

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



BÁO CÁO ĐỒ ÁN
MÔN HỆ THỐNG TƯ VẤN

Project: Video Games Recommendation System

Giảng viên môn học: Huỳnh Thanh Sơn

Nhóm 3

22280018 - Chiêm Huỳnh Giao

22280020 - Nguyễn Ngọc Bảo Hân

22280037 - Nguyễn Thị Xuân Hương

Tp. Hồ Chí Minh, 29 tháng 06 năm 2025

Mục lục

1. Giới thiệu dataset:	3
1.1. Nguồn gốc dataset.....	3
1.2. Khám phá dataset.....	3
1.3. Đề xuất phương pháp phù hợp với bộ dataset.....	8
2. Xây dựng hệ thống tư vấn.....	9
2.1. Quy trình giải quyết bài toán.....	9
2.2. Metrics đánh giá.....	9
2.3. Triển khai các phương pháp.....	11
2.3.1. Triển khai không giảm chiều.....	13
2.3.1.1. Cosine Similarity.....	13
2.3.1.2. KNN.....	19
2.3.1.3. Nhận xét chung kết quả Cosine Similarity vs. KNN.....	24
2.3.2. Triển khai kết hợp Truncated SVD.....	24
2.4. Triển khai theo hướng lọc theo điều kiện đầu vào.....	27
2.4.1. Đề xuất các game gần với từng thuộc tính riêng lẻ của game đầu vào.....	27
2.4.2. Đề xuất các game gần với các thuộc tính của game đầu vào.....	28
2.4.3. Đề xuất các game gần với các thuộc tính của game đầu vào tính bằng weighted score.....	29
2.5. Kết luận.....	30
3. Một số phương hướng phát triển bài toán.....	31

1. Giới thiệu dataset:

1.1. Nguồn gốc dataset

Bộ dữ liệu được cào từ Giant Bomb sử dụng GiantBomb API

Giant Bomb: là cơ sở dữ liệu trò chơi điện tử khổng lồ, gồm các thông tin về các trò chơi điện tử. Bộ dữ liệu nhóm sử dụng bao gồm thông tin các video games có tổng cộng 11 thuộc tính và 37026 records:

- `id`: Mã định danh - định danh duy nhất cho từng game trong bộ dữ liệu
- `name`: tên game
- `original_game_rating`: độ tuổi quy định được phép chơi game - đánh giá theo chuẩn Mỹ
- `original_release_date`: ngày phát hành game
- `platform`: nền tảng tương thích để chơi game
- `developers`: nhà phát triển game
- `genre`: các thể loại game được gắn mác
- `theme`: chủ đề của trò chơi
- `concept`: phong cách, ý tưởng trò chơi
- `franchise`: series game có game
- `image_url`: link liên kết đến poster game

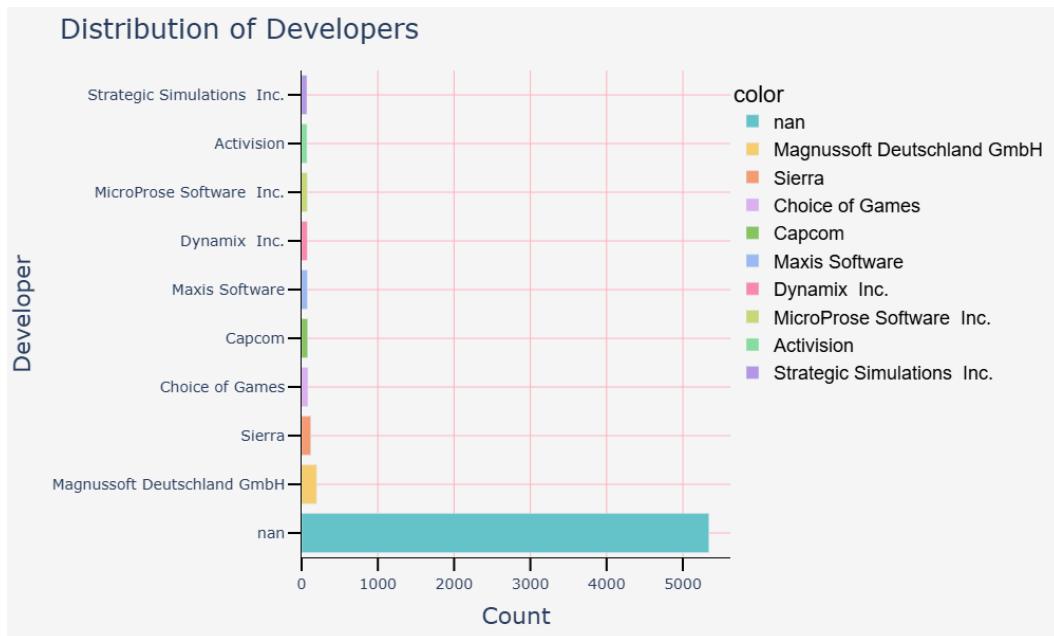
1.2. Khám phá dataset

Khám phá một số đặc trưng nổi bật:

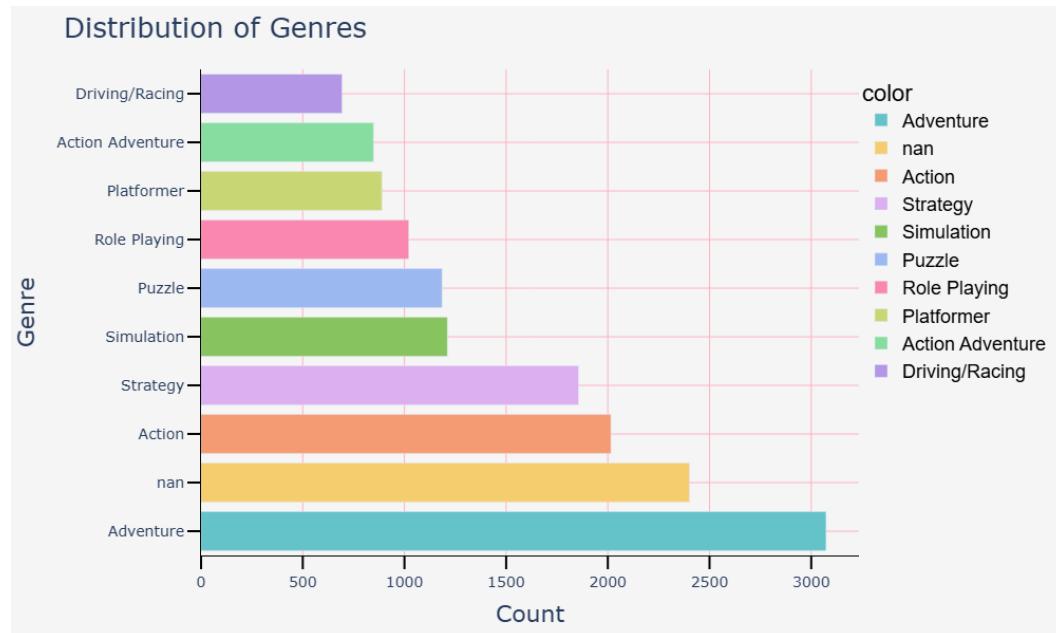
- `platform`: có tổng cộng 61 platforms khác nhau

```
Counter({'PC': 20694, 'Mac': 7414, 'Amiga': 1547, 'Xbox 360': 541, 'PlayStation 2': 508, 'PlayStation': 498, 'Xbox 360 Games Store': 347, 'Apple II': 264, 'Game Boy Advance': 245, 'PlayStation Portable': 181, 'Amstrad CPC': 134, 'Game Boy': 134, 'Commodore 64': 103, 'Wii': 100, 'Genesis': 97, 'PlayStation Network (PS3)': 85, 'Arcade': 83, 'PlayStation 3': 69, 'Xbox': 66, 'Super Nintendo Entertainment System': 63, 'Nintendo DS': 62, 'Dreamcast': 41, 'Atari 2600': 39, 'MSX': 28, 'Intellivision': 28, '3DO': 25, 'Atari ST': 25, 'Nintendo Entertainment System': 24, 'Saturn': 23, 'Game Boy Color': 22, 'Game Gear': 19, 'Wii Shop': 18, 'Nintendo 64': 13, 'Sega CD': 11, 'Game Cube': 10, 'CD i': 9, 'Atari 8 bit': 9, 'iPod': 8, 'ZX Spectrum': 7, 'Sega Master System': 6, 'TurboGrafx CD': 5, 'Neo Geo': 5, 'N Gage': 5, 'TurboGrafx 16': 5, 'Dragon 32/64': 3, 'Apple IIgs': 3, 'Amiga CD32': 3, 'Atari Lynx': 3, 'Jaguar': 2, 'TRS 80': 2, 'Commodore PET/CBM': 2, 'TI 99/4A': 2, 'Atari 7800': 2, 'Sega 32X': 2, 'NUON': 1, 'Game.Com': 1, 'Famicom Disk System': 1, 'WonderSwan Color': 1, 'Zodiac': 1, 'ColecoVision': 1, 'Neo Geo Pocket Color': 1})
```

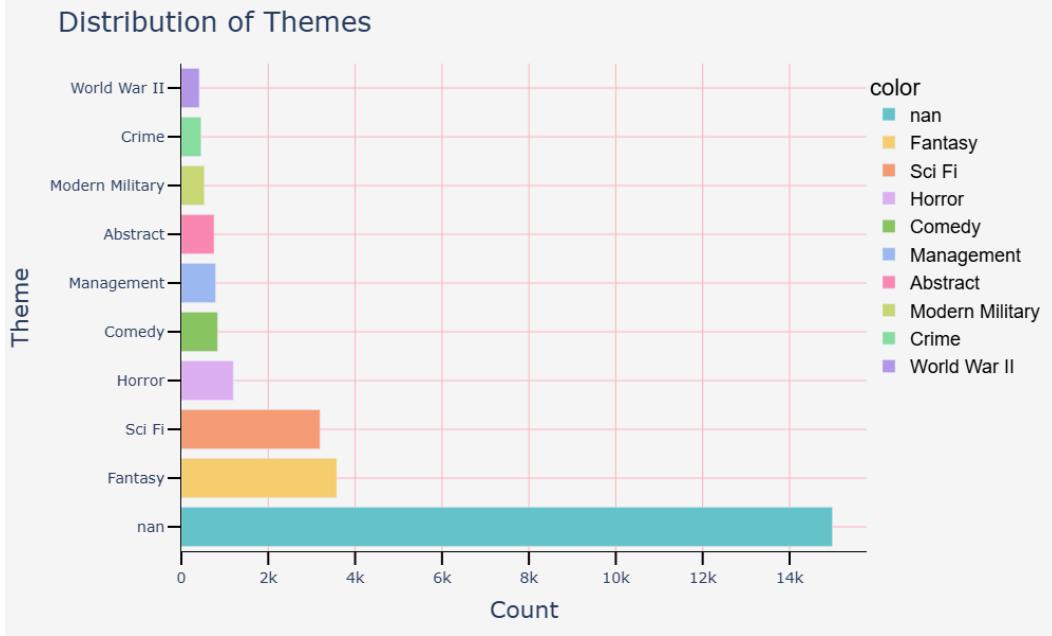
- `developers`:
 - + Có tổng cộng **13515** developers khác nhau
 - + 2401 giá trị là NaN



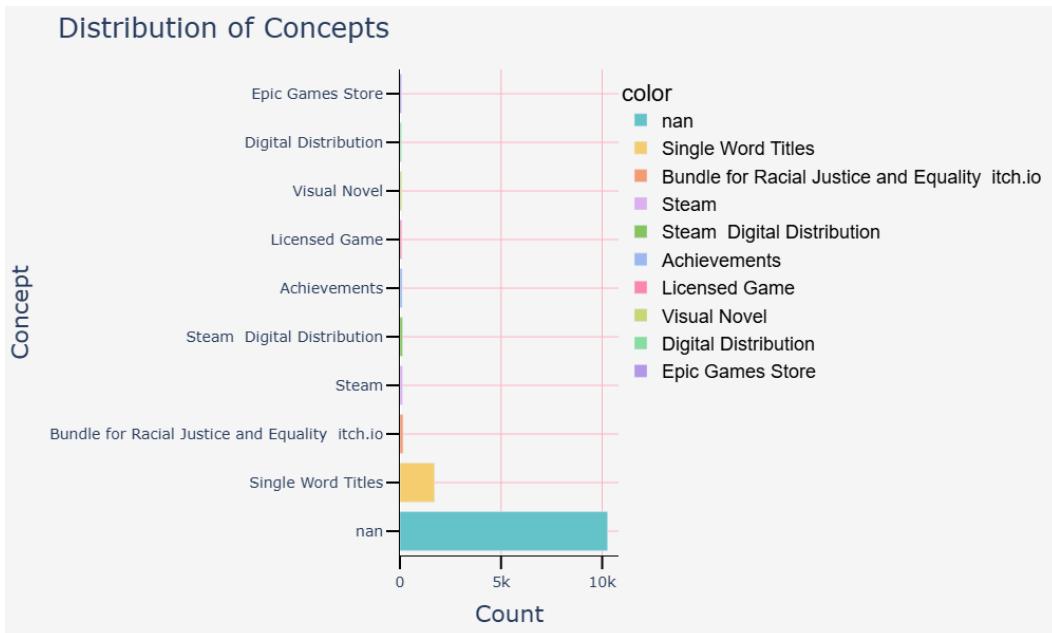
- `genre`:
 - + Có **1344** thể loại khác nhau, trong đó có 5342 nan values
 - + Thể loại adventure chiếm phần lớn trong tập dữ liệu



- `theme`:
 - + Có **693** themes khác nhau
 - + Trong đó có **14986** giá trị là NaN
 - + Số lượng lớn các game có theme Fantasy và Sci-Fi



- `concept`:
 - + Có **17277** concept khác nhau, trong đó có **14986** nan values (45.82%)



- `franchise`:
 - + Có **3251** giá trị khác nhau, trong đó có **23608** nan values (63,76%)

Nhận xét sơ bộ về bộ dữ liệu:

- Bộ dữ liệu không có thông tin người dùng như hành vi, lịch sử tương tác do đó không tính được sự tương đồng của các người dùng khác nhau.
 - Bộ dữ liệu không có thông tin về rating nên không xác định mức độ yêu thích hay hài lòng của các khách hàng.

- Bộ dữ liệu khá thưa do có nhiều missing values.

1.3. Đề xuất phương pháp phù hợp với bộ dataset

Những phương pháp không phù hợp với bộ dữ liệu này:

- **Phương pháp collaborative filtering**
 - + Không có thông tin người dùng.
 - + Không có dấu vết tương tác của người chơi nên không so sánh được các người dùng với nhau.
- **Matrix Factorization**

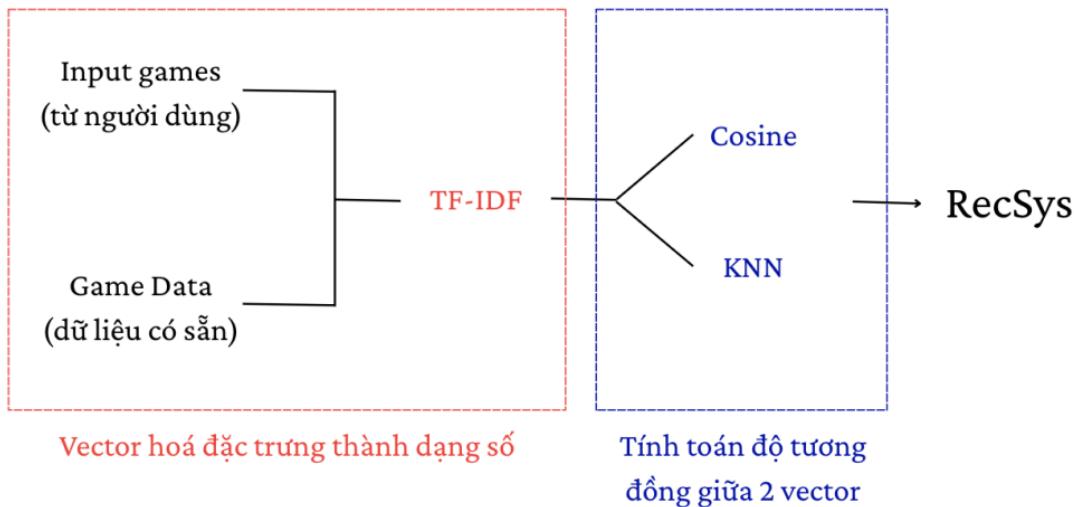
Không có ma trận rating hay giá trị thể hiện mối quan hệ người dùng-game.
- **Model-Based:** Yêu cầu dữ liệu người dùng như lịch sử tương tác, rating,..vv.
- **Association Rule (Apriori, FP-Growth):** Yêu cầu lịch sử giao dịch, chuỗi lựa chọn của người dùng.

Kết luận: phương pháp phù hợp với bộ dữ liệu này nhất là *Content-based filtering*, vì những nguyên nhân:

- Chỉ cần thông tin game, không cần user.
- Phù hợp với dữ liệu thưa.
- Dễ triển khai.
- Phương pháp vector hóa được chọn là TF-IDF:
 - + Vì dữ liệu ngắn, chủ yếu là danh sách tag không cần sử dụng các phương pháp tốn kém hơn.
 - + Làm nổi bật các tag ít phổ biến hơn tốt hơn so với *Count Vectorizer*.
 - + Thuật toán đơn giản giúp tối ưu quá trình tính toán.

2. Xây dựng hệ thống tư vấn

2.1. Quy trình giải quyết bài toán



2.2. Metrics đánh giá

Đối với bài toán này không thể dùng các metrics thông thường để đánh giá

- Do không có ground truth để đánh giá bằng các chỉ số truyền thống như Precision/Recall, RMSE/MAE.
- Không thể đánh giá gợi ý đúng hay sai vì không có thông tin của user.

Những metrics phù hợp để tối ưu hóa trải nghiệm người dùng:

- **Thời gian chạy hiệu quả** là rất quan trọng, vì thời gian chạy luôn là một chỉ số quan trọng trong triển khai ứng dụng để tối ưu hóa trải nghiệm người dùng.
- **Sự liên quan của đề xuất** cũng là một metric quan trọng để đánh giá vì các đề xuất cần có sự liên quan và nhất quán với những game được nhập bởi người dùng.

6 test case để đo lường hiệu quả của các thuật toán: mỗi test-case là danh sách tối đa 5 games đầu vào.

Các test-case chủ yếu đánh giá mức độ liên quan các games dựa vào cột genre (thể loại).

Test case 1: 1 Action Platformer - 1 Action Adventure

Mục đích là kiểm tra *mức độ phân biệt thể loại gần giống nhau*, đánh giá hệ thống gợi ý có thể nhận diện điểm chung là **Action** nhưng cũng hiểu được sự khác biệt giữa *Platformer* và *Adventure* không.

Kỳ vọng thể hiện của hệ thống là gợi ý trò chơi sẽ thiên về **Action**, nhưng hệ thống nên **đa dạng** về *Platformer* và *Adventure*, không chỉ gợi ý một nhóm.

Test case 2: 4 Platformer - 1 Action Adventure

Mục đích là kiểm tra khả năng *tập trung gợi ý vào chủ đề nổi trội*, không bị lệch khi có nhiều nhẹ bằng cách thêm vào bởi 1 game có thể loại khác. Hay nói cách khác là kiểm tra “*sự mềm dẻo*” của hệ thống khi Action Adventure chỉ khác một chút.

Kỳ vọng thể hiện của hệ thống là gợi ý phải ưu tiên gợi ý *Platformer/ Action Platformer*.

Test case 3: 5 Action Platformer/Platformer

Mục đích là đánh giá *độ chính xác* và *độ tập trung chủ đề* của hệ thống. Kiểm tra xem hệ thống có thể nhận biết rằng người dùng yêu thích một thể loại rất cụ thể không.

Kỳ vọng thể hiện của hệ thống là các gợi ý nên cùng thể loại *Action Platformer/Platformer*.

Test case 4: 5 MMORPG

Mục đích là đánh giá *độ nhận dạng chính xác* một thể loại cụ thể.

Kỳ vọng thể hiện của hệ thống là các gợi ý cũng phải thuộc thể loại *MMORPG*.

Test case 5: 4 Real Time Strategy - 1 Action-Adventure

Mục đích là kiểm tra độ *chóng nhiễu mạnh* khi có 1 game khác *hắn thẻ loại*, kiểm tra tính *ổn định và nhất quán*, test case này khác test case 2 ở chỗ là nó lệch *hắn* về 1 thẻ loại.

Kỳ vọng thể hiện của hệ thống là vẫn phải gợi ý chủ yếu là *Real Time Strategy*.

Test case 6: 5 Shooter nhưng có 1 có dạng cụ thể của Shooter là First Person Shooter

Mục đích là kiểm tra *khả năng nhận diện* một nhóm thẻ loại rộng (Shooter) - thẻ loại chính, và hiểu được sự đa dạng nội tại trong nhóm đó - dạng phụ của chính nó. Hay nói cách khác là, kiểm tra “độ sắc sảo” của hệ thống.

Kỳ vọng thể hiện của hệ thống là gợi ý thuộc cùng lớp thẻ loại “Shooter”, lý tưởng nhất là *First Person Shooter* nhưng có thẻ thêm vài dạng phụ của Shooter như *Third-person shooter*.

2.3. Triển khai các phương pháp

Giới thiệu về TF-IDF (Term Frequency-Inverse Document Frequency)

- Hiển thị **mức độ liên quan** của các từ khóa với một vài tài liệu cụ thể hoặc những từ khóa giúp xác định hoặc phân loại một số tài liệu cụ thể
- Là sự kết hợp của 2 độ đo khác nhau: **TF** nói lên tần suất một thuật ngữ xuất hiện trong một tài liệu và **IDF** - gán trọng số nhỏ hơn cho những từ xuất hiện thường xuyên và trọng số lớn hơn cho những từ ít xuất hiện

$$TF = \frac{Term}{TotalWords}$$

$$IDF = \log_e \frac{TotalDocuments}{DocumentFrequency}$$

TF-IDF là sự kết hợp của tần suất thuật ngữ xuất hiện và tần suất tài liệu ngược, nó thể hiện “**sức nặng**” của mỗi thuật ngữ với văn bản. Đảm bảo chỉ những từ xuất hiện ở mức độ vừa phải (không quá nhiều hoặc quá ít) mới được điểm cao

$$TF - IDF = TF * IDF$$

Phân tích thuật toán

Thuật toán được chọn dựa trên khả năng ứng dụng của nó trong Information Retrieval vì vấn đề cần giải quyết liên quan đến mô tả content của sản phẩm chứ không dựa vào lọc cộng tác dựa trên người dùng.

Phương pháp này bao gồm xác định sự tương đồng các cặp điểm trong các điểm dữ liệu, vì vậy *cosine similarity* và *k-nearest neighbor* là những lựa chọn hàng đầu.

Triển khai

Triển khai hàm tf-idf để chuyển đổi dataframe thành dạng vector, nhóm thực hiện code from scratch tất cả các hàm trong bài toán chứ không dùng thư viện, triển khai các công thức theo định nghĩa. Hàm bao gồm hai tham số chính:

- *Max features*: chọn n thuộc tính có điểm số được tính bằng tf-idf cao nhất để cân bằng giữa tốc độ chạy và độ tin cậy của gợi ý.
- *Stop word list*: gồm những từ mà sẽ bị loại bỏ ra khỏi tính toán tf-idf, những từ không có ý nghĩa đáng kể trong phân tích content của video game.

2.3.1. Triển khai không giảm chiều

2.3.1.1. Cosine Similarity

Đo lường sự tương đồng giữa 2 vector trong không gian tích vô hướng, xác định xem liệu 2 vector có chỉ gần như cùng một hướng không

$$sim(x, y) = \frac{xy}{|x||y|}$$

Cách hàm recommendation hoạt động

Đầu vào: `df` , `tfidf_matrix` , `idf_list` , `game_1` , `game_2` , `game_3` , `game_4` , `game_5`

- + ‘df’: dataframe đầu vào gồm danh sách cách video games
- + `tfidf_matrix`: ma trận điểm cosine similarity
- + `idf_list`: idf dictionary
- + `game_1` , `game_2` , `game_3` , `game_4` , `game_5`: danh sách tên game đầu vào

Nhóm đã có sự thay đổi so với project gốc để tối ưu thời gian phản hồi cho khách hàng:

Đầu vào có thêm tham số (`tfidf_matrix` , `idf_list`) tức đưa ma trận cosine similarity - được tính ở bước offline của toàn bộ các game trong dataset và idf dictionary cho các từ tương ứng để giảm thời gian tính toán

Thực hiện:

- Trích xuất thông tin các game đầu vào:
game_1 , game_2 , game_3 , game_4, game_5 ['total_contents'] =
['original_game_rating'] + " " + ['developer'] + " " + ['genre'] + " "
+ ['theme'] + " " + ['concept'] + " " + ['franchise'] + " " + ['platform']

(cột *total_content* đã được xử lý trong phần tiền xử lý dữ liệu và lưu trong df; *total_content* đại diện cho thông tin của từng game)

- Thông tin của users là tổng hợp các đặc trưng của các game đầu vào
- Tính toán cosine similarity của vector đầu vào với vector tương ứng trong ma trận tf-idf của các game
- Gợi ý cho người dùng top k game tương đồng nhất là top k game có cosine similarity score cao nhất

Kết quả bài toán

Test case 1: 1 Action Platformer - 1 Action Adventure

```
# 1 Action Platformer and 1 Action Adventure
test_case_1 = cs_game_recommendations(df = df, tfidf_matrix = tmp_df, idf_dict = idf,
                                         game_1 = '30XX',
                                         game_2 = 'Batman: Arkham City', game_3 = None,
                                         game_4 = None, game_5 = None)

test_case_1
```

Time Elapsed: 12.63 seconds

	name	genre
0	Batman: Arkham Asylum	Action Adventure
1	Batman: Arkham Knight	Action Adventure
2	Batman: Arkham Origins	Brawler Action Adventure
3	Tyranny	Role Playing
4	Crawl	Role Playing Action Adventure
5	Batman: Arkham Origins Blackgate	Action Adventure
6	Horizon Zero Dawn	Role Playing Shooter Action Adventure
7	LEGO Batman 2: DC Super Heroes	Action Adventure
8	Splasher	Action Platformer
9	Costume Quest	Role Playing

Test case 2: 4 Platformer - 1 Action Adventure

```
# 4 Action Platformers and 1 ActionAdventure
test_case_2 = cs_game_recommendations(df = df, tfidf_matrix = tmp_df, idf_dict = idf,
f, game_1 = '30XX',
game_2 = 'Castlevania', game_3 = 'Fumiko!', game_4 = '99 Levels To Hell', game_5 = 'Batman: Arkham Asylum')

test_case_2
```

Time Elapsed: 12.89 seconds

	name	genre
0	Batman: Arkham City	Action Adventure
1	Batman: Arkham Knight	Action Adventure
2	Batman: Arkham Origins	Brawler Action Adventure
3	Battle Chasers: Nightwar	Action Role Playing
4	Costume Quest	Role Playing
5	The Elder Scrolls IV: Oblivion	Action Role Playing
6	The Surge	Action Role Playing
7	Injustice: Gods Among Us	Fighting
8	Tomb Raider	Shooter Puzzle Platformer Action Adventure
9	Tomb Raider	Shooter Platformer Action Adventure

Test case 3: 5 Action Platformer/Platformer

```
# 5 Action Platformers
test_case_3 = cs_game_recommendations(df = df, tfidf_matrix = tmp_df, idf_dict = idf,
f, game_1 = '30XX',
game_2 = 'A.R.E.S. Extinction Agenda EX', game_3 = 'Fumiko!', game_4 = '99 Levels To Hell', game_5 = 'Zack Zero')

test_case_3
```

Time Elapsed: 12.46 seconds

	name	genre
0	Hard Room	Platformer
1	Berserk Boy	Action Platformer
2	Orbital Gear	Action Shooter Platformer
3	Curse of the Crescent Isle	Platformer
4	Solar Gun	Puzzle Platformer
5	WarpThrough	Action Platformer
6	Ball Kicker	Action Platformer
7	CreatorCrate	Action Platformer
8	Featherpunk Prime	Action Platformer
9	Space Gladiators	Action Platformer

Test case 4: 5 MMORPG

```
# 5 MMORGPS
test_case_4 = cs_game_recommendations(df = df, tfidf_matrix = tmp_df, idf_dict = idf,
                                       game_1 = 'Albion Online',
                                       game_2 = 'ArcheAge', game_3 = 'World of Warcraft',
                                       game_4 = 'City of Heroes', game_5 = 'City of Villains')

test_case_4
```

Time Elapsed: 12.61 seconds

	name	genre
0	Guild Wars	Action Strategy Adventure Role Playing
1	World of Warcraft: The Burning Crusade	MMORPG
2	Guild Wars: Nightfall	Action Strategy Adventure Role Playing MMORPG
3	Dark Age of Camelot	Role Playing MMORPG
4	EverQuest II	MMORPG
5	Guild Wars: Factions	Action Strategy Adventure Role Playing MMORPG
6	Anarchy Online	Role Playing MMORPG
7	Helbreath	Action MMORPG
8	The Lord of the Rings Online: Shadows of Angmar	Role Playing MMORPG
9	Dark Age of Camelot: Shrouded Isles	Role Playing MMORPG

Trong 4 test case đầu hầu hết các game được gợi ý có cùng thể loại hoặc gần thể loại với các game đầu vào.

Test case 5: 4 Real Time Strategy - 1 Action-Adventure

```
# 4 Action Platformers and 1 ActionAdventure
test_case_5 = cs_game_recommendations(df = df, tfidf_matrix = tmp_df, idf_dict = idf,
game_1 = '8-Bit Hordes',
game_2 = '8-Bit Invaders!', game_3 = '9th Company: Roots of Terror',
game_4 = 'A Game of Thrones: Genesis', game_5 = 'Batman: Arkham Asylum')

test_case_5
```

Time Elapsed: 12.52 seconds

	name	genre
0	Batman: Arkham City	Action Adventure
1	Batman: Arkham Knight	Action Adventure
2	DwarfCorp	Simulation Real Time Strategy
3	Warcraft III: Reforged	Real Time Strategy
4	C-Dogs	Action
5	Batman: Arkham Origins	Brawler Action Adventure
6	The Seven Years War (1756-1763)	Real Time Strategy
7	Circle Empires	Real Time Strategy
8	Armored Moon	Real Time Strategy
9	Battle for Enlor	Real Time Strategy

- Gợi ý của hệ thống cho kết quả các game khá ổn.
- Phù hợp với xu hướng có vẻ thích *Real Time Strategy* hơn của người dùng: kết quả gợi ý có thể loại *Real Time Strategy* chiếm phần lớn và cũng có thể loại *Action/Adventure* - đầu vào có đến 4/5 game đầu vào là thể loại *Real Time Strategy*.

Test case 6: Shooters

```
# 5 Shooters
test_case_6 = cs_game_recommendations(df = df, tfidf_matrix = tmp_df, idf_dict = idf,
game_1 = '8bit Killer',
game_2 = 'Alien Swarm', game_3 = 'Doom VFR',
game_4 = 'Earth Defense Force 5', game_5 = 'Fortnite')

test_case_6
```

Time Elapsed: 12.51 seconds

	name	genre
0	Doom	First Person Shooter
1	Duke Nukem 3D	First Person Shooter
2	Gears of War 4	Shooter
3	Left 4 Dead	Action First Person Shooter
4	Halo 5: Guardians	First Person Shooter
5	Gears 5	Action
6	Halo: Combat Evolved	First Person Shooter
7	Doom 3	First Person Shooter
8	Doom II: Hell on Earth	First Person Shooter
9	Resident Evil 6	Action Shooter Brawler

Hầu hết các game được gợi ý có cùng hoặc gần với thể loại Shooter, trong đó là FPS chiếm số lượng 7/10 game, hệ thống đạt đúng như độ “*lý tưởng*” mong muốn.

Nhận xét kết quả khi dùng Cosine Similarity

- **Thời gian trung bình chương trình xử lý và đưa ra đề xuất:** 12-13 s/truy vấn, thời gian trên vẫn còn khá chậm đối với một hệ thống đề xuất cần trả về kết quả gợi ý tức thì
- **Mức độ liên quan của các đề xuất:**
 - + Hệ thống đã đưa ra đề xuất các game gần với các game được người dùng nhập (nói cụ thể ở đây chỉ mới so sánh genre)
 - + Kết quả đề xuất đáp ứng được kỳ vọng xây dựng hệ thống
 - + Kết quả tỷ lệ các game trả về phù hợp với tỷ lệ game đầu vào (ví dụ số lượng game đầu vào thiên về hẳn một thể loại thì đầu ra cũng tương tự)

Kết luận: hệ thống hiểu được mức độ ưu tiên mong muốn của người dùng.

2.3.1.2. KNN

Vì bài toán cần trả về các đối tượng tương tự nhất, không phải các nhãn như các bài toán classification nên không thể dùng KNeighborsClassifier hay RadiusNeighbors Classifier, do đó chọn KNearestNeighbors.

Ý tưởng thuật toán

- **Cách hoạt động:**

- Tính khoảng cách từ X_{new} (thông tin đặc trưng các game đầu vào) đến toàn bộ tập train đã lưu,
- Tìm ra K điểm gần nhất,
- Trả về K điểm, và khoảng cách đến K điểm (nếu cần).

- **Điểm đặc biệt của thuật toán:** NN không huấn luyện trên tập train khi fit() như các thuật toán khác mà chỉ lưu lại tập train. Khi có input dữ liệu mới vào (X_{new}) mới thực hiện tính toán để chọn ra K neighbor gần nhất bằng cách hoạt động bên trên.

Triển khai thuật toán:

Đi sâu vào từng parameter của thuật toán NN để có thể hiểu rõ thuật toán:

- `kd_tree`, `ball_tree`:
KDTree, BallTree là cấu trúc dữ liệu cây dùng để *tăng tốc độ tìm kiếm cho NN* trong không gian nhiều chiều bằng cách giảm số lượng phép tính khoảng cách cần thực hiện.
 - Tại mỗi bước duyệt cây được tạo từ KDTree và BallTree: chỉ cần truy cập vào các nhánh có khả năng chứa điểm gần hơn điểm hiện tại,
 - Nếu khoảng cách từ điểm đang xét đến một nhánh đã lớn hơn khoảng cách đến k điểm gần nhất hiện tại thì bỏ qua nhánh đó luôn mà không cần xét.

- **Cách hoạt động:**

Xây dựng cây *nhi phân* để chia nhỏ không gian:

- KDTree chia không gian dựa vào trục toạ độ, BallTree chia dựa vào các vùng cầu (ball) chứa tâm và bán kính bao quanh các nhóm điểm.
- BallTree khác KDTree ở chỗ có bước clustering (dùng kmeans hoặc các heuristics) để dựng cây thay vì dựa vào giá trị trung vị của chiều có độ phân tán lớn như KDTree.
- `leaf_size` để giới hạn độ sâu của cây.

Do KDTree, BallTree khá khó để triển khai thủ công, vì nó liên quan đến tối ưu máy, và nó cũng không phải là bản chất cần quan tâm của thuật toán NN nên nhóm sẽ dùng thư viện có sẵn.

- `brute`: sử dụng brute-force search, đơn giản là tìm trên toàn bộ bộ tập dữ liệu.
- `auto`: sẽ tự động chọn algorithm phù hợp dựa vào n_samples và n_features của bộ dữ liệu train.

Kết quả thuật toán:

Test case 1: 1 Action Platformer - 1 Action Adventure

```
# 1 Action Platformer and 1 Action Adventure
test_case_1 = knn_game_recommendations(df = df, tfidf_matrix = tfidf_matrix, tf = tf,
                                         game_1 = '30XX', game_2 = 'Batman: Arkham City',
                                         game_3 = None, game_4 = None, game_5 = None)

test_case_1
```

Time Elapsed: 6.609687566757202 seconds

	name	genre
2911	Batman: Arkham Asylum	Action Adventure
2913	Batman: Arkham Knight	Action Adventure
2914	Batman: Arkham Origins	Brawler Action Adventure
30959	Tyranny	Role Playing
13466	Horizon Zero Dawn	Role Playing Shooter Action Adventure
25862	Splasher	Action Platformer
7038	Darksiders	Action Action Adventure
19596	Observation	Action Adventure
6310	Crawl	Role Playing Action Adventure
15908	LEGO Batman 2: DC Super Heroes	Action Adventure

Các game được gợi ý khá gần với các game đầu vào.

Test case 2: 4 Platformer - 1 Action Adventure

```
# 4 Action Platformers and 1 ActionAdventure
test_case_2 = knn_game_recommendations(df = df, tfidf_matrix = tfidf_matrix, tf=tf,
                                         game_1 = '30XX', game_2 = 'Castlevania', game_3 = 'Fumiko!',  
                                         game_4 = '99 Levels To Hell', game_5 = 'Batman: Arkham Asylum')
```

```
test_case_2
```

```
Time Elapsed: 6.614527225494385 seconds
```

	name	genre
2912	Batman: Arkham City	Action Adventure
2913	Batman: Arkham Knight	Action Adventure
2914	Batman: Arkham Origins	Brawler Action Adventure
2942	Battle Chasers: Nightwar	Action Role Playing
29458	The Surge	Action Role Playing
6230	Costume Quest	Role Playing
25796	Spider-Man 3	Action Adventure
7218	Dead Rising 2	Role Playing Action Adventure
3502	BioShock Infinite	First Person Shooter Action Adventure
28301	The Elder Scrolls IV: Oblivion	Action Role Playing

Phần nhỏ gợi ý có vẻ không quá gần với các genre của các game input đầu vào.

Test case 3: 5 Action Platformer/Platformer

```
# 5 Action Platformers
test_case_3 = knn_game_recommendations(df = df, tfidf_matrix = tfidf_matrix, tf=tf,  
                                         game_1 = '30XX', game_2 = 'A.R.E.S. Extinction Agenda EX', game_3 = 'Fumik  
o!',  
                                         game_4 = '99 Levels To Hell', game_5 = 'Zack Zero')
```

```
test_case_3
```

```
Time Elapsed: 6.647079706192017 seconds
```

	name	genre
3303	Berserk Boy	Action Platformer
12741	Hard Room	Platformer
32138	WarpThrough	Action Platformer
23100	Robot Exploration Squad	Platformer Action Adventure
30968	U.F.O - Unfortunately Fortunate Organisms	Action
6794	Cyjin: The Cyborg Ninja	Platformer
9096	Echelon: Wind Warriors	Action Simulation
28742	The Last Federation	Strategy
5690	Cloudbuilt	Action Platformer
9092	Echelon	Action Simulation

Hầu hết các game có sự tương đồng (Action/Platformer).

Test case 4: 5 MMORPG

```
# 5 MMORPGS
test_case_4 = knn_game_recommendations(df = df, tfidf_matrix = tfidf_matrix, tf=tf,
                                         game_1 = 'Albion Online', game_2 = 'ArcheAge', game_3 = 'World of Warcraft',
                                         game_4 = 'City of Heroes', game_5 = 'City of Villains')
```

```
test_case_4
```

```
Time Elapsed: 6.651864528656006 seconds
```

	name	genre
32882	World of Warcraft: The Burning Crusade	MMORPG
12453	Guild Wars	Action Strategy Adventure Role Playing
6907	Dark Age of Camelot	Role Playing MMORPG
28844	The Lord of the Rings Online: Shadows of Angmar	Role Playing MMORPG
12460	Guild Wars: Nightfall	Action Strategy Adventure Role Playing MMORPG
12976	Hellbreath	Action MMORPG
9754	EverQuest II	MMORPG
1566	Anarchy Online	Role Playing MMORPG
12459	Guild Wars: Factions	Action Strategy Adventure Role Playing MMORPG
20648	Perfect World International	MMORPG

Tất cả các đề xuất đều có sự liên quan đến các game đầu vào về thể loại.

Test case 5: 4 Real Time Strategy - 1 Action-Adventure

```
# 4 Action Platformers and 1 ActionAdventure
test_case_5 = knn_game_recommendations(df = df, tfidf_matrix = tfidf_matrix, tf=tf,
                                         game_1 = '8-Bit Hordes', game_2 = '8-Bit Invaders!', game_3 = '9th Company:
Roots of Terror',
                                         game_4 = 'A Game of Thrones: Genesis', game_5 = 'Batman: Arkham Asylum')
```

```
test_case_5
```

```
Time Elapsed: 6.743025779724121 seconds
```

	name	genre
2912	Batman: Arkham City	Action Adventure
2913	Batman: Arkham Knight	Action Adventure
8951	DwarfCorp	Simulation Real Time Strategy
2914	Batman: Arkham Origins	Brawler Action Adventure
15289	Kingdom Elemental	Strategy Real Time Strategy
2915	Batman: Arkham Origins Blackgate	Action Adventure
32018	Warcraft III: Reforged	Real Time Strategy
6014	Conan Unconquered	Real Time Strategy
7445	Defend The Keep	Real Time Strategy
26458	Stellar Warfare	Real Time Strategy

Hầu hết là các game Real Time Strategy và phần nhỏ thể loại Action Adventure được gợi ý → kết quả khá đúng với kỳ vọng.

Test case 6: Shooters

```
# 5 Shooters
test_case_6 = knn_game_recommendations(df = df, tfidf_matrix = tfidf_matrix, tf=tf,
                                         game_1 = '8bit Killer', game_2 = 'Alien Swarm', game_3 = 'Doom VFR',
                                         game_4 = 'Earth Defense Force 5', game_5 = 'Fortnite')

test_case_6
```

Time Elapsed: 6.6309263706207275 seconds

	name	genre
16715	Madden NFL 19	Football
16716	Madden NFL 20	Football
19195	NFL Quarterback Club 96	Football
18897	NBA 2K21	Sports Basketball
16723	Madden NFL 21	Football
8314	Doom	First Person Shooter
11653	Gears of War 4	Shooter
18896	NBA 2K20	Simulation Basketball
15815	Left 4 Dead	Action First Person Shooter
8797	Duke Nukem 3D	First Person Shooter

Tuy các kết quả gợi ý có vẻ không liên quan đến loại đầu vào là Shooter, nhưng các game này đều là game liên quan đến thể thao.

Nhận xét kết quả khi dùng KNN:

- **Thời gian trung bình chương trình xử lý và đưa ra đề xuất:**
7-8s/truy vấn, thời gian này vẫn chưa đủ nhanh cho một hệ thống nhóm mong muốn xây dựng.
- **Mức độ liên quan của các đề xuất:**
 - Hệ thống đã đưa ra đề xuất các game khá gần genre với các game được người dùng nhập.
 - Gợi ý có vẻ không quá gần, nhưng có một số đề xuất khá thú vị, ví dụ như test case 6, không có cùng thể loại nhưng các đề xuất có điểm chung.

2.3.1.3. Nhận xét chung kết quả Cosine Similarity vs. KNN

- **Thời gian xử lý:** KNN chạy 6 đến 7 giây nhanh hơn *Cosine Similarity* chạy từ 12 đến 13 giây
- **Kết quả đề xuất:**
 - + **Mức độ liên quan đến các game đầu vào:** *Cosine Similarity* cho những đề xuất gần hơn với game đầu vào.
 - + **Mức độ đa dạng đề xuất:** *KNN* cho kết quả đa dạng hơn, tuy có vẻ nhiều trường hợp không thực sự tương đồng với input đầu vào nhưng lại cho người dùng đa dạng lựa chọn để trải nghiệm.

2.3.2. Triển khai kết hợp Truncated SVD

Vì sao lại chọn Truncated SVD

Ma trận TF-IDF có rất nhiều chiều, điều đó làm tăng thời gian tính toán để tìm ra các game tương đồng. Việc giảm chiều dữ liệu là cần thiết để giảm thiểu thời gian phản hồi. Truncated SVD là một phương pháp hiệu quả giúp giảm chiều dữ liệu mà vẫn giữ lại những thông tin quan trọng.

Triển khai SVD

SVD phân rã ma trận ban đầu thành 3 thành phần chính, như sau:

$$A = U\Sigma V^T \quad (1)$$

Với,

A: Ma trận TF-IDF, ví dụ có kích thước m x n.

U: ma trận trực giao đặc trưng cho các mẫu, kích thước m x m.

V: ma trận trực giao đặc trưng cho các feature, kích thước n x n.

Σ : ma trận đường chéo không vuông chứa các giá trị suy biến (singular values), có kích thước m x n, với các phần tử trên đường chéo

$$\lambda_1 \geq \lambda_2 \geq \dots \geq 0.$$

Từ (1), dễ dàng phân tích được:

$$\begin{aligned} A^T A &= (U \Sigma V^T)^T U \Sigma V^T \\ &= V \Sigma^T U^T U \Sigma V^T \\ &= V \Sigma^T \Sigma V^T \end{aligned}$$

$\Sigma^T \Sigma$ là một ma trận đường chéo không vuông, với các phần tử riêng trên đường chéo là $\lambda_1^2, \lambda_2^2, \dots$ là trị riêng của ma trận $A^T A$. Mỗi cột của V chính là một vector riêng của $A^T A$, ta gọi mỗi cột này là *right-singular vector* của A .

Triển khai thuật toán:

Tham số chính: `n_components`: số chiều muốn giữ lại

- Sử dụng hàm np.linalg.eigh trong thư viện linalg để phân tích các vector riêng và trị riêng của ma trận $A^T A$,
- Lấy ra n các trị riêng cao nhất lớn hơn 0 (n là số chiều muốn giữ lại) và các vector riêng tương ứng trong ma trận V ,
- Để giảm chiều, áp dụng công thức sau:

$$A_{reduced} = A V_k$$

, k là số chiều muốn giữ lại.

Vai trò của Truncated SVD trong thuật toán:

Ưu điểm:

- Giảm số chiều của ma trận TF-IDF, làm tăng tốc độ tính toán: Ma trận TF-IDF có thể có hàng ngàn cột, với mỗi cột là một từ. Vì thế, giảm số chiều của ma trận TF-IDF có thể làm cho các phép tính như cosine

similarity, KNN,... tăng tốc độ đáng kể.

- Loại bỏ từ nhiều và từ ít quan trọng: Dataframe sau khi thực hiện TF-IDF sẽ có thể chứa nhiều từ không có giá trị cao. Vì thế, Truncated SVD giúp loại bỏ các giá trị này bằng cách loại bỏ những vector riêng có giá trị của trị riêng tương ứng nhỏ.

Nhược điểm:

- Chi phí tính toán cao Việc phân rã ma trận TF-IDF tốn nhiều thời gian tính toán và tài nguyên
- Khó giải thích kết quả Sau khi giảm chiều thì mỗi trực không gian mới là một tổ hợp tuyến tính của các từ nên sẽ không thể giải thích được từng chiều đó nói lên điều gì.

Nhận xét:

- Thời gian truy vấn: Truncated SVD giúp 2 thuật toán **Cosine Similarity** và **KNN** cải thiện tốc độ đáng kể .
 - + *Truncated SVD kết hợp Cosine Similarity*: 0.18 giây đến 0.20 giây cho một truy vấn
 - + *Truncated SVD kết hợp với KNN*: 0.05 giây đến 0.07 giây cho một truy vấn
- Độ “chất lượng” của đề xuất: Kết quả đề xuất cũng không quá khác với khi chưa giảm chiều bằng Truncated SVD.

Kết luận: KNN hiệu quả đối với bài toán này

2.4. Triển khai theo hướng lọc theo điều kiện đầu vào

2.4.1. Đề xuất các game gần với từng thuộc tính riêng lẻ của game đầu vào

Đề xuất những game tương tự các game đầu vào theo từng thuộc tính khác nhau: `genre`, `theme`, `concept`, `developer`, `franchise`

Cách thực hiện:

- Lấy thông tin các game đầu vào từ bộ dataset ban đầu,
- Lấy random 2 tag của mỗi đặc điểm,
- Lọc tìm danh sách gợi ý các game chứa ít nhất 1 tag được chọn.

Mỗi lần người dùng reload, hệ thống sẽ trả về các đề xuất các game khác nhau nhưng vẫn đảm bảo có sự tương đồng với các game đã nhập, từ đó người dùng có đa dạng sự lựa chọn hơn.

Nhận xét:

- Ưu điểm:
 - + Kết quả thực thi nhanh chóng (do không tính toán phức tạp)
 - + Kết quả giải thích được
 - + Người dùng có nhiều lựa chọn theo mong muốn
- Nhược điểm:
 - + Không học được mối quan hệ phức tạp giữa các thuộc tính. Ví dụ, có những game không cùng theme/genre nhưng mang lại cho người dùng cùng cảm giác khi chơi.
 - + Không tối ưu trải nghiệm của người dùng: Trả về quá nhiều kết quả tương ứng, người dùng phải lọc lại để tìm được game mong muốn.

2.4.2. Đề xuất các game gần với các thuộc tính của game đầu vào

Tương tự như ý tưởng của phương pháp Single Filtering nhưng lọc kết hợp các đặc trưng. Những đặc trưng được chọn: `genre`, `theme`, `concept`, `original_release_date` là đặc trưng phụ dùng để ưu tiên các game mới hơn trong trường hợp còn nhiều option để chọn

Ý tưởng triển khai:

- Lấy danh sách các đặc trưng để lọc bằng cách truy xuất ngược lại đặc trưng từ các game đầu vào,
- Xem mỗi danh sách đặc trưng từ các game đầu vào là điều kiện lọc của mỗi lớp, và các lớp được xếp thứ tự ưu tiên theo mức độ quan trọng - càng quan trọng càng được lọc trước,
- Tính tỷ lệ xuất hiện đầu vào của các thuộc tính để output trả về có tỷ lệ tương ứng,
- Lọc theo thứ tự ưu tiên các thuộc tính: `genre` → `theme` → `concept` → `original_release_date` ,
- Trả về top K kết quả.

Điểm đặc biệt của hàm:

- Tăng tính đa dạng của recommendation bằng cách lấy số lượng mẫu sau khi lọc $2 * \text{top_k}$ games.
- Không lọc cứng: Các game không cần phải sở hữu toàn bộ các đặc trưng giống với game đầu vào mà chỉ cần chứa một trong các đặc trưng trong danh sách đặc trưng đầu vào là được chấp nhận. Điều này làm tăng tính đa dạng, tránh đưa những gợi ý quá giống với game đầu vào, người dùng có cơ hội tìm kiếm nhiều game có thể loại khác hoặc tương tự.

2.4.3. Đề xuất các game gần với các thuộc tính của game đầu vào tính bằng weighted score

Tính điểm có trọng số các games có trong bộ dữ liệu dựa vào độ tương đồng của chúng với các đặc điểm của 5 game đầu vào, cụ thể ở đây ta chọn 3 thuộc tính, theo quan điểm của nhóm là có tính giải thích và đặc trưng nhất cho 1 game: `genre`, `theme` và `concept`.

Ý tưởng triển khai: Điểm tương đồng được tính bằng độ trùng khớp của các đặc tính `genre`, `theme`, `concept` của các game này với các game đầu vào, tính bằng cách:

- Với mỗi thuộc tính, đếm số lượng phần tử trong thuộc tính của game đang xét mà có nằm trong list thuộc tính tương ứng của game đầu vào,
- Số lượng phần tử trùng trong mỗi thuộc tính sẽ được nhân với trọng số tương ứng của thuộc tính đó,
- Dựa vào điểm tương đồng vừa tính, chọn ra top K các game gần nhất với danh sách game được người dùng nhập để đưa ra gợi ý.

Nhận xét hai phương pháp Combined Filtering và Weighted Scoring:

Nhận xét chung:

- **Ưu điểm:**
 - + Thời gian thực thi tức thì do không cần tính toán, chỉ cần lọc theo điều kiện.
 - + Trả về kết quả gần với các thuộc tính của các game đầu vào.
 - + Dễ triển khai và dễ giải thích.
- **Nhược điểm:**
 - + Không tận dụng được tối đa thông tin vì chỉ lọc theo một vài thuộc tính nhất định.
 - + Không đa dạng lựa chọn, người dùng không có cơ hội trải nghiệm các game có thuộc tính khác nhưng có thể họ sẽ thích.

Kết luận: kết quả đề xuất có thể chưa thực sự phù hợp với người dùng.

So sánh:

Tiêu chí	Filtering	Weighted Scoring
Tốc độ xử lý	Rất nhanh do lọc theo điều kiện.	Tính toán tốn thời gian hơn
Tính tổng thể	Không đánh giá toàn cục.	Có thể sắp xếp toàn bộ theo mức độ phù hợp
Tính linh hoạt	Lớp đầu (<i>'genre'</i>) quá mạnh, làm loại bỏ game khác phù hợp hơn.	Linh hoạt theo điểm tổng hợp các thuộc tính.
Tính đa dạng	Có nhưng hạn chế (lọc từng lớp)	Dễ mất đa dạng nếu <i>top_k</i> game trội có điểm quá cao

Tùy tình huống, có thể kết hợp cả hai phương pháp để tạo hệ thống gợi ý cân bằng giữa độ chính xác và tính khám phá.

2.5. Kết luận

Sử dụng phương pháp **Content-based filtering** là phù hợp nhất với bộ dữ liệu này, vì:

- *Tận dụng được hết tất cả các thuộc tính:* Mô tả nội dung chi tiết của các game.
- *Tính toán linh hoạt:* Cho phép so sánh độ tương đồng mềm vì cosine similarity giúp nhận ra mức độ gần nhau về nội dung.
- *Tránh tình trạng overfitting theo từng thuộc tính:* Tổng hợp nhiều đặc trưng nên kết quả đa dạng hơn, tránh rơi vào chỉ một nhánh.
- *Thời gian xử lý đạt tối ưu:* Sau khi đã áp dụng phương pháp giảm chiều SVD

Kết luận phương pháp tối ưu: Cả hai phương pháp Cosine Similarity và KNN sau khi giảm chiều bằng SVD đều có thời gian phản hồi lý tưởng; tuy nhiên KNN có kết quả phản hồi tốt hơn hẳn.

- **Chọn SVD + Cosine Similarity:** Ưu tiên là gợi ý cho người dùng những game có khả năng cao họ sẽ thích - **lựa chọn an toàn**.
- **Chọn SVD + KNN:** ưu tiên về trải nghiệm người dùng, thời gian đưa ra đề xuất **tức thì**, đồng thời giúp người dùng **khám phá** nhiều thể loại game, giúp họ có trải nghiệm phong phú hơn.

Hạn chế của thuật toán:

- Do tự triển khai các thuật toán nên tốc độ xử lý chưa tối ưu ở một số hàm.
- Chưa chọn được những thuộc tính đủ tốt để đại diện cho sản phẩm nên các gợi ý chưa thật sự “chất lượng”.
- Không linh hoạt khi input có ít thông tin. Ví dụ: Người dùng chỉ nhập vào 1 game, hệ thống cho ra gợi ý thiếu tin cậy do quá ít thông tin đầu vào.
- TF-IDF phụ thuộc chặt vào từ trùng khớp, những từ có gần nghĩa không được xử lý tốt.

3. Một số phương hướng phát triển bài toán

- **Thu thập thêm dữ liệu:** Kết hợp thêm các thông tin về rating, đánh giá của user,... để triển khai bài toán theo hướng Collaborative Filtering/ Model Based, hướng tới Hybrid Recommendation.
- **Cá nhân hóa theo lựa chọn của người dùng:** Xây dựng chương trình người dùng lựa chọn đặc điểm game họ muốn chơi; cho phép người dùng điều chỉnh sau mỗi lần gợi ý. Từ đó, hệ thống gợi ý sẽ cải thiện kết quả đến khi người dùng chọn được game mong muốn.
- **Ưu tiên gợi ý game phổ biến:** Thu thập thêm những thống kê tổng thể về các trò chơi (độ thảo luận, điểm đánh giá trung bình từ các trang review/những người chơi khác, số người đã bỏ vào wishlist...). Từ đó,

hệ thống sẽ ưu tiên gợi ý các game đang phổ biến này cho người dùng hơn.