

Дивнич М.В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Голубєв Л.П.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Ківа І.Л.

Таврійський національний університет імені В.І. Вернадського

ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ФІЛЬТРАЦІЇ ІГРОВОГО КОНТЕНТУ З УРАХУВАННЯМ СОЦІАЛЬНОЇ АКТИВНОСТІ КОРИСТУВАЧІВ

У статті розглядається проєктування та реалізація інформаційної системи для фільтрації ігрового контенту з урахуванням соціальної активності користувачів. Актуальність розробки зумовлена швидким зростанням ігрової індустрії, оскільки існує потреба в інструментах, які полегшують користувачам швидкий перегляд великої кількості доступних цифрових ігор, вибір персоналізованого контенту та взаємодію з іншими користувачами на платформі. Система в першу чергу орієнтована на веб-середовище та реалізована з урахуванням сучасних підходів відносно масштабованості, взаємодії з користувачем та безпеки. На відміну від класичних механізмів фільтрації, запропонована система враховує не лише індивідуальні вподобання, але й поведінку друзів у мережі, зокрема зв'язок між їхніми іграми, оцінками тощо.

Особлива увага приділяється архітектурним та інженерним рішенням. Інтерфейс створено за допомогою мови програмування TypeScript та бібліотеки Angular. Для спрощення розробки веб-системи, використано набір UI-компонентів під назвою Angular Material. Усі функціональні компоненти системи реалізовано за допомогою лінивого завантаження (lazy loading), що оптимізує продуктивність. Також компоненти використовують стратегію OnPush, що зменшує кількість зайвих перерендерень. Застосовано механізм автентифікації на основі JWT-токенів, що зберігаються в локальному сховищі (localStorage) з указаним терміном дії, який безпосередньо забезпечує безпечний доступ до персоналізованих функцій. Клієнтські та серверні частини системи розташовані на різних доменах. Даний розподіл надає змогу враховувати питання безпеки та гнучкості архітектури для мікросервісного розділення.

Основна функціональність системи полягає в можливості переглядати ігри, розрізняти їх жанри, оцінки, ціни. Також користувач повинен мати змогу додавати певну гру в бібліотеку, створювати власні списки побажань і відображати свої рекомендації друзям. На поточному етапі, функціональність системи визначається шляхом спостереження та оцінки інтерфейсу до неї, через обмежену кількість користувачів. Даний підхід дозволяє збирати високоякісний зворотний зв'язок, який допоможе в наступних етапах розробки.

Ключові слова: інформаційна система, рекомендаційна система, фільтрація контенту, соціальна активність, ігрові платформи, матрична факторизація, JWT-автентифікація.

Постановка проблеми. В сучасному світі, кількість цифрового контенту щодо ігор стрімко зростає. Щороку випускаються тисячі нових ігор на різних платформах і кожна з них охоплює різні жанри, методи, теми тощо. Дане середовище є досить насичене і користувачі все частіше стикаються зі складним завданням вибору з безлічі ігор, які не тільки технічно відповідають певним

вимогам, але й мають особисту привабливість для гравця. Традиційні інструменти пошуку, які пропонують жанрову, тематичну чи цінову фільтрацію, більше не ефективні для задоволення зростаючих запитів користувачів, оскільки вони не мають соціального контексту, моделей поведінки чи рекомендацій від довірених осіб (родини, друзів чи знайомих зі схожими вподобаннями).

На практиці, користувачі здебільшого покладаються на своє особисте ігрове коло, оскільки дуже часто обирають гру, яку вже придбали друзі або яка часто обговорюється у спільнотах чи на форумах. Дана поведінка свідчить про необхідність розширення функціональних платформ – від традиційної каталогізації до персоналізованої фільтрації, яка враховує соціальну активність. Це питання є особливо проблемним у сфері ігрових веб-платформ, які використовуються як ринки, джерела інформації та простори для спільноти гравців. Незважаючи на безліч вже існуючих рішень, таких як Steam, GOG, Epic Games Store, Itch.io, кожне з них має свої недоліки, через які, користувачі змушені залишатися з великим обсягом інформації наодинці.

Це свідчить про те, що наразі відсутній єдиний універсальний підхід, який би поєднував в собі багатоаспектну фільтрацію, індивідуальні рекомендації, аналіз вподобань друзів та соціальну взаємодію. Дана проблематика підкреслює важливість розробки нової інформаційної системи, яка має змогу об'єднати сильні сторони готових систем, одночасно усуваючи їхні ключові недоліки.

Проблема полягає у необхідності розробки архітектури та алгоритмічного ядра для такої системи. Зокрема, мова йде про впровадження гібридних моделей рекомендацій, що в змозі інтегрувати колаборативну фільтрацію (collaborative filtering), подібність контенту та соціальні зв'язки користувачів. Також не менш важливим завданням є розробка адаптивного інтерфейсу, яка безпосередньо ґрунтується на сучасних фронтенд-технологіях і забезпечує швидкий доступ до релевантного контенту, не перевантажуючи користувача.

Отже, предмет дослідження даної статті полягає не тільки у створенні зручної платформи для пошуку ігор, а й у реалізації нового типу рекомендаційної системи, яка базується на моделях соціальної взаємодії та структурованій каталогізації у поєднанні з масштабованими веб-технологіями. Це все відкриває новий шлях до розвитку не лише нової інформаційної системи, але й цілої екосистеми для інтелектуального споживання ігрового контенту.

Аналіз останніх досліджень і публікацій. Протягом двох останніх десятиліть тематика рекомендаційних систем набула ключового значення в сфері інтелектуального аналізу даних, особливо в контексті цифрових платформ, медіа, комерції та соціальних сервісів. Однією з фундаментальних моделей, яка заклала основу для сучасних рекомендаційних систем, є матрична факторизація. В роботі Y. Koren, R. Bell та C. Volinsky [1, с. 30-37]

була запропонована модель розкладання матриці оцінок користувачів на латентні фактори, що надає змогу брати до уваги неявні взаємозв'язки між користувачем та контентом. Такий підхід знайшов успішне застосування, зокрема у рамках Netflix Prize, де модель продемонструвала значне покращення точності рекомендацій порівняно з традиційними методами.

Однак, модель від Y. Koren не передбачає врахування жодних зовнішніх чи соціальних даних. Дану проблему частково вирішують соціально-орієнтовані моделі, які досліджені в роботі Jamali та Ester [2, с. 135-142]. Вони представляють комбіновану модель матричної факторизації з урахуванням довіри між користувачами. Їхній підхід «Trust-aware Recommender Systems» показує, що інтеграція соціальних зв'язків у розрахунок подібності здатна суттєво покращити якість персоналізованих рекомендацій. Схожі дослідження проводилися F. Tang, X. Wang, C. S. Norman [3, с. 1060-1073], де пропонувалася модель, яка модифікує функцію втрат з врахуванням соціального впливу.

Великий відсоток сучасних робіт зосереджений на гібридних рекомендаційних системах [4, с. 331-370]. В дослідженнях G. Adomavicius та A. Tuzhilin [5, с. 734-749] було закладено теоретичний фундамент для комбінування collaborative filtering (CF) та content-based filtering (CBF). Згідно з їхньої класифікацією є кілька методів поєднання підходів, а саме: послідовне комбінування, об'єднання результатів і модель на основі метарівня. Даного роду архітектури використовуються у великих сервісах, наприклад Amazon, YouTube та інших. Однак, ці підходи не часто реалізуються у відкритому вигляді для спеціалізованих застосувань, особливо у сфері відеоігор.

Окрему увагу в науковому світі останніми роками приділяють моделям, що базуються на графах та графових нейронних мережах (Graph Neural Networks). L. Zhang та співавтори [6, с. 482-494], у своєму огляді розглянули напрямок graph-based recommendation, в якому користувачі та об'єкти виступають як вузли графа, а взаємодії – як ребра. Такі моделі дозволяють глибше вивчати структуру зв'язків, зокрема формальні й не формальні соціальні групи, ступінь впливу тощо. Проте, дані моделі в більшості випадків лишаються теоретичними або знаходять своє застосування в соціальних мережах (Facebook, Twitter), а не в системах каталогізації ігрового контенту.

Варто також відзначити книгу від F. Ricci, L. Rokach та B. Shapira під назвою «Recommender

Systems Handbook» [7, с. 1-35]. Вона вважається фундаментальним джерелом, в якому детально розкладено класифікацію методів рекомендацій, починаючи від евристичних підходів і закінчуючи машинним навчанням. В даному виданні також акцентується увага на проблемі «data sparsity» (розрідженості даних), яка є дуже актуальною згідно нових платформ, які мають низьку активність користувачів і системи потребують створення надійного профілю користувача.

Попри наявність широкого спектру підходів, аналіз наукових праць демонструє брак досліджень, які б інтегрували повноцінну функціональність каталогізації ігор з багатовимірною фільтрацією та соціальними механізмами впливу. Існуючі рішення або орієнтовані на рекомендації без урахування соціального контексту (як Netflix), або ж на соціальні зв'язки без структурованої каталогізації (як Discord). У сфері геймінгу, дослідження зосереджуються переважно на алгоритмах Steam, які не є відкритими та не передбачають гнучку фільтрацію, що інтегрована з аналізом активності друзів.

Отже, проблема полягає у відсутності комплексних підходів, які б одночасно:

- Враховували вподобання користувача на основі жанрових і цінових характеристик;
- Моделювали вплив соціального оточення;
- Працювали на структурованих даних про ігри;
- Були реалізовані в адаптивному інтерфейсі для веб-платформи.

Запропонована в цій статті система має вирішити дану проблему, поєднуючи всі зазначені компоненти та відкриваючи можливості для розвитку в напрямку сучасних AI-моделей і графових методів.

Постановка завдання. Основна ціль статті полягає в дослідженні, проектуванні та реалізації прототипу веб-орієнтованої інформаційної системи, яка дає можливість ефективно каталогізувати ігровий контент, забезпечує багатофакторну фільтрацію та формує персоналізовані поради, базуючись на аналізі не тільки особистих вподобань користувача, а й соціальної активності його контактів (друзів, приятелів, знайомих тощо). Такий підхід має на меті збільшити релевантність рекомендацій та вдосконалити користувацький досвід у контексті цифрових ігрових платформ.

Задля досягнення поставленої мети, в рамках аналізу, визначено ряд конкретних задач, кожна з яких відображає логіку поетапного проекту-

вання та впровадження системи. Виглядають вони наступним чином:

1) Розробити логічну й програмну архітектуру системи, яка включає клієнтську та серверну частини, а також структуровану базу даних для збереження ігор, профілів користувачів, соціальних зв'язків та історії взаємодій.

2) Сформулювати математичну модель гібридної рекомендаційної системи, що поєднує елементи контентно-орієнтованої фільтрації, матричної факторизації та соціального контексту. Ключова задача моделі – визначити релевантність кожної гри для користувача, спираючись на аналіз як його попередніх уподобань, так і активності його друзів у системі.

3) Реалізувати програмну логіку рекомендаційного модуля, беручи до уваги підрахунок схожості між іграми і налаштувати параметри алгоритмів під поведінкові особливості гравців.

4) Підготувати систему до первинного тестування, яке включає огляд взаємодії огляд взаємодії з інтерфейсом цільовою аудиторією, збір якісних відгуків щодо зручності пошуку, відповідності видачі контенту, а також зручності соціальних функцій. На основі отриманих даних – визначити шляхи для покращення системи та алгоритмічної складової.

Отже, головне завдання статті полягає у тому, щоб створити повноцінну технічну та методологічну основу для інформаційної системи, яка зможе поєднати переваги сучасних технологій веб-розробки з науково-обґрунтованими моделями персоналізованих рекомендацій та аналізу соціального впливу в ігровому просторі.

Виклад основного матеріалу. Розроблена інформаційна система для фільтрації ігрового контенту побудована за модульною клієнт-серверною архітектурою, що дає змогу досягти високої масштабованості, гнучкості розгортання, відмовостійкості та розмежування функціональних обов'язків. Дана система є орієнтованою на сучасні веб-технології та розроблена з врахуванням найкращих практик frontend- та backend-розробки, враховуючи при цьому безпечну взаємодію між складовими.

Інформаційна система складається з трьох ключових компонентів:

– Клієнтська частина (Frontend) – реалізована з використанням фреймворку Angular, що забезпечує створення SPA (Single Page Application) з високою продуктивністю. Для забезпечення оптимізації продуктивності, усі UI-компоненти організовані за модульним принципом з вико-

ристанням лінивого завантаження (lazy loading). Для виявлення змін застосовується стратегія під назвою OnPush, Вона зменшує навантаження на рендеринг та підвищує реактивність інтерфейсу.

– Серверна частина (Backend) – розроблена за допомогою платформи Node.js та її бібліотеки Express. Вона обробляє усю бізнес-логіку, керує обліковими записами користувачів та оперує запити до бази даних. API (Application Programming Interface) спроектовано як REST-сервіс, що функціонує за протоколом HTTP для взаємодії між клієнтом та сервером. Безпека забезпечується за допомогою JWT-автентифікації. При вході користувач отримує access- та refresh-токен, термін дії яких є обмеженим. Дані токени зберігаються у localStorage браузера.

– База даних (Database) – основа зберігання всієї інформації, що необхідна для функціонування системи. Вона містить відомості про користувачів, ігри, жанри, дружні зв'язки, списки бажаного та запити на додавання в друзі. Для сховища даних використовується документо-орієнтована база даних MongoDB, яка працює з JSON-подібними документами.

Веб-клієнт та серверна частина спілкуються один з одним за допомогою HTTP-запитів. Елементи системи розташовані на різних доменах, що відповідає сучасним підходам розділення frontend та backend як окремих сервісів, з підтримкою CORS (Cross-Origin Resource Sharing). Це забезпечує не тільки безпеку та здатність до масштабування, але й дає можливість окремо розгортати та підтримувати кожну з частин системи.

Повну блок-схему системи можна побачити на рисунку нижче (рис. 1).

Клієнт в даному випадку відповідає за візуалізацію ігор, застосування фільтрів, перегляд рекомендацій і списку друзів; сервер – за авторизацію, маршрутизацію запитів, підрахунок рейтингів і рекомендацій, а також контроль доступу; БД – за повторюваним зберіганням структурованих і пов'язаних даних (користувачів, їхніх вподобань тощо).

Також на наступному зображенні (рис. 2) відображено модель «сутність-зв'язок» (ER-модель) системи. Вона пояснює концептуальні схеми, вдаючись до узагальнених блочних конструкцій. Основними сутностями виступають користувачі, ігри, запити на дружбу та взаємодія між ним. Структура базується на принципі вкладеності, що дозволяє безпосередньо вбудовувати зв'язки у форматі масивів об'єктів.

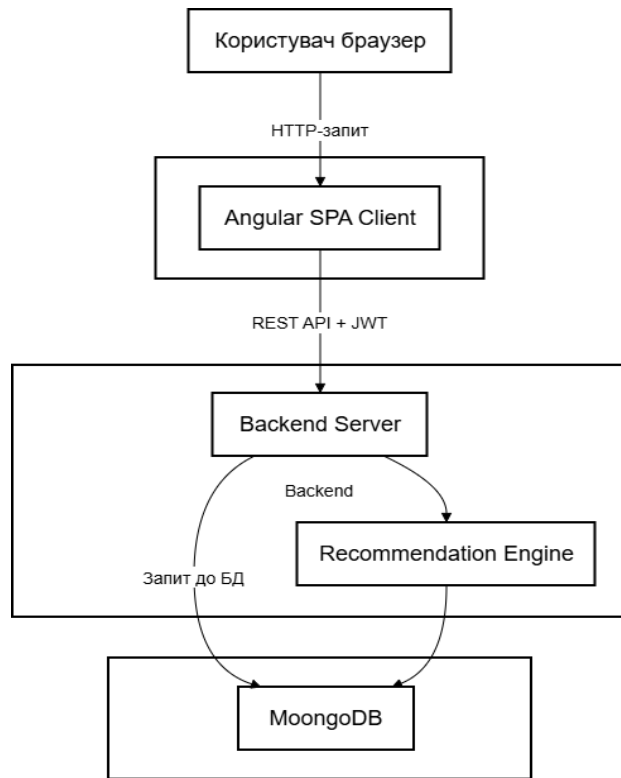


Рис. 1. Блок-схема системи (архітектурна взаємодія)

В рамках проєкту, за основу взято модель рекомендаційної системи, яка використовує матричну факторизацію. Даний метод дозволяє виявити приховані закономірності у вподобаннях користувачів, навіть за умов неповних даних. На відміну від класичних методів, що використовують сусідство користувачів чи предметів, підхід на основі latent factor models надає змогу зменшити розмірність вхідної матриці і виділити при цьому також суттєві поведінкові патерни.

Основою для навчання рекомендаційної системи є матриця оцінок $R \in \mathbb{R}^{(n \times m)}$, де n – кількість користувачів, m – кількість ігор, а кожен елемент r_{ui} – оцінка гри i користувачем u . В даному випадку оцінки формуються на основі дій користувачів. Це може бути як додавання гри до бібліотеки, так і тривалість перегляду її опису.

Основним завданням є факторизація матриці R у дві низьковимірні матриці: U (матр. латентних ознак користувачів) та V (матр. латентних ознак ігор):

$$\hat{R} = U \cdot \Sigma \cdot V^T$$

де $U \in \mathbb{R}^{(n \times k)}$, $V \in \mathbb{R}^{(m \times k)}$, k – кількість латентних факторів (зазвичай 20-100).

Наступним кроком є застосування методу SVD (Singular Value Decomposition). Даний метод дозволяє подати високо-розмірну, розріджену

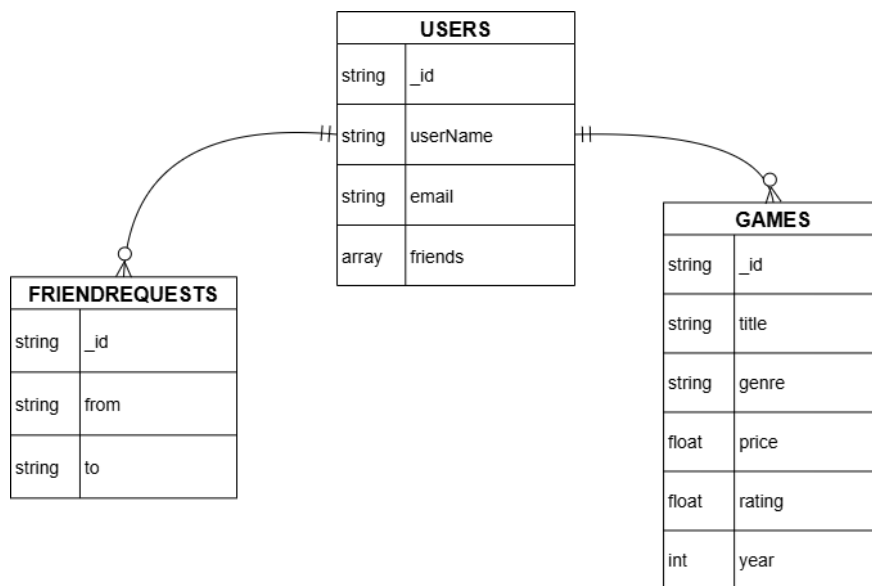


Рис. 2. ER-модель системи

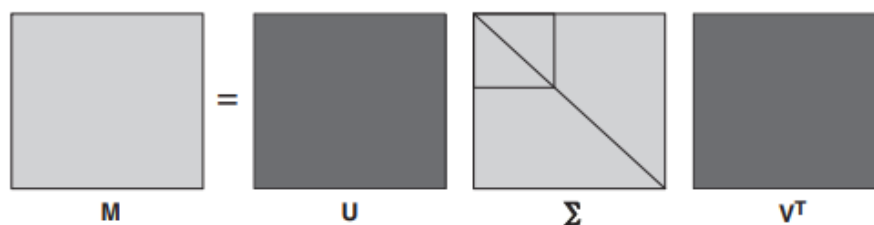


Рис. 3. Розклад матриці за допомогою методу SVD

матрицю у вигляді добутку трьох меншого розміру матриць (див. рис. 3). SVD дозволяє розкласти матрицю $R \in R^{(n \times m)}$ наступним чином:

$$\hat{R} = U \cdot \Sigma \cdot V^T$$

Тут $\Sigma \in R^{(k \times k)}$ – діагональна матриця сингулярних значень. Використовуються лише перші k найбільших сингулярних значень, що дозволяють зменшити розмірність (зазвичай $k = 10 \div 100$), зберігши найбільш значущі закономірності у поведінці користувачів.

Розглянемо приклад матриці оцінок п'яти користувачів до п'яти ігор (табл. 1).

Таблиця 1

Початкова матриця оцінок R					
	g_1	g_2	g_3	g_4	g_5
u_1	5	3	0	1	0
u_2	4	0	0	1	2
u_3	1	1	0	5	0
u_4	0	0	5	4	0
u_5	0	1	5	4	0

Застосовуючи SVD з обмеженням до двох латентних факторів, виконаємо факторизацію та отримаємо наступний результат (табл. 2):

Таблиця 2

Матриця передбачених оцінок $\hat{R}$

	g_1	g_2	g_3	g_4	g_5
u_1	4.98	1.99	-0.54	1.65	0.87
u_2	3.57	1.43	-0.39	1.18	0.62
u_3	1.83	1.03	2.09	2.98	0.30
u_4	-0.37	0.43	4.47	4.46	-0.10
u_5	-0.07	0.55	4.46	4.58	-0.05

Отримані передбачені значення оцінювання використовуються для створення персоналізованого списку рекомендацій відносно кожного окремого користувача. Наприклад, для користувача u_3 найвищою оцінкою модель визначає гру g_4 , а також g_3 – незважаючи, навіть, на відсутність взаємодії між ними.

В підсумку, SVD-факторизація дає змогу системі робити обґрунтовані передбачення щодо можливих вподобань користувача на основі даних схожих користувачів та структури ігор. Такий під-

хід дозволяє закласти фундамент для подальшого включення соціальних та контентних модифікацій.

Окрім звичайної факторизації матриці оцінок з використанням сингулярного розкладу (SVD), в роботі було досліджено Funk SVD (див. рис. 4). Вона являє в собі покращену версію SVD, яка тренується за допомогою стохастичного градієнтного спуску (Stochastic Gradient Descent). Даний варіант активно використовується в завданнях персоналізованої фільтрації. Він дозволяє навчатися безпосередньо на розріджених даних.

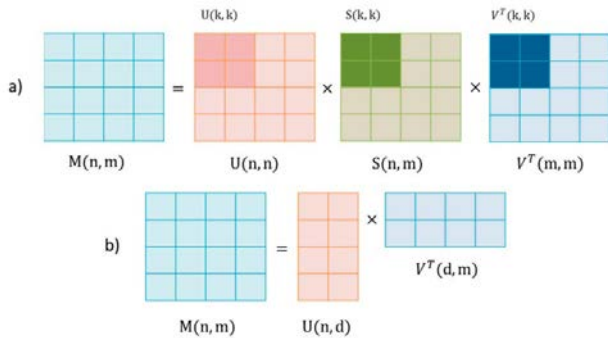


Рис. 4. Відмінність роботи алгоритму SVD (a) та Funk SVD (b)

Головна концепція полягає в тому, щоб представити кожного користувача і кожну гру у вигляді векторів латентних особливостей фіксованої розмірності k , та навчити їх таким чином, щоб скалярний добуток цих векторів найточніше відповідав відомим рейтингам.

Для кожного користувача u будується латентний вектор $p_u \in R^k$, а для кожної гри i – вектор $q_i \in R^k$. Передбачена оцінка обчислюється за формулою:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot p_u$$

де μ – середнє значення всіх оцінок, b_u , b_i – відповідно зміщення користувача та гри.

Слід зазначити, що у нас можлива наявність факторів користувачів і також певного відхилення. В такому випадку, прогноз оцінки являє в собі суму чотирьох речей, які показані детальніше на рис. 5.

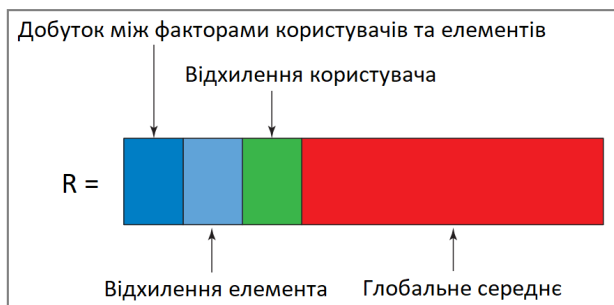


Рис. 5. Прогноз оцінки – це комбінація чотирьох речей

Навчання здійснюється за допомогою стохастичного градієнтного спуску. Для кожного спостереження $(u,i) \in k$ (де k – множина відомих оцінок), оновлення параметрів відбувається за наступними правилами:

$$e_{ui} = r_{ui} - \hat{r}_{ui}$$

$$b_u \leftarrow b_u + \gamma \cdot (e_{ui} - \lambda \cdot b_u)$$

$$b_i \leftarrow b_i + \gamma \cdot (e_{ui} - \lambda \cdot b_i)$$

$$p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u)$$

$$q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i)$$

Тут γ – швидкість навчання (learning rate), а λ – коефіцієнт регуляризації.

Однією із важливих проблем під час створення рекомендаційних систем, базованих на латентних моделях – є перенавчання. Даний процес виникає, коли модель дуже точно запам'ятовує існуючі (навчальні) оцінки користувачів, але втрачає здатність до узагальнення. Можна сказати, що вона не може якісно передбачати на нових, раніше невідомих прикладах. У разі розріджених матриць, де бракує більшості оцінок, перенавчання особливо небезпечне. Для того, щоб уникнути даної проблематики, до функції втрат додається коефіцієнт регуляризації, котрий штрафуватиме надмірно великі значення параметрів моделі. У випадку Funk SVD, будуть регуляризуватися латентні вектори користувачів та ігор, а також зміщення. Мінімізація виражається у формулі:

$$\min \sum_{(u,i) \in k} (r_{ui} - \hat{r}_{ui})^2 + \lambda \cdot (\|p_u\| + \|q_i\| + b_u^2 + b_i^2)$$

Високе значення λ зменшує гнучкість моделі, але підвищує її здатність до узагальнення, тоді як занадто низьке може призвести до перенавчання. Сама ідея полягає в тому, щоб алгоритм знаходив найкращі U і V , не дозволяючи при цьому жодному з них ставати занадто великим [8, с. 342-367]. Графік MSE (міра похибки алгоритмів прогнозування) тестових та навчальних даних для кращого розуміння наведено нижче (рис. 6).

В умовах розробки масштабованих рекомендаційних систем першорядне значення мають дві ключові характеристики, а саме: точність відтворення даних та швидкість обчислень. Тому, було проведено аналіз двох методів матричної факторизації, зосереджуючись на їх продуктивності у задачах фільтрації, заснованих на користувацьких вподобаннях. На наведеному графічному матеріалі (див. рис. 7) показано тривалість обчислень для обох алгоритмів, в залежності від розміру вхідної матриці $k \times d$. Funk SVD демонструє

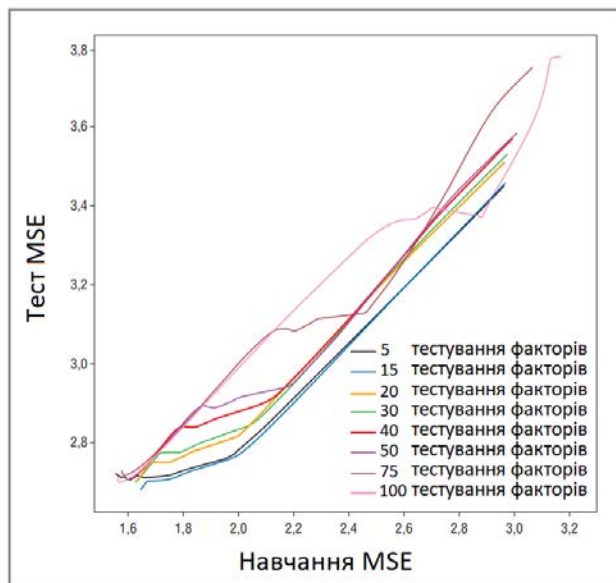


Рис. 6. Графік MSE тестових і навчальних даних, який демонструє момент перенавчання

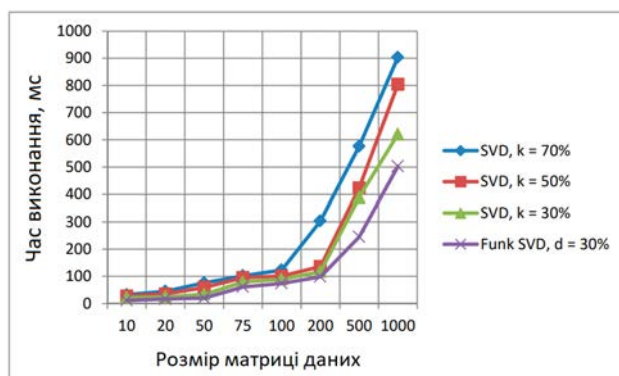


Рис. 7. Порівняння тривалості обчислень алгоритмів SVD та Funk SVD

постійно менший час обробки даних у порівнянні з традиційним SVD в середньому на 20%. Це пояснюється тим, що Funk SVD не потребує повної матриці для виконання.

В даному випадку, можна сказати, що алгоритм особливо ефективний при роботі з великими обсягами інформації, де переважна більшість комірок є незаповненими.

Додатково, окрім показників продуктивності, досить важливою характеристикою вважається оцінка якості відтворення вихідних даних після процедури факторизації. Згідно з проведенням дослідженням (див. рис. 7), метод Funk SVD не демонструє втрати у точності порівняно з класичним SVD. Іншими словами, навіть при суттєвому зменшенні обчислювальних витрат, якість апроксимації (відновлення) залишається майже не змінною. Відносно отриманого результату,

варіант Funk SVD є привабливим рішенням у системах реального часу, де швидкість формування персоналізованих рекомендацій є найважливішим аспектом.

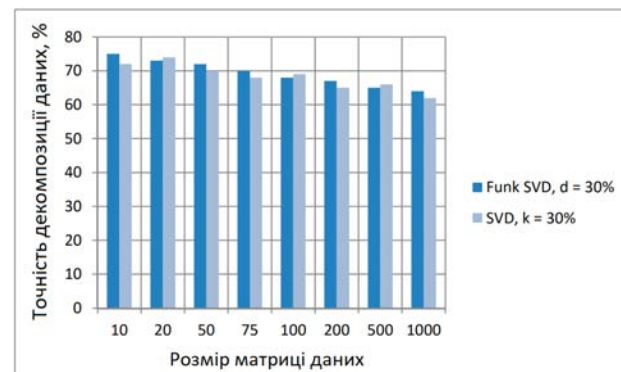


Рис. 8. Порівняння точності відновлення початкової матриці після факторизації для SVD та Funk SVD

Завдяки своїй простоті й адаптивності, Funk SVD є оптимальним вибором для онлайн-рекомендацій, коли необхідно швидко реагувати на дії користувача. Крім того, його реалізація дозволяє легко інтегрувати соціальні механізми впливу.

В умовах сучасних рекомендаційних систем, зокрема тих, що функціонують на цифрових платформах із соціальними аспектами (як Steam, GOG), соціальний вплив відіграє ключову роль [9, с. 56-58]. Дослідження демонструють, що користувачі з більшою ймовірністю зацікавляються продуктами, які вже вподобали їхні друзі або отримали рекомендації у процесі соціальної взаємодії. Врахування цього ефекту дозволяє підвищити якість рекомендацій, особливо у випадках холодного старту (cold start) або за обмеженої інформації про користувача.

Соціальний компонент інтегрується до моделі Funk SVD у вигляді дружнього зваженого впливу на остаточну оцінку гри. Суть полягає в тому, щоб скоригувати передбачувану оцінку гри для конкретного користувача, беручи до уваги, як її оцінили його друзі. Формула для підсумкового прогнозу виглядає наступним чином:

$$\hat{r}_{ui}^{social} = \hat{r}_{ui} + \alpha \cdot \frac{1}{|F(u)|} \sum_{f \in F(u)} (\hat{r}_{fi} - \mu_f)$$

де $\hat{r}_{ui}$ – базова оцінка, яка отримана через Funk SVD; $F(u)$ – множина друзів користувача u ; μ_f – середня оцінка, яку зазвичай ставить користувач f ; α – коефіцієнт соціального впливу (емпірично встановлено в межах 0.1-0.3).

Дана формула надає змогу «піднести» гру вище в списку, коли друзі користувача їй дають

високу оцінку і також навпаки – опустити, якщо гра не цікавить його оточення. Розглянута варіація відповідає природній поведінці гравців, оскільки користувач з більшою вірогідністю обере гру, якщо бачитиме, що в неї активно грають або її обговорюють знайомі.

Наведемо приклад використання даного методу. Для початку виконаємо припущення, що користувач u_3 , має двох товаришів – u_1 , u_2 . Прогнозована оцінка гри g_3 для u_3 без соціального впливу становить $\hat{r}_{ui} = 3.8$. Його друзі оцінили гру наступним чином: $\hat{r}_{fi} \in \{4.9, 5.0\}$, а їхні середні оцінки – $\mu_f \in \{3.5, 4.0\}$. Беручи до уваги коефіцієнт соціального впливу ($\alpha = 0.2$) отримаємо:

$$\hat{r}_{ui}^{social} = 3.8 + 0.2 \cdot \frac{(4.9 - 3.5) + (5.0 - 4.0)}{2} = 3.8 + 0.2 \cdot 1.2 = 4.04$$

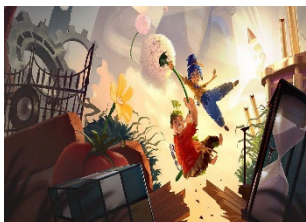
Отже, як бачимо, рекомендація посилюється, що відповідає нашому очікуванню. Гра буде показана користувачу як більш релевантна.

Виконуючи ряд досліджень, у таблиці порівняння (див. табл. 4) наведено приклад рекомендованих ігор, що були отримані за допомогою п'яти різних методів, а саме: асоціативних правил, контентної та колаборативної фільтрації, матричної факторизацією та Funk SVD із соціальним компонентом. Добірка сформована відносно ігор жанру RPG/Action Adventure ілюструє, як різні методи формують рекомендації залежно від структури даних, історії вподобань та взаємодії користувачів.

Стосовно кожного методу можна навести короткий аналіз, щоб розуміти, якому віддавати перевагу в даному випадку. Пояснення наведено у таблиці нижче (табл. 5).

Таблиця 4

Порівняння ігрових рекомендацій різними методами

Association Rules			
Content-Based			
Collaborative Filtering			
Matrix Factorization (SVD)			
Matrix Factorization (Funk SVD) + Social Layer			

Також, в ході роботи, було отримано порівняльну характеристику (див. табл. 6) метрик якості (Recall@10, NDCG@10, RMSE) для різних методів рекомендацій. Вона базується на частково реалізованому тестуванні і виглядає наступним чином.

Додатково побудовано графік (рис. 9) відносно метрик Recall@10 та NDCG@10.

Проведене порівняння засвідчило, що модель Funk SVD з врахуванням соціального впливу досягла найвищих показників Recall@10 та NDCG@10. Даний результат вказує на її найкращу здатність формувати рекомендації, що є максимально релевантними та адаптованими для конкретного користувача. Такий підхід не просто

бере до уваги особисті вподобання, а й враховує вплив оточення користувача, зокрема його друзів, що особливо важливо для ігрових платформ, де вибір часто базується на колективній думці.

На противагу класичним підходам, таким як асоціативні правила чи контентна фільтрація, які продемонстрували помітно меншу точність через неврахування особливостей поведінки користувачів, алгоритми колаборативної фільтрації забезпечили кращу якість. Проте, вони виявилися досить чутливими відносно проблеми «cold-start». Стандартна SVD-модель показала добрі результати, проте поступається Funk SVD. Інтеграція соціального компонента з Funk SVD виявилася найефек-

Таблиця 5

Порівняльний аналіз різних методів

Association Rules	Асоціативні правила спираються на частоту, з якою різні гравці обирають певні ігри разом. Пропонують базові, загальні рекомендації.
Content-Based	Контентно-орієнтовані аналізують жанри, рік випуску, ціни та рейтинги ігор. Добре працює для новачків, але не бере до уваги смаки інших користувачів.
Collaborative Filtering	Колаборативна фільтрація базується на подібності профілів гравців. Ефективний метод, але вразливий до проблеми cold-start [10, с. 107-144].
Matrix Factorization (Funk SVD)	Матрична факторизація за допомогою Funk SVD – це модель, що навчається на вже поставлених оцінках, виявляє приховані симпатії. Надає персоналізовані, детальні поради.
Matrix Factorization (Funk SVD) + Social Layer	Funk SVD із соціальним компонентом включає у рекомендації дії друзів. Дає більш соціально-значущі результати, оскільки гравець бачить те, що обирають його друзі.

Таблиця 6

Порівняльна характеристика метрик якості

Метод рекомендації	Recall@10	NDCG@10	RMSE
Association Rules	0.112	0.108	–
Content-Based Filtering	0.178	0.161	–
User-Based CF	0.219	0.198	0.983
Item-Based CF	0.224	0.204	0.965
Classical SVD	0.242	0.227	0.936
Funk SVD	0.259	0.238	0.911
Funk SVD + Social Layer	0.274	0.245	0.928

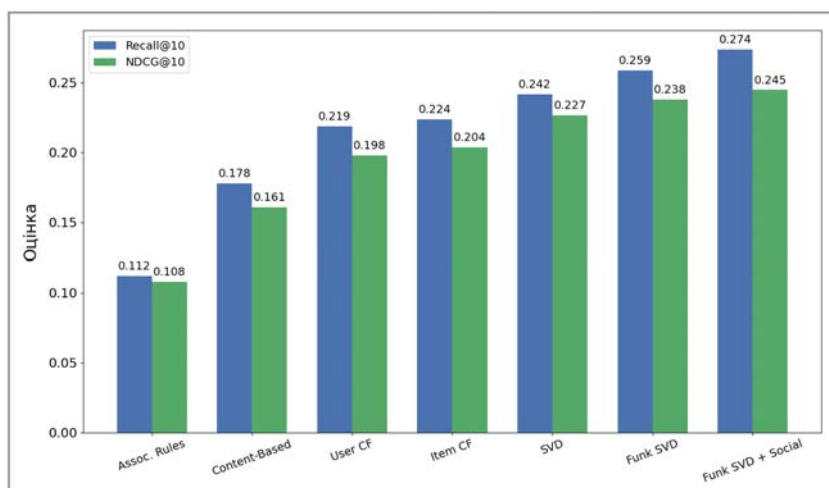


Рис. 9. Порівняння ефективності методів рекомендації

тивнішою, особливо для нових або неактивних користувачів, для яких складно сформувати рекомендації, виходячи лише з їхньої особистої історії взаємодій (ефект «friend boosting»).

Отримані в процесі дослідження результати демонструють наукову новизну, котра полягає в інтеграції латентно-факторних моделей рекомендацій із компонентами соціального впливу в рамках ігрової інформаційної системи. Зокрема, застосовано варіацію класичної факторизації Funk SVD, в якій передбачене значення оцінки $\hat{r}_{ui}$ доповнюється соціальним зсувом, котрий бере до уваги поведінку найближчих друзів у даній структурі

Загалом, сформовано наступну модель передбачення:

$$\hat{r}_{ui}^{social} = \mu + b_u + b_i + q_i^T \cdot p_u + \alpha \cdot \frac{1}{|F(u)|} \sum_{f \in F(u)} (\hat{r}_{fi} - \mu_f)$$

З позиції метричних оцінок, було продемонстровано, що отримана модифікація збільшує показники ранжування (Recall@10 та NDCG@10) приблизно на 5-6% у порівнянні зі звичайним методом Funk SVD. Дане емпіричне покращення підтверджує припущення, щодо значення врахування взаємодій типу «friend-to-friend» в розріджених системах з численними профілями «cold-start».

Отже, запропонований метод відкриває перспективу розробки гібридних friend-aware рекомендаційних систем з адаптивним контролем довіри та соціального впливу.

Висновки. Під час дослідження було розроблено прототип інформаційної системи, який призначений для каталогізації ігор, що оснащений багатокритеріальною фільтрацією, функцією персоналізованих рекомендацій та можливостями соціальної взаємодії між користувачами. Рекомендаційна підсистема базується на модифікованій моделі Funk SVD, що адаптована під врахування соціального компонента (вплив друзів на формування оцінок). Побудова латентно-факторної моделі з коригуванням передбачених оцінок на основі соціального фактора призвела до значного покращення показників точності.

Отримані результати підтверджують релевантність рекомендацій, які враховують не лише історію взаємодії користувача, але й його соціальне середовище.

Також, система включає багатокритеріальну фільтрацію ігор, за такими параметрами, як жанр, рейтинг, рік випуску та інші. Для забезпечення зручності користувача, в інтерфейсі використано сучасні технології: фреймворк Angular, реалізовано модулі lazy loading, аутентифікацію JWT, та застосовано UI-компоненти Angular Material. Окрему увагу було приділено структуруванню даних в MongoDB, створенню матриць схожості та аналізу патернів поведінки в системі.

Дана робота демонструє потенціал для покращення класичних рекомендаційних моделей шляхом інтеграції соціальних факторів, що відкриває нові перспективи для досліджень у сфері персоналізації.

У подальших дослідженнях варто розглянути наступні можливості:

- Вдосконалення соціальної моделі, додавши механізми вимірювання довіри (trust score) між користувачами;
- Запровадження динамічного зважування соціального впливу, яке буде змінюватися в залежності від кількості взаємодій у мережі;
- Інтегрування механізму нейронних рекомендацій (Neural Collaborative Filtering) для підвищення точності в умовах складних взаємозв'язків;
- Реалізація соціального графу у режимі реального часу з візуалізацією зв'язків та впливу друзів на рекомендації;
- Інтегрування додаткових видів взаємодії: коментарі, рецензії, спільні обговорення, що дозволить покращити аналіз соціального контексту.

Підсумовуючи, можна сказати, що представлена система є ефективною як у технічному, так і в поведінковому плані. Вона закладає основу для створення ігрових платформ наступного покоління з інтелектуальним ядром персоналізованого контенту.

Список літератури:

1. Koren Y., Bell, R., Volinsky C. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer*. 2009. Vol. 42, No. 8. P. 30–37.
2. Jamali M., Ester M. A matrix factorization technique with trust propagation for recommendation in social networks. *Proceedings of the fourth ACM conference on Recommender systems*. 2010. P. 135–142.
3. Tang F., Wang X., Norman C. An investigation of the impact of media capabilities and extraversion on social presence and user satisfaction. *Behaviour & Information Technology*. 2013. Vol. 32. P. 1060–1073.
4. Burke R. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*. 2002. Vol. 12, No. 4. P. 331–370.

5. Adomavicius G., Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*. 2005. Vol. 17, No. 6. P. 734–749.
6. Zhang S., Yao L., Sun A., Tay Y. Deep Learning based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys*. 2017. Vol. 52, No. 1. P. 482–494.
7. Ricci F., Rokach L., Shapira B. *Recommender Systems Handbook*. Springer, 2010. P. 1–35.
8. Falk K. *Practical Recommender Systems*. Shelter Island, NY: Manning Publications, 2019. P. 342–367.
9. Resnick P., Varian H. R. Recommender systems. *Communications of the ACM*. 1997. Vol. 40, No. 3. P. 56–58.
10. Desrosiers C., Karypis G. A Comprehensive Survey of Neighborhood-Based Recommendation Methods. *Recommender Systems Handbook*. Springer, 2011. P. 107–144.

Dyvnych M.V., Holubiev L.P., Kiva I.L. AN INFORMATION SYSTEM FOR GAME CONTENT FILTERING BASED ON USERS SOCIAL ACTIVITY

The article explores the design and implementation of an information system for filtering gaming content while incorporating users' social activity. The relevance of this development is driven by the rapid growth of the gaming industry and the increasing need for tools that assist users in browsing vast collections of digital games, selecting personalized content, and interacting with other users on the platform. The system is primarily web-oriented and is implemented using modern approaches to scalability, user interaction, and security. Unlike traditional filtering mechanisms, the proposed solution takes into account not only individual preferences but also the behavior of users' friends – including correlations between games, ratings, and social interactions.

Special attention is paid to architectural and engineering decisions. The interface is developed using TypeScript and the Angular framework. To accelerate development, a set of UI components from Angular Material was used. All functional modules are implemented with lazy loading to optimize performance, and OnPush change detection is employed to minimize unnecessary re-renderings. The authentication mechanism is based on JWT tokens stored in localStorage with an expiration time, ensuring secure access to personalized features. The client and server parts of the system are hosted on different domains, which increases security and allows for scalable, microservice-based architecture.

The core functionality includes browsing games by genre, rating, and price. Users can add games to their personal libraries, create wishlists, and share recommendations with friends. At the current stage, the system's evaluation is conducted through user interface observation and feedback from a limited group of test users. This approach enables the collection of high-quality insights that will guide further system development.

Key words: *information system, recommender system, content filtering, social interaction, game platforms, matrix factorization, JWT-authentication.*

Дата надходження статті: 14.11.2025

Дата прийняття статті: 02.12.2025

Опубліковано: 30.12.2025