Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра прикладної математики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Проєктування рекомендаційної системи відеоігор
(тема)

Виконав:
здобувач 2 курсу, групи САУм-23-2
Дюг Р.В
(прізвище, ініціали)
Спеціальність
124 Системний аналіз
(код і повна назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма
Системний аналіз і управління
(повна назва освітньої програми)
Керівник доц. Матвієнко О.І.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ПМ _____ Сидоров М.В.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _інформаційно-аналітичних технологій та менеджменту
Кафедра _прикладної математики
Рівень вищої освіти _другий (магістерський)
Спеціальність _124 Системний аналіз
(код і повна назва)
Тип програми _освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма _Системний аналіз і управління
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ _____
(підпис)
“_25_” _листопада_ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _Дюгу Роману Володимировичу
(прізвище, ім’я, по батькові)

1. Тема роботи _Проєктування рекомендаційної системи відеоігор

затверджена наказом по університету від 22 листопада 2024 р. № 1228 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 6 січня 2025 р.

3. Вихідні дані до роботи _дані про відеоігри

4. Перелік питань, що потрібно опрацювати в роботі
1. Системний аналіз предметної області
2. Вибір і обґрунтування методу розв’язання
3. Програмна реалізація
4. Результати обчислювального експерименту
5. Аналіз можливих застосувань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

1. Актуальність теми роботи
2. Постановка задачі
3. Системний аналіз предметної області
4. Метод чисельного аналізу
5. Результати обчислювального експерименту

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	25 листопада – 1 грудня 2024 р.	виконано
2	Вибір та обґрунтування методу	2 – 8 грудня 2024 р.	виконано
3	Розробка алгоритму і програми	9 – 22 грудня 2023 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	23 – 29 грудня 2024 р.	виконано
5	Робота над текстом пояснювальної записки	30 грудня 2024 р. – 9 січня 2025 р.	виконано
6	Представлення роботи на рецензію в ЕК	10 січня 2025 р.	виконано

Дата видачі завдання _25_ листопада 2024 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Матвієнко О.І.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 62 с., 3 табл., 13 рис., 1 дод., 11 джерел.

АСОЦІАТИВНІ ПРАВИЛА, ВІДЕОІГРИ, РЕКОМЕНДАЦІЙНА СИСТЕМА, JAVA SCRIPT, MTMR, SQLITE.

Об’єкт дослідження – система рекомендації відео ігор.

Мета роботи – системний аналіз проблеми проєктування рекомендаційної системи відео ігор.

Методи дослідження – методи системного аналізу, математичного та комп’ютерного моделювання.

У кваліфікаційній роботі досліджується та вдосконалюється метод побудови рекомендаційної системи відеоігор на основі аналізу контенту.

Проведено системний аналіз предметної області. Виконано огляд і аналіз сучасного стану рекомендаційних систем із фокусом на об'єктивних і незалежних підходах до формування рекомендацій. Основними методами дослідження є використання асоціативних правил і методу TF-IDF для аналізу контенту відеоігор.

Створено програмне забезпечення для реалізації та тестування алгоритмів на основі контенту, яке не використовує персональні дані користувачів. Ефективність розробленої системи перевірено на наборах тестових даних, що включають текстовий опис, жанри, ключові слова та інші характеристики відеоігор.

Розроблене програмне забезпечення може бути використано платформами для продажу та рекомендації відеоігор, які прагнуть зберігати об'єктивність і незалежність рекомендацій. Отримані результати можна адаптувати для впровадження в інших сферах, де необхідні контент-орієнтовані рекомендації, такі як електронна комерція або освітні платформи.

ABSTRACT

Introductory note: 62 pages, 3 tables, 13 figures, 1 appendix, 11 sources.

ASSOCIATIVE RULES, JAVA SCRIPT, MTMR, RECOMMENDATION SYSTEM, SQLITE, VIDEO GAMES.

Object of research – a video game recommendation system.

Purpose of work – a systematic analysis of the problem of designing a video game recommendation system.

Methods of research – methods of systems analysis, mathematical and computer modeling.

The qualification work investigates and improves the method of building a video game recommendation system based on content analysis.

A systematic analysis of the subject area has been conducted. A review and analysis of the current state of recommendation systems have been carried out, focusing on objective and independent approaches to generating recommendations. The primary research methods include the use of association rules and the TF-IDF method for analyzing video game content.

Software has been developed for implementing and testing content-based algorithms that do not rely on users' personal data. The effectiveness of the developed system was tested on datasets containing textual descriptions, genres, keywords, and other characteristics of video games.

The developed software can be used by platforms for selling and recommending video games that aim to maintain objectivity and independence in recommendations. The obtained results can be adapted for implementation in other areas where content-oriented recommendations are required, such as e-commerce or educational platforms.

ЗМІСТ

С.

Перелік скорочень, умовних познак, одиниць і термінів 08

Вступ 09

1 Системний аналіз предметної області та постановка задач дослідження 12

1.1 Системний аналіз проблеми проєктування системи рекомендаційної системи відео ігор 12

1.1.1 Вербальна модель системи 12

1.1.2 Морфологічний опис системи 13

1.1.3 Функціональна модель системи 13

1.1.4 Інформаційна модель 17

1.2 Аналіз сценарію вирішення задачі проектування рекомендаційної системи, що базується на контенті, а не на персональних даних 18

1.2.1 Аналіз основних факторів, що впливають на рекомендації відеоігор 18

1.3 Формальна та змістовна постановка задачі 20

1.3.1 Формальна постановка задачі 20

1.3.2 Змістовна постановка задачі 22

1.4 Постановка задач дослідження 22

2 Вибір та обґрунтування методу розв’язання 23

2.1 Асоціативні правила 23

2.2 Пошук важливих слів TF-IDF 24

2.3 Розрахунок відстані за допомогою L2 норм 26

Висновки за розділом 2 27

3 Програмна реалізація 28

3.1 Вибір бази даних 28

3.2 Вибір технологій реалізації 30

3.2.1 Back-end 30

3.2.2 Front-end 31

3.3 Наповнення БД 32
3.4 Користувацький інтерфейс UI 33
Висновки за розділом 3 36
4 Результати обчислювального експерименту та їх аналіз 38
4.1 Моделювання роботи рекомендаційної системи 38
4.2 Формування рекомендаційних правил 39
4.3 Пошук важливих слів в описах сутностей 40
Висновки за розділом 4 44
Висновки 45
Перелік джерел посилання 46
Додаток А. Лістинг програми 47

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ

БД – база даних;

JS – java script;

SQL – structured query language;

HTML – hypertext markup language;

CSS – cascading style sheets;

API – application programming interface;

MTMR – many-to-many relationships.

ВСТУП

Актуальність теми. У 2023 році дохід відеоігрової індустрії сягнув 390 млрд доларів. Щорічно цей показник зростає в середньому на 30 млрд доларів, що робить цей ринок розваг одним із найбільш розвинених. З кожним роком цей вид розваг усе більше охоплює наше повсякденне життя, перетворюючись із нішового хобі для підлітків на кіберспортивні змагання, трансляції яких дивляться по всьому світу. Виграшні дивіденди на цих змаганнях стрімко зростають щорічно.

Наприклад, останній турнір з популярної гри Dota 2 “The International” приніс команді-переможцю 2 245 000 доларів за перше місце в турнірній сітці.

Нестихаючий інтерес до відеоігор змушує корпорації вкладатися у бізнес із розробки ігор, що провокує велику кількість пропозицій у цьому сегменті. Наразі існує велика кількість популярних онлайн-крамниць, де люди можуть придбати цифрові копії відеоігор, які їх цікавлять.

Проте, хоч ці крамниці й мають інструменти, за допомогою яких користувачі можуть впливати на рейтинги відеоігор, нерідко крамниці йдуть на поступки великим корпораціям для зменшення кількості негативних відгуків чи просування власного продукту в чартах, що збільшує рівень продажів.

З огляду на це, незалежні рекомендаційні платформи, які виступають у ролі зручного інструменту для пошуку та незалежної оцінки ігрових продуктів, здобуватимуть усе більшу популярність у мережі Інтернет. Це гарантує актуальність та практичну необхідність подібних програмних продуктів.

Мета і завдання кваліфікаційної роботи. Метою кваліфікаційної роботи є проєктування рекомендаційної системи з урахуванням кількох незалежних параметрів відеоігор, що робить їх максимально бажаними для кінцевого споживача. Для досягнення поставленої мети необхідно:

– опрацювати джерела;

– визначити ступінь подібності між двома сутностями (відеогра);

– визначити, наскільки вагомим є глобальний рейтинг сутності щодо середньостатистичного користувача (гравця);

– визначити, наскільки точною може бути у своїх прогнозах рекомендаційна система, яка базується на контенті, а не на персональних вподобаннях користувача;

Для будь-якої рекомендаційної системи важливо мати велику кількість даних, які вона могла б опрацьовувати та використовувати для надання рекомендацій. Звісно, ці дані повинні володіти повнотою та актуальністю, оскільки застарілі дані підходять хіба що для тестування, але аж ніяк не для актуальної системи, що надає рекомендації. Важливою характеристикою даних також є їх відкритість та об'єктивність.

Саме тому використання API (application programming interface) популярних сайтів чи онлайн-крамниць на цю роль є не найкращим вибором, адже вони часто пов’язані з великими компаніями, які мають свій інтерес. Високі рейтинги стимулюють продажі, а наша рекомендаційна система стає залежною від їхніх API, що викликає сумніви щодо її незалежності.

Саме для таких цілей був розроблений «IGDB» – відкрите, безоплатне API для некомерційного використання, яке постійно актуалізується та містить вичерпну інформацію про кожну відеогру. Воно включає такий контент, як відгуки, оцінки критиків та гравців, опис, обкладинки, посилання на сайт розробника та навіть відео ігрового процесу. Цих даних більш ніж достатньо для побудови рекомендаційної системи, що базується на контенті.

Натомість користувач усе ж повинен вказати мінімальні характеристики відеогри, яку шукає. Припускаю, що вказівки ігрової платформи та жанру гри буде достатньо.

Важливою складовою частиною кваліфікаційної роботи є розробка вебзастосунку для взаємодії з користувачем за допомогою UI-інтерфейсу та генерації рекомендацій згідно із заданими користувачем характеристиками очікуваної гри.

Об’єкт дослідження: система рекомендації відео ігор.

Предмет дослідження: математичні моделі та методи оптимізації рекомендаційної системи відео ігор.

Методи дослідження: методи системного аналізу, математичного та комп’ютерного моделювання.

Публікації. Результати, отримані у роботі “Designing a video game recommendation system”, були представлені на 13-ї міжнародній науково-технічній конференції «Інформаційні системи та технології ІСТ-2024» (м. Харків 26-28 листопада 2024р.) [1].

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Системний аналіз проблеми проєктування системи рекомендаційної системи відео ігор.

1.1.1 Вербальна модель системи

Вербальна модель системи – це опис принципів роботи системи словами, без використання складних технічних термінів, формул чи кодів. Вона пояснює, як працює система, які дані використовуються, які етапи обробки виконуються та як отримується результат, у простій і зрозумілій формі.

Вербальна модель потрібна, щоб зробити складну технічну концепцію зрозумілою для людей, які не мають глибоких знань у цій сфері, наприклад, для бізнес-замовників, користувачів або інших зацікавлених сторін.

Рисунок 1.1 – Модель типу «чорний ящик»

Модель типу «чорний ящик» (рис. 1.1) як система взаємодіє з зовнішнім середовищем. Розглядається поведінка та призначення системи.

1.1.2 Морфологічний опис системи

Зробимо морфологічний опис системи на основі зовнішнього середовища, що приведено на рис. 1.2.

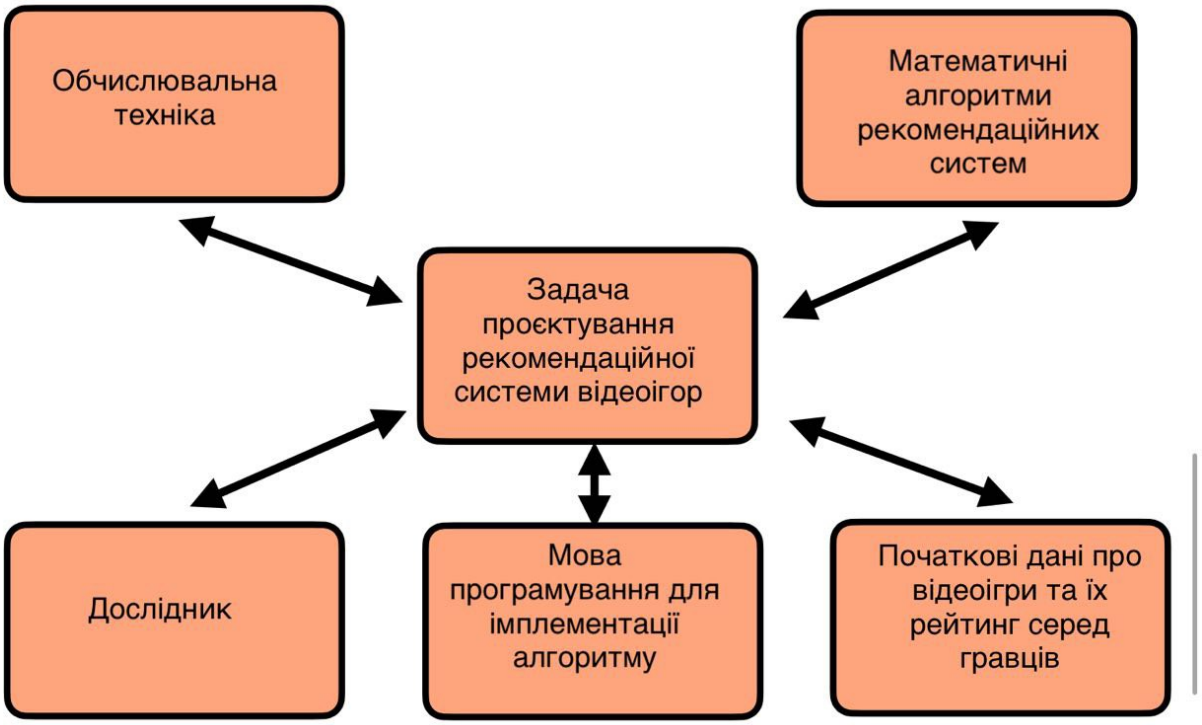

Рисунок 1.2 – Модель зовнішнього середовища системи

Зовнішнє середовище – це сукупність усіх об'єктів системи, зміни властивостей, чиї властивості змінюються в результаті поведінки системи, а також тих об'єктів, яких впливають на систему.

1.1.3 Функціональна модель системи

Розробимо IDEF0 для означення функцій та вимог системи, яка відповідає вимогам та реалізує усі функції.

IDEF0 може застосовуватися для аналізу функцій системи, для відображення механізмів виконання цих функцій.

IDEF0 процес представляє собою набір взаємодіючих між собою елементів та ресурсів, що споживаються кожною роботою.

Діаграма один (контекстна діаграма) представляє собою загальний вид системи та взаємодію її із зовнішнім середовищем.

На рис. 1.3 приведена контекстна діаграма, яка показує функціонування системи в цілому, а після цього на рис. 1.4 проводиться її декомпозиція на фрагменти.

На рис. 1.5 проводиться декомпозиція кожного великого фрагменту системи на дрібні. І далі ще дрібніші для більшої деталізації.

Для того, щоб дати можливість системним аналітикам описати послідовність виконання процесів та всі об’єкти процесу, використовують метод IDEF3, який допомагає розібратися спеціалістам.

Це можна побачити на рис. 1.6 – 1.7.

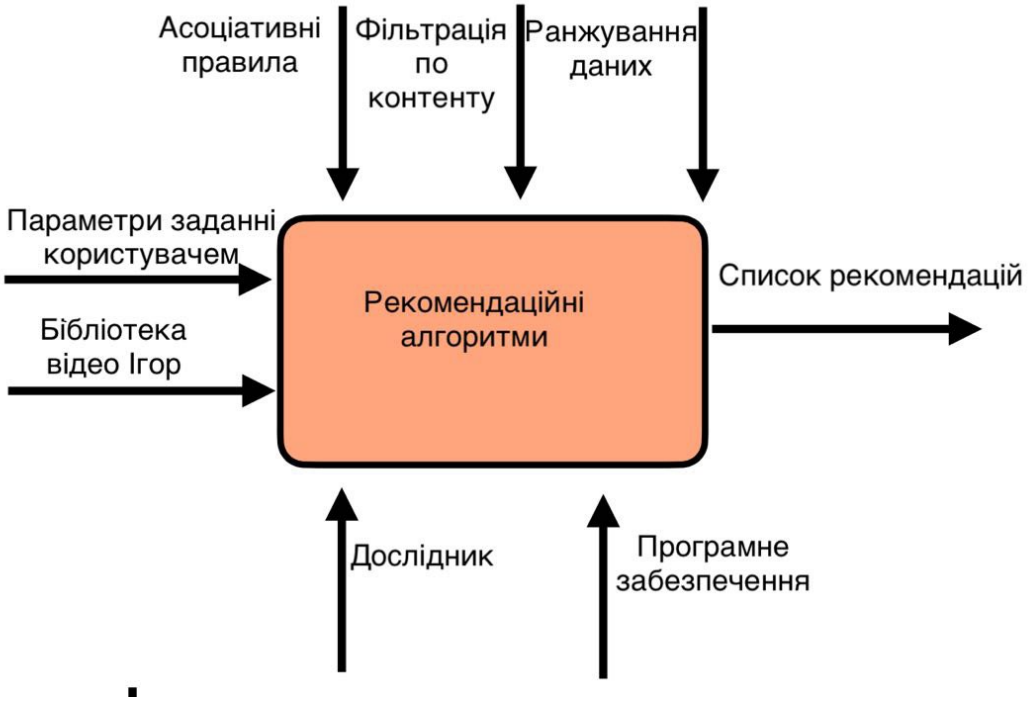

Рисунок 1.3 – Контекстна діаграма (рівень А-0)

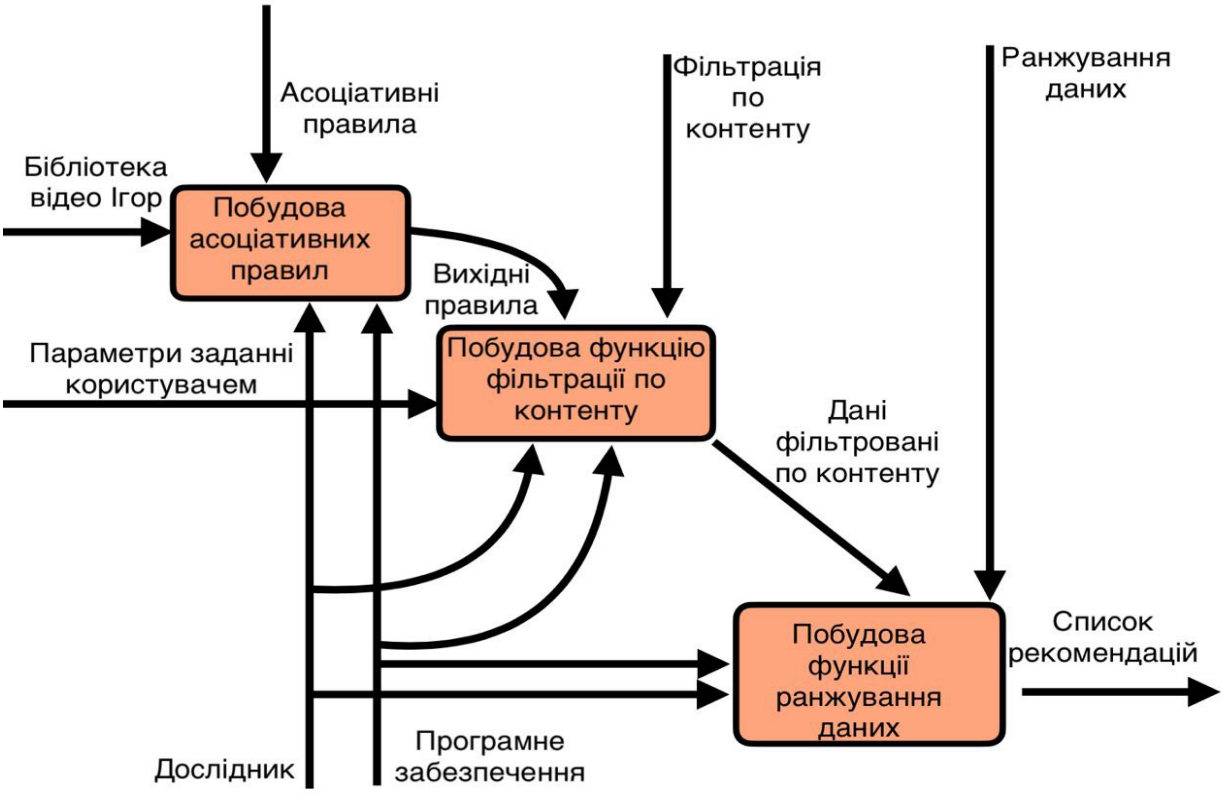

Рисунок 1.4 – Декомпозиція роботи «Рекомендаційний алгоритм»: (рівень А0)

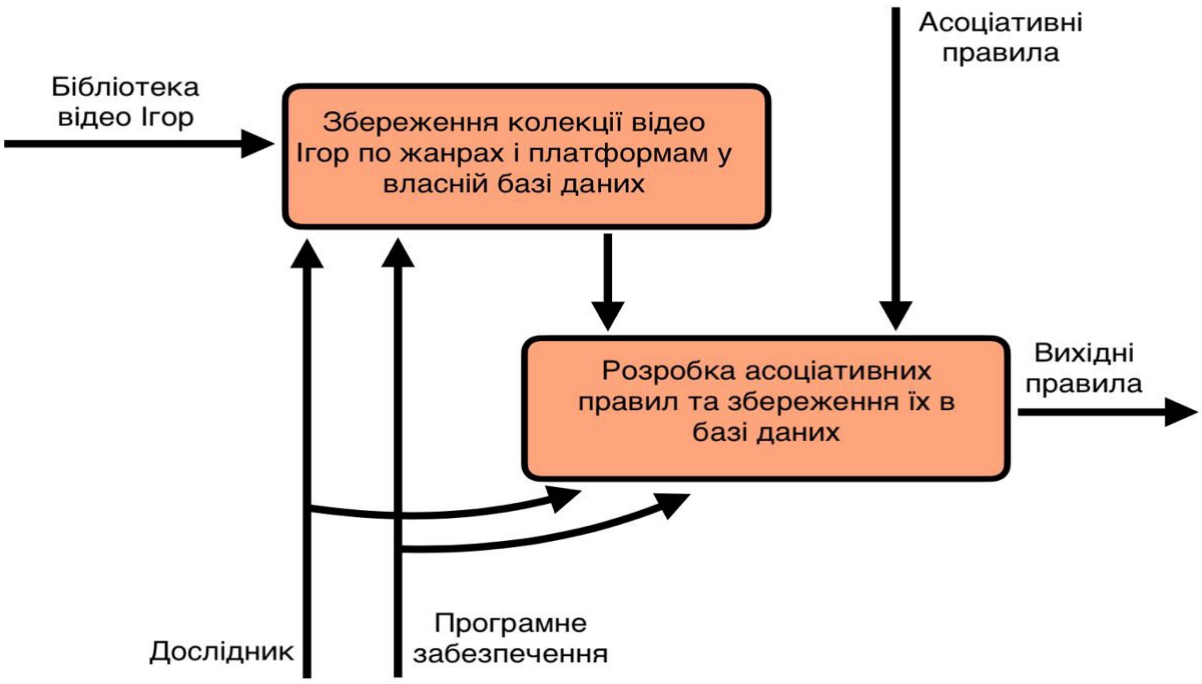

Рисунок 1.5 – Декомпозиція роботи «Вихідні правила»: (рівень А1)

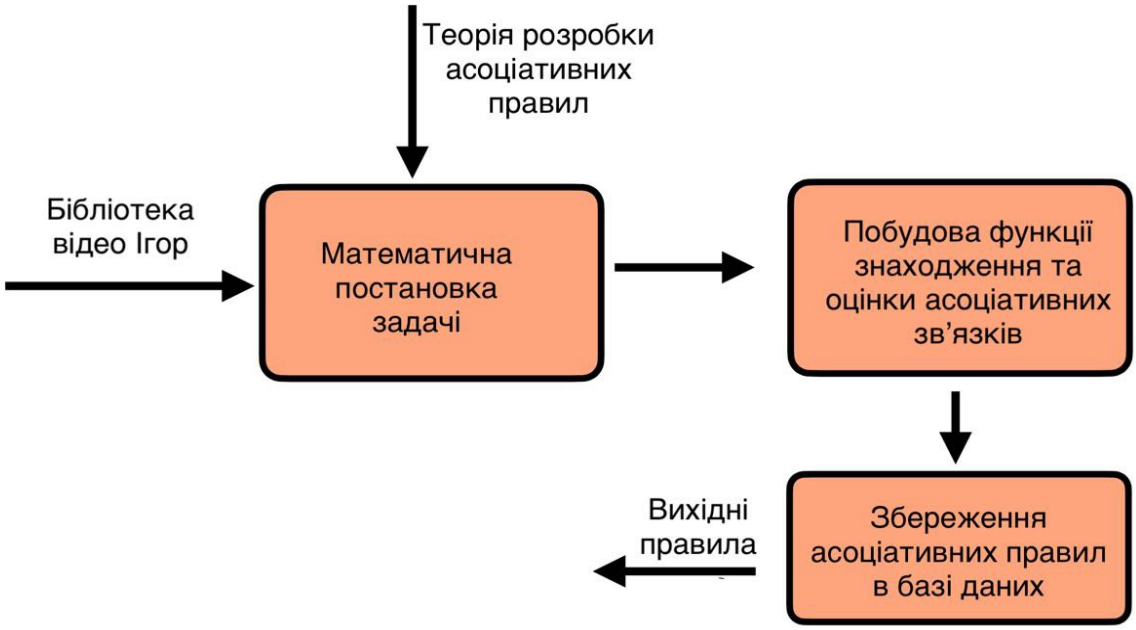

Рисунок 1.6 – Опис роботи «Побудова асоціативних правил»: рівень A0 (в нотації IDEF3)

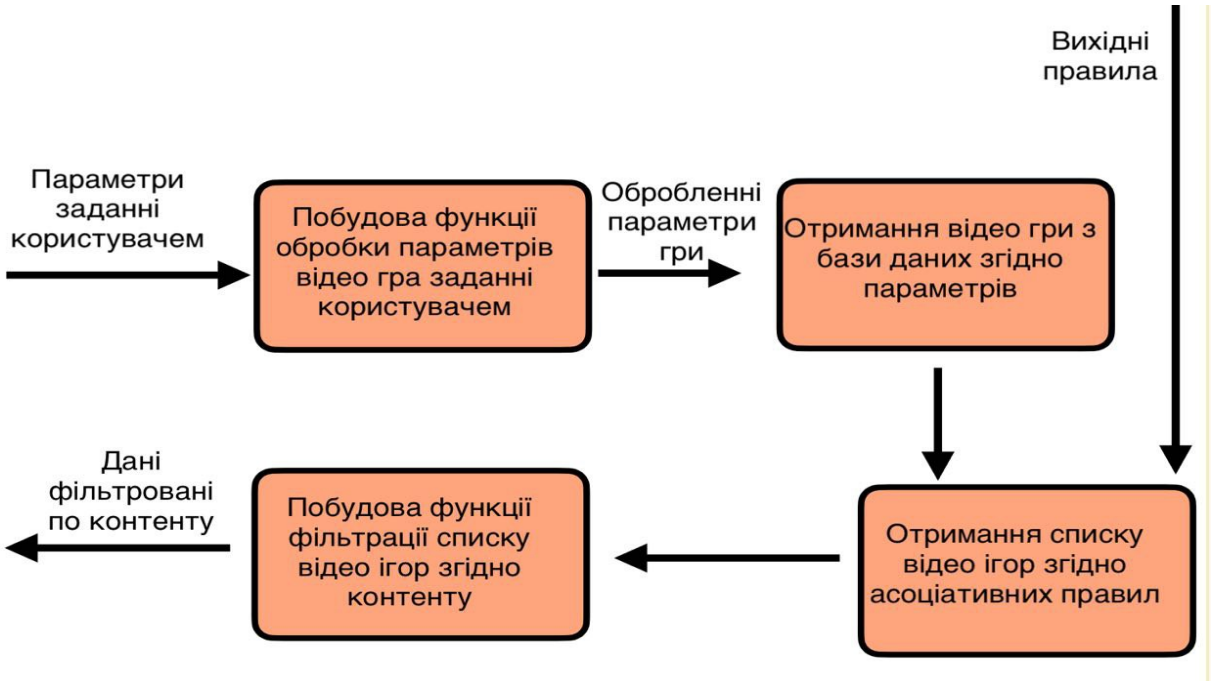

Рисунок 1.7 – Опис роботи «Дані Фільтровані по контенту»: рівень A0 (в нотації IDEF3)

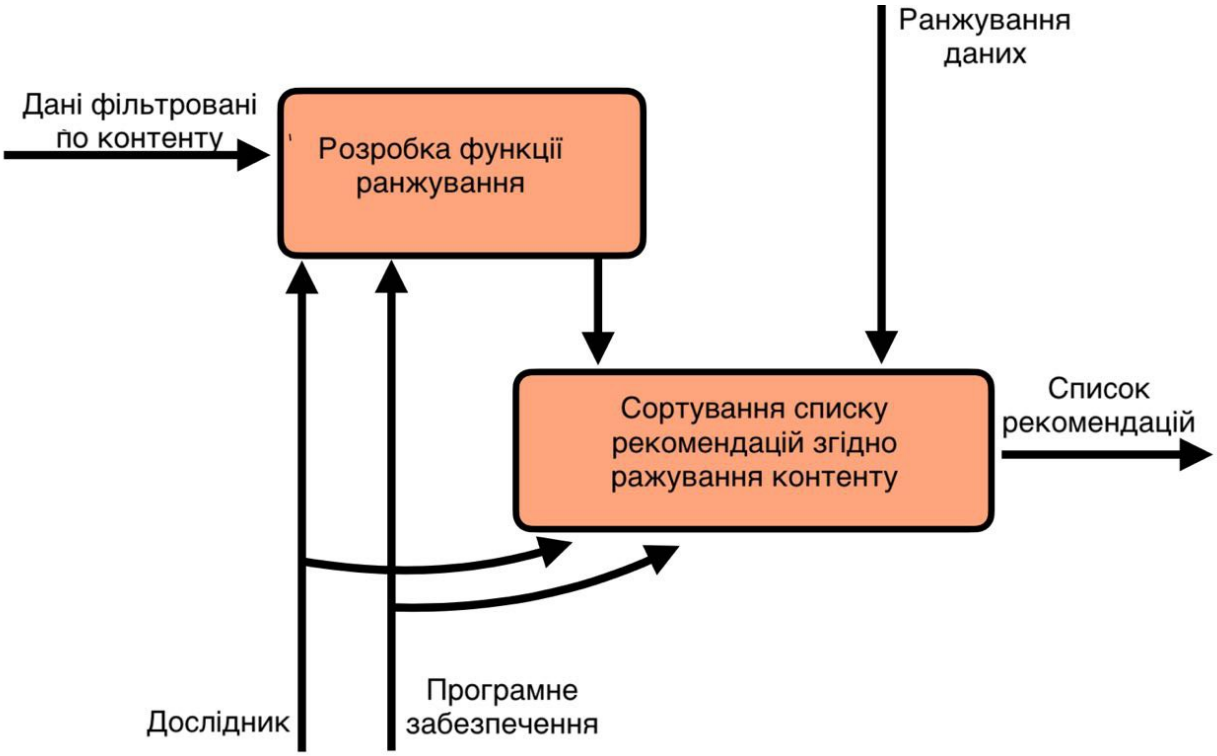

Рисунок 1.8 Опис роботи «Побудова функції ранжування даних»: (рівень A1) (в нотації IDEF3)

1.1.4 Інформаційна модель

Призначенням інформаційної моделі системи є відображення зв’язків між її елементами у виді структур даних, звертаючи увагу дослідника на взаємозв’язки та склад потоків даних.

Для обробки інформацій та опису документообігу використовуються діаграми потоків даних (Data Flow Diagramming, DFD), які також є модельними системами та зображують систему, що досліджується у виді пов’язаних між собою робіт.

Вони використовуються додатково до моделі IDEF0 для гарного відображення поточних операцій. На рис. 1.9 наведена DFD-діаграма.

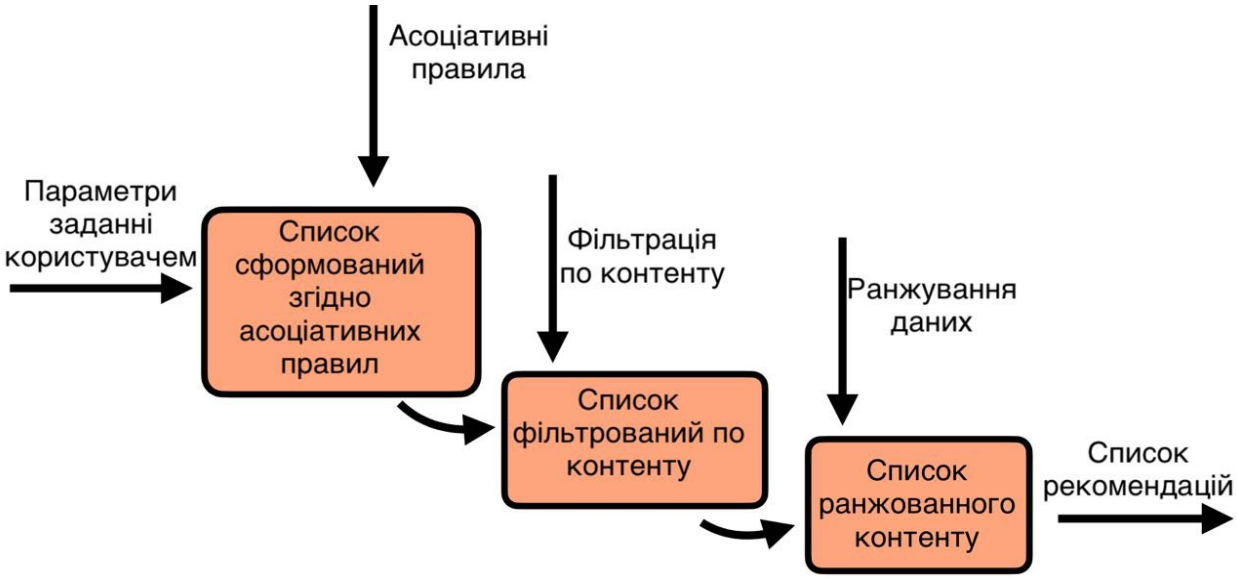

Рисунок 1.9 – DFD-Діаграма

1.2 Аналіз сценарію вирішення задачі проектування рекомендаційної системи, що базується на контенті, а не на персональних даних

1.2.1 Аналіз основних факторів, що впливають на рекомендації відео ігор

Основним фактором, що впливає на вибір відеоігор, які в результаті будуть запропоновані користувачу, є основні характеристики відеоігор, які користувач обере для пошуку. Їх не може бути надто багато, адже це зробить рекомендаційну систему надто нав’язливою та відштовхуватиме потенційних користувачів. Тому пропоную обрати два фактори пошуку, які будуть запропоновані користувачу: «Платформа» та «Жанр».

У ролі платформи виступає пристрій, якому користувач надає перевагу для комфортної гри. Це може бути консоль Xbox від компанії Microsoft чи PlayStation від компанії Sony, або навіть персональний комп’ютер. Що стосується жанру, то, подібно до кінофільмів, кожна гра має свій жанр, що характеризує ігровий процес. Наприклад, «Shooter», де основний акцент зроблено на симуляцію стрілянини, або «Rhythm», де гравцям пропонується скористатися

своїм відчуттям музичного ритму, щоб виконати завдання, поставлені грою.

Беручи за основу цю інформацію, вказану користувачем через вебінтерфейс, ми можемо суттєво зменшити обсяг контенту, який міг би зацікавити користувача.

Іншими факторами, які повинні враховуватися під час формування рекомендаційного списку, є, оцінки відеоігор від користувачів та відеоігрових критиків. Однак базувати рекомендації лише на оцінці відеогри іншими користувачами – це недосконалий підхід, оскільки смаки кожного користувача відрізняються. Утім, це хороший старт для початку формування вибірки навколо однієї з найвище оцінених ігор та пошуку факторів, які роблять інші ігри схожими на неї. Уже на основі знайдених закономірностей формується рекомендаційний список відеоігор для кінцевого користувача.

До непрямих факторів, що впливають на рекомендації, можна віднести поведінку користувача в рамках нашої системи. Наприклад, в онлайн-магазині ми могли б зберігати інформацію про товари, які користувач переглядає чи купує, і пропонувати йому «подібні» товари. Таким чином, ми не набридатимемо користувачеві постійними опитуваннями, але водночас намагатимемося пропонувати товари, що будуть йому «цікавими». У випадку рекомендаційної системи відеоігор, де користувач нічого не купує, ми можемо отримувати інформацію про вподобання користувача непрямим способом – аналізуючи, які ігри він переглядає, а які ігнорує. Останні, звісно, повинні з’являтися в списку запропонованих якомога рідше або взагалі виключатися з нього.

Будь-яка індустрія, що приносить великий прибуток і стрімко розвивається, шукає все більш оптимальні та продуктивні способи масштабування та розвитку. Відеоігрова індустрія не є винятком, і ми бачимо це в збільшенні кількості нових ігрових студій та самих ігор у геометричній прогресії щороку. Це призводить до необхідності регулярного оновлення відеоігрової бази даних, яка швидко поповнюється новими ігровими проєктами.

Звісно, щоб лишатися цікавою та актуальною, рекомендаційна система зобов’язана регулярно оновлювати свою базу даних. Тут стоїть питання не ли-

ше в нових проєктах, адже до чинних регулярно виходять оновлення (DLC), які інколи докорінно змінюють ігровий процес, що впливає на рейтинг самої гри в спільноті гравців. Також бувають випадки, коли гра отримує прохолодні відгуки на старті продажів, часто через невдало обрану дату релізу. Але з часом гравці можуть змінити свою думку про гру, інколи навіть радикально, що впливає на її рейтинг.

У такій індустрії, що стрімко розвивається, залишатися актуальною є критично необхідним для рекомендаційної системи. Найкращим графіком оновлень можна вважати «щоденний». Утім, процес наповнення бази даних сутностями або їх актуалізації, а також побудова асоціативних правил на основі контенту займають багато часу та ресурсів, які маленька незалежна платформа для рекомендацій далеко не завжди може собі дозволити. Тому оптимальним є «щотижневий» графік оновлень даних у базі сутностей та їх асоціативних правил.

Для ефективного проєктування рекомендаційної системи відеоігор необхідно покроково змоделювати роботу системи та її взаємодію з користувачем для наочної демонстрації принципу її роботи та алгоритмів розрахунку рекомендацій.

1.3 Формальна та змістовна постановка задачі

1.3.1 Формальна постановка задачі

Важливим елементом кожної рекомендаційної системи є широкий асортимент сутностей та їх актуальність. Важко уявити собі ефективну рекомендаційну систему, яка формує рекомендаційний список для користувача лише з десятка сутностей. Звісно, це залежить від специфіки та області застосування рекомендаційної системи, але у такій сфері, як відеоігри, із її динамічним зростанням, необхідно розглядати сотні або навіть тисячі сутностей.

З іншого боку, зберігання, обробка та обчислення асоціативних правил для великої кількості сутностей вимагає значних обчислювальних ресурсів і часу. Оскільки метою цієї роботи є проєктування, а не створення фінального продукту, необхідно визначити оптимальну кількість сутностей. Вона повинна бути достатньою для формування рекомендаційного списку, але не надто великою, щоб не ускладнювати процес збору даних, обробки та тестування алгоритмів.

Оптимальною кількістю пропоную прийняти 100 сутностей для кожної пари (платформа, жанр). Цього буде більш ніж достатньо для створення рекомендацій із достатньою точністю, без значних витрат часу на збір та агрегацію даних. Як зазначалося раніше, для збору інформації про сутності було обрано відкриту безоплатну базу даних відеоігор – IGDB.

Актуальність даних є надзвичайно важливою, тому необхідно забезпечити регулярне оновлення бази сутностей. Пропонується здійснювати оновлення щотижня. Для цього слід розробити спеціальний скрипт, який виконуватиметься за розкладом, встановленим менеджером завдань операційної системи. Це дозволить автоматизувати процес оновлення даних у базі сутностей.

Оскільки розроблена рекомендаційна система орієнтується на алгоритми, що працюють із контентом, а не з особистими вподобаннями конкретного користувача, це ускладнює надання персоналізованих рекомендацій. Однак, певне уявлення про смаки користувача можна отримати, відслідковуючи, які відеоігри вже були рекомендовані йому під час взаємодії з вебінтерфейсом, а також найбільш подібні до них. Це дозволить звузити коло сутностей для майбутніх рекомендацій і підвищити їхню якість залежно від часу, який користувач проводить у застосунку.

При цьому система не потребує від користувача проходження опитувань для отримання ефективних рекомендацій. Окрім цього, користувач зможе оцінювати запропоновані відеоігри за допомогою функції «лайків». Це допоможе зробити майбутні рекомендації більш персоналізованими та точними.

1.3.2 Змістовна постановка задачі

Метою проєктування цієї рекомендаційної системи є допомога користувачам у пошуку відеогри, яка могла б їх зацікавити, виключаючи вплив факторів на кшталт «накручених» рейтингів та «відбілювання» негативних відгуків.

Система має надавати релевантні рекомендації швидко та ефективно, не збираючи жодної зайвої приватної інформації про користувачів. У сучасному світі, де майже кожен застосунок прагне зібрати якомога більше персональних даних, ця характеристика є надзвичайно важливою.

1.4 Постановка задач дослідження

Постановка задачі проєктування рекомендаційної системи відеоігор полягає в чіткій формулюванні завдань розробки цієї системи, правильному виборі алгоритмів обробки даних та асоціативних правил для побудови релевантних рекомендаційних списків.

Необхідно регулярно оновлювати список сутностей для забезпечення актуальності інформації. Не менш важливим фактором є створення зручного інтерфейсу, який дозволить користувачу ефективно взаємодіяти з рекомендаційною системою.

Задачі, які будуть вирішені в процесі роботи над кваліфікаційною роботою:

– аналіз літератури за даною предметною областю;

– проєктування рекомендаційної системи;

– розробка схеми для бази даних;

– створення програмного застосунку (рекомендаційної системи);

– створення веб-інтерфейсу для застосунку.

– тестування застосунку.

2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ’ЯЗАННЯ

2.1 Асоціативні правила

У випадку, коли система абсолютно нічого не знає про вподобання відвідувача сайту чи користувача застосунку, найпростішим кроком буде продемонструвати, скажімо, "топ 10 товарів". Це базується на припущенні, що найбільш привабливий товар для більшості користувачів буде таким самим і для нового користувача.

Цей підхід добре працює для холодних користувачів (нових користувачів, про яких мало інформації). Проте ми можемо покращити цей підхід, аналізуючи приховані зв'язки між сутностями, створені іншими користувачами. Наприклад, можна проаналізувати товарні чеки з супермаркету за певний період часу:

а) хліб, йогурт;

б) морква, хліб, морква;

в) хліб, морква;

г) хліб, молоко;

ґ) молоко, фініки, морква;

д) молоко, фініки, йогурт, хліб.

Кожний рядок – це список покупок одного окремого покупця. Використовуючи цю інформацію, ми маємо змогу побудувати асоціативні правила, які будуть для нас орієнтиром щодо того, "які товари найчастіше купують в один продуктовий кошик". Основними величинами асоціативних правил є впевненість та підтримка [1].

Скориставшись формулою 2.1 ми можемо розрахувати наскільки сильний зв’язок в товарному чеку між хлібом та молоком, що становить 0.6:

$$c(X \to Y) = \frac{\left|T\left(X \ AND \ Y\right)\right|}{T\left(X\right)}, \tag{2.1}$$

де c – впевненість;

$T()$ – кількість усіх ігор, що беруть участь у побудові асоціативних правил.

Що говорить нам: "З ймовірністю 60 відсотків ви знайдете в кошику покупця хліб, якщо там є молоко". Проте цей показник недосконалий, і якщо ми проведемо цей розрахунок з парою (фініки, морква), то отримаємо "гарний" результат – 0.5. Проаналізувавши товарні чеки, знову ми зрозуміємо, що ця оцінка є результатом низької кількості товару "фінік" у споживчих кошиках. Тому для перевірки правдоподібності даної оцінки ще використовують підтримку [1]:

$$s(X \to Y) = \frac{\left|T\left(X \ AND \ Y\right)\right|}{T\left(\ \right)}, \tag{2.2}$$

де s – підтримка.

Розрахувавши підтримку для пари (молоко, хліб), отримуємо 0.5, що є все ще високою оцінкою. У випадку пари (фініки, морква) отримуємо більш реалістичний результат – 0.16, саме тому підтримка повинна мати більш високий пріоритет ніж впевненість.

Звісно, розрахунок асоціативних правил для великої кількості сутностей це дуже трудомістка операція і потребує великої кількості часу, тож зазвичай асоціативні правила розраховуються заздалегідь і зберігаються в базі даних в окремій таблиці чи документі.

Правильно побудовані асоціативні правила допомагають будувати зв’язки між сутностями та давати більш релевантні результати під час сортування за рейтингом. Саме тому асоціативні правила є чудовим інструментом для рекомендацій, побудованих на контенті, а не на персональних даних користувача [8].

2.2 Пошук важливих слів TF-IDF

Для покращення якості рекомендацій за контентом в рекомендаційних

системах широко застосовують алгоритми подібності сутностей на основі схожості їхніх описів. Для цієї мети добре підходить модель TF-IDF, за допомогою якої можна оцінити важливість кожного окремого слова в групі текстів, пов'язаних між собою спільною тематикою. Скориставшись цією моделлю, ми могли б зрозуміти, які слова мають відношення до теми автомобілів, а які не мають жодного відношення, проаналізувавши список описів автомобілів.

Аналіз "важливості" кожного окремого слова в групі текстів робиться в декілька етапів. Першим з яких буде розрахунок частоти появи терміну.

$$tf\,(слово, документ) = \frac{кількість\ появ\ слова\ в\ описі}{кількість\ слів\ в\ описі}. \qquad (2.3)$$

Чим частіше слово з'являється в одному документі, тим вища ймовірність того, що воно важливе (за умови, що ви видалили всі слова, які слід ігнорувати). Але, як уже згадувалося раніше, це має значення лише в тому випадку, якщо слово зустрічається лише в кількох документах. Щоб визначити це, ви можете використати обернену частоту документа (IDF у рівнянні) – кількість усіх документів, поділена на кількість документів, які містять це слово. Разом це дає нам TF-IDF [3].

$$idf\,(термін) = \log\left(\frac{усього\ документів}{кількість\ документів\ з\ терміном}\right). \qquad (2.4)$$

Річ у тім, що коли ми знаходимо слова, які мають велике значення TF-IDF, ці слова можна додати до списку ознак подібності:

$$tf - id\,(слово, документ) = tf\,(слово, документ) \times idf\,(слово, документ). \qquad (2.5)$$

Отримавши вектори лексичної подібності з описів до відеоігор, які ми порівнюємо після використання асоціативних правил, ми ще сильніше звужує-

мо коло подібних відеоігор. Це, своєю чергою, дає нам змогу робити більш точні рекомендації, відштовхуючись не тільки від рейтингів, асоціативних правил, але й від опису сутностей. Проте отримати вектори лексичної подібності ще недостатньо – далі нам необхідно знайти спосіб для коректного порівняння останніх.

2.3 Розрахунок відстані за допомогою L_2 норм

L_2 норма або евклідова відстань, є одним із найпоширеніших способів вимірювання відстані між точками у багатовимірному просторі. Вона базується на класичній геометрії та визначає фізичну відстань між двома точками в просторі. Цей метод є фундаментальним у математиці, статистиці, машинному навчанні, комп'ютерному зорі та інших галузях [1].

Розрахувавши вектори лексичної подібності описів до відео ігор ми можемо скористатися L_2 нормою для отримання відстані між описами до відео ігор (велична що вказує в числовому форматі наскільки два тексти є подібними):

$$відстань(век_1, век_2) = r_{век_1} - r_{век_2} = \sqrt{\sum_{i=q}^{n} \left(r_{(век_1, i)} - r_{(век_2, i)} \right)^2} . \qquad (2.6)$$

Отримавши “відстань”, необхідно пам'ятати, що в такому випадку “позитивною” оцінкою треба вважати близьку до нуля, оскільки нас цікавлять саме “найближчі” елементи, подібно до елементів на карті. Тому цей параметр потребує нормалізації для коректного порівняння з іншими величинами подібності, що розглядаються в цій роботі (“підтримка”, “впевненість”). Таким чином, кожна отримана оцінка відстані буде приведена до нормалізованої форми [2].

$$L_{2нор} = 1 - L_2 . \quad (2.7)$$

У випадку, якщо відстань між лексичними векторами двох відеоігрових описів дорівнює 0.28 (що вказує на високу подібність між описами), функція нормалізації поверне число 0.72, що є еквівалентом високої оцінки.

Якщо ж відстань виявиться більшою за 1, то доведеться прийняти її на рівні 0.1, що відповідатиме вкрай низькій оцінці [6].

Висновки за розділом 2

У цьому розділі обговорено три підходи, що є основою для створення рекомендаційних систем:

Асоціативні правила визначають взаємозв'язки між об'єктами на основі історичних даних. Використання показників впевненості та підтримки дозволяє оцінювати силу зв'язків, що особливо ефективно для користувачів, про яких мало інформації.

TF-IDF допомагає аналізувати текстові описи об'єктів, виділяючи найважливіші терміни, які визначають тематику [9]. Це дає змогу побудувати лексичні вектори, що використовуються для пошуку подібних об'єктів.

L_2-норма дозволяє обчислювати відстань між текстовими векторами, забезпечуючи кількісну оцінку схожості між об'єктами. Нормалізація результатів робить можливим їх інтеграцію з іншими метриками.

Ці методи доповнюють одне одного, забезпечуючи багатогранний підхід до створення персоналізованих рекомендацій, які враховують як поведінкові, так і контекстуальні особливості.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Вибір бази даних

База даних є важливою складовою будь-якого програмного продукту, що потребує зберігання даних. Неправильний вибір бази даних може призвести до втрати продуктивності та суперечливих рішень в архітектурі всього застосунку. Тому до вибору баз даних слід підходити з особливою обережністю, враховуючи специфіку додатку та види зв’язків між сутностями, над якими додаток буде здійснювати маніпуляції.

Говорячи про рекомендаційну систему відеоігор, ми можемо охарактеризувати зв’язки між сутностями як MTMR (“Many-to-Many Relationships”)[11]. Це випадок, коли кожна сутність може мати декілька незалежних зв’язків з іншими сутностями. Наглядним прикладом таких зв’язків є відношення між фільмами та акторами, які грають у них. Актор може зніматися у декількох фільмах, так само як і фільм може мати багато акторів у своєму складі. Такого роду зв’язки також притаманні відеоіграм у відношенні до жанрів та ігрових платформ.

Бази даних сімейства SQL добре зарекомендували себе для роботи та зберігання сутностей із такими зв’язками. Саме тому для цієї рекомендаційної системи була обрана база даних SQLite.

SQLite – це компактна реляційна база даних із відкритим кодом, яка відома своєю легкістю інтеграції та тим, що не потребує серверної інфраструктури. Вона чудово підходить для застосунків, яким потрібне локальне зберігання даних без складних налаштувань. Ця база даних поєднує простоту розгортання та повну підтримку базових команд SQL [6]. Часто її обирають для стартапів чи тестування прототипів програмних застосунків. Незважаючи на це, SQLite широко використовується і у великих корпоративних застосунках завдяки своїй простоті, стабільності роботи та ефективності.

Ключовою сутністю для рекомендаційної системи відеоігор беззаперечно

є ігри. Інші сутності, такі як жанр, платформа та асоціативні правила, є службовими сутностями, які допомагають охарактеризувати ігри та структурувати їх для рекомендаційної системи.

Для рекомендаційної системи була розроблена відповідна схема бази даних для зберігання та структурування сутностей.

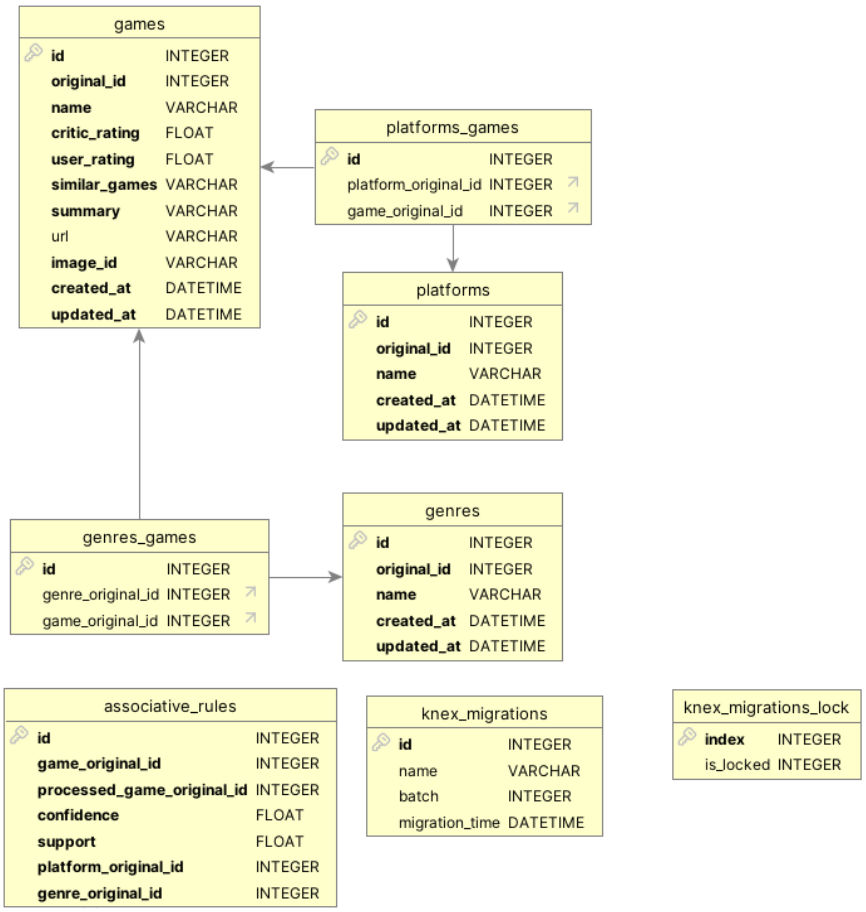

Рисунок 3.1 – Схема ДБ

Як видно з рисунка 3.1, база даних містить основні таблиці, такі як platforms, genres, а також службові таблиці genres_games і platforms_games. Останні необхідні для реалізації зв’язків типу MTMR. Таблиця associative_rules слугуватиме сховищем асоціативних правил, які система використовуватиме для надання рекомендацій користувачам. Таблиці з префіксом knex потрібні для коректної роботи бібліотеки Knex, яка забезпечує зручний інтерфейс для взаємодії з базою даних [12].

3.2 Вибір технологій реалізації

Для ефективної роботи будь-якого вебзастосунку прийнято розділяти додаток на дві частини, які розподіляють між собою функціонал та відповідальність. Back-end, як правило, реалізує бізнес-логіку, яка маніпулює сутностями та надає програмний інтерфейс для безпечної взаємодії з ними користувачем. У свою чергу, Front-end відповідає за інтерактивний користувацький інтерфейс, який відображає сутності для користувача та елементи взаємодії з ними (кнопки, чекбокси, поля). Таким чином, логіка формування контенту та його відображення поєднуються лише спільними контрактами взаємодії.

3.2.1 Back-end

Для даної частини додатку була використана така технологія, як Node.js. Node.js – це середовище виконання JavaScript, яке дозволяє запускати цей код на сервері [10]. Раніше JavaScript використовувався лише на стороні клієнта (у браузері), але з появою Node.js він став популярним інструментом для створення серверних додатків завдяки можливості обробляти велику кількість запитів при споживанні відносно малих ресурсів системи. Водночас Node.js легко масштабувати для збільшення продуктивності, додаючи обсяг доступної пам’яті для додатку чи налаштовуючи роботу Node.js у багатоядерному режимі. Останній надає можливість завантажувати кожне ядро процесора паралельно, оптимально розподіляючи навантаження на кожне.

Важливою частиною Node.js є npm (Node Package Manager). Що являє собою сховищем що надає доступ до пакетів (програмних бібліотек). Які в свою чергу значно спрощують розробку програмних продуктів надаючи готові рішення (бібліотеки), до прикладу бібліотека knex що надає зручний інтерфейс для роботи з різними ДБ, чи fastify – швидкий мінімалістий фраемворк що, надає функціонал для розгортання ефективного http/https серверу [5]. Використавши який ми створимо

канал спілкування між Back-end та Frone-end частинами рекомендаційної системи.

Основними сферами використання Node.js є чати в реальному часі, API (інтерфейси для обміну даними між різними системами), інтернет-магазини, системи потокового відео чи музики.

Універсальність та продуктивність роблять цю технологію передбачуваним вибором для таких продуктових компаній, як LinkedIn (соцмережа для пошуку роботи), PayPal (платіжна система), Netflix (стримінговий сервіс), eBay (торговельний майданчик). Ці приклади показують, що Node.js добре працює як у малих стартапах, так і в системах, які повинні витримувати високі серверні навантаження (кількість запитів на секунду від користувачів.

3.2.2 Front-end

Розробка веб-інтерфейсу передбачає використання кількох технологій. Серед найважливіших з них можна виділити JavaScript, HTML та CSS. Ці три елементи працюють разом, щоб створити інтерактивні, естетичні та функціональні вебсайти. Кожен з них має свою специфічну роль, і їх поєднання дозволяє створювати повноцінний користувацький досвід.

HTML (HyperText Markup Language) є основою кожної веб-сторінки. Це мова, що використовується для структуризації та організації контенту на сторінці. HTML визначає основні елементи сторінки, такі як заголовки, абзаци, зображення, посилання та форми. Уявіть собі HTML як каркас вебсайту. Без HTML вебсторінка була б порожнім місцем без структури чи змісту.

Елементи HTML позначаються за допомогою тегів. Наприклад, тег <h1> використовується для визначення основного заголовка, а тег <p> – для абзацу тексту. Ці теги розміщуються в документах HTML, які потім інтерпретуються браузером для відображення сторінки.

CSS (Cascading Style Sheets) використовується для управління візуальним виглядом веб-сторінки. Поки HTML відповідає за структуру контенту, CSS визна-

чає, як цей контент має відображатися. Наприклад, CSS дозволяє змінювати колір текстів, розмір зображень, розташування елементів на сторінці та багато іншого.

CSS працює, вибираючи елементи HTML і застосовуючи до них стилі. Це може включати колір, розмір, позицію та шрифт елементів. Наприклад, за допомогою CSS можна встановити, що всі заголовки рівня 1 (теги <h1>) повинні бути синіми та жирними. Це дозволяє зробити сторінку привабливою та зручною для користування.

JavaScript – це мова програмування, яка дозволяє додавати інтерактивні функції на вебсторінку. На відміну від HTML та CSS, які є статичними, JavaScript дозволяє зробити сторінку динамічною. Наприклад, за допомогою JavaScript можна створювати випадаючі меню, інтерактивні форми або навіть онлайн-ігри. JavaScript дозволяє реагувати на дії користувача. Якщо хтось натисне кнопку, JavaScript може бути використаний для відображення повідомлення, зміни кольору елемента або завантаження інформації без перезавантаження сторінки. Це робить вебсайти набагато більш інтерактивними та цікавими для користувачів. Де-факто JavaScript є стандартною мовою будь-якого вебдодатку, що робить його однією з найпопулярніших мов програмування у світі. Це, у свою чергу, забезпечує стрімкий розвиток його екосистеми та синтаксису. Сі-подібний синтаксис та динамічна типізація цієї мови програмування забезпечують плавний поріг навчання та зрозумілий вихідний код.

HTML надає структуру сторінки, CSS відповідає за дизайн, а JavaScript додає інтерактивність. Використовуючи ці три складові ми розробили інтуітивно зрозумійлий користувацький інтерфейс для запиту , отримання та відображення рекомендацій для кінцевого користувача.

3.3 Наповнення БД

Для будь-якої рекомендаційної системи критично важливою є наявність актуальних даних, оскільки неактуальні дані ставлять під сумнів працездатність

системи.

Для збору даних була обрана IGDB (відкрита безоплатна база даних відеоігор). Однією з основних причин вибору цієї бази даних є її незалежність та активна підтримка спільнотою. Окрім того, ця база даних надає надзвичайно важливу інформацію для рекомендаційної системи, зокрема поле similar_games. Це поле є властивістю кожної гри й містить масив ідентифікаторів відеоігор, що мають схожі характеристики. Використовуючи цю властивість, ми можемо формувати асоціативні правила, які, своєю чергою, спрощують створення рекомендацій для користувача.

Для наповнення бази даних рекомендаційної системи був розроблений скрипт, який робить запити до API IGDB і зберігає результати в базу даних рекомендаційної системи. Структура цієї бази представлена на рисунку 3.1.

Розроблений скрипт для наповнення бази даних спочатку виконує запити для отримання всіх актуальних відеоігрових платформ і жанрів. Після цього, використовуючи цикл, він здійснює запити для кожної пари "платформа-жанр", таким чином рівномірно наповнюючи базу даних всіма актуальними платформами та жанрами відеоігор. Оптимальною кількістю ігор для кожної пари "платформа-жанр" було обрано 100. Отриманий список сортується за рейтингом користувачів, що дозволяє отримати не лише найбільш актуальні, а й найцікавіші проєкти для кожної пари.

Нерідко відеогра має декілька жанрів та кілька платформ. У цьому випадку в базу даних вноситься запис для всіх цих складних відношень, тому була обрана модель зв’язків MTMR. Зважаючи на це, стає зрозуміло, що база даних рекомендаційної системи буде наповнена значно більшою кількістю сутностей, ніж 100 одиниць для кожної пари "платформа-жанр".

3.4 Користувацький інтерфейс UI

Для розробки користувацького інтерфейсу були використані зазначені

вище технології: HTML, CSS, JavaScript.

Розроблена рекомендаційна система не збирає та не зберігає жодних персональних даних користувачів, тому необхідність логінитися чи реєструватися в ній відсутня. Користувач одразу потрапляє на головну сторінку, де може почати роботу з системою відразу після її завантаження.

Головна сторінка рекомендаційної системи, як показано на рис. 3.2, виконана в мінімалістичному дизайні. Вона пропонує користувачеві одразу після завантаження обрати бажану платформу та жанр відеоігри, що дозволяє системі робити більш точні рекомендації, звужуючи коло сутностей (відеоігор).

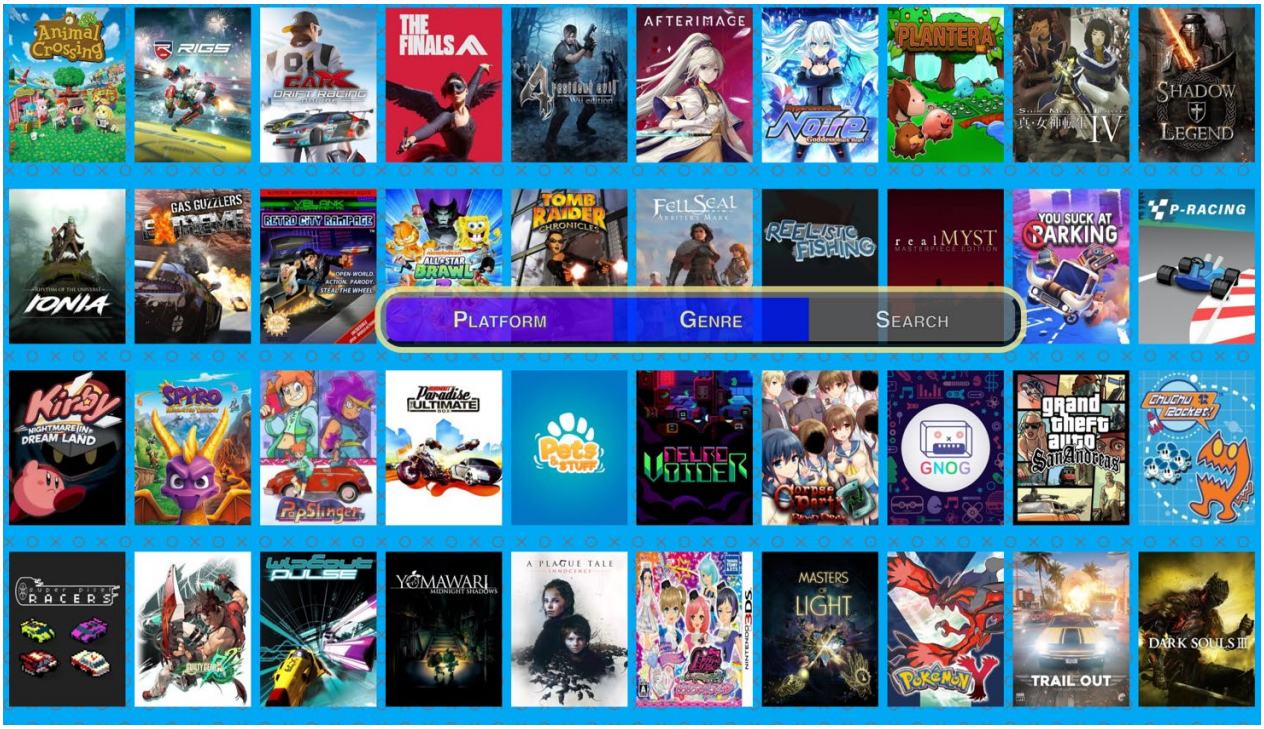

Рисунок 3.2 – Головна сторінка UI

Головна сторінка також демонструє обкладинки відеоігор як шпалери сайту, що підкреслює відеоігрову тематику. Дані шпалери вебінтерфейсу є динамічними та змінюються при кожному перезавантаженні сторінки.

Кнопка пошуку набуває статусу «активна» лише в тому випадку, коли користувач обрав платформу та жанр. Це унеможливлює потенційні помилки під час формування запиту до Back-End.

Список платформ та жанрів UI отримує, зробивши запит до Back-End, який, у свою чергу, отримує відповідні сутності, скориставшись бібліотекою Knex. Ця бібліотека надає зручний інтерфейс для роботи з базами даних. Таким чином, список платформ та жанрів із бази даних рекомендаційної системи відображається користувачеві через UI.

Рис. 3.3 демонструє, що користувач обрав платформу "PC (Microsoft Windows)" та жанр "Puzzle". Натиснувши кнопку "Search", користувач ініціює запит до Back-End, який, у свою чергу, повертає рекомендації відповідно до обраної платформи та жанру. Алгоритм формування рекомендацій буде розглянуто в наступному розділі.

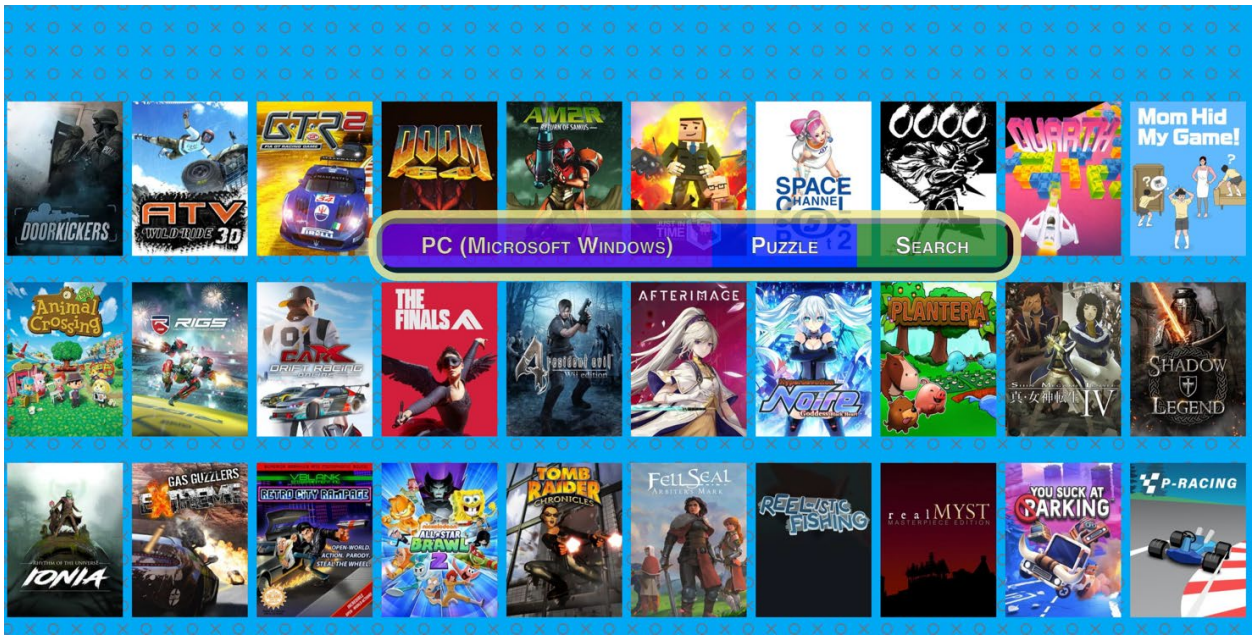

Рисунок 3.3 – Стан UI активний пошук

Отримавши список рекомендацій, UI демонструє їх користувачеві у вигляді карток, що проілюстровано на рис. 3.. Кожна картка містить інформацію про рекомендовану відеогру, а саме: ім’я, обкладинку, опис, посилання на сайт для ознайомлення з детальною інформацією, а також рейтинг від користувачів і критиків, розташовані по обидва боки від посилання.

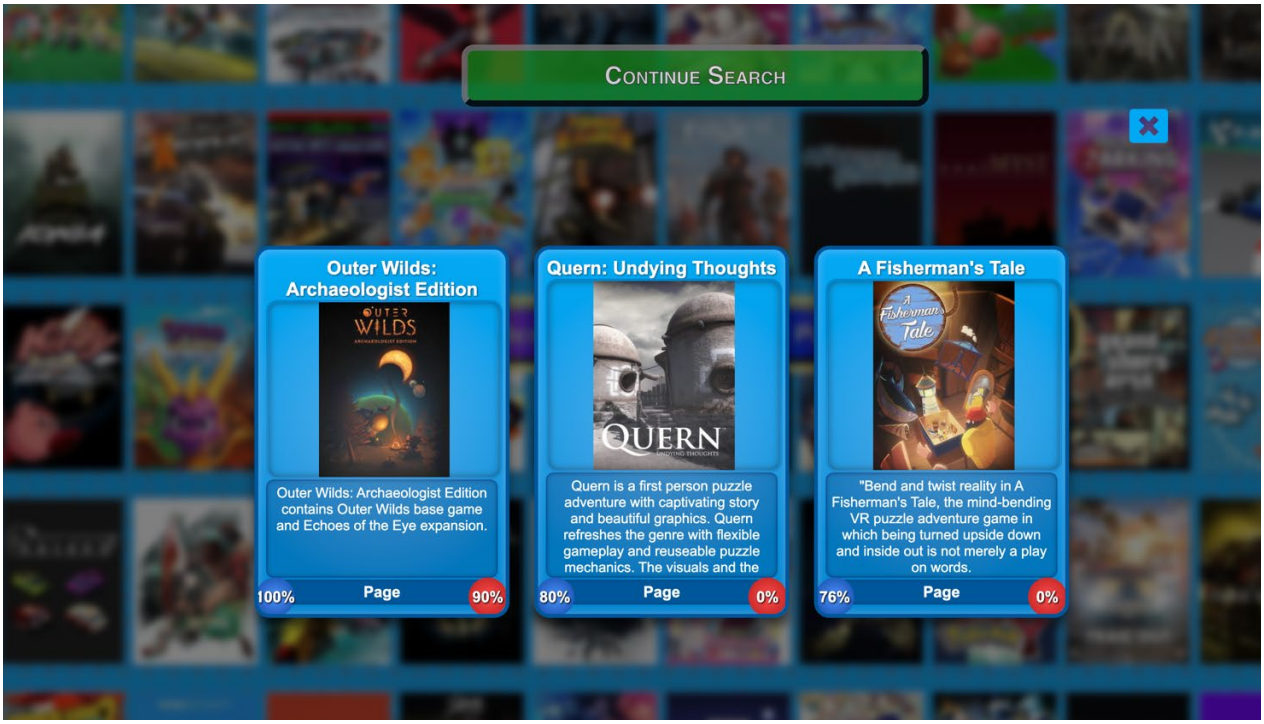

Рисунок 3.4 – Стан UI демонструє список рекомендацій

Користувачеві надається можливість продовжити пошук, натиснувши кнопку "Continue search" що, в свою чергу зробить запит на сервер. Це призведе до повторного пошуку рекомендацій, зберігаючи налаштування платформи та жанру.

Важливим аспектом роботи рекомендаційної системи є те, що вона зберігає інформацію про переглянуті відеоігри та виключає їх зі списку рекомендацій під час повторного пошуку. Це робить підбір рекомендацій більш точним і позбавляє користувача необхідності переглядати одні й ті самі ігри при наступному пошуку. Список переглянутих ігор зберігається в браузері користувача, і останній може в будь-який момент видалити ці дані через налаштування власного браузера. Таким чином, забезпечується приватність що є однією з головним вимог до застосунку.

Висновки за розділом 3

У цьому розділі було детально розглянуто технології, обрані для розроб-

ки рекомендаційної системи відеоігор. Зокрема, було обґрунтовано вибір типу бази даних та моделі зв’язків сутностей, що в ній використовується. Представлено структуру зв’язків сутностей у базі даних.

Також було описано алгоритм наповнення бази даних сутностями. Особливу увагу приділено вирішенню можливих проблем, пов’язаних із нерівномірним розподілом даних, дублюванням сутностей та забезпеченням їхньої актуальності.

Розглянуто необхідність розподілу системи на дві незалежні частини: Back-end та Front-end. Такий підхід дозволяє забезпечити модульність, гнучкість та легкість масштабування системи. Back-end відповідає за обробку запитів, роботу з базою даних та реалізацію алгоритмів формування рекомендацій. Front-end забезпечує зручний користувацький інтерфейс, що сприяє комфортній взаємодії користувача із системою.

Особливу увагу було приділено взаємодії системи з користувачем через інтерфейс. У деталях розглянуто дизайн і функціонал користувацького інтерфейсу, включно з елементами, які допомагають користувачеві здійснювати точні та зручні пошуки.

Додатково було проаналізовано аспекти забезпечення приватності користувачів. Зокрема, система не зберігає персональні дані користувачів, а інформація про переглянуті ігри зберігається локально в браузері й може бути видалена в будь-який момент.

Таким чином, цей розділ узагальнює ключові аспекти архітектури, технологій та взаємодії, що є основою роботи рекомендаційної системи відеоігор.

4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ ТА ЇХ АНАЛІЗ

4.1 Моделювання роботи рекомендаційної системи

Припустимо, що користувач за допомогою вебінтерфейсу розглянутої рекомендаційної системи обрав бажану платформу “PC” та жанр відеогри “Shooter ”. У такому випадку першим кроком буде обрати зі сховища п’ять ігор з найвищим рейтингом від користувачів та відео ігрових критиків, що відповідають цим параметрам (табл. 4.1).

Таблиця 4.1 – Початкові дані, отриманні з бази даних згідно вказаних користувачем параметрів.

Id	Назва	Оцінка критиків	Оцінка гравців	Опис	Список подібних
1	DOOM Eternal	94	92	Динамічний шутер з епічними битвами проти демонів…	[2,3,4]
2	Call of Duty: Modern Warfare	88	85	Реалістичний військовий шутер з насиченим сюжетом...	[1,3,5]
3	Borderlands 3	82	80	Гумористичний шутер з елементами RPG…	[1,2,5]
4	Apex Legends	89	87	Королівська битва з унікальними героями, що мають власні здібності…	[1,3]
5	Far Cry 6	80	78	Відкритий світ, революція на тропічному острові, і розмаїття зброї…	[2,3,4]

Згідно з отриманими початковими даними бачимо, що найбільш високо оціненою грою є гра з ідентифікатором 1, тож ця гра потрапить до списку рекомендованих в будь-якому випадку. Тепер, згідно з алгоритмами рекомендацій, що базуються на контенті, необхідно знайти сутності з отриманого списку, що є найбільш наближеними до еталону, в нашому випадку це гра з ідентифікатором 1.

4.2 Формування рекомендаційних правил

Для початку треба визначити за якими ознаками ми будемо формувати асоціативні правила. Зазвичай в рекомендаційних системах для цього обираються чеки користувачів, або найбільш популярні послуги що замовляють разом, щоб аналізувати які товари люди обирають зазвичай до пари. В нашому випадку “igdb” (відкрита безоплатна база даних відеоігор) надає нам список ідентифікаторів ігор, що асоціюються для гравців. Нашим головним завданням на цьому етапі буде зрозуміти наскільки міцні ці асоціативні зв’язки, або наскільки вони випадкові.

Для прикладу розрахуємо асоціативні параметри c,s (впевненість, підтримка) для елементів з ідентифікаторами 1 та 2. Використаємо формули 2.1 та 2.2:

$$c(1 \to 2) = \frac{\left|T\left(1\ AND\ 2\right)\right|}{T\left(1\right)} = \frac{3}{4} = 0,75. \tag{4.1}$$

$$s(1 \to 2) = \frac{\left|T\left(1\ AND\ 2\right)\right|}{T\left(\ \right)} = \frac{3}{5} = 0,6. \tag{4.2}$$

Розрахунки всіх пар представимо в таблиці 4.2.

Таблиця 4.2 – Асоціативні параметри для обраної вибірки

Confident,Support	DOOM Eternal	Call of Duty: Modern Warfare	Borderlands 3	Apex Legends	Far Cry 6
DOOM Eternal	-	0.75,0.6	1,0.8	0.5,0.4	0.5,0.4
Call of Duty: Modern Warfare	0.75,0.6	-	1,0.8	0.5,0.4	0.75,0.6
Borderlands 3	0.8,0.8	0.8,0.8	-	0.6,0.6	0.6,0.6
Apex Legends	0.6,0.4	0.6,0.4	1,0.6	-	0.3,0.2
Far Cry 6	0.6,0.4	1,0.6	1,0.6	0.3,0.2	-

4.3 Пошук важливих слів в описах сутностей

Для більш точної оцінки подібності між сутностями ми також можемо скористатися описами, які ми отримуємо про кожну відеогру. Для досягнення цієї мети нам необхідний алгоритм для аналізу подібності між собою двох текстів, використовуючи пошук ключових слів. Скористаємося методом пошуку ключових слів TF-IDF.[1]

Розрахуємо наскільки слово “шутер” є ключовим для опису відео гри з ідентифікатором 2. Розрахувавши за формулою (2.3), отримуємо:

$$tf(слово, документ) = \frac{m}{n} = \frac{1}{6} = 0{,}166, \tag{4.3}$$

де m – кількість появи слова в описі;

n – кількість слів в описі.

Цей розрахунок необхідно повторити для кожного слова в описі до гри з ідентифікатором під номер 2. В результаті отримаємо числовий вектор.

$$tf[1] = [0, 0.166, 0.166, 0, 0, 0, 0, 0.166, 0.166, 0.166,$$
$$0.166, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].$$

Довжина вектору відповідає кількості унікальних слів серед усіх розглянутих описів, в нашому випадку довжина вектору 30. Якщо слова немає в описі, відповідне його значення в числовому векторі дорівнює нулю. Нам необхідно звести всі вектори до однакової довжини, що дозволить нам виконувати коректі операції для порівняння їх між собою.

Обчислимо зворотну частоту появи слів в описі для вже використаного слова “шутер”. Використовуючи формулу (2.4), отримуємо:

$$idf\,(термін) = \log\left(\frac{cd}{cdt}\right) = \log\left(\frac{3}{5}\right) = 1,40\,, \tag{4.4}$$

де cd – загальна кількість документів;

cdt – кількість документів з термінами.

Після чого отримаємо числовий вектор зі зворотною частотою появи для кожного унікального слова в межах розглянутих відеоігрових описів.

Вище приведені розрахунки необхідні для обчислення TF-IDF для терміну. Що вважається високою оцінкою “ваги” кожного унікального слова в тексті.

Проведемо розрахунок TF-IDF для слова “шутер” та описом гри, що має ідентифікатор 2. Для розрахунку скористаємось формулою (2.5):

$$tf - id(слово, документ) =$$
$$= tf\,(слово, документ) \times idf\,(слово, документ) = 2,23\,. \tag{4.5}$$

Обчисливши TF-IDF для кожного опису ми отримали числові вектори однакової довжини де кожне число в ряді відповідає важливості кожного окремого слова саме для конкретного документа, враховуючи “вагу” слова для всіх документів в цілому.

Можемо скористатися L_2 для визначення відстані між двома векторами, що дозволить нам зробити висновок наскільки два описи є “подібними” (формула (2.6)).

Проте, даний параметр потребує нормалізації позаяк у випадку відстані, саме мала відстань між об’єктами буде вважатися більш позитивним результатом на відміну від великої відстані, тож цей параметр потребує нормалізації для коректного співвідношення з іншими.

Згідно формули (2.7), у випадку якщо L_2 більше одиниці, то значення $L_{2нор}$ буде вказуватися мінімальним і дорівнювати 0.1 що дасть справедливу оцінку надто великій різниці між векторами.

4.4 Формування остаточної оцінки подібності

В ході розрахунків ми отримали декілька різних характеристик (підтримка, впевненість, векторна відстань) що вказують, що вказують на різні аспекти подібності між сутностями (відеоіграми). Проте для формування остаточної оцінки необхідно сформувати єдине співвідношення цих показників для отримання остаточної оцінки подібності.

Для цієї мети чудово підходить такий інструмент як ранжування, формула якої приведена нижче.

$$f(\arg_1, \arg_2) = w_0 \times \arg_1 + w_1 \times \arg_2, \tag{4.6}$$

де w_0 та w_1 – вагові коефіцієнти.

Як ми бачимо з приведеної формули нам необхідно визначитися з параметрами, які будуть брати участь в ранжуванні та ваговими коефіцієнтами для кожного параметра.

Важливим параметром слід вважати рейтинг гри від критиків та гравців. Та високі вагові коефіцієнти для них не слід обирати, оскільки ми вже першочергово обирали гру з найвищим рейтингом. Далі наша мета обрати найбільш подібні до неї, тож високі вагові коефіцієнти для них не бажані, водночас інші параметри повинні мати більший вплив на остаточну оцінку, отже повинні мати вищі вагові коефіцієнти. Пропонується обрати ваговий коефіцієнт для рейтингу від критиків на рівні 0.04, для рейтингу, що встановили для гри користувачі, на рівні 0.06, та як, як правило, гравців більше і вони частіше ставлять рейтинг відеоіграм.

Далі слід розглянути сформовані нами асоціативні правила c, s (впевненість, підтримка). Ці параметри повинні мати значно вищі ваги через те, що вони вказують пряме відношення між різними відеоіграми. Проте c мусить мати більший пріоритет тому, що вказує на пряму залежність появи одної сутності в списку при наявності іншої, на відміну від s, яка вказує на зв’язок між двома

сутностями та всіма комбінаціями. Тож пропоную встановити вагові коефіцієнти відповідним чином 4 для s та 6 для c.

Останнім отриманим параметром подібності між сутностями була векторна відстань між описами до ігор. Ця величина повинна мати найвищий пріоритет, ніж виставлені оцінки. Схожі описи між двома іграми – це пряма ознака подібності між ними. Тож пропонується встановити цей ваговий коефіцієнт на рівні 8.

Обчислимо остаточну оцінку подібності між сутностями 1 та 2.

$$f(score_{critic}, score_{user}, s, c, L_{2нор}) = 0,04 \times score_{critic} \times 0,06 \times score_{user} + \\ +4 \times s + 6 \times c + 8 \times L_{2нор} = 15,73. \quad (4.7)$$

Представимо результати розрахунку остаточної оцінки для кожної з відібраних відеоігор (табл. 1.3).

Таблиця 4.3 – Оцінки подібності до елемента з ідентифікатором 1.

Id	Назва	Оцінка критиків	Оцінка гравців	s	c	$L_{рнор}$	Оцінка
2	Call of Duty: Modern Warfare	88	85	0.6	0.75	0.030	15.73
3	Borderlands 3	82	80	0.8	1	0.005	17.31
4	Apex Legends	89	87	0.4	0.5	0.037	13.64
5	Far Cry 6	80	78	0.4	0.5	0.010	12.55

Підсумовуючи наші розрахунки ми можемо припустити, що елементи з ідентифікаторами 2 та 3 мають найкращі показники подібності до елемента з Id 1. Тож користувач, який зробив запит з параметрами платформа “PC” та жанр відеогри “Shooter ”, отримає рекомендацію з елементів що мають Id 1, 2 та 3. Які будуть повернуті користувачу додатку що зробив запит.

Висновки за розділом 4

У розділі проведено моделювання роботи рекомендаційної системи для відеоігор жанру "Shooter" на платформі "PC". Розрахунки включали аналіз рейтингів критиків і гравців, асоціативних зв’язків між іграми та текстової подібності описів за допомогою TF-IDF. На основі комплексної оцінки, що враховує всі параметри з відповідними ваговими коефіцієнтами, визначено, що ігри з Id 1, 2 та 3 є найбільш подібними та рекомендованими для користувача.

ВИСНОВКИ

Наразі непросто знайти безоплатну рекомендаційну систему, яка б не накопичувала персональні дані користувачів і водночас була повністю незалежною. Тож метою моєї роботи є розробка рекомендаційної системи, яка не вимагала б жодних персональних даних від користувачів, але водночас генерувала найбільш релевантні рекомендації для них, базуючись виключно на аналізі сутностей, а не на вподобаннях користувачів. Ці вподобання враховуються в алгоритмах рекомендацій лише опосередковано.

Під час проєктування рекомендаційної системи були обрані алгоритми, що дозволяють формувати рекомендації, орієнтуючись на контент. Об’єднавши кілька алгоритмів у ланцюг для розрахунку подібності контенту, ми створили гібридну рекомендаційну систему. Вона використовує сильні сторони різних підходів, забезпечуючи кращий відбір сутностей без необхідності проведення прямого опитування користувачів.

Створена рекомендаційна система, що базується лише на аналізі контенту, надає релевантні та точні рекомендації, забезпечуючи високий рівень приватності для користувачів.

Вважаю цю тему надзвичайно перспективною, із широкими можливостями для вдосконалення алгоритмів рекомендацій шляхом впровадження технологій машинного навчання та використання більшого обсягу ресурсів для поглибленої аналітики.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Matviienko O., Diuh R. Designing a video game recommendation system. *13th International Scientific and Technical Conference «Information Systems And Technologies» IST-2024*. 2024, №13. P. 89 – 90.

2. Фальк К. Рекомендаційні системи на практиці. “ДМК Пресс”, 2020. 448 с.

3. Aggarwal, C. C. Recommender Systems: The Textbook. Springer, 2016. 498 с.

4. Manning, C. D., Raghavan, P., & Schütze, H. Introduction to Information Retrieval: Cambridge University Press, 2008. 482 с.

5. IGDB. URL: https://www.igdb.com/about (дата звернення 23.10.2024)

6. Node js. – Run JavaScript Everywhere. URL: https://nodejs.org/en (дата звернення 25.10.2024)

7. SQLite. URL: https://www.sqlite.org/ (дата звернення 25.10.2024)

8. Germán Cheuque, José Guzmán, Denis Parra. Recommender Systems for Online Video Game Platforms: the Case of STEAM. WWW '19: *Companion Proceedings of The 2019 World Wide Web Conference.* 2019. P. 763 – 771.

9. Ben Medler. Using Recommendation Systems to Adapt Gameplay. *International Journal of Gaming and Computer-Mediated Simulations (IJGCMS)*. 2009. P. 13.

10. Proposal For A Game Recommendation System Based On Review Keywords / Seo J., Yoo H., Jung W., Lee W., Kwak M. *ICIC Express Letters Part B: Applications*. 2023. №6. P. 619 – 624.

11. Pedro Teixeira. Professional Node.js: Building Javascript Based Scalable Software. Wrox, october 23, 2012. 351 p.

12. Alan Beaulieu. Learning SQL: Master SQL Fundamentals. O'Reilly, 2020. 380 p.