GAME Recommendation System

I R S P R O J E C T R E P O R T

Fu Chentao A0285737M

Deng Yuxuan A0285746M

Lin Fanzhi A0285725U

Tu Runzhang A0285767H

Content

Executive Summary

The gaming industry is witnessing an exponential growth in the number and diversity of available games. With this abundance of options, players often find it challenging to navigate through the vast catalog of games and identify those that align with their preferences. Players are faced with an overwhelming amount of game-related information, including game descriptions, reviews, and ratings. Navigating through this flood of information to make informed decisions becomes time-consuming and cumbersome. To address these, there is a need for an intelligent game recommendation system that can alleviate information overload, provide personalized recommendations, facilitate game exploration, and optimize the search process.

This project focuses on implementing a game recommendation system based on a dataset consisting of over 4,000 popular games. The system incorporates two main recommendation techniques: description-based recommendation and tags recommendation. The description-based recommendation approach utilizes natural language processing (NLP) techniques to extract relevant information from game descriptions. By analysing the textual content, the system identifies similarities between games and recommends titles that share similar themes, genres. In addition to description-based recommendation, the system also incorporates tag recommendation. Tags are descriptive labels associated with games that provide additional information about their characteristics. By analysing the tags assigned to each game in the dataset, the system can identify games with similar tags and recommend them to users based on their preferences. Overall, this project combines various technologies, including JavaScript, CSS, HTML, React framework, and Flask lightweight framework, to develop this robust game recommendation system.

1.Project Background

The gaming industry has witnessed unprecedented growth, resulting in a vast and ever-expanding library of available games. While this abundance of choices is exciting, it has also given rise to some significant challenges for users.

One of the major difficulties faced by both gamers and game developers is the issue of game discovery. Despite the presence of numerous high-quality games,

many struggle to gain attention and visibility in the highly competitive market. This is often due to promotional restrictions, budget limitations, or lack of marketing resources. As a result, these hidden gems, which could provide unique and enriching experiences, go unnoticed by the majority of players. Furthermore, even after a game is launched, users encounter challenges in assessing its initial evaluation and quality. Reviews and ratings can be unreliable or biased, and it becomes difficult for players to make informed decisions about which games to explore further. This lack of comprehensive and trustworthy information hinders the discovery process, leaving users uncertain about the games they should invest their time and money in. Another aspect that contributes to the difficulty in game discovery is the limitations of traditional filtering systems. Standard filters, such as game type or age rating, are often too simplistic and fail to capture the nuanced preferences and interests of individual users. Different players have diverse gaming preferences, and relying solely on basic filters neglects the intricacies of their gaming tastes. Consequently, there exists a significant gap between what users are seeking and the games they come across, leading to frustration and missed opportunities for both gamers and developers.

2.Demand Analysis

2.1 Market Search

By analyzing and summarizing the recommendation models of existing game recommendation systems in the market. Major platforms such as Steam, PlayStation Network, Taptap, Xbox and Epic Games store all have their own recommendation method. Some based on user purchase history and preference of game, some based on their social connections and community interests, but most of them do not have a personalized recommendation function for users' personal preferences and game preferences. Users cannot get the game recommendations they want properly through their own descriptions. It is difficult to satisfy individual needs simply by relying on basic filters. So we need an intelligent system that can recommend games based on users' own descriptions.

For the market, the gaming community is seeking more personalized and tailored experiences. Users are overwhelmed by the vast number of games available and are looking for efficient ways to discover content that aligns with

their specific interests and preferences. For users, users crave a game recommendation system that understands their unique requirements and can provide accurate suggestions based on their descriptions. They desire a system that goes beyond basic tag filters and takes into account their individual gaming tastes, gameplay styles, and desired experiences.

2.2 System Demand

According to the market demands and user demands and in order to develop an effective game recommendation system, the design requirements have been defined as follows:

- Accuracy and Relevance: The system should provide accurate and relevant game recommendations to users. The recommendations should align with the user's preferences.
- Scalability: The system should be designed to handle a large volume of game data.As the game catalog, the system should efficiently process and recommend games without compromising performance.
- Multiple Recommendation Approaches: The system should employ multiple recommendation approaches, such as game name-based, user description-based, and game tag-based recommendations. Combining these approaches will enhance recommendation accuracy and cater to different user preferences and discovery patterns.
- User-Friendly Interface: The system should have a user-friendly interface that allows users to easily navigate and interact with the recommendation features.

By addressing these system design requirements, the game recommendation system aims to provide an efficient, personalized, and user-centric experience, enabling users to discover and enjoy games that align with their interests and preferences.

3 Recommendation System Design

3.1 System Architecture

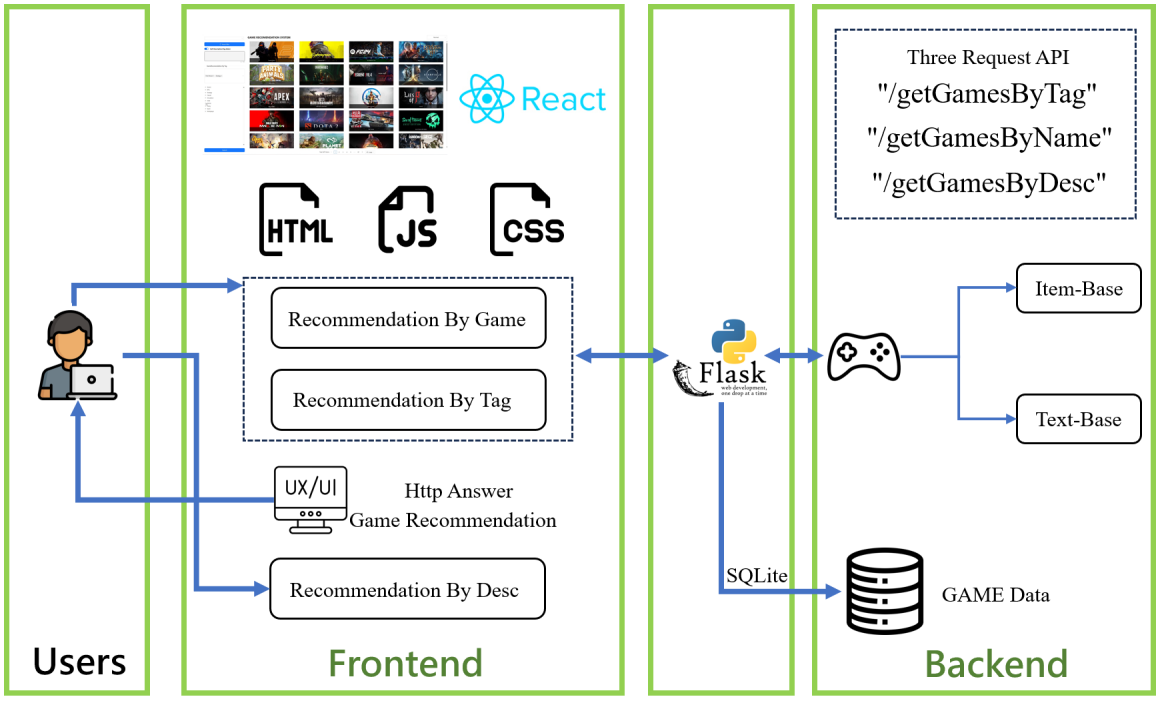


Figure 3.1: System Architecture

Through the user interface, the user and the system are connected. The game recommendation model, which is the heart of the system, is built on the user description, game similarity, and Tags & Wilson interval. Databases provide each model with the necessary data.

The user interface is the web. The site implements all of the functions for our recommendation models based on the user-friendly principle. The model can rapidly obtain user requirements through the user interface and provide precise recommendations to users. The user may easily and rapidly obtain the desired games that meet the specifications.

The data that is being mined from Steam is stored in the database. All of the models are supported by the database with the necessary data respectively. They require different data for their different recommendation models. The fundamental component of the system is the database. Increasing the database's quantity is crucial for improving the web functionality.

The core of the web is the recommendation model. The three models that make up the recommendation system are: recommendation based on user description, recommendation based on game similarity, and recommendation based on tag and Wilson interval. Based on user input and recommendation algorithms, models generate suggested games.

3.2 Algorithm For Recommendation

Our game recommendation system has three recommendation modes, which are based on the Wilson interval of Users' Positive Ratings, based on game similarity and based on user description. With these three recommendation modes, users can get the most popular games under a certain type by ticking the tabs on our website. They can also click on the game they are interested in to get multiple game recommendations that are most similar to that game, and they can use a chatbot to describe their favorite game as well.

3.2.1 Recommendation based on tags and wilson interval

The recommendation model based on tags and Wilson interval recommends games by games' tags that the user is interested in, and then ranks the games that satisfy the tag requirements based on the number of reviews and positive ratings of the games, and ultimately recommends multiple games with higher rankings to the user.

In order to use tags to recommend games, we build a knowledge graph. The graph shows as below:

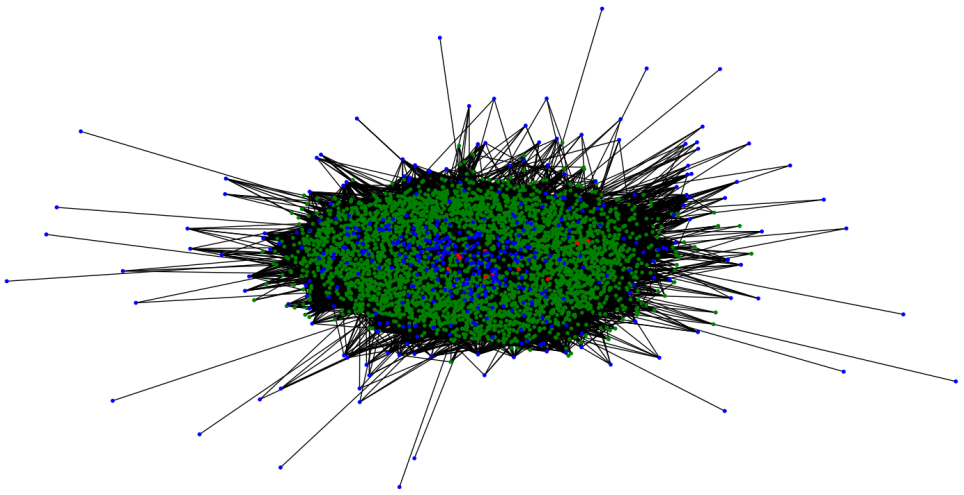


Figure 3.2: Knowledge Graph

In this graph, the red nodes represent the type of game, the blue nodes represent the tag of game and the green nodes represent games. By using the knowledge graph of this, we can get which type a game belongs to and which tags it has, and can also get how many games there are under a genre or tag.

After constructing the knowledge graph, we need a metric to know which games we should prioritize for recommendation. Ranking products directly based on positive feedback is the ranking strategy adopted by the majority of online gaming sites: the higher rating, the higher ranking of the product. There is an obvious flaw in this ranking method, which ignores the influence of the total number of reviews on the positive rating. The impact is more in the credibility of the reviews.

Table 3-1 Ranking of products based on positive ratings (A - E)

Ranking	Game	Positive Evaluation Rate/%	Comment Number
1	A	100	2
2	B	100	5
3	C	98	26
4	D	95	100
5	E	85	300

The above table gives an example: game A has a positive rating of 100% and a review count of 2, game C has a positive rating of 98% and a review count of 26. If based on the traditional evaluation method of good rating, game A is ranked before game C. However, according to the statistical sampling theory, it can be concluded that when evaluating using the positive evaluation rate, game C is more reliable than game A. In addition, comparing game A and game B, both of them have 100% positive feedback rate, but the number of reviews of game B is much larger than that of game A. If based on the traditional evaluation method of positive evaluation rate, it is impossible to distinguish between the two games. Therefore, for any two games, if both have the same positive rating, their respective number of positive reviews should also be considered for a combined judgment. If they have different positive ratings, the number of reviews will also affect the final ranking to a certain extent.

To solve this problem, we use Wilson intervals algorithm. The Wilson interval is a correction formula for the confidence interval problem for the binomial distribution proposed by American mathematician Edwin Bidwell Wilson in 1927. The calculation of Wilson's confidence interval relies on the game's good reviews ratings and the number of reviews. It is calculated by the following formula:

$$Score = \frac{p + \frac{1}{2n}z_{1-\alpha/2}^2 \pm z_{1-\alpha/2}\sqrt{\frac{p(1-p)}{n} + \frac{z_{1-\alpha/2}^2}{4n^2}}}{1 + \frac{1}{n}z_{1-\alpha/2}^2}$$

p denotes the rate of positive reviews, n denotes the number of reviews, $z_{1-\alpha/2}^2$ denotes the z-statistic for a certain confidence level. The lower limit of the wilson interval is:

$$Score = \frac{p + \frac{1}{2n}z_{1-\alpha/2}^2 - z_{1-\alpha/2}\sqrt{\frac{p(1-p)}{n} + \frac{z_{1-\alpha/2}^2}{4n^2}}}{1 + \frac{1}{n}z_{1-\alpha/2}^2}$$

According to the formula for wilson's confidence interval, it can be determined that the confidence the width of the interval is:

$$width = \frac{2z_{1-\alpha/2}\sqrt{\frac{p(1-p)}{n} + \frac{z_{1-\alpha/2}^2}{4n^2}}}{1 + \frac{1}{n}z_{1-\alpha/2}^2}$$

The mean of this confidence interval is:

$$mean = \frac{p + \frac{1}{2n}z_{1-\alpha/2}^2}{1 + \frac{1}{n}z_{1-\alpha/2}^2}$$

When n is large enough, the mean tends to p, which means that when the number of game reviews is large enough, the narrower the confidence interval is. The lower limit is close to the rate of good reviews and the ranking of games

is mainly based on the rate of good reviews. On the contrary, when n is less than a certain value, the mean is much less than p. At this time, the wider the confidence interval, the greater the difference between its lower limit and the good rate, and the result is that the effect of the good rate on the game ranking is weakened.

Figure 3.3 gives the top ten outputs for the input labels 'Action', 'RPG' and 'Casual':

```
['id:', 1794680, 'name:', 'Vampire Survivors', 'wilson_score:', 0.8296159647743553]
['id:', 1942280, 'name:', 'Brotato', 'wilson_score:', 0.7399898321718621]
['id:', 333980, 'name:', "AKIBA'S TRIP: Undead & Undressed", 'wilson_score:', 0.675237152127812]
['id:', 840260, 'name:', 'Endless World Idle RPG', 'wilson_score:', 0.6299209388207752]
['id:', 2342950, 'name:', 'God Of Weapons', 'wilson_score:', 0.5621183167638024]
['id:', 204360, 'name:', 'Castle Crashers®', 'wilson_score:', 0.5427327772615373]
['id:', 704850, 'name:', 'Thief Simulator', 'wilson_score:', 0.5308328816987251]
['id:', 17390, 'name:', 'SPORE™', 'wilson_score:', 0.5255393363554683]
['id:', 1402110, 'name:', 'Eternights', 'wilson_score:', 0.4977450973072504]
['id:', 454650, 'name:', 'DRAGON BALL XENOVERSE 2', 'wilson_score:', 0.4871840283499129]
```

Figure 3.3: Wilson Score Output Result

3.2.2 Recommendation based on game similarity

In this part, we hope that our system can recommend some games based on the game similarity. Users can choose some games which they are interested in and then the web page will return some game recommendations which are similar to those of user interest. We use two algorithms to measure the similarity between the games, which are Jaccard index and semantic similarity.

The Jaccard index, also known as the Jaccard similarity coefficient, is used to compare similarities and differences between finite sample sets. Given two sets A and B, the Jaccard coefficient is defined as the ratio of the size of the intersection of A and B to the size of the concatenation of A and B, defined as follows:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Spacy is able to compare two objects, and make a prediction of how similar they are. Predicting similarity is useful for building recommendation systems or

flagging duplicates. The similarity method in spacy is used to calculate the similarity between two texts which can used to be the semantic similarity.

The underlying principle of similarity calculation is typically based on the concept of word embeddings or word vectors. Word embeddings are real-valued vectors that map words into a high-dimensional space, where semantically similar words are close to each other. One commonly used word vector model is Word2Vec.

In spacy, the nlp.similarity method uses a pre-trained word embedding model (such as Word2Vec) to calculate the similarity between texts. It employs cosine similarity as the metric for similarity measurement. Cosine similarity measures the cosine of the angle between two vectors in a high-dimensional space, and it ranges from -1 to 1. A value closer to 1 indicates higher similarity.

Word2vec model is a model for efficiently training word vectors proposed by Tomas Mikolov et al. in 2013, the basic starting point is that two words with similar contexts should have similar word vectors. Word2vec consists of two structures, CBOW and Skip-Gram. CBOW predicts the probability of the current word from the words that appear in the context, while Skip-Gram predicts the probability of each word in the context based on the current word. The essence of Word2vec is to train the weights of the neural network in a shallow neural network model, so as to maximize the overall probability of the generation of all the words in the corpus. The weight matrix obtained in the model is the vector representation of each word. Figure 3.4 gives the structure of the Word2vec model.

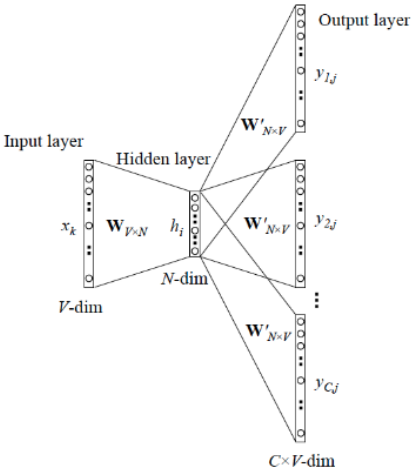


Figure 3.4: Word2vec model

In this project, we tried both Jaccard index and semantic similarity. As a result, we found that the jaccard measure is faster to compute and the semantic similarity has a better match performance, but it takes longer to compute.

Figure 3.5 gives the top ten outputs for the input games Counter-Strike 2:

```
['id:', 222880, 'name:', 'Insurgency']
['id:', 209160, 'name:', 'Call of Duty®: Ghosts']
['id:', 107410, 'name:', 'Arma 3']
['id:', 16900, 'name:', 'GROUND BRANCH']
['id:', 359550, 'name:', "Tom Clancy's Rainbow Six® Siege"]
['id:', 578080, 'name:', 'PUBG: BATTLEGROUNDS']
['id:', 500, 'name:', 'Left 4 Dead']
['id:', 24240, 'name:', 'PAYDAY™ The Heist']
['id:', 291480, 'name:', 'Warface']
['id:', 218620, 'name:', 'PAYDAY 2']
```

Figure 3.5: Similarity Output Result

3.2.3 Recommendation based on user description

The recommendation model based on the user's description is a preferable option when users are unable to separate their needs into certain game categories or tags. Based on the user's written description, this model makes recommendations.

The game description is one of the features of the game in the database. Numerous game details, such as the game's category, backdrop, and elements, are included in the game description. It is possible and effective to make the recommendation by comparing the text similarity between the game description and the user description. On the other hand, there are also cases where the game description section only displays the game's awards and not the game elements. This has implications for recommendations.The game tags supplement the game description in order to address this issue. As of right now, the game corpus consists of game tags and game descriptions.

Term Frequency - Inverse Document Frequency (TF-IDF) is a widely used statistical method in natural language processing. It measures how important a term is within a document relative to a collection of documents (i.e., relative to a corpus). Words within a text document are transformed into importance numbers by a text vectorization process.

Term Frequency: TF of a term or word is the number of times the term appears in a document compared to the total number of words in the document.

$$TF = \frac{number\ of\ times\ the\ term\ appears\ in\ the\ document}{total\ number\ of\ terms\ in\ the\ document}$$

Inverse Document Frequency: IDF of a term reflects the proportion of documents in the corpus that contain the term. Words unique to a small percentage of documents (e.g., game elements) receive higher importance values than words common across all documents (e.g., a, the, and).

$$IDF = log(\frac{number\ of\ the\ documents\ in\ the\ corpus}{number\ of\ the\ documents\ in\ the\ corpus\ contain\ the\ term})$$

The TF-IDF of a term is calculated by multiplying TF and IDF scores

$$TF - IDF = TF * IDF$$

Cosine Similarity is commonly used to calculate text similarity and is a method for measuring the angle between two vectors. Cosine Similarity is not affected by the absolute length of the vector, only the direction of the vector. This makes sense for texts, which can vary greatly in length, but the similarity between them mainly depends on shared keywords and semantics, not the length of the text. At the same time, in text processing, a large number of features are often used to represent words or phrases, resulting in a high-dimensional space. Cosine Similarity is still effective in high-dimensional space, and the calculation is relatively simple.

First, text preprocessing is performed on user input, including lowercase text, stop word removal, and tokenization. And preprocess the game tags, and combine the game tags with the game description accordingly. Through the TfidfVectorizer, user input TF-IDF matrix and game corpus TF-IDF matrix are generated. Next, use the cosine similarity technique to get the similarity scores. And arrange the similarities to create the list of game recommendations.

4.System Development

4.1 Database

SQLite is a lightweight embedded relational database management system (RDBMS). It's an open-source, zero-configuration, serverless SQL database engine. Unlike most traditional database management systems, SQLite doesn't require a separate server process or system configuration; it can directly access regular disk files as the database storage.

SQLite has some key features, such as zero configuration, lightweight, transaction support, untyped and standard SQL support. SQLite is primarily used in scenarios involving lightweight applications, embedded devices, and mobile applications. It's popular for local storage on mobile devices and in small-scale applications.

Our database table has 14 attributes, including id, name, class, label, language, date, PER, comment number, wilson score, processor, memory, storage, image url and game description. Some of the dataset are shown below:

id	name	class	label	language	date
过滤	过滤	过滤	过滤	过滤	过滤
949230	Cities: Skylines II	['Simulation']	['Simulation', 'City Builder',…	['English', 'French', 'Italian…	['24 Oct, 2023']
2252570	Football Manager 2024	['Simulation', 'Sports', …	['Simulation', 'Sports', …	['English', 'French', 'Italian…	['6 Nov, 2023']
2131680	METAL GEAR SOLID: MASTER …	['Action']	['Action', 'Adventure', 'Actio…	['English', 'French', 'Italian…	['24 Oct, 2023']
2072450	Like a Dragon: Infinite Wealth	['Action', 'Adventure']	['Action', 'Adventure', 'Actio…	['English', 'French', 'Italian…	['25 Jan, 2024']
1849250	WRC	['Racing', 'Simulation', …	['Racing', 'Simulation', …	['English', 'French', 'Italian…	['2 Nov, 2023']
553850	HELLDIVERS™ 2	['Action']	['Action', 'Third-Person …	['English', 'French', 'Italian…	['8 Feb, 2024']
1778820	TEKKEN 8	['Action']	['Action', 'Fighting', '3D …	['English', 'French', 'Italian…	['25 Jan, 2024']
2161700	Persona 3 Reload	['Adventure', 'RPG', 'Strategy']	['RPG', 'Adventure', 'Strategy…	['English', 'French', 'Italian…	['1 Feb, 2024']
2375550	Like a Dragon Gaiden: The Man …	['Action', 'Adventure']	['Action', "Beat 'em up", …	['English', 'Japanese', …	['8 Nov, 2023']
2328310	CUSTOM MECH WARS	['Action']	['Mechs', 'Action', 'Third-…	['English', 'Japanese']	['14 Dec, 2023']
2238900	STAR OCEAN THE SECOND STORY R	['Action', 'RPG']	['RPG', 'JRPG', 'Action', …	['English', 'French', 'Italian…	['2 Nov, 2023']
2022670	Sonic Superstars	['Action']	['Action', 'Adventure', …	['English', 'French', 'Italian…	['16 Oct, 2023']
1658280	Eiyuden Chronicle: Hundred …	['Action', 'Adventure', 'Indie…	['RPG', 'JRPG', '2.5D', 'Story…	['English', 'French', 'Italian…	['23 Apr, 2024']
2144740	Ghostrunner 2	['Action']	['Action', 'Cyberpunk', 'Fast-…	['English', 'French', 'Italian…	['26 Oct, 2023']
2254740	Persona 5 Tactica	['RPG', 'Strategy']	['RPG', 'Strategy', 'Action', …	['English', 'French', 'Italian…	['16 Nov, 2023']
1643320	S.T.A.L.K.E.R. 2: Heart of …	['Action', 'Adventure', 'RPG']	['Open World', 'FPS', 'Post-…	['English', 'French', 'Italian…	['Q1 2024']
2478970	Tomb Raider I-III Remastered	['Action', 'Adventure']	['Action', 'Action-Adventure',…	['English', 'French', 'Italian…	['14 Feb, 2024']
1150530	Wizard with a Gun	['Action', 'Adventure', 'Indie']	['Action', 'Adventure', 'Actio…	['English', 'French', 'German'…	['17 Oct, 2023']
1485590	ENDLESS™ Dungeon	['Action', 'Strategy']	['Action', 'Roguelike', …	['English', 'French', 'German'…	['19 Oct, 2023']
2243980	Marshmallow Imouto Succubus	['Adventure', 'Casual']	['Casual', 'Visual Novel', …	['English', 'Japanese', …	['22 Sep, 2023']
1493640	Banishers: Ghosts of New Eden	['Action', 'Adventure', 'RPG']	['Adventure', 'Action', 'Story…	['English', 'French', 'Italian…	['13 Feb, 2024']
1844380	Warhammer Age of Sigmar: Realm…	['Action', 'Strategy']	['Strategy', 'Action', 'Action…	['English', 'French', 'Italian…	['17 Nov, 2023']
2051120	HOT WHEELS UNLEASHED™ 2 - …	['Action', 'Adventure', …	['Arcade', 'Family Friendly', …	['English', 'French', 'Italian…	['19 Oct, 2023']
706270	Space Zombies Invasion	['Action', 'Adventure', …	['Action', 'Adventure', 'Indie…	['English', 'Spanish - Spain']	['2 Mar, 2018']
1333200	Song of Nunu: A League of …	['Adventure']	['Adventure', 'Puzzle', 'Story…	['English', 'French', 'Italian…	['1 Nov, 2023']
2238430	Mon-Yu: Defeat Monsters And Ga…	['RPG']	['RPG', 'JRPG', 'Dungeon …	['English', 'Japanese']	['21 Sep, 2023']

Figure 4.1: Dataset

4.2 Backend

4.2.1 Overview

The backend of the game recommendation system is built using Flask, a lightweight web framework for Python, which facilitates seamless communication between the frontend and backend components. The backend is responsible for handling various API requests from the frontend, implementing the necessary logic, and interacting with the SQLite database for data retrieval and storage.

4.2.2 Flask Integration

Flask is utilized to create a restful API that forms the communication bridge between the frontend and backend. It enables the handling of HTTP requests, such as GET and POST, and allows for the exchange of data in JSON format. Flask's routing mechanism is leveraged to define the API endpoints and associate them with the appropriate backend functions.

4.2.3 SQLite Database Integration

The backend utilizes the SQLite database for efficient data storage and retrieval. SQLite is a lightweight, serverless database engine that provides a relational database management system. It offers simplicity, portability, and ease of integration with Python applications. The backend interacts with the database to fetch game information and other relevant data required for generating recommendations.

4.2.4 API Implementation

API (Application Programming Interface) is a collection of definitions, programs, and protocols that enable communication between computer software. APIs are primarily used in the backend to provide services to the frontend. By making API calls, the frontend can send requests and parameters, and subsequently accomplish complex tasks by utilizing the data returned by the backend. In our system, the backend implements three APIs to handle different aspects of the game recommendation system. These APIs include:

- "/getGamesByName" : This API is used for the search box in the head of our website. When a user wants to find a game but could not remember the whole name, he can enter an incomplete name of this game in the search box. When the user selects the searched game, various information about the selected game will appear on the right end of the web page. And in the middle, other games related to the selected game will be displayed.

Implementation process:

Inputs are the name of the game the user is interested in. Outputs are recommended games information. In this function, we need to calculate semantic similarity between the input game and other games. And then compare the semantic similarity values, the larger the value, the higher the game's recommended ranking. After ranking, information of top 50 similar games would be picked and returned to the frontend.

- "/getGamesByDesc" : This API is used for the recommendation based on description. There is a text-input area on our website. In this area, users could input their own description about the game they want like “I wanted an adrenaline-pumping action shooter set in an ultra-modern city with mecha elements. Featuring exciting combat, advanced weapons and driving controls.”, then our system will recommend games based on the description entered. The game recommend will show in the middle.

  Implementation process:

  The input is the user’s text description of the desired game. The output is recommended game information. In this function, we need to calculate the text similarity between the input description and all game descriptions. Then the similarity is sorted to obtain the game recommendation ranking. After the ranking is completed, the top 100 game information will be selected and returned to the front end.

- "/getGamesByTag" : This API is used for the tag recommendation. There are ten classes and hundreds of tags in the list shown on the right of our website. Users could choose tags which are relative to the game they want such as FPS, Strategy and Multiplayer. And our system will return the games relative to these tags.

  Implementation process:

  Inputs are the tags selected by the user on the web page. Outputs are recommended games information. In this function, we need to find all related games with input tags from the knowledge graph. And then calculate games’ wilson score according to the comment number and

positive evaluation rate. Finally, sorting games according to the wilson score and returning the sorted games to the frontend.

4.3 Frontend

The frontend of the game recommendation system is built using React, a popular JavaScript library for building user interfaces. React provides a modular and component-based approach to develop interactive and responsive web applications.React enables the creation of a dynamic and interactive user interface. The frontend utilizes HTML, CSS, and JavaScript to design and style the components, ensuring a visually appealing and user-friendly interface. In the frontend, we also incorporate CSS frameworks like Antd to facilitate responsive design and consistent styling across the application. The frontend is responsible for presenting the user interface, interacting with the users, and communicating with the backend to fetch data and display game recommendations.

In addition, the frontend communicates with the backend using Axios, a popular JavaScript library for making HTTP requests. Axios provides a straightforward and flexible way to send requests to the backend API endpoints.

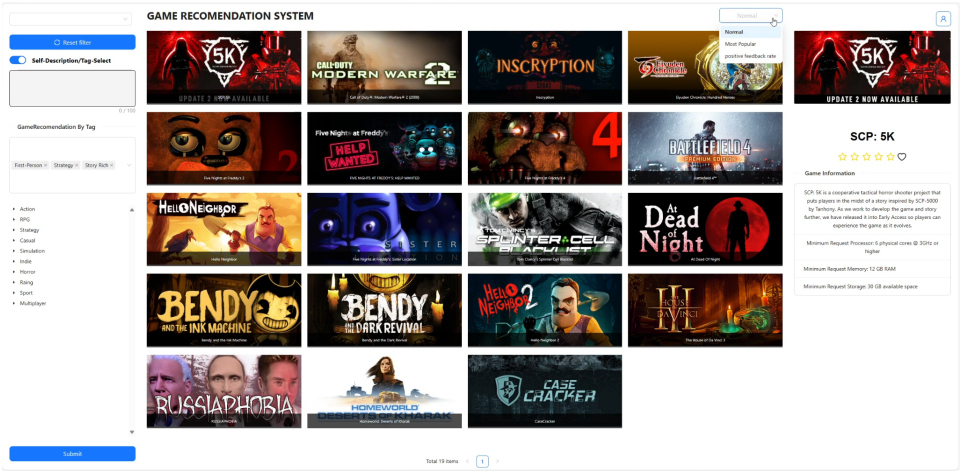


Figure 4.3: Web page

5.conclusion

5.1.Summary

Our team has developed a Games Recommendation web application that caters to diverse needs of gamers. The application offers three distinct recommendation modes to accommodate different user preferences. The first mode is designed for avid gamers who already have a collection of favorite games. We can recommend games that share the same content and genres with the game they choose. The second mode targets users who are unsure about their preferred game name. For these individuals, our system can recommend games through the tags they choose. The third mode caters to users with specific criteria for their ideal games. Users can provide a description of their ideal game, specifying their desired features, genres, or gameplay elements. Leveraging this information, our system utilizes intelligent algorithms to suggest games that meet their requirements.

5.2 Future improvement

Despite a successful implementation for a minimum viable product, there are areas for future improvements:

- A function block can be added: users can input all games they have played before and their ratings for these games. Based on this information, our recommendation system can conduct a comprehensive analysis to provide game recommendations;
- Using a more refined algorithm to enhance the computational performance of game similarity: the present similarity calculation algorithm simply merges the input labels into a string and applies the word2vec model to transform it into an embedding vector. After that the similarity value is obtained by cosine similarity. We would like to preprocess the tags to get a more reasonable string representing the game, and use a more advanced model to compute the textual similarity between the two strings in order to recommend more accurate games.
- User data storage: The system allows users to log in and record user game data into the database, and recommend games based on user similarity.

6.Appendix

6.1. Proposal

Date of proposal : 10/01/2023

Project Title : Game Recommendation

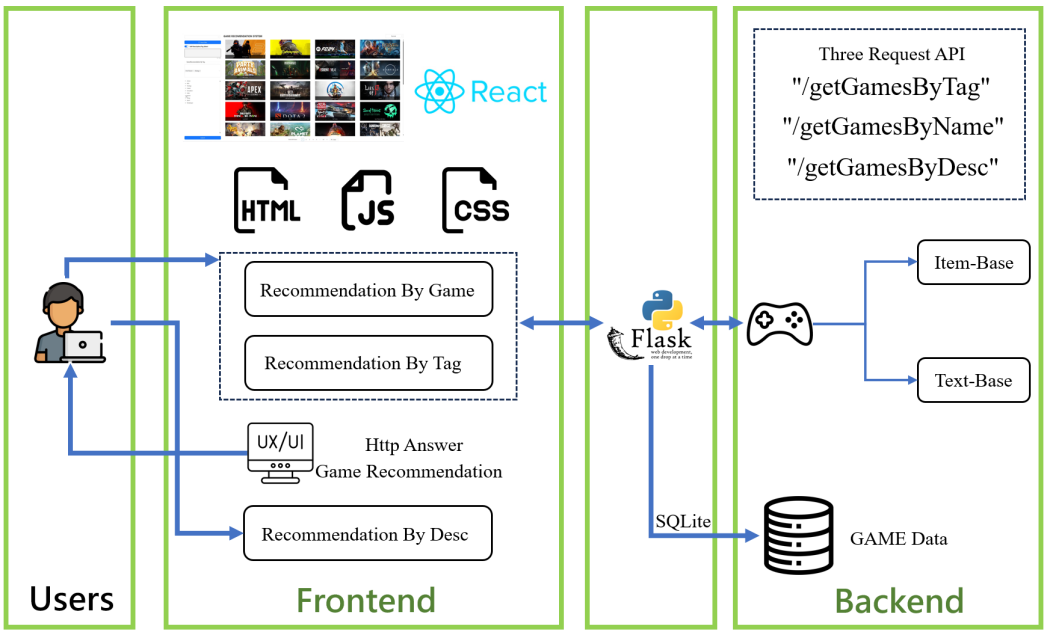
TEAM NUMBER:

- FU CHENTAO A0285737M
- DENG YUXUAN A0285746M
- LIN FANZHI A0285725U
- TU RUNZHANG A0285767H

Background/Aims/Objectives:

The gaming industry is witnessing an exponential growth in the number and diversity of available games. With this abundance of options, players often find it challenging to navigate through the vast catalog of games and identify those that align with their preferences. Players are faced with an overwhelming amount of game-related information, including game descriptions, reviews, and ratings. Navigating through this flood of information to make informed decisions becomes time-consuming and cumbersome. To address these, there is a need for an intelligent game recommendation system that can alleviate information overload, provide personalized recommendations, facilitate game exploration, and optimize the search process.

System Architecture:



Requirements Overview:

Research ability Programming ability Two user function: 1. Game-based recommendations 2. User-desc recommendations
Resource Requirements (Hardware, Software and any other resources) Local Server: Windows 11-- Intel 13980HX CPU 32GB Memory Reasoning systems: Statistics-based recommendation, NLP Language: Python, HTML, JavaScript, CSS Natural language processing tools: Dialog Flow, NLTK Database: MySQL or SQLite Framework: Flask Tools: Pycharm, VSCode

6.2 System Functionalities Mapping

Courses	**Knowledge and Techniques**	**Project**
Machine Reasoning (MR)	Text Pre-Processing	Recommendation based on user description
Reasoning Systems (RS)	Knowledge Graph Cosine Similarity Jaccard index	Recommendation based on tags and wilson interval Recommendation based on user description Recommendation based on game similarity
Cognitive Systems (CGS)	NLP	Recommendation based on user description

6.3 Installation and User Guide

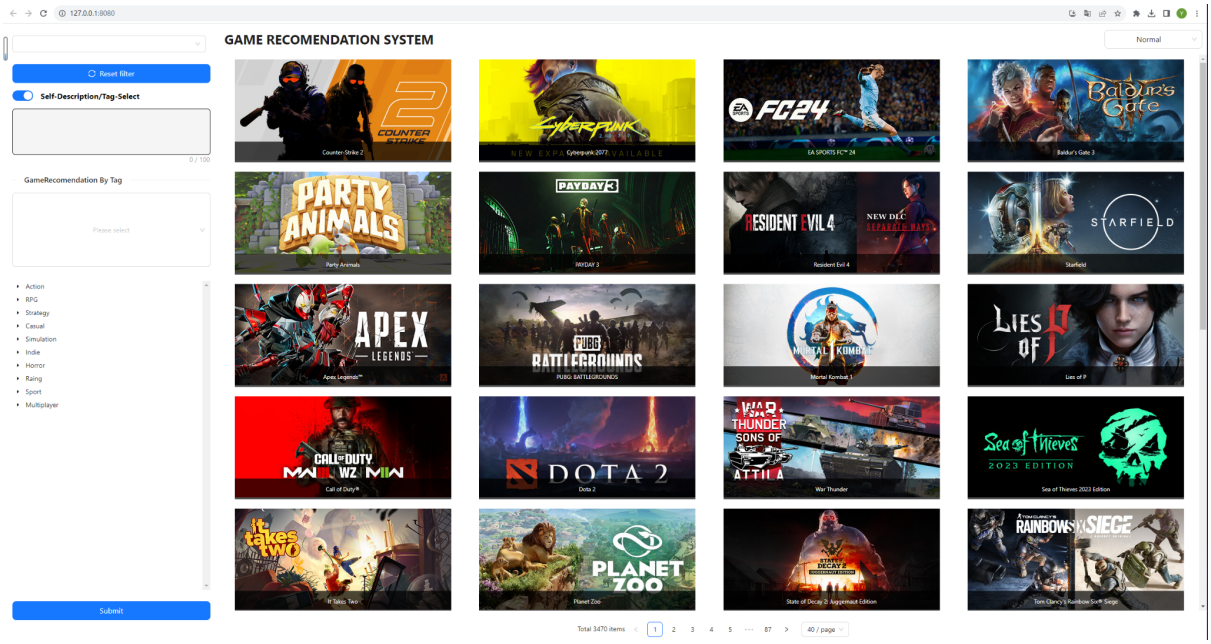
6.3.1 Local Installation Guide

1. Step1. Clone our project from github

- Git clone https://github.com/facerrr/IRS-PM-2023-10-29-GRP7-GamesRecommendation.git
- Cd backend

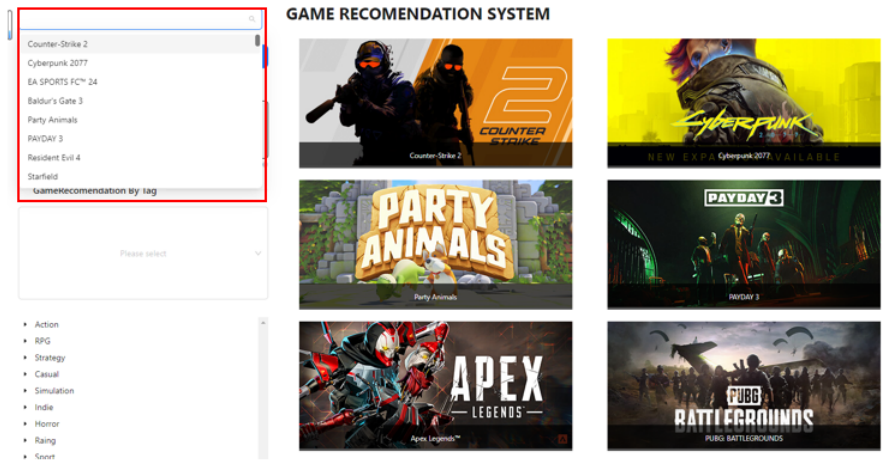
2. Step2. Install environment via requirement.txt
    - Pip install -r requirements.txt
3. Step3. Run the project
    - Python main.py

Open the url: http://127.0.0.1:8080/ and you will see the website

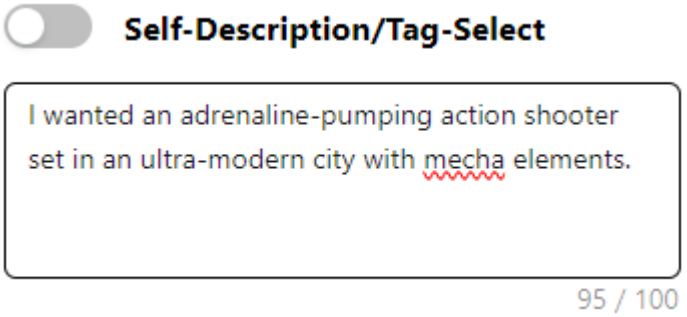


6.3.2 User Guide

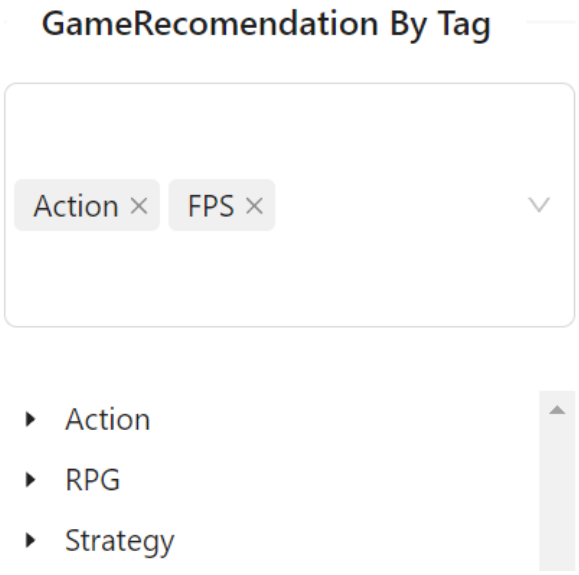
1. Get Recommendation through game name
   In the search box, you can search a game that is sorted in our database.



2. Get Recommendation through self description.
   In the text area, you can input your own description about the game you want.



3. Get Recommendation through tags, choose tags relative to the game they want such as Action and FPS.



6.4 Individual Reports

Deng Yuxuan

1.Personal contribution to group project

- Help design the overall system architecture, including front-end, back-end, and algorithm implementations.
- Help determine the technology stack, frameworks, and languages to be used.
- Developed the front-end and backend api that connect to the frontend.
- Created the installation and user guide for the project.
- Responsible for writing specific sections of the project report

2.What you learn is most useful for you.
The most useful learning experience for me during this project was gaining hands-on experience in designing and implementing the overall system. This involved considering various factors such as scalability, performance, and integration of front-end, back-end, and algorithms. By being actively involved in this project, I developed a deep understanding of how different components of a system interact and contribute to its overall functionality. I developed some skills in NLP and designing recommendation system, and also gained hands-on experience in creating visually appealing and interactive user interfaces through independent front-end development.

3.How you can apply the knowledge and skills in other situations or your workplaces

- The experience in designing and implementing an end-to-end system can be helpful for taking up similar projects in the future that involve building products or solutions from scratch. The learnings around system design, integration of components, and delivering full stack solutions will be broadly applicable.
- The NLP skills developed can be applied in other projects that involve text processing, classification or extraction of insights from textual data. Knowledge of working with NLP libraries and models can be directly useful.
- Experience in building a complete web application, and knowing how to write backend can be applied in any other project involving UI/UX development, which is needed in any industry.

Fu Chentao

1.Personal contribution to group project

- Participate in the overall design of the project and product concept proposal and functional design
- Help design backend system framework Flask and algorithm deployment.
- Test project vulnerability and put forward suggestion

- Make a video of project introduction and demonstration of this game recommendation system

2.What you learn is most useful for you.
In this project, I learned the importance of team communication and collaboration during the development process. I also learned how to use Flask as a backend framework, which is a lightweight Python web framework. Learned how to create routes, handle HTTP requests and responses, connect to databases, and write API endpoints, etc. Flask provides a clean and flexible way to build server-side applications. Mastered the fundamentals and implementation of game recommendation algorithms. This involves using techniques such as machine learning or collaborative filtering to recommend games that match their preferences based on user input or selected tags. I learned to process game data, feature extraction, similarity calculation and generation of recommendation results. I learned how to communicate between the front-end and back-end, usually using a RESTful API. I learned how to design and define API endpoints so that the front-end can send requests to the back-end and receive responses. I Used JSON format for data transfer and learned to handle cross-domain requests, error handling, and authentication. Finally, I also learned about project management and team communication and collaboration during the development process.

3.How you can apply the knowledge and skills in other situations or your workplaces

- I have learned about game recommendation algorithms and database management, and can apply them to other data analysis and recommendation system development. I can use similar algorithms and techniques to build a recommendation system for personalized recommendation of products, music, movies, etc. Data analysis techniques can also be applied to process and analyze a large amount of data, extracting valuable information and insights from it.
- For the skills of front-end and back-end development, I can apply them to other web development projects. I can use Flask or other backend frameworks to handle business logic and data interaction. This is suitable for building various types of web applications, such as e-commerce platforms, social media applications, blog systems, etc

- Project management and team collaboration are very important everywhere. I can use project management tools and version control systems to track tasks and progress, communicate and collaborate with team members. This is a beneficial skill for any project that requires collaborative development.

Lin Fanzhi

1. Personal contribution to group project

- Assist in establishing the project scope and create the system architecture.
- Develop and code the recommendation model based on the user description.
- Assist with the overall assembly and test.
- Complete the appropriate report and presentation sections.

2. What you learn is most useful for you.

For me, building a recommendation model based on user descriptions has significantly helped and improved my capabilities. I am now well-versed in text pre-processing, TD-IDF, and similarity algorithms. Even if the code part is simple, I get more knowledge about text similarity comparison, am familiar with the TD-IDF's algorithm principle, understand why cosine similarity is used for text similarity calculation, and so on. In addition to my core expertise, I am aware of the value of teams and how members interact and collaborate with one another. This is also very important for future work

3. How can you apply the knowledge and skills in other situations or your workplaces?

Text similarity comparison can be used in search engine optimization to improve visibility in search engine results by comparing text content on websites to optimize keyword selection and page ranking. It can be used in companies that need to do content recommendation, social media analysis, and even legal document comparison.

Tu Runzhang

1. Personal contribution to group project

- Using crawler to get game information from steam and some other game platform and preprocess the dataset.
- Designing database structures and back-end recommendation algorithms.
- Developed the recommendation based on tags and wilson interval api.
- Developed the recommendation based on game similarity api.
- Responsible for writing specific sections of the project report.

2.What you learn is most useful for you.

Through this project, I learnt how to use a crawler to get the required information from a website, how to design and build a database, and how to clean and pre-process data from raw data. I gained a deeper understanding of the use of sql and learnt how to call databases using python. I also learnt the Wilson interval algorithm and some text processing methods like word embedding, word2vec models and so on. Moreover, I have a better understanding of how to design a system from the whole to the local level. I also verified what I had learned in class in practice and realized the significance of these algorithms in practical application. For example, I know using spaCy for NLP is a great way to analyze user input and make recommendations based on user input. Overall, through this project, I learned how to combine the AI technology I learned with real life, and how to apply AI in product development.

3.How can you apply the knowledge and skills in other situations or your workplaces?

- In this project, I learnt how to design and develop a game recommendation programme. In my future study and work, I can apply the experience in this project to other scenarios, which may no longer be game recommendation, but may be food recommendation, shopping recommendation, film recommendation and so on. Maybe the recommendation methods and goals are different, but I think the design concepts and methods are similar.
- In this project, I learnt a lot of methods about data processing and text processing. These methods can probably be applied to my future projects. For example, if I need to design an NLP system in the future, I might

apply the text preprocessing and word embedding methods used in this project to that system.

- Through this project experience, I realized the importance of teamwork and communication. Teamwork must be essential in other jobs as well. This experience allowed me to learn how to better communicate with my team members, divide up the work, and design and complete a complex project development task quickly and efficiently.