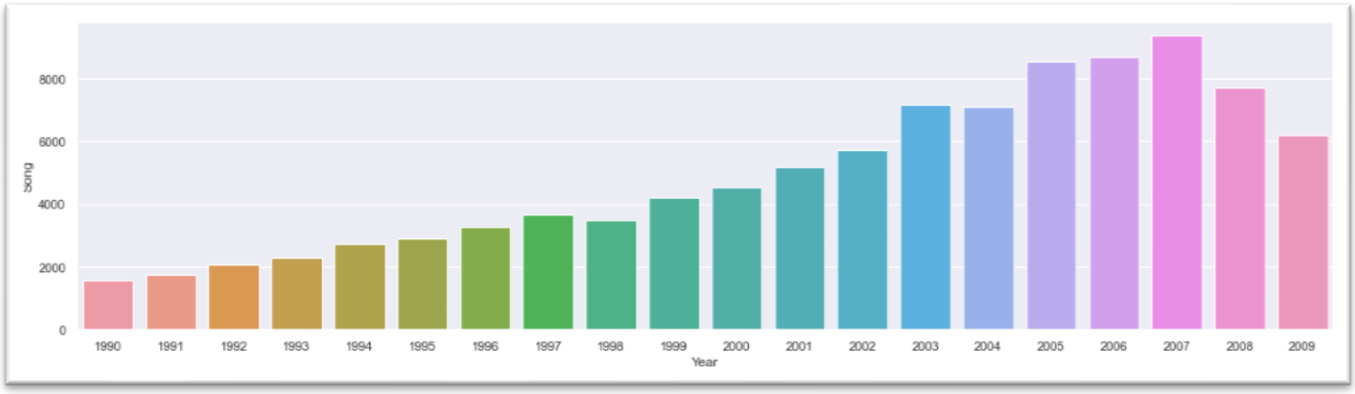

*Hình 5: Thống kê số lượng user theo tỉ lệ thể loại lớn nhất đã nghe*

- Lượng lớn người nghe thích nghe nhạc rock, pop và electronic.




*Hình 6: Thống kê số lượng user nghe theo thể loại nhạc*

- Càng về sau, số lượng bài hát được phát hành mỗi năm có xu hướng tăng lên.




*Hình 7: Thống kê theo năm phát hành của bài hát*

### 3. Tách dữ liệu cho huấn luyện, tối ưu, thử nghiệm

Dữ liệu chúng em sử dụng trong lọc cộng tác ở đây là lịch sử nghe nhạc của người dùng gồm có thông tin người dùng nghe bài hát bao nhiêu lần. Chúng em nhóm các tương tác lại theo người dùng, sau đó chia các tập dữ liệu theo người dùng.

Sử dụng 70% người dùng và toàn bộ lịch sử tương tác của họ để làm dữ liệu huấn luyện

Tập thử nghiệm và tối ưu mỗi tập được tách ra từ 15% số người dùng.

Trong quá trình đánh giá chúng em sẽ cho 80% lượng tương tác của người dùng qua mô hình để dự đoán rating cho người dùng đó và 20% lượng tương tác còn lại sẽ dùng để đánh giá chất lượng gợi ý.

Kết quả sau khi tách; tập huấn luyện có 24523 người dùng cùng tương tác của họ, tập thử nghiệm và tập tối ưu có 5334 và 5266 người dùng.

## III. Mô hình lựa chọn

Các vấn đề mà các hệ gợi ý hay gặp phải đó chính là dữ liệu thưa (số lượng user và item lớn nhưng số lượng tương tác lại ít) và cold-start (hệ thống không gợi ý được cho các user và item mới). Một nhóm các mô hình hệ gợi ý dựa trên auto-encoder đang cho thấy hiệu quả rất tốt trong giải quyết hai vấn đề trên.

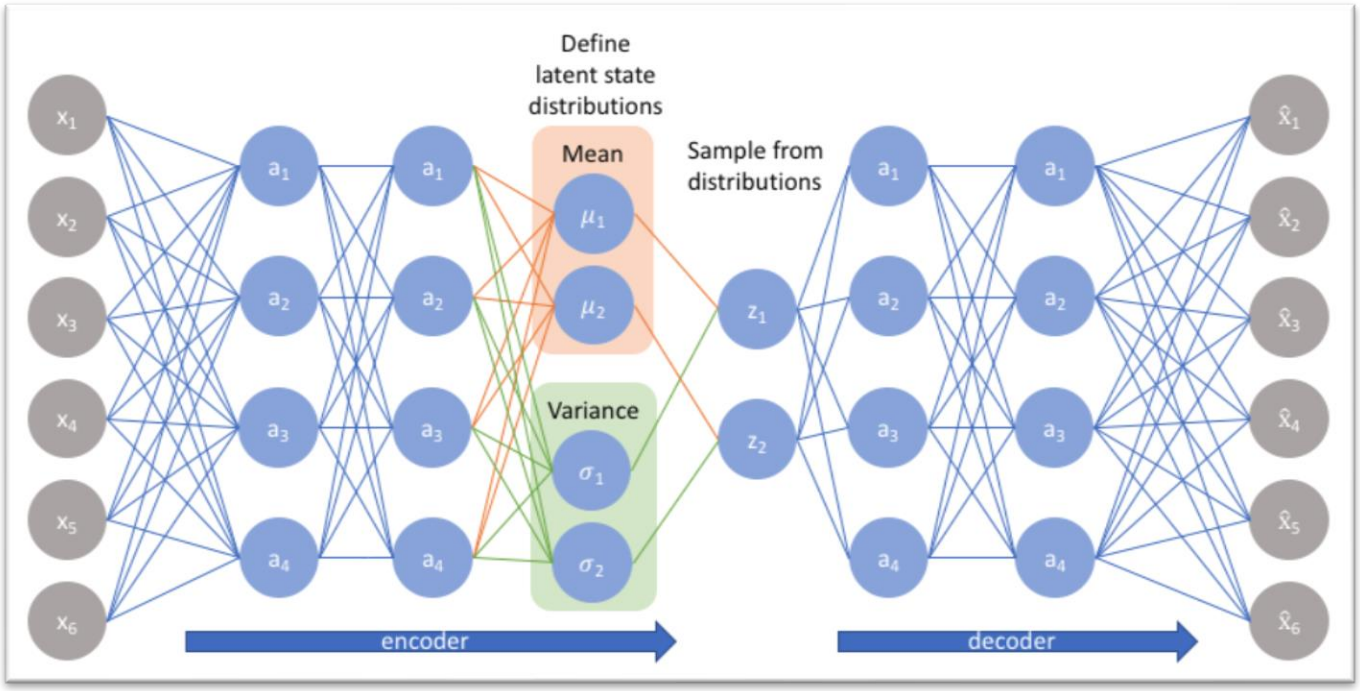
Auto-encoder là một phương pháp học không giám sát sử dụng mạng neural, với ý tưởng là ánh xạ đầu vào có số chiều lớn thành một biểu diễn ẩn ở chiều không gian nhỏ hơn và cố gắng khôi phục lại đầu vào từ biểu diễn ẩn đó. Auto-encoder đang được ứng dụng rất rộng rãi trong nhiều bài toán học không giám sát và gần đây đã được đưa vào sử dụng trong hệ gợi ý qua các mô hình: AutoRec, DeepRec, Collaborative Denoising Auto-encoder (CDAE) và Multinomial Variational Auto-encoder (Mult-VAE).

Các mô hình gợi ý bằng auto-encoder có khả năng lấp đầy các tương tác còn thiếu trong ma trận tương tác giữa user và item từ đó rất thích hợp trong việc đưa ra top K recommendation. Không những có thể làm việc tốt với dữ liệu thưa mà auto-encoder còn có thể giải quyết vấn đề cold-start với user.

Multinomial Variational Auto-encoder (Mult-VAE) đang là một trong các phương pháp có ảnh hưởng nhất trong số các phương pháp dựa trên hướng tiếp cận lọc cộng tác sử dụng auto-encoder. Vì vậy nhóm chúng em quyết định xây dựng hệ gợi ý nhạc dựa trên Mult-VAE. Mô hình và cài đặt thực nghiệm chúng em tham khảo từ bài báo Variational Autoencoders for Collaborative Filtering của Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, Tony Jebara [1].

## 1. Variational Autoencoders (VAE)

Variational Autoencoders (VAE) là một phương pháp giảm chiều dữ liệu bằng cách sử dụng mạng neural và phương pháp variational inference để xác định các tham số.




*Hình 8: Kiến trúc của VAE [2]*

Kiến trúc của VAE gồm hai mạng encoder và decoder được mô tả như ở Hình 8. Đầu vào $X$ ở chiều không gian lớn hơn được ánh xạ sang $Z$ có số chiều nhỏ hơn bởi encoder. Nhiệm vụ của decoder là khôi phục lại $X$ ban đầu từ $Z$. Với hướng tiếp cận Bayesian, $Z$ được coi như một biến ngẫu nhiên tuân theo phân phối Gauss với mean và variance được học từ mạng encoder. Như vậy với mỗi đầu vào $X$, ta sẽ tìm được một phân phối cho biểu diễn ẩn $Z$ tương ứng với $X$. Từ phân phối tìm được sẽ lấy mẫu để sinh ra $Z$. Từ $Z$ thông qua decoder sẽ khôi phục lại được $X$.

Hàm loss của VAE:

$$\mathcal{L}_\beta(\mathbf{x}_u;\theta,\phi) \equiv \mathbb{E}_{q_\phi(\mathbf{z}_u \mid \mathbf{x}_u)}[\log p_\theta(\mathbf{x}_u \mid \mathbf{z}_u)] - \beta \cdot \mathrm{KL}(q_\phi(\mathbf{z}_u \mid \mathbf{x}_u) \| p(\mathbf{z}_u)).$$

## 2. Lọc cộng tác sử dụng Mult-VAE

Hệ thống có *N* user và *M* bài hát được đánh chỉ số. Mỗi user được biểu diễn bởi 1 vector $x_u = \{x_{u1}, x_{u2}, \ldots, x_{uM}\}$ có số chiều là số bài hát *M*, trong đó thành phần tại một vị trí $x_{ui} = 1$ nếu user nghe bài hát đó và $x_{ui} = 0$ nếu user không nghe bài hát đó. Mô hình VAE sẽ được huấn luyện để học khôi phục lại biểu diễn của mỗi user và dự đoán hành vi của user với những bài hát chưa được tương tác để đưa ra gợi ý. Ở đây tác giả của [1] coi phân phối của các vector biểu diễn user xu tuân theo phân phối Multinomial:

$$\mathbf{z}_u \sim \mathcal{N}(0, \mathbf{I}_K), \quad \pi(\mathbf{z}_u) \propto \exp\{f_\theta(\mathbf{z}_u)\},$$
$$\mathbf{x}_u \sim \text{Mult}(N_u, \pi(\mathbf{z}_u)).$$

Từ đó ta có thể tính được log-likelihood cho mỗi user $u$ như sau:

$$\log p_\theta(\mathbf{x}_u \mid \mathbf{z}_u) \stackrel{c}{=} \sum_i x_{ui} \log \pi_i(\mathbf{z}_u).$$

Do ở đây chúng em thấy rằng chuyển từ dữ liệu explicit (số lượt nghe) sang thành dữ liệu dạng implicit (nghe hay không nghe) có thể làm mất một số các thông tin quan trọng. Vậy nên chúng em đã thêm vào biểu thức log-likelihood độ tin cậy của dữ liệu tiềm ẩn giống như ở trong hàm loss của Weighted Matrix Factorization (Yifan Hu el at -2008). Độ tin cậy được tính bằng hàm sigmoid của số lượt nghe của user đối với bài hát. Lý do chúng em sử dụng hàm sigmoid vì nó chuyển được dữ liệu lượt nghe về dạng xác suất trong khoảng [0, 1], lượt nghe càng nhiều thì độ tin cậy càng lớn. Như vậy biểu thức log-likelihood sẽ được tính bằng:

$$log\ p_\theta(x_u|z_u) = \Sigma\ c_i\ log\ \pi_i(z_u)$$

Trong đó $c_i$ = *Sigmoid(số lần user u nghe bài hát i)* là độ tin cậy của bài hát i đối với user $u$. Chúng em đã cài đặt thử nghiệm và thấy rằng việc tận dụng dữ liệu explicit cải thiện được chất lượng gợi ý của mô hình hơn một chút.

Như vậy đối với user mới vào hệ thống, sau một thời gian có một lượng nhất định các tương tác của user, ta có thể đưa ra gợi ý cho họ bằng VAE mà không cần phải huấn luyện lại toàn bộ hệ thống như đối với các phương pháp lọc cộng tác khác như phân tích ma trận.

## IV. Kết quả thực nghiệm

Kiến trúc mạng VAE chúng em sử dụng có dạng M-600-200-600-M (số neural tương ứng ở các tầng). Đầu vào được dropout với xác suất 0.5. Chúng em huấn luyện mô hình với batch size = 500, learning rate = 0.001, beta = 0.2 (tham số regular của KL) huấn luyện trong 100 epoch và sử dụng tối ưu Adam.

Mô hình được huấn luyện và đánh giá dựa trên bộ dữ liệu đã được chia ở trên. Với kịch bản đánh giá trên tập Valid và tập Test đều là các người dùng không có trong tập huấn luyện. Chúng em đưa một lượng tương tác nhất định của người dùng vào coi như phần đã biết và cho gợi ý ra top K bài hát. Sau đó dùng phần tương tác còn lại như dữ liệu trong tương lai để đánh giá.

Chúng em sử dụng NDCG và Recall để đánh giá chất lượng của mô hình.

| Mult-VAE | NDCG@20 | NDCG@100 | Recall@20 | Recall@100 |
|---|---|---|---|---|
| Chỉ sử dụng implicit | Valid: 0.30893<br>Test: 0.30877 | Valid: 0.40195<br>Test: 0.40283 | Valid: 0.31457<br>Test: 0.31650 | Valid: 0.43749<br>Test: 0.43673 |
| Kết hợp explicit | Valid: 0.30974<br>Test: 0.31124 | Valid: 0.40218<br>Test: 0.40450 | Valid: 0.31504<br>Test: 0.31775 | Valid: 0.43486<br>Test: 0.43738 |
| Người dùng mới chưa có tương tác | Valid:<br>Test: 0.08997 | Valid:<br>Test: 0.08269 | Valid:<br>Test: 0.07888 | Valid:<br>Test: 0.07136 |

*Bảng 1: Bảng kết quả thực nghiệm*

Nhìn vào bảng trên ta thấy rằng việc sử dụng kết hợp thêm dữ liệu explicit là số lượng lượt nghe khiến cho mô hình có hiệu quả tốt hơn so với chỉ dùng dữ liệu implicit ở cả 2 độ đo đánh giá là NDCG và Recall. Tuy nhiên sự chênh lệch là không nhiều do đa phần người dùng chỉ nghe mỗi bài hát 1 lần.

Với các người dùng mới, chỉ cần có một lượng vừa đủ tương tác hệ thống có thể gợi ý tương đối tốt. Vậy nếu người dùng mới chưa có bất cứ tương tác nào thì sao?

Chúng em lại dùng một tập 2524 user mới không có trong các tập trên có trên 50 tương tác và coi toàn bộ tương tác đó là dữ liệu trong tương lai. Chúng em đưa vào mô hình vector 0 để cho mô hình gợi ý và thu được kết quả như trong bảng. Trong tương lai chúng em có thể sẽ kết hợp thêm gợi ý những bài hát hot nhất để làm tăng hiệu quả.

## V. Tài liệu tham khảo

[1] Variational Autoencoders for Collaborative Filtering by Dawen Liang, Rahul G. Krishnan, Tony Jebara, Matthew D. Hoffman https://arxiv.org/pdf/1802.05814.pdf

[2] Variational autoencoders. Jeremy Jordan https://www.jeremyjordan.me/variational-autoencoders/