

Розроблення рекомендаційної системи на основі колаборативної фільтрації та machine learning з врахуванням особистих потреб користувача

В. В. Литвин, В. А. Висоцька, В. В. Шатських, І. В. Когут, О. С. Петрученко, Л. В. Дзюбик, В. В. Бобрівець, В. М. Панасюк, С. І. Саченко, М. П. Комар

Проведено дослідження рекомендаційних алгоритмів та виявлення переваг та недоліків. Вдосконалено метод формування рекомендацій на основі колаборативної фільтрації як Content-Based Filtering (CBF), Collaborative Filtering (CF) та гібридних методів Machine Learning (ML). Описано принципи проектування та функціональні вимоги до рекомендаційної системи у вигляді веб-додатка для вибору необхідного для користувача контенту на прикладі фільмів. Основні дослідження зосереджені на вирішенні проблем холодного старту та масштабованості в методі колаборативної фільтрації. Для ефективного вирішення цих проблем, використані гібридні методи навчання. Здійснена практична реалізація гібридної рекомендаційної системи (ГРС) надання релевантних рекомендацій контенту на прикладі фільмів з врахуванням особистих потреб користувача на основі розробленого гібридного методу. Удосконалено алгоритм формування рекомендацій контенту на основі колаборативної фільтрації та Machine Learning для спільної фільтрації показників подібності між користувачами або між товарами. Гібридний алгоритм приймає вхідну інформацію у різному вигляді, нормалізує її та формує відповідні рекомендації на основі комбінації методів CF та CBF. Machine Learning здатне визначати чинники, що впливають на підбір релевантних фільмів, що сприяє поліпшенню надання рекомендацій конкретному користувачу. Для вирішення цих завдань пропонується новий удосконалений метод, на відміну від наявних систем рекомендацій, в основі якого лежать гібридні методи та Machine Learning. Дані для Machine Learning розробленої ГРС взято із MovieLens. Проаналізовано методи формування рекомендацій користувачеві, проведений огляд наявних рекомендаційних систем. Експериментальні результати показують, що показники роботи запропонованої ГРС на основі технології CF+CBF+ML кращі, ніж у двох одиничних моделей, CF та CBF, та їх комбінацій як CF+CBF, CF+ML та CBF+ML. ГРС рекомендується використовувати для збору даних про вподобання людей у виборі товару та надання релевантних рекомендацій

Ключові слова: комерційний контент, персоналізація, Machine Learning, SEO-технологія, метрики пошуку, електронна комерція, NLP

1. Вступ

Кількість інформації у всьому світі зростає швидкими темпами протягом значного проміжку часу [1]. Люди кожного дня сприймають та фільтрують вхідний потік інформації, що надходить з різних джерел: робота, соціальні мережі, електронна пошта, онлайн кінотеатри, популярні джерела інформації тощо [2]. Після

винайдення мережі Інтернет кількість такої інформації стала стрімко зростати, з'явилася велика кількість сервісів для надання користувачам всього необхідного для комфортного життя [3]. За останні роки набули значної популярності Інтернет-сервіси, що пропонують товари всіх можливих видів (Інтернет-магазини), інформацію на будь-який смак (Інтернет-журнали, фільми, новини, форуми, книги, серіали, статті) тощо [4]. Користувачу стало надзвичайно важко орієнтуватися в каталогах фільмів та списках серіалів, навіть із вбудованим пошуком та фільтрацією, оскільки дуже важко зробити вибір при настільки великому обсягу інформації [5]. Всесвітня мережа Інтернет дає можливість зручного збору інформації про вподобання користувачів онлайн-ресурсів [6]. У всесвітній мережі Інтернет налічується величезна кількість пошукових систем, каталогів, рейтингів, файлообмінників тощо. І ця кількість постійно збільшується. Тому кількість контенту експотенціально зростає, оскільки кожен сайт/сервіс в процесі функціонування породжує ще більші обсяги контенту [7]. Основне, але не єдине завдання рекомендаційних сервісів – допомогти споживачеві контенту зорієнтуватися у результатах пошуку. Історично люди покладалися на своїх однолітків або на пропозиції експертів для підтримки прийняття рішень та рекомендацій щодо товарів, новин, розваг тощо [8]. Експоненціальне зростання цифрової інформації за останні 25 років, особливо в Інтернеті, створило проблему інформаційного перевантаження. Цей термін визначається як «стрес, індукований отриманням більшої кількості інформації, ніж необхідно для прийняття рішення, і спробами боротися з нею за допомогою застарілих методів управління часом». Ця проблема обмежує здатність переглядати специфікації та вибирати між численними альтернативами на онлайн-ринку. З іншого боку, інформаційна наука і технологія відповідно реагували, розробляючи інструменти фільтрації інформації для вирішення цієї проблеми [9]. Рекомендаційні системи (РС) і є такими інструментами. РС виникли в середині 90-х років [10]. Зазвичай визначаються як програмні засоби та методи, які використовуються для надання пропозицій користувачам та іншим зацікавленим особам [1].

Рекомендаційні системи є програмними засобами та методами, які надають пропозиції щодо предметів, які найбільш ймовірно представлятимуть інтерес для конкретного користувача [11]. Пропозиції стосуються різних процесів прийняття рішень, таких як те, що потрібно купити, яку музику прослуховувати або які фільми чи телесеріали переглядати [1].

Зазвичай рекомендації фільмів різних онлайн-сервісів наявних рекомендаційних систем не завжди є релевантними для користувачів, оскільки часто базуються на непродуктивних та застарілих алгоритмах [12]. Проблем створення системи прогнозування для рекомендацій фільмів чимало, тож дуже цікаво подивитися, як саме різні підходи до створення рекомендаційних систем вирішують ці проблеми та покращують якість рекомендацій [13]. Тематика є актуальною, оскільки в Інтернет недостатньо належних сервісів для надання релевантних рекомендацій фільмів [14]. Всі вони є закритими комерційними проектами. І для розробників аналогічних проектів необхідно заново придумувати структуру та архітектуру для власних рекомендаційних систем. Також мусять провадити власні аналізи апробацій для вибору або вдосконалення або розроблення нових найкращих методів та технологій формування рекомендацій кінцевому користувачу.

Персоналізований підхід до користувача Web-ресурсу значно збільшує коефіцієнти продаж. Тобто, клієнти, які не можуть знайти шукану ними інформацію, залишають Web-ресурс незадоволеними і ймовірніше ніколи не переглядатимуть його знову, з огляду на надзвичайно конкурентний Інтернет-ринок. Згідно з [2] до 40 % відвідувачів зазвичай використовують функцію пошуку Web-ресурсу, тим самим демонструючи намір придбати продукцію за назвою чи кодом. Отже, одержання необхідних результатів персоналізованого пошуку та подальший аналіз засобами штучного інтелекту на основі колаборативної фільтрації та Machine Learning необхідне для успішного розвитку е-бізнесу. В е-комерції звертаються до Machine Learning для покращення рекомендацій для споживачів як постійних/потенційних відвідувачів Web-ресурсу. Методи Machine Learning значно покращують аналіз результатів попереднього персоналізованого пошуку при кожному відвідуванні користувача Web-ресурсу. Генерування рейтингу тематичного пошуку дозволяє на Web-ресурсі відсортовувати результати за релевантністю чи оціночною актуальністю. Ця оцінка враховує специфічні пошукові терміни, а також особливості відповідного профілю користувача (наприклад, віковий діапазон, стать, попередні замовлення, смакові переваги та попередні пошукові терміни). Алгоритми персоналізації дозволяють пов'язати кожного користувача з наймовірнішим списком товарів/послуг згідно його потреб та зацікавлень. Ці алгоритми також дозволяють прогнозувати побажання клієнтів, навіть якщо про це ще не знають або не здогадуються. Окрім звичайного текстового введення категорій та тегів на основі зображень та опису товарів/послуг все більш життєздатним варіантом є додавання процесів автоматизації та систем прийняття рішень для ідентифікації побажань користувача Web-ресурсу. Наукові досягнення щодо розпізнавання контексту за допомогою Machine Learning мереж тепер забезпечують технологію автоматичного додавання тегів в описи товару Web-ресурсу е-комерції.

2. Аналіз літературних даних і постановка проблеми

Рекомендаційні системи відіграють важливу роль у популярних Web-сайтах, таких як YouTube, Netflix, Last.fm, Spotify, Facebook, Tripadvisor, LinkedIn та IMDb [15]. Більшість медіа-компаній розробляють і розгортають РС як частину послуг, які надають своїм абонентам [16]. Наприклад, Netflix, онлайн-провайдер потокових мультимедійних даних за запитом, присудив премію у мільйон доларів команді, якій вперше вдалося значно покращити продуктивність своєї системи рекомендацій [1]. В роботі [17] наведені результати досліджень, чому постачальники послуг можуть захотіти використовувати цю технологію. Показано, що існують певні фактори як:

- збільшення кількості проданих товарів;
- продаж більш різноманітних предметів;
- підвищення задоволеності користувачів;
- збільшення вірності користувачів;
- краще розуміння того, що хоче користувач.

Перший чинник, який слід врахувати при розробленні РС, є область застосування, оскільки має значний вплив на алгоритмічний підхід, який необхідно

прийняти [18]. Існують ресурси, які надають таксономію РС і класифікують наявні РС на конкретні домени додатків [1]. Виходячи з цих конкретних доменів, можна визначити більш загальні класи доменів для найбільш поширених систем застосування [7]:

- розваги – рекомендації для фільмів, музики, ігор та IPTV [19];
- контент – персоналізовані газети, рекомендації щодо документів, вебсторінок, програм для е-навчання та фільтри е-пошти [20];
- е-комерція – рекомендації для покупців (книги, камери, ПК тощо) [21];
- послуги – рекомендації туристичних послуг, експертів для консультацій, будинків для оренди або послуг для знайомств [22];
- соціальні – рекомендації людей та контенту у соціальних мережах, таких як твіти, канали Facebook, оновлення LinkedIn тощо [23].

Але залишилися невирішеними питання, пов'язані з вибором конкретного алгоритмічного підходу для конкретної області застосування РС [24]. Причиною цього можуть бути об'єктивні та суб'єктивні труднощі, пов'язані з:

- відсутністю досліджень та експериментів для аналогічних проектів в цих доменах для широкого кола користувачів (більшість проектів є приватними одноразовими закритими комерційними, результати роботи яких не підлягають розголосу) [25];
- принципова неможливість отримати доступ як до архітектури таких закритих комерційних проектів, так і до методології розроблення, результатів аналізу функціонування тощо [26];
- витратна частина в плані проведення власних експериментів як на чужих проєктах, так і створення власних декількох проєктів, що робить відповідні дослідження недоцільними, і т. п. [27].

Варіантом подолання відповідних труднощів може бути лише спроба проведення власних експериментів на власне розроблених проєктах [28]. Однак є один суттєвий недолік – відсутність порівняння результатів з аналогами [29]. Тому доцільним є проведення дослідження, присвяченого аналізу рейтингів товару е-комерції для формування рекомендацій кінцевому користувачу на основі особистих потреб та вподобань.

На рис. 1 подана загальна архітектура РС. Фактично, рейтинги є найпопулярнішою формою даних про транзакції, які збирає РС [30]. Ці рейтинги можуть збиратися явно або неявно [31]. У явному наборі оцінок користувачу пропонується надати думку про предмет у рейтинговій шкалі [32]. Інша форма оцінки користувача складається з тегів, асоційованих користувачем з елементами, які система подає [33]. Наприклад, у MovieLens (<https://movielens.org/>) теги РС показують, як користувачі MovieLens описують фільм, наприклад: «затягнутий» або «фантастичний» [2]. Технічно в РС використовують декілька рекомендаційних стратегій в редакційних алгоритмах як Content-Based Filtering (CBF), Collaborative Filtering (CF), Demographic Filtering (DF), Knowledge-Based Filtering (KBF) тощо [34]. Також існує клас підходів на поєднанні різних методів – Hybrid Filtering (HF) [35].

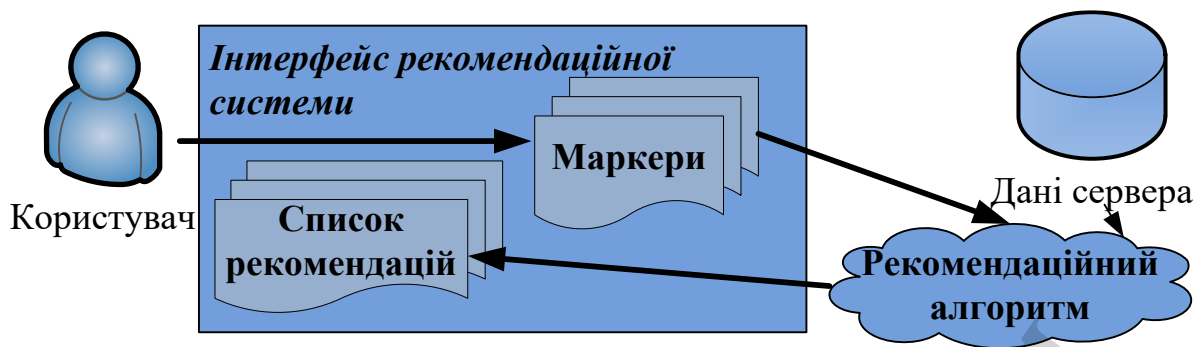


Рис. 1. Загальна архітектура РС

Загальні кроки для формування рекомендацій (рис. 2) [3]:

- 1) попереднє опрацювання, що включає пошук подібності даних, визначення вибірки, розмірності даних [36];
- 2) аналіз даних, який містить класифікацію та кластеризацію даних [37];
- 3) подання результатів [38].

Сьогодні РС зустрічаються повсюдно, допомагаючи користувачам в пошуку різних видів товарів/послуг [39]. Також є помічниками з продажу для підприємств, збільшуючи їхній прибуток [40]. Щоб зрозуміти, як описувати методи надання рекомендацій потрібно відповісти на ряд питань [41]:

- які проблеми стоять перед дослідниками у цій галузі?
- які методи інтелектуального аналізу даних і Machine Learning використовують в гібридних РС (ГРС)?
- які ці методи об'єднуються і які проблеми вирішують?
- які для оцінки використовують методології та метрики?
- які характеристики РС оцінюються і які метрики використовують?
- які набори даних використовують для навчання та тестування ГРС?
- які напрямки найбільш перспективні для майбутніх досліджень?

Проблеми методів генерації рекомендацій у РС [42]:

– *Холодний старт* має відношення до рекомендацій для нових користувачів або товарів при відсутності відповідної статистичної інформації про них (уподобання нового користувача та рейтингів для нового товару) [43]. Тоді часто використовують імовірнісну модель для вилучення прихованих ознак з елементів [44]. Використовуючи приховані функції, генерують точні псевдорейтинги, навіть у ситуації холодного старту, коли мало або немає жодної оцінки [45]. Іншим прикладом вирішення є поєднання CF та CBF [46]. Проблему холодного старту також вирішують за допомогою математичної техніки об'єднання зважених виходів різних стратегій рекомендацій з використанням методу Ordered Weighted Averaging (OWA) [47].

– *Розрідженість даних* зазвичай виникає при оцінці користувачами обмеженої кількості доступних елементів, особливо коли каталог великий [48]. Результатом є розріджена матриця рейтингу користувачів з недостатніми даними для ідентифікації подібних користувачів або товару, що негативно впливає на якість рекомендацій [49]. Розрідженість даних переважає в CF РС, які покладаються на зворотний зв'язок з колегами для надання рекомендацій [50]. Розрі-

дженість даних міждоменних рекомендацій часто вирішують використанням моделі факторизації тріадового відношення user-item-domain [51]. Також розглядають кожний рейтинг користувачького елемента як предикат інших відсутніх оцінок [52].

– *Точність рекомендацій* є здатністю РС релевантно передбачити переваги елемента для певного користувача [53]. Підвищенню точності рекомендацій завдяки приділяється велика увага [54]. Це особливо необхідно в ситуаціях розрідженості даних [55].

– *Масштабованість* є важкодоступною характеристикою, яка пов'язана з кількістю користувачів і товару РС [56]. При формуванні рекомендацій кількох пунктів для декількох сотень користувачів, ймовірно, система не зможе запропонувати мільйонам людей сотні товару, якщо не розрахована на високу масштабованість [57]. Розробники системи Nured для усунення такої проблеми поєднали модифіковану кореляцію Пірсона CF з граничною відстанню CBF [58]. Тобто знаходять найближчих і далеких сусідів кожного користувача для зменшення набору даних [59]. Використання стислого набору даних покращує масштабованість, зменшує розрідженість та час обчислень системи [60].

– *Різноманітність* є бажаною характеристикою, яка останнім часом привертає особливу увагу [8]. Важливо мати різноманітні рекомендації, оскільки це допомагає уникнути упередженості рекомендацій [61]. Таким може бути список рекомендацій з подібними елементами (наприклад, всі епізоди дуже популярної саги) [62]. Користувач, не зацікавлений в одному з них, ймовірно, не зацікавлений у жодному, не отримує користі з рекомендації [63].

– Іншими проблемами є *відсутність персоналізації* [64], *збереження конфіденційності* [65], *зменшення шуму* [66], *інтеграція джерел даних* [67], *відсутність новизни* [68] та *адаптивність переваг користувача* [69].

Вирішення хоча би частини проблем проектування доменних РС значно покращить якість та ефективність обслуговування кінцевого користувача е-комерції в сучасному світі прогресивних Інтернет-технологій. Але основною проблемою є вибір найкращого алгоритму формування рекомендацій для доменних РС конкретного е-бізнесу. Проаналізуємо поширенні та популярні рекомендаційні алгоритми як CBF, CF, KBF, DF та HF [70].

CBF базується на припущенні: якщо вподобали товар з певними атрибутами в минулому, також вподобають такий товар в майбутньому [71]. Використовує функції елементів для порівняння елемента з профілями користувачів та надання рекомендацій (рис. 2) [72]. Якість рекомендацій обмежена вибраними функціями рекомендованих елементів. CBF страждає від проблеми холодного старту (новий користувач або новий елемент), «сірий олівець» (користувачі, які не вписуються в будь-який кластер вподобань) [73].

Основне припущення CF: подібні смаки людей в минулому, ймовірно, будуть такими ж в майбутньому [74]. Одним з визначень є «співпраця між людьми, щоб допомогти один одному виконувати фільтрацію, записуючи їхні реакції на документи, які читають» [3]. Використовує рейтинги або інші форми зворотного зв'язку, згенеровані користувачем, для виявлення спільних смаків між групами користувачів (рис. 3). Далі генерує рекомендації на основі подібності

між користувачами [2]. Як і CBF, CF страждає від проблеми холодного старту [75]. Яскравий приклад CBF ілюструє діаграма Венна для користувачів системи (рис. 4), де подані зв'язки між «подібністю» та «відмінністю» двох користувачів однієї системи. Як бачимо, «відмінності» є чудовим джерелом для генерування рекомендацій [6]. Алгоритми CBF поділяють на категорії як Memory-based, Model-based та Hybrid (рис. 5) [76].

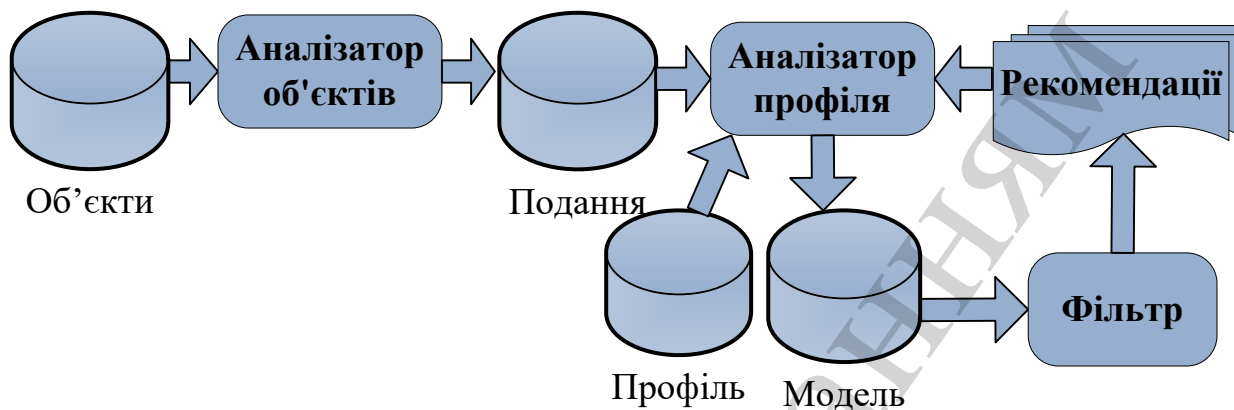


Рис. 2. Схема утворення рекомендації CBF

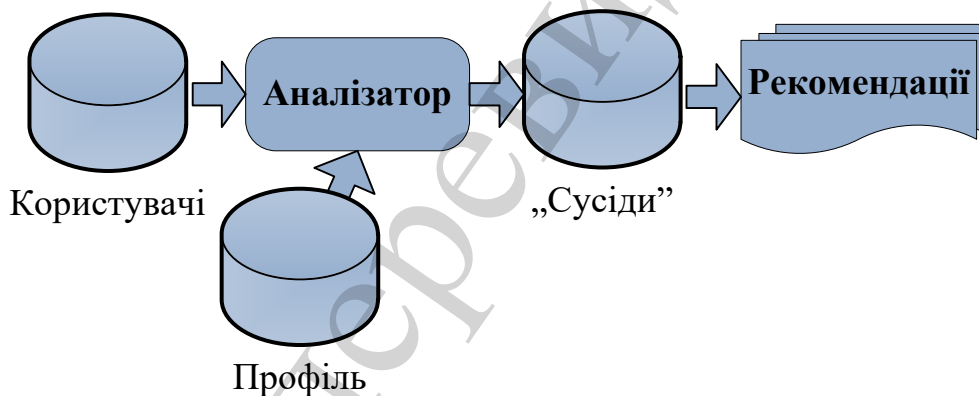


Рис. 3. Схема утворення рекомендації CF

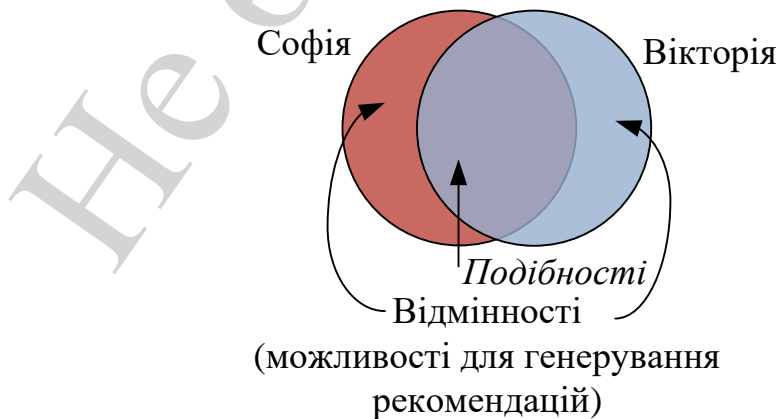


Рис. 4. Діаграма Венна CBF

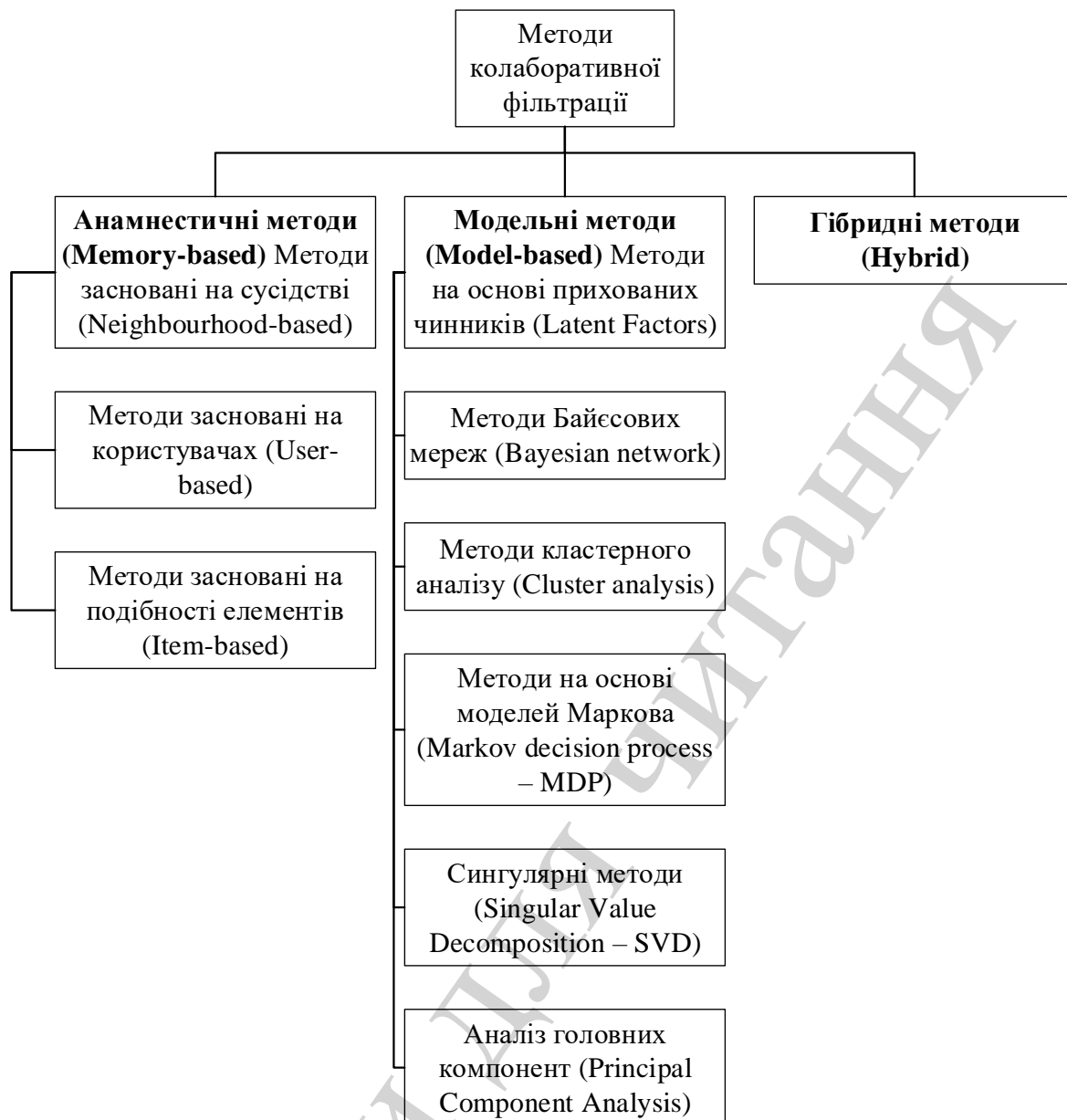


Рис. 5. Методи СВФ

КВФ використовує знання про користувачів та товар для обґрунтування, які товари відповідають вимогам користувачів, і генерують відповідні рекомендації [4]. Особливим типом КВФ є РС, що базуються на обмеженнях [72]. Здатні рекомендувати складні товари, які рідко купують (наприклад, автомобілі, будинки), і виявляють важливі обмеження для користувача (наприклад, ціну) [5]. Неможливо успішно використовувати CF або СВФ для цих товарів, оскільки переважно є мало даних про взаємодію користувачів системи (люди рідко купують будинки) [73].

DF використовує демографічні дані як вік, стать, освіта тощо для визначення категорій користувачів [74]. Не страждає від проблеми холодного старту, оскільки не використовує рейтинги [75]. Проте, складно зібрати достатню кількість необхідної демографічної інформації через проблеми конфіденційності в Інтернет, що обмежує використання DF. Поєднують з іншими методами для покращення точності рекомендацій [76].

В HF при поєднанні CBF та CF може бути більш ефективним у деяких випадках [77]. HF реалізують кількома способами: шляхом окремо окреслених прогнозів на основі контенту та спільних. А потім їх об'єднують шляхом додавання можливостей, заснованих на контенті, до підходу, що базується на спільній роботі (і навпаки) [78]. Кілька досліджень емпірично порівнюють продуктивність HF з чистими методами CBF та CF і демонструють, що HF може забезпечити більш точні рекомендації, ніж чисті підходи [79]. Ці методи також можуть бути використані для подолання деяких поширених проблем РС, таких як холодний старт і проблема розрідженості даних. Netflix є гарним прикладом використання ГРС [80]. Сайт надає рекомендації, порівнюючи звички перегляду та пошуку подібних користувачів (наприклад, спільну фільтрацію), а також пропонуючи фільми, які поділяють характеристики з фільмами, які користувач високо оцінив (на основі вмісту) [81].

Кожен з методів CBF, CF, KBF, DF має недоліки [82]. Тому використання HF вирішує проблемі місця кожного чистого методу окремо [83]. ГРС є такою, яка об'єднує кілька методів разом для досягнення певної синергії між ними [84]. Термін ГРС використовується тут для опису будь-якої СР, яка об'єднує декілька методів рекомендацій разом для отримання ефективного та якісного результату [85]. Немає жодних причин, чому кілька різних методів одного типу не можуть бути гібридизовані, наприклад, дві різні рекомендації на основі CBF та CF можуть працювати разом [86]. Існує декілька методів гібридизації [87]:

- зважування: чисельна оцінка різних компонентів рекомендацій;
- перемикання: вибір одного варіанту серед компонентів рекомендацій;
- змішані: одночасне використання рекомендації різних рекомендацій;
- комбінація ознак: функції, отримані з різних джерел знань, об'єднуються разом і надаються уніфікованому алгоритму рекомендації;
- збільшення функцій: одна методика рекомендацій використовується для обчислення набору функцій як частини вхідних даних для наступної техніки;
- каскад: рекомендаціям надається чіткий пріоритет, при цьому найнижчі пріоритети порушують зв'язки у виграші вищих;
- мета-рівень: застосовується одна методика рекомендацій і генерується модель як вхідна інформація для наступної техніки.

І лише практичне дослідження для конкретної доменної ГРС дозволить виявити найкращий варіант для формування релевантних рекомендацій кінцевому користувачу згідно особистих потреб та вподобань.

3. Мета і завдання дослідження

Метою дослідження є розроблення системи прийняття рішень для формування рекомендацій контенту на основі колаборативної фільтрації та Machine Learning з врахуванням особистих потреб користувача

Для досягнення мети були поставлені такі завдання:

- розробити загальні функціональні вимоги до архітектури системи формування рекомендацій контенту для поширення комерційного контенту в Інтернет-просторі;

- розробити метод персоналізації комерційного контенту згідно потреб користувача;
- розробити програмне забезпечення поширення комерційного контенту в Інтернет-просторі на основі колаборативної фільтрації та Machine Learning;
- здійснити аналіз результатів експериментальної апробації запропонованого методу персоналізації комерційного контенту згідно потреб користувача.

4. Загальні функціональні вимоги до архітектури системи формування рекомендацій контенту для поширення контенту

Для наближення до реалізації мети, опишемо генеральну мету, аспекти генеральної мети та підаспекти генеральної мети системи, а також критерії. Для цього побудуємо дерево цілей СППР (рис. 6).

Формування найбільш релевантних рекомендацій користувачеві є третім аспектом генеральної мети системи. У ньому виділимо окремий підаспект, мета якого – обрати зі всіх релевантних рекомендацій 10 рекомендацій з найвищим рейтингом. Для реалізації системи скористаємося МАІ (методом аналітичної ієрархії) з такими 4 альтернативними варіантами (табл. 1–3):

- v_1 – комбінованого типу;
- v_2 – орієнтована на дані;
- v_3 – орієнтована на знання;
- v_4 – орієнтована на моделі.

Таблиця 1

Заповнені матриці попарних порівнянь на основі аналізу робіт [1–13]

	Критерій 1 Точність рекомендування				Критерій 2 Ліквідація холодного старту				Критерій 3 Масштабованість			
	v_1	v_2	v_3	v_4	v_1	v_2	v_3	v_4	v_1	v_2	v_3	v_4
v_1	1	1	6	3	1	6	2	6	1	2	3	3
v_2	1	1	8	5	1/6	1	4	2	1/2	1	1	5
v_3	1/6	1/8	1	2	1/2	1/4	1	4	1/3	1	1	6
v_4	1/3	1/5	1/2	1	1/6	1/2	1/4	1	1/3	1/5	1/6	1
	Критерій 4 Розрідженість даних				Критерій 5 Збільшення продажів				Критерій 6 Різноманітність			
	v_1	v_2	v_3	v_4	v_1	v_2	v_3	v_4	v_1	v_2	v_3	v_4
v_1	1	4	6	2	1	3	6	1	1	6	3	6
v_2	1/4	1	6	4	1/3	1	7	3	1/6	1	2	3
v_3	1/6	1/6	1	2	1/6	1/7	1	3	1/3	1/2	1	7
v_4	1/2	1/4	1/2	1	1	1/3	1/3	1	1/6	1/3	1/7	1
	Критерій 7 Релевантність рекомендацій				Критерій 8 Збільшення постійних користувачів				Критерій 9 Ліквідація шуму			
	v_1	v_2	v_3	v_4	v_1	v_2	v_3	v_4	v_1	v_2	v_3	v_4
v_1	1	3	7	3	1	2	3	2	1	2	6	2
v_2	1/3	1	2	3	1/2	1	9	1	1/2	1	6	3
v_3	1/7	1/2	1	6	1/3	1/9	1	6	1/6	1/6	1	2
v_4	1/3	1/3	1/6	1	1/2	1	1/6	1	1/2	1/3	1/2	1



Рис. 6. Дерево цілей для надання релевантних рекомендацій

Результати експертного оцінювання на основі аналізу робіт [1–13] та наступне застосування МАІ дозволяють зробити висновок про доцільність обрання варіанту побудови РС комбінованого типу (табл. 3). Для прикладу оберемо домен ГРС – розваги, піддомен – фільми(рис. 7).

Користувач задіяний у процесах надання вподобань жанрів фільмів та улюблених фільмів. Користувач повинен отримати релевантні рекомендації, а

також має можливість переглянути усі доступні рекомендації. Окремим варіантом використання є можливість додати рекомендований фільм у обране.

Таблиця 2

Порівняння критеріїв за якісними аспектами на основі аналізу робіт [1–13]

Фактор 1 Підвищення задоволеності користувача				Фактор 2 Адаптивність переваг користувача				Фактор 3 Розуміння потреб користувача			
1	3	6	1	1	6	2	6	1	4	6	2
1/3	1	7	3	1/6	1	4	2	1/4	1	6	4
1/6	1/7	1	3	1/2	1/4	1	4	1/6	1/6	1	2
1	1/3	1/3	1	1/6	1/2	1/4	1	1/2	1/4	1/2	1

Таблиця 3

Вектор пріоритетів, розрахований за методом аналітичної ієрархії

Альтернативи	v_1	v_2	v_3	v_4	Значення ваг	Варіанти вибору
v_1	1	2	6	2	0,42596	1
v_2	1/2	1	6	3	0,256486	2
v_3	1/6	1/6	1	2	0,252045	3
v_4	1/2	1/3	1/2	1	0,065509	4

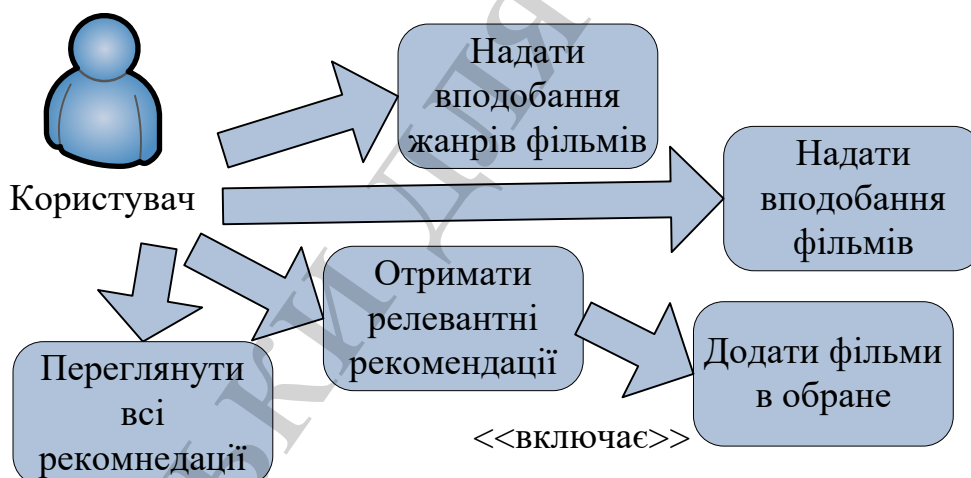


Рис. 7. Діаграма варіантів використання РС комбінованого типу

На діаграмі діяльності видно (рис. 8), що є три процеси – «Надати вподобання жанрів фільмів», «Оновити вподобання» та «Переглянути усі рекомендації». Також система матиме можливість робити також інші атомарні дії, як видалення нерелевантних рекомендацій, видалення та додавання користувачів системи тощо.

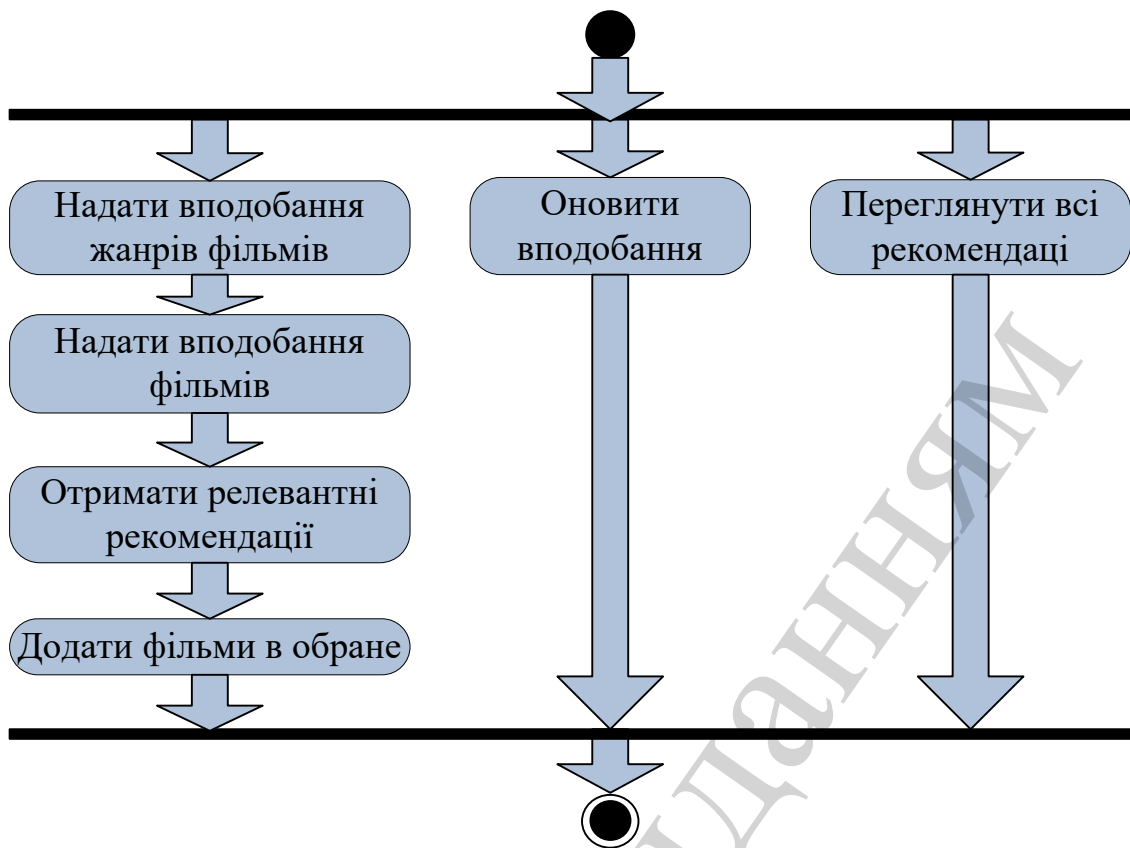


Рис. 8. Діаграма діяльності РС комбінованого типу

На діаграмі послідовності (рис. 9) 9 повідомлень взаємодіють з 3 об'єктами: користувачем, сервісом управління користувачами та генерування релевантних рекомендацій, базою даних фільмів та рейтингів.

З діаграми переходів станів (рис. 10) випливає, що якщо релевантні рекомендації уже були сформовані, то система одразу переходить у кінцевий стан. Якщо рекомендації все ще потрібно згенерувати, то система проходить 4 етапи: попередню обробку даних, колаборативну фільтрацію, гібридний метод генерування рекомендацій та передавання результатів користувачу. Саме етап застосування гібридного методу забезпечує на виході системи релевантні рекомендації фільмів для користувача, оскільки застосування лише колаборативної фільтрації не дає бажаного результату.



Рис. 9. Діаграма послідовності РС комбінованого типу

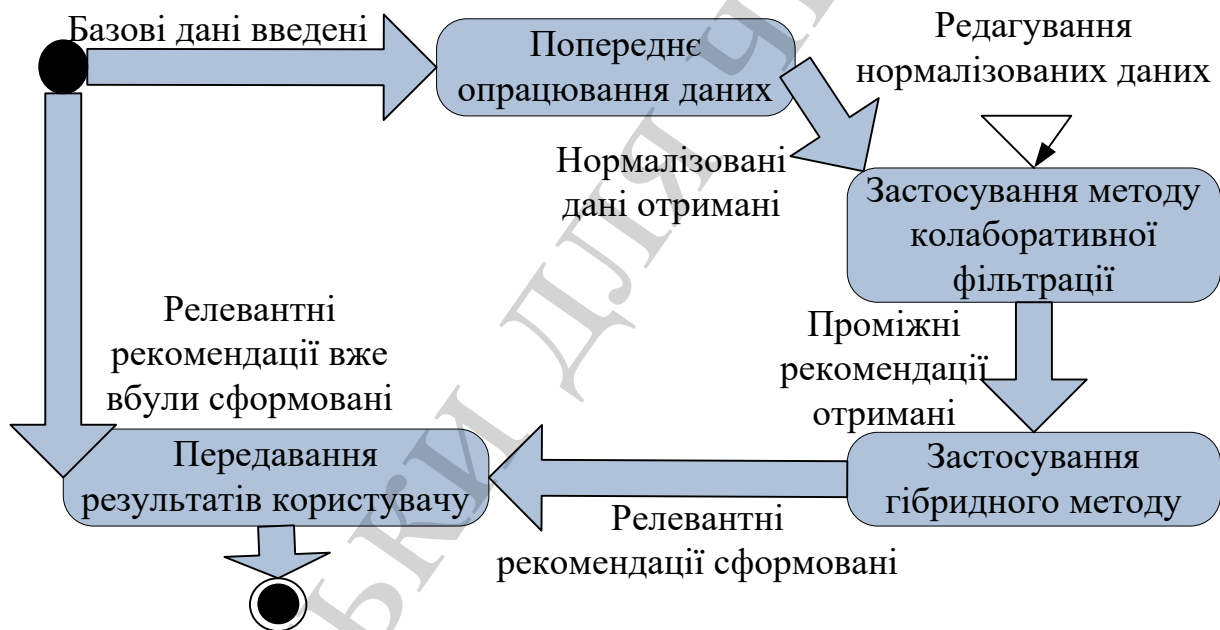


Рис. 10. Діаграма переходів станів РС комбінованого типу

Діаграма класів (рис. 11) складається із 6 сутностей як колоборативна фільтрація InvertedKNN, користувач User, надання рекомендацій RecommenderSystem, рейтинги та оцінки MovieRating, база даних MovieDB та список обраного Wishlist. Діаграма розгортання для розробленої системи зображена на рис. 13. Видно, що з апаратного забезпечення потрібно: браузер користувача, сервер управління користувачами, сервер генерування релевантних рекомендацій та сервер бази даних.

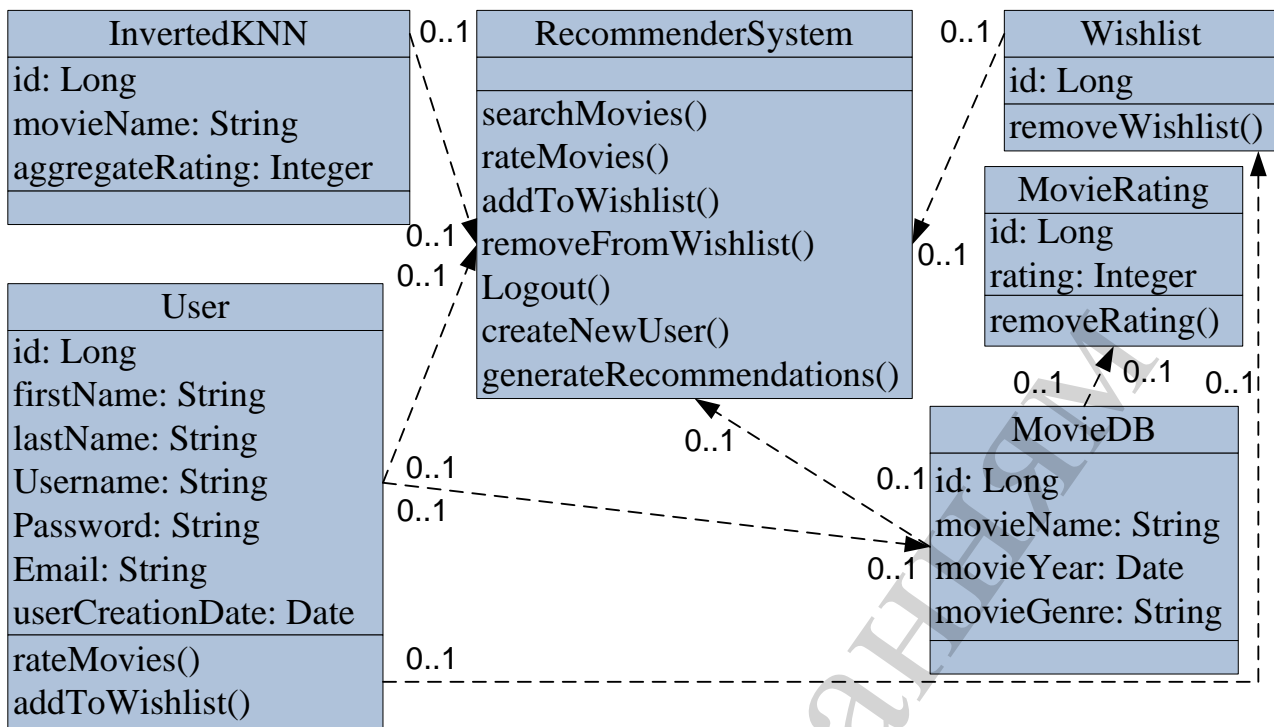


Рис. 11. Діаграма класів PC комбінованого типу

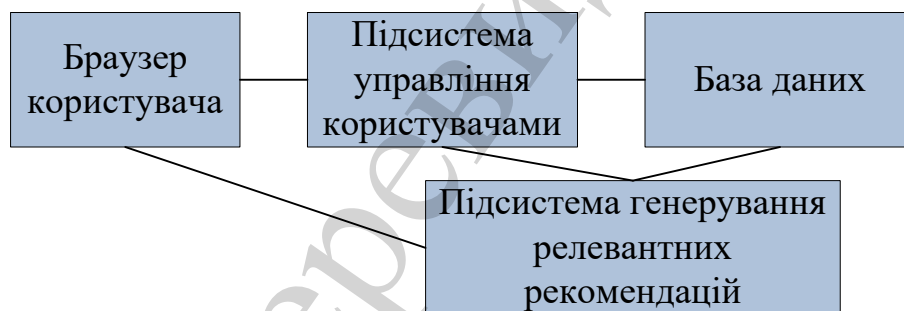


Рис. 12. Діаграма розгортання PC комбінованого типу

З причин безпеки системи, не рекомендується використовувати один сервер для опрацювання облікових записів користувачів та для генерування рекомендацій. Оскільки це різні завдання системи, необхідні окремі сервери.

5. Метод персоналізації контенту згідно потреб користувача.

Наявні системи надання рекомендацій фільмів, такі як IMDb, Netflix та Rotten Tomatoes, ефективно функціонують для індивідуального відстеження особистих потреб користувачів. Ці системи допомагають у підборі фільмів, надаючи ефективні рекомендації, які враховують потреби конкретного користувача, включаючи частоту перегляду фільмів, вподобання користувача у жанрах, акторах та часових проміжках. Для того, щоб система змогла підтримувати прийняття рішень для надання користувачам релевантних рекомендацій фільмів, необхідно об'єднати наявні алгоритми рекомендацій на основі особистих потреб користувача та побудувати гібридні, що і є метою розроблення системи. Що стосується двох типів проблем холодного старту, то проблема нового кори-

стувача вважається більш складною [10]. Відповідно, деякі дослідження запропонували гібридні методи, які в основному поєднують фільтрацію на основі контенту і спільну фільтрацію для посилення обмежень за допомогою алгоритмів [9]. Через зростання обсягу інформації та даних, виникають різноманітні проблеми у методах фільтрації, включаючи щільність і масштабованість. Новітні тенденції в Machine Learning використовують різні шари обробки, що допомагає в навчанні за допомогою складних контекстних особливостей. Machine Learning здатне визначати чинники, що впливають на підбір релевантних фільмів, що сприяє поліпшенню надання рекомендацій конкретному користувачу [10]. Для вирішення цих завдань пропонується новий удосконалений метод навчання, на відміну від наявних систем рекомендацій, в основі якого лежать гібридні методи, а базується на Machine Learning. Основним призначенням системи є надання релевантних рекомендацій фільмів з врахуванням особистих потреб користувача. Створену РС рекомендується використовувати для збору даних про вподобання різних людей у виборі фільмів та надання відповідних щоденних рекомендацій. Також систему можна легко адаптувати для рекомендацій музики, книг, криптовалют чи навіть біржових ринків. Гібридний алгоритм буде приймати вхідну інформацію у різному вигляді, нормалізувати її, та видавати відповідні рекомендації. Зважаючи на це, систему можна застосовувати як у повсякденному житті для вибору фільмів, так і у великих корпораціях у робочих процесах. Проектована система – це РС фільмів на основі гібридних алгоритмів Machine Learning, включаючи алгоритм User-based колаборативної фільтрації. Веб-додаток призначений для надання рекомендацій по вибору фільмів. Це веб-додаток, який є кросбраузерним та мультиплатформним. Його можна використовувати з будь-якого пристрою з браузером та доступом у Інтернет. NoSQL [11] використовується для зберігання даних, що допомагає інтегрувати контекстну інформацію в дві підсистеми – підсистему профілю користувача та підсистему контенту. Кожна техніка фокусується на різних аспектах за допомогою методів Machine Learning. Гібридні методи навчання використовують для персоналізованого рекомендатора і ґрунтуються на зворотному зв'язку. Є рекомендації, які можуть бути подані як оцінка відповіді нового користувача, вибір конкретних алгоритмів, механізм навчання та еволюція продуктивності [12]. Зміст РС підтримує запис рутинної діяльності та надає послуги з надання рекомендацій. Опрацювання вимог до даних здійснюється за допомогою передачі джерел даних у багатьох аспектах. Існує інтеграція даних, пов'язаних з контекстом користувачів і засобів рекомендацій. Опрацювання даних ґрунтується на алгоритмах Machine Learning з вчителем, які вивчають рефлексивні функції, що, в свою чергу, допомагає у прогнозуванні релевантності рекомендацій для кожного користувача. Проблема холодного старту виникає, коли профіль користувача не існує в РС, і немає попереднього рейтингу [9]. Концептуальна модель системи зображена на рис. 13.

Для вирішення цих проблем пропонується новий метод навчання, що базується на Machine Learning для вдосконалених рекомендацій, який ґрунтується на гібридних методах навчання для вдосконалення наявних систем рекомендацій з використанням спільної фільтрації. Основні дослідження зосереджені на

вирішенні проблем холодного старту та масштабованості в методі колаборативної фільтрації. Для ефективного вирішення цих проблем, досліджуються гібридні методи навчання. ГРС розроблена для дослідження, як комбінована фільтрація покращує вирішення важливих проблем холодного старту, використовуючи концепт профілю навчання в циклі. Тип відгуку користувачів покращує точність рекомендації та підвищує ефективність роботи системи на основі методів Machine Learning. Щодо задоволення вимог, методи гібридного навчання інтегровані з різними алгоритмами, включаючи:

- перемикання між фільтрацією на основі контенту та колаборативною фільтрацією для вирішення проблеми холодного старту нових користувачів;
- виявлення контексту користувача в межах динамічної фільтрації;
- інтегрування методів навчання на основі профілю для відображення відгуків користувачів.

Найважливішою частиною ГРС є профілі користувачів і релевантний контент. На рис. 14 подана моделі ГРС і визначені компоненти з рис. 13 [32]:

- браузер користувача – компонента «Досвід користувача та середовище»;
- підсистема управління користувачами – компонента «Інженіринг профайла користувача»;
- база даних – компонента «Сховище даних»;
- підсистема генерування релевантних рекомендацій – компонента «Гібридний метод навчання реоменування».

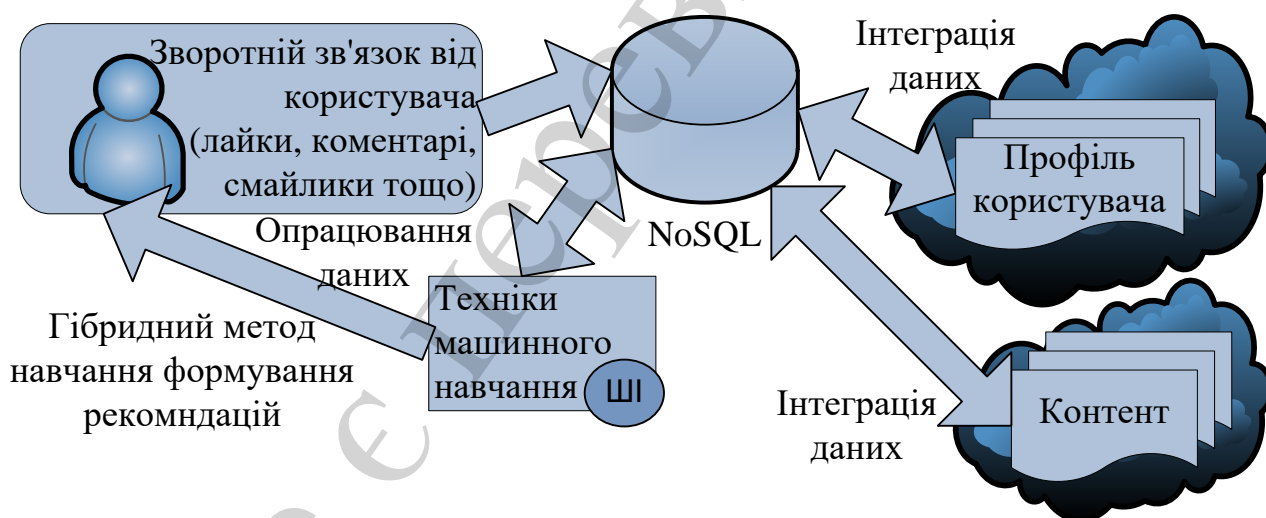


Рис. 13. Концептуальна модель РС комбінованого типу

Потоки службових даних ГРС строго регламентовані основними етапами функціонування системи з кроками 1–12. Попереднє опрацювання потоку даних користувача відбувається на етапах 1 та 3 відповідно (підкроки А–D). Обмін даними між основними компонентами ГРС описано та позначено # [32].

На першому етапі користувачі відповідають на просте опитування, щоб визначити свої вподобання та очікування у жанрах фільму. Ця інформація збережена в історії профілю користувача для фільтрації та створення початкового набору рекомендацій. Користувачі використовують адаптивний візуальний інтерфейс для

вираження особистих вподобань фільмів на основі аналізу cookies браузера. Визначені переваги використовуються для подальшої переоцінки представлених рекомендацій. Першою підсистемою моделі системи є збір даних, що включає в себе сукупність інформації, яка зазначено в кроках 1 і 2 із створенням профайлу при реєстрації <A>. На другому етапі на основі опрацювання потоку службових даних на кроках <3–5> [32] інформація передається та зберігається як попередньо опрацьований потік даних користувача, вказаний в <B–D> в алгоритмі профілю користувача. На третьому етапі – застосування гібридних методів навчання на кроках <6–10>, необхідних для попереднього опрацювання потоку даних контенту, згаданого в <#>, за допомогою Machine Learning (ML) для навчання та тестування набору даних та оцінки моделі навчання. Ця необхідно для підвищення точності гібридної фільтрації та зберегти результати фільтрації на основі *k*-найближчих сусідів для алгоритму класифікації. Для побудови списку вподобань користувача використовують метадані фільмів на кроках <11–12> [32].



Рис. 14. Сервісна модель РС комбінованого типу [32]

Запропонована архітектура системи подана на рис. 15. Методи навчання у функціях виведення ґрунтуються на алгоритмі k -найближчих сусідів для обробки переваг у рейтингах користувачів та ранжуванні релевантного вмісту [13]. На наступному етапі система надає користувачам релевантні рекомендації.



Рис. 15. Архітектура РС комбінованого типу [32]

На рис. 16 поданий процес функціонування ГРС з використанням техніки змішаного ML [32]. Користувач спочатку повинен створити профіль. Далі користувачу пропонують перелік фільмів, який базується на його оцінках. На наступному етапі система створює модель тренувальних даних, беручи вхідні дані зі спостережень. Далі у цьому процесі є фільтрування даних і вибір елементів, вподобані або оцінені користувачем. Цей же процес повторюється для створення подібної множини службового контенту. Це є змішаним ML, тому результати обох ітерацій об'єднуються. Процес гібридного ML продовжує перемикатися з однієї вибірки на іншу, щоб вирішити проблему холодного старту для нового користувача. Як подано на схемі, в роботі застосовується кероване ML.

Вхідними даними програми є база даних фільмів та рейтингів. Для того, щоб розробити інтерфейс веб-додатка дружнім, вхідні дані було спрощено до вибору трьох жанрів та трьох фільмів відповідних жанрів. Вихідними даними є список із десяти найбільш релевантних для користувача фільмів.

Створена ГРС розв'язує класи завдань, пов'язаних з прогнозуванням для рекомендацій фільмів, вибором кінофільмів, які будуть релевантними для кожного окремого користувача на основі його вподобань.

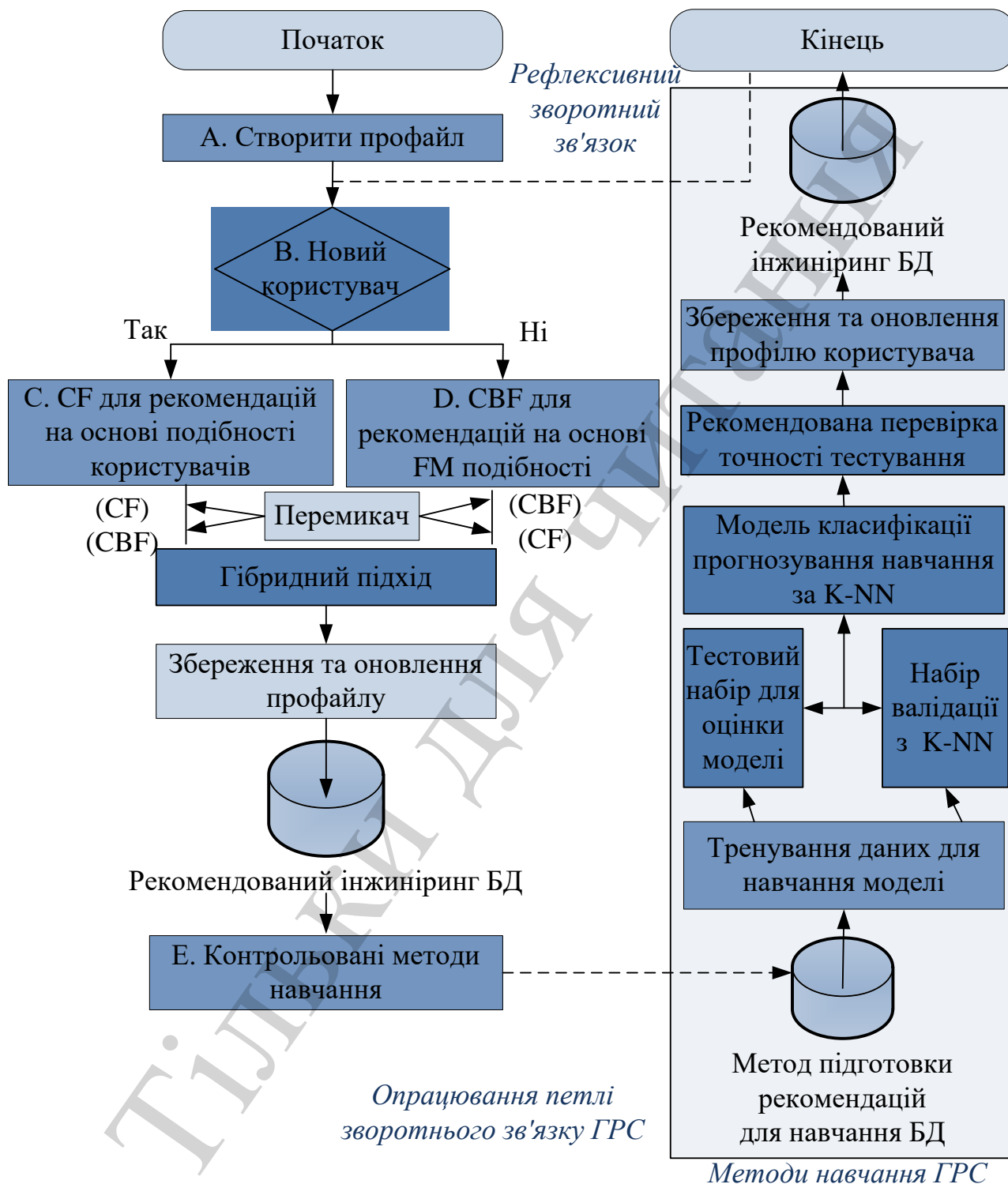


Рис. 16. Процес роботи РС комбінованого типу [32]

- У програмній системі для вирішення завдань застосовуються такі методи:
- User-based колаборативна фільтрація.
 - Item-based колаборативна фільтрація.

- Інтелектуальний аналіз даних.
- Machine Learning.
- Гібридні методи.

Програма пропонує релевантні рекомендації фільмів на основі вибору трьох улюблених жанрів та трьох улюблених фільмів. Використовуючи цю систему, користувач набагато пришвидшить пошуки нових фільмів для перегляду. Система розміщена в хмарі на серверах та працює в цілодобовому режимі. Адміністративна панель сервісу Shiny записує постійні логи роботи веб-додатка, які можна переглядати у будь-який момент. Якщо веб-додаток використовує забагато трафіку, що може спричинити його повільну роботу, то сервіс повідомляє про це розробнику чи адміністратору, використовуючи електронну пошту. Для стабільності роботи з веб-додатком рекомендується використовувати якісне та швидке Інтернет з'єднання, наприклад Wi-Fi чи 4G та пристрій з достатньою кількістю оперативної пам'яті. Додаток можна використовувати з будь-якого пристрою з браузером та доступом у мережу Інтернет. Периферійні пристрої не потрібні для роботи з системою. Для того, щоб розгорнути додаток на веб-серверах Shiny, потрібно мати локально встановлене середовище розробки мови R, таке як RStudio IDE [28].

6. Програмне забезпечення поширення комерційного контенту в Інтернет-просторі на основі колаборативної фільтрації та Machine Learning

У результаті виконання практичної частини роботи розроблено систему у вигляді веб-додатка «VikToFilm» – рекомендаційну систему фільмів на основі гібридних алгоритмів Machine Learning, включаючи алгоритм User-based колаборативної фільтрації. Серверна частина написана на мові програмування R, а клієнтська частина, тобто інтерфейс користувача, створена з використанням фреймворків Bootstrap та Shiny. Оскільки ця система є веб-додатком, розробленим на мові програмування R з використанням фреймворку Shiny, то програмним забезпеченням, необхідним для її функціонування, є веб-сервер shinyapps [29] з розгорнутим на ньому додатком. Служба Shiny працює в хмарі на серверах, які є власністю корпорації RStudio. Кожна програма є автономною і працює з будь-якими даними, завантаженими разом з додатком, або даними, які витягуються зі сховищ даних третіх сторін, таких як бази даних або веб-служби. Основним призначенням системи є надання релевантних рекомендацій фільмів з врахуванням особистих потреб користувача. Створену систему рекомендується використовувати для збору даних про вподобання різних людей у виборі фільмів та надання відповідних щоденних рекомендацій. Також систему можна легко адаптувати для рекомендацій музики, книг, криптовалют чи навіть біржових ринків. Гібридний алгоритм приймає вхідну інформацію у різному вигляді, нормалізує її, та видає відповідні рекомендації. Зважаючи на це, систему можна застосовувати як у повсякденному житті для вибору фільмів, так і у великих корпораціях у робочих процесах. Основні R-пакети, які використані під час розроблення системи, перелічені нижче: shiny версії 1.3.2; shinydashboard версії 0.7.1; knitr версії 1.22; markdown версії 0.9; proxy 0.4-17; recommenderlab 0.2-4; reshape2 1.4.3; ggplot2 3.1.1; ggvis 0.4.4; data.table 1.12.2;

DBI 1.0.0; openssl 1.3; htmltools 0.3.6; jsonlite 1.6; rconnect 0.8.13. Додаток побудований на основі гібридних алгоритмів Machine Learning, включаючи алгоритм User-based колаборативної фільтрації. Основною ідеєю системи CFR (колаборативної фільтрації) є те, що, якщо два користувачі мали однакові інтереси у минулому, наприклад, вподобали той же фільм, то ці користувачі будуть мати подібні смаки у майбутньому. Якщо, наприклад, користувач А і користувач Б мають подібну історію переглядів фільмів і користувач А нещодавно переглянув та високо оцінив фільм, який користувач Б ще не бачив, то цей фільм буде запропонований користувачу Б. Дані для опрацювання взято із *MovieLens* [30]. Тут фільмам наданий рейтинг за 5-бальною шкалою. Із-за неможливості мати доступ до потужних серверів для обрахунків у проекті використано найменший доступний варіант (*ml-latest-small.zip*), який на час завантаження містив 100000 рейтингів та 3600 ключових слів (тегів) для 9000 фільмів. Ці дані були створені 600 користувачами у період з січня 1995 року до вересня 2018 року [22]. Усі вибрані користувачі вже оцінили не менше, як 20 фільмів. Створена ГРС використовує файли *movies.csv* (табл. 4) і *ratings.csv* (табл. 5) з *MovieLens* для генерування рекомендацій.

Таблиця 4

Частина *movies.csv* з ідентифікаторами, назвами та жанрами фільмів

1	movieId,title,genres
2	1,Toy Story (1995),Adventure Animation Children Comedy Fantasy
3	2,Jumanji (1995),Adventure Children Fantasy
4	3,Grumpier Old Men (1995),Comedy Romance
5	4,Waiting to Exhale (1995),Comedy Drama Romance
6	5,Father of the Bride Part II (1995),Comedy
7	6,Heat (1995),Action Crime Thriller
8	7,Sabrina (1995),Comedy Romance
9	8,Tom and Huck (1995),Adventure Children
10	9,Sudden Death (1995),Action
11	10,Golden Eye (1995),Action Adventure Thriller
.....	

Короткий опис даних про фільми згідно вмісту файлу *movies.csv*:

> summary(movies)

movieID	title	genres
Min. : 1	Length : 9125	Length : 9125
1st Qu. : 2850	Class : character	Class : character
Median : 6290	Mode : character	Mode : character
Mean : 31123	3td Qu. : 56274	Max. : 164979

Таблиця 5

Частина вмісту файлу ratings.csv з унікальними ідентифікаторами користувачів та фільмів, рейтингами кінокартин від користувачів та мітками часу

1	userId,movieId,rating,timestamp
2	1,31,2.5,1260759144
3	1,1029,3.0,1260759179
4	1,1061,3.0,1260759182
5	1,1129,2.0,1260759185
6	1,1172,4.0,1260759205
7	1,1263,2.0,1260759151
8	1,1287,2.0,1260759187
9	1,1293,2.0,1260759148
10	1,1339,3.5,1260759125
.....	

Короткий опис даних рейтингів фільмів згідно вмісту файлу ratings.csv:

> summary(ratings)

userID	movieId	rating	timestamp
Min. : 1	Min. : 1	Min. : 0.500	Min. : 7.897e+08
1st Qu. : 182	1st Qu. : 1028	1st Qu. : 3.000	1st Qu. : 9.658e+08
Madian : 367	Madian : 2406	Madian : 4.000	Madian : 1.110e+09
Mean : 347	Mean : 12549	Mean : 3.544	Mean : 1.130e+09
3td Qu. : 520	3td Qu. : 5418	3td Qu. : 4.000	3td Qu. : 1.296e+09
Max. : 671	Max. : 163949	Max. : 5.000	Max. : 1.477e+09

Перед створенням системи рекомендації потрібно зробити попереднє опрацювання наявних даних. Перш за все було реорганізовано інформацію про жанри фільмів таким чином, щоб дозволити майбутнім користувачам шукати фільми, які їм подобаються в межах певних жанрів. З точки зору дружнього інтерфейсу, це набагато простіше для користувача в порівнянні з вибором фільму з одного списку всіх доступних фільмів [23]. У ході розроблення системи створено матрицю відповідних жанрів для кожного фільму. Наступний крок – це розроблення матриці пошуку, яка дозволяє без особливих труднощів здійснювати пошук фільму за будь-яким з його жанрів (табл. 6). Як бачимо, кожен фільм може відповідати ≥ 1 жанру.

Для того, щоб використовувати дані рейтингів для створення рекомендаційного двигуна з допомогою бібліотеки recommenderlab, перетворено матрицю рейтингів на розріджену матрицю типу *realRatingMatrix*:

671 x 9066 rating matrix of class 'realRatingMatrix' with 100004 ratings.

Пакет recommenderlab містить деякі параметри алгоритму рекомендацій:

```

> names(recommender_models)
[1] "ALS_realRatingMatrix" "ALS_implicit_realRatingMatrix"
"IBCF_realRatingMatrix"
[4] "POPULAR_realRatingMatrix" "RANDOM_realRatingMatrix"
"RERECOMMEND_realRatingMatrix"
[7] "SVD_realRatingMatrix" "SVDF_realRatingMatrix" "UBCF_realRatingMatrix"
> lapply(recommender_models, "[", "description")
$ALS_realRatingMatrix
[1] "Recommender for explicit ratings based on latent factors, calculated by alternating
least squares algorithm."
$ALS_implicit_realRatingMatrix
[1] "Recommender for implicit data based on latent factors, calculated by alternating
least squares algorithm."
$IBCF_realRatingMatrix
[1] "Recommender based on item-based collaborative filtering."
$POPULAR_realRatingMatrix
[1] "Recommender based on item popularity."
RANDOM_realRatingMatrix
[1] "Produce random recommendations (real ratings)."
RERECOMMEND_realRatingMatrix
[1] "Re-recommends highly rated items (real ratings)."
SVD_realRatingMatrix
[1] "Recommender based on SVD approximation with column-mean imputation."
SVDF_realRatingMatrix
[1] "Recommender based on Funk SVD with gradient descend."
UBCF_realRatingMatrix
[1] "Recommender based on user-based collaborative filtering."

```

Таблиця 6
Частина матриці пошуку за жанрами фільмів

N	movie Id	title	Action	Adventure	Animation	Children	Comedy	Crime	Documentary	...
1	1	Toy Story (1995)	0	1	1	1	1	0	0	...
2	2	Jumanji (1995)	0	1	0	1	0	0	0	...
3	3	Grumpier Old Men (1995)	0	0	0	0	1	0	0	...
4	4	Waiting to Exhale (1995)	0	0	0	0	1	0	0	...
5	5	Father of the Bride Part II (1995)	0	0	0	0	1	0	0	...
6	6	Heat (1995)	1	0	0	0	0	1	0	...
.....										

У ході розроблення системи прийнято рішення використовувати моделі IBCF та UBCF [25], які мають наступні параметри:

```
> recommender_models$IBCF_realRatingMatrix$parameters
$k [1] 30
$method [1] "cosine"
$normalize [1] "center"
$normalize_sim_matrix [1] FALSE
$alpha [1] 0.5
$na_as_zero [1] FALSE
> recommender_models$UBCF_realRatingMatrix$parameters
$method [1] "cosine"
$nn [1] 25
$sample [1] FALSE
$normalize [1] "center"
```

7. Результати досліджень реалізації системи поширення комерційного контенту в Інтернет-прострі згідно потреб користувач

Веб-додаток «VikToFilm» – це рекомендаційна система фільмів на основі гібридних алгоритмів Machine Learning, включаючи алгоритм User-based колаборативної фільтрації. Інтерфейс користувача створений за допомогою фреймворка Shiny (рис. 17):

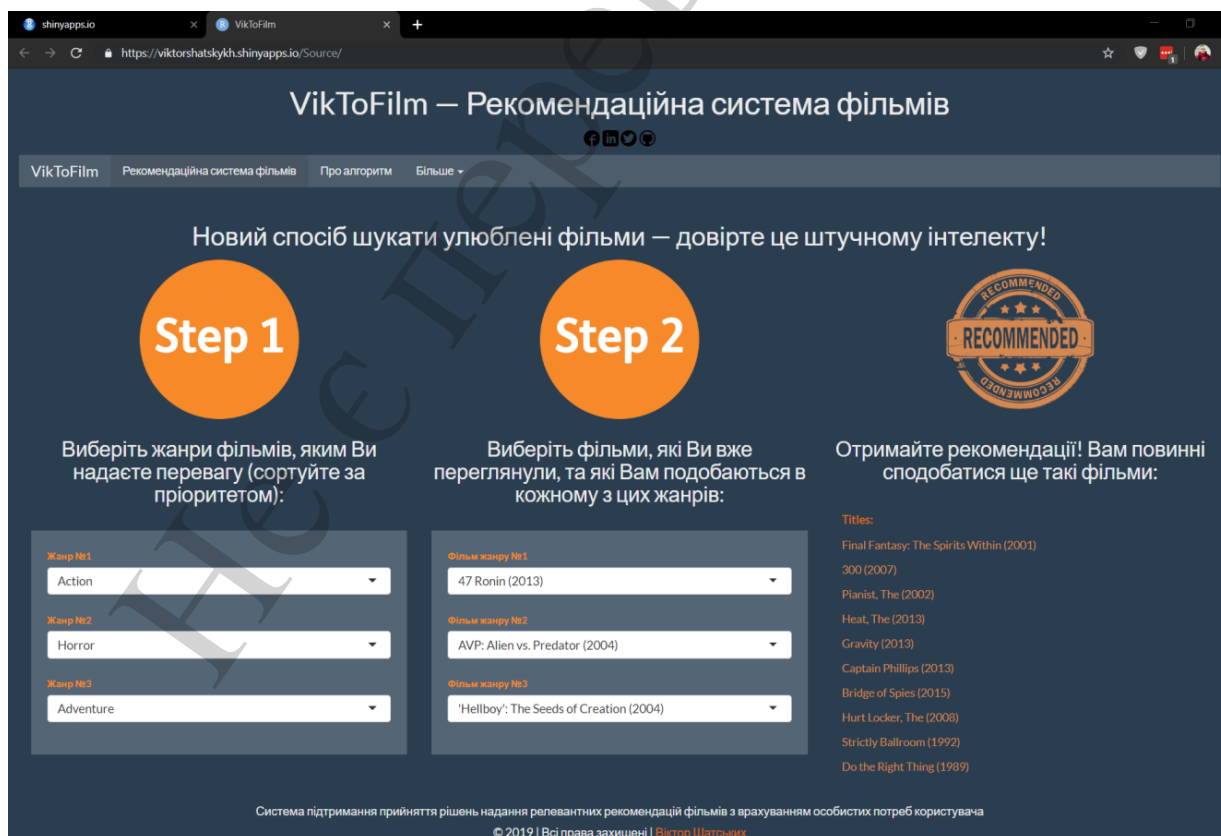


Рис. 17. Веб-додаток

Під час розробки системи використані наступні середовища та мови програмування:

- мова програмування R версії 3.3.3;
- середовище RStudio IDE версії 1.2.1335;
- фреймворки Shiny версії 1.3.2, HTML5/CSS3 та Bootstrap версії 4.3.1;
- бібліотеки jQuery версії 3.3.1 та SockJS;
- операційна система Windows 10;
- веб-переглядач Google Chrome версії 74.0.3729.131.

Основним призначенням системи є надання релевантних рекомендацій фільмів з врахуванням особистих потреб користувача. Створену систему рекомендується використовувати для збору даних про вподобання різних людей у виборі фільмів та надання відповідних щоденних рекомендацій.

Також систему можна легко адаптувати для рекомендацій музики, книг, криптовалют чи навіть біржових ринків. Гібридний алгоритм буде приймати вхідну інформацію у різному вигляді, нормалізувати її, та видавати відповідні рекомендації. Зважаючи на це, систему можна застосовувати як у повсякденному житті для вибору фільмів, так і у великих корпораціях у робочих процесах. Оскільки цей веб-додаток є кросбраузерний та мультиплатформний, то його можна використовувати з будь-якого пристрою, який містить браузер та має доступ до Інтернет (рис. 18).

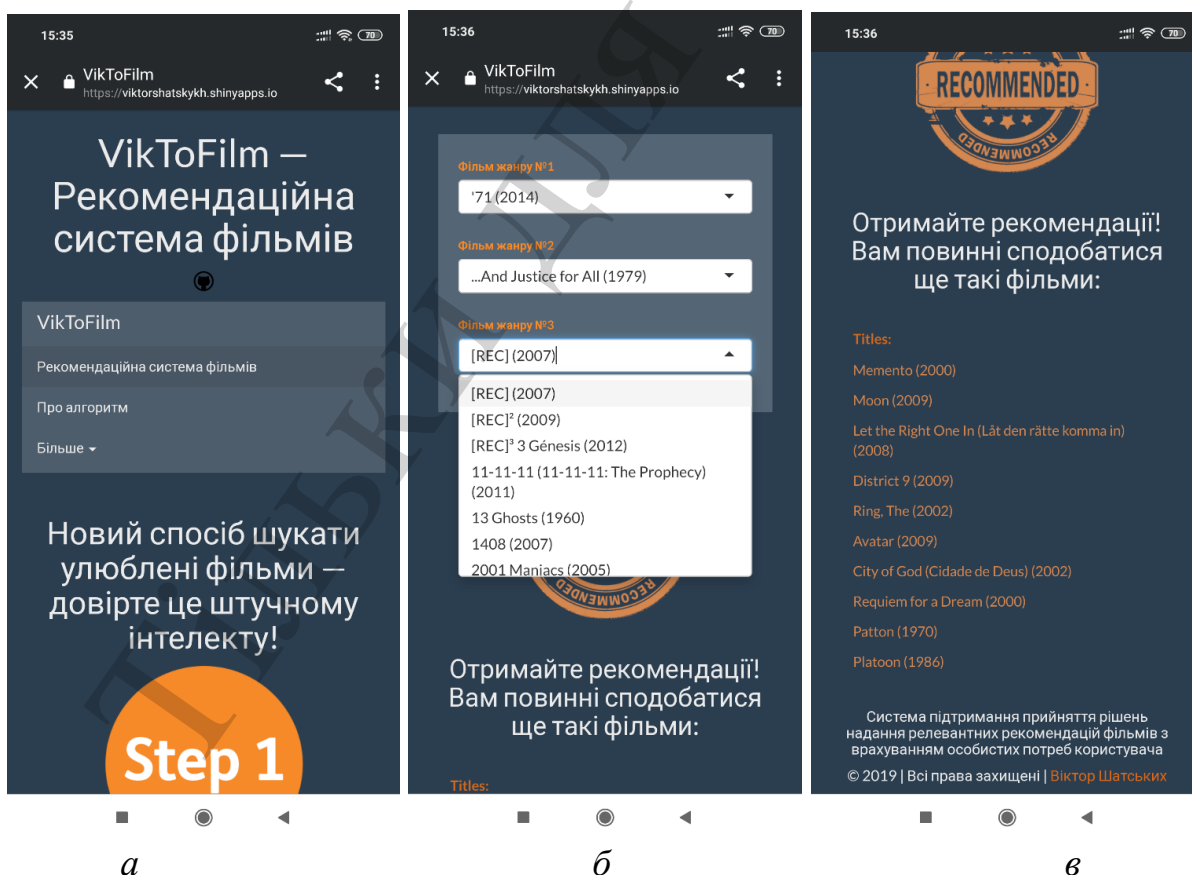


Рис. 18. Веб-додаток на Android: *a* – меню; *б* – жанри; *в* – рекомендації

Отримати доступ до додатка можна з будь-якого пристрою з браузером та доступом у мережу Інтернет. Вхідною точкою у веб-додаток є його головна сторінка (<https://viktorshatskykh.shinyapps.io/Source/>). Вхідними даними програми є база даних фільмів та рейтингів, яка міститься у файлах `movies.csv` і `ratings.csv`. Для того, щоб розробити інтерфейс веб-додатка дружнім, вхідні дані було спрощено до вибору трьох жанрів та трьох фільмів відповідних жанрів. Вихідними даними є список із десяти найбільш релевантних для користувача фільмів. Щоб підтвердити працездатність розробленого веб-додатка та відповідність результатів функціонування системи поставленій задачі, був проведений контрольний приклад (рис. 19, 20). Користувач, зайшовши на веб-сторінку додатку (<https://viktorshatskykh.shinyapps.io/Source/>) у своєму улюбленому веб-переглядачі (у цьому тесті використано веб-переглядач Google Chrome версії 74.0), бачить перед собою головну сторінку веб-додатку. Після обрання трьох улюблених жанрів та трьох улюблених фільмів, система генерує йому релевантні рекомендації. Цими рекомендаціями є 10 назв фільмів та роки, коли ці фільми були зняті та випущені на екрани світу.

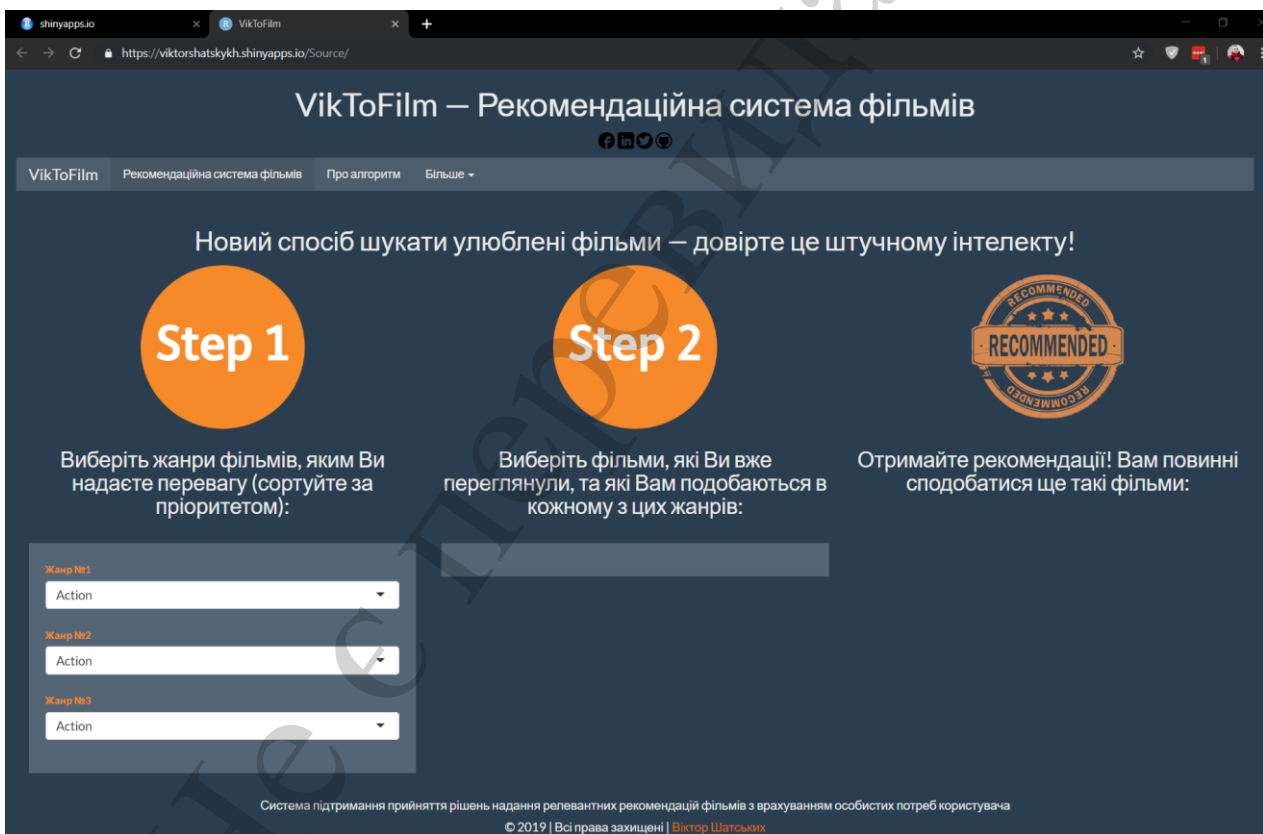


Рис. 19. Початок роботи

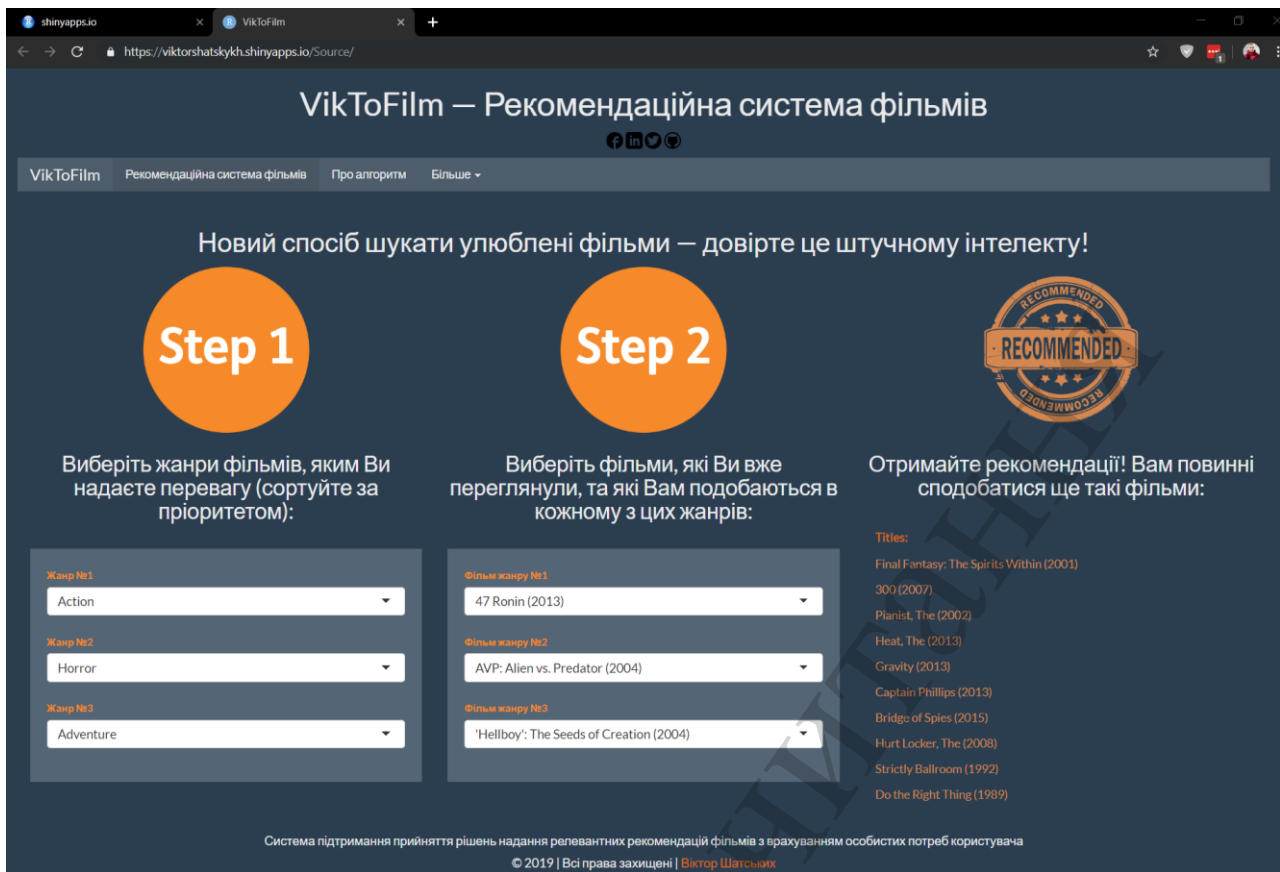


Рис. 20. Отримані рекомендації

Як видно з рис. 21, система працює та надає доволі релевантні рекомендації по вибору фільмів. Якщо користувач хоче отримати інші рекомендації, йому достатньо змінити один з трьох вибраних фільмів (є можливість змінити вибір на будь-якому етапі роботи з системою).

8. Обговорення результатів досліджень автоматичного поширення комерційного контенту на основі SEO-технологій та Machine Learning

Алгоритми спільної фільтрації засновані на мірі подібності між користувачами або між елементами. Для цього *recommenderlab* містить функцію подібності. Підтримуються наступні методи для обчислення подібності:

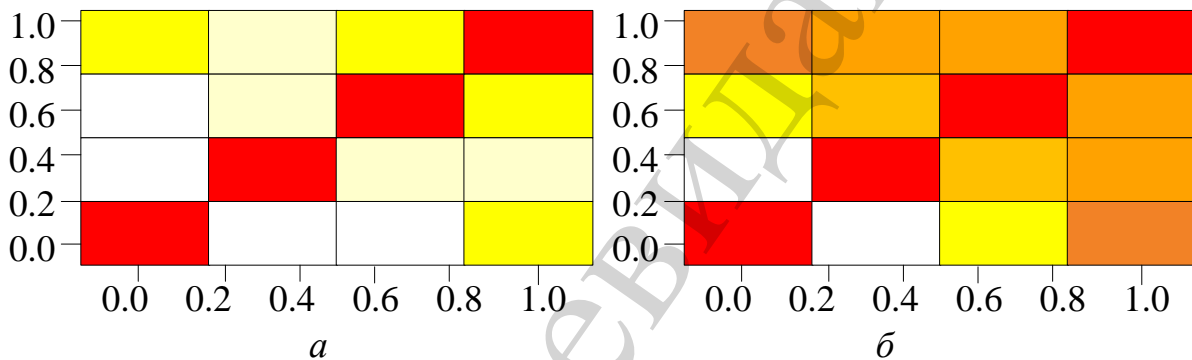
- косинусний коефіцієнт (Cosine Similarity) – cosine;
- коефіцієнт кореляції Пірсона (Pearson correlation coefficient) – pearson;
- коефіцієнт Жаккара (Jaccard index) – jaccard.

Щоб з'ясувати наскільки перші чотири користувачі/фільми подібні один з одним створено і візуалізовано матрицю подібності, яка використовує косинусний коефіцієнт (табл. 7, рис. 21). У матриці кожен рядок і кожен стовпець відповідає користувачеві/фільму, а кожна клітинка відповідає подібності між двома користувачами/фільмами. Чим червоніша клітинка, тим більше подібні вподобання двох користувачів (зміст двох фільмів). Варто звернути увагу на те, що діагональ – червоного кольору, містить порівняння кожного користувача/фільму з самим собою.

Таблиця 7

Матриці подібності користувачів та фільмів на основі косинусного коефіцієнту

Матриця подібності користувачів				
	1	2	3	4
1	0.00000000	NA	NA	0.07448245
2	NA	0.000000	0.12429498	0.11882103
3	NA	0.124295	0.00000000	0.08163991
4	0.07448245	0.118821	0.08163991	0.00000000
Матриця подібності фільмів				
	1	2	3	4
1	0.00000000	0.3945115	0.365159	0.1336141
2	0.3945115	0.00000000	0.2174915	0.1646513
3	0.3065159	0.2174915	0.00000000	0.1770118
4	0.1336141	0.1646513	0.1770118	0.00000000

Рис. 21. Подібність : a – користувачів; b – фільмів згідно з MovieLens

Після візуалізації матриці подібності перших чотирьох фільмів досліджено вагомість рейтингів:

```
> vector_ratings <- as.vector(ratingmat@data)
> unique(vector_ratings) # які значення рейтингів – унікальні
[1] 0.0 3.0 4.0 5.0 2.0 3.5 1.0 2.5 4.5 1.5 0.5
> table_ratings <- table(vector_ratings) # яка кількість кожного з рейтингів
> table_ratings
vector_ratings
0      0.5    1     1.5    2     2.5    3     3.5    4     4.5    5
5983282 1101 3326 1687 7271 4449 20064 10538 28750 7723 15095
```

Загалом існують 11 унікальних значень оцінок рейтингу. Більш низькі значення означають нижчі рейтинги і навпаки. Якщо не враховувати рейтинг 0 (частина фільмів не оцінена користувачами), із оцінених фільмів найчастіше рейтинг є 4 (28750), а найменш вживаний рейтинг є 0.5 (1101). Згідно з документацією, якщо рейтинг дорівнює 0, то це означає відсутність оцінки, тому вилу-

чимо з набору даних перед візуалізацією результатів (рис. 22). Як показують дані, є менш низькі (менше 3-ох) рейтингові оцінки, проте, більшість фільмів оцінені балом 3-и або вище. Найбільш поширеним є рейтинг 4-ри бали.

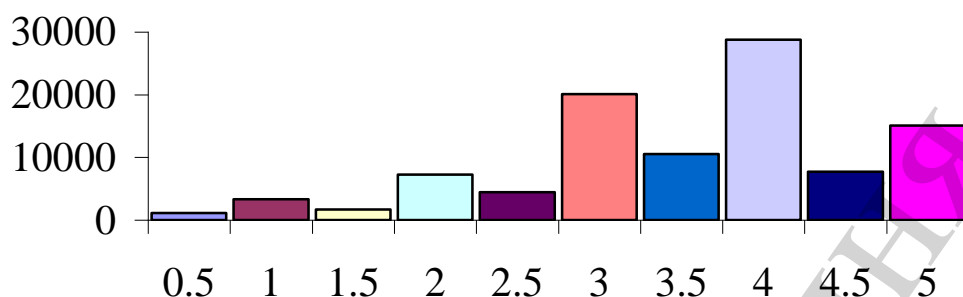


Рис. 22. Розподіл рейтингів фільмів згідно з MovieLens (ml-latest-small.zip)

Розглянемо, які фільми переглянуті користувачами найбільшу кількість разів (табл. 8) згідно з MovieLens (ml-latest-small.zip). Отриманий результат пояснюється тим, що дані створені 600 користувачами у період з січня 1995 року до вересня 2018 року та на момент тестування не оновлені. Тому найпопулярніші фільми після 2015 року ще не брали відповідного рейтингу в MovieLens. Статистичні дані ще обмежені малою кількістю користувачів та їх регіональністю. Але навіть на такій маленькій вибірці можна побачити, що фундаментальні фільми Forrest Gump (1994) Shawshank Redemption, The (1994), які змінюють наше суспільство, увійши у перші позиції популярних фільмів із 9000 фільмів у маленької аудиторії із 600 людей серед 100000 рейтингів (біля 170 на людину) та 3600 ключових слів. І скорше всього навіть з оновленою вибіркою не гарантовано що на перших позиціях, але такі фільми всетаки будуть входити в список популярних фільмів для євро-американського суспільства. На інших теренах (Азії, Африки тощо) користувачі скорше всього оберуть зовсім інші фільми, а перелічені в табл. 8 навіть можуть не потрапити в список рейтингова них фільмів. В межах статті не провадилось демографічне та регіональне сатистичне дослідження вподобань користувачів.

Таблиця 8

Найпопулярніші фільми згідно з MovieLens (ml-latest-small.zip)

№	movie	views	title
1	356	341	Forrest Gump (1994)
2	296	324	Pulp Fiction (1994)
3	318	311	Shawshank Redemption, The (1994)
4	593	304	Silence of the Lambs, The (1991)
5	260	291	Star Wars: Episode IV – A New Hope (1977)
6	480	274	Jurassic Park (1993)

Після створення рейтингу найбільшої кількості переглядів визначено фільми з найкращим рейтингом і обчислено середню оцінку для кожного з них (рис. 24). Рис. 24, *a* показує розподіл середнього рейтингу фільмів. Найбільше значення становить близько 3, і є кілька фільмів, чий рейтинг дорівнює 1 або 5. Можливо, що ці фільми отримали рейтинг усього від кількох людей, а отже, не треба брати їх до уваги. Після видалення фільмів, які мають кількість переглядів 50 та нижче, створено підмножину тільки релевантних фільмів. Рис. 23, *б* показує розподіл відповідних релевантних середніх оцінок. Усі рейтинги між 2.16 і 4.55. Як і очікувалося, екстремуми видалені. Найбільше значення рейтингу змінилося, і тепер становить близько 4.

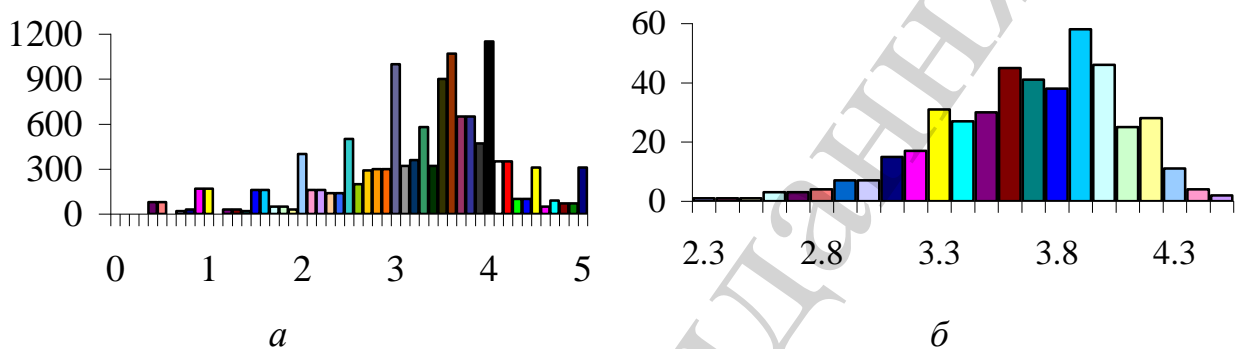


Рис. 23. Розподіл згідно з MovieLens (ml-latest-small.zip) : *a* – середнього рейтингу фільмів; *б* – середнього релевантного рейтингу фільмів

Графічно подамо матрицю оцінок шляхом побудови теплокарти (heatmap) матриці рейтингів, колір якої є рейтинги. Теплокарта – графічне подання даних, при якому окремі значення що містяться в матриці подамо у вигляді кольорів. Кожен рядок матриці відповідає користувачеві, а кожен стовпець – фільму, а відтак, кожна клітинка на перетині – відповідному рейтингу (рис. 24). Оскільки наведена чимала кількість користувачів і елементів, прочитання графіка з усіма 600 користувачами та 9000 фільмами є ускладненим. Тому для прикладу побудований графік у збільшеному масштабі і акцентує увагу на перших даних.

Деякі користувачі переглядали більше фільмів, ніж інші. Таким чином, замість відображення будь-яких випадкових користувачів і фільмів, оберемо найбільш релевантних користувачів та фільми. Для того, щоб виявити і обрати найбільш релевантних користувачів та фільми, виконаємо наступні дії (рис. 25):

- Визначити мінімальну кількість фільмів для кожного користувача.
- Визначити мінімальну кількість користувачів для кожного фільму.
- Обрати користувачів і фільми, які відповідають цим критеріям.

[1] “Мінімальна кількість фільмів для кожного користувача:”

99% 1131.4

[1] “Мінімальна кількість користувачів для кожного фільму:”

99% 123

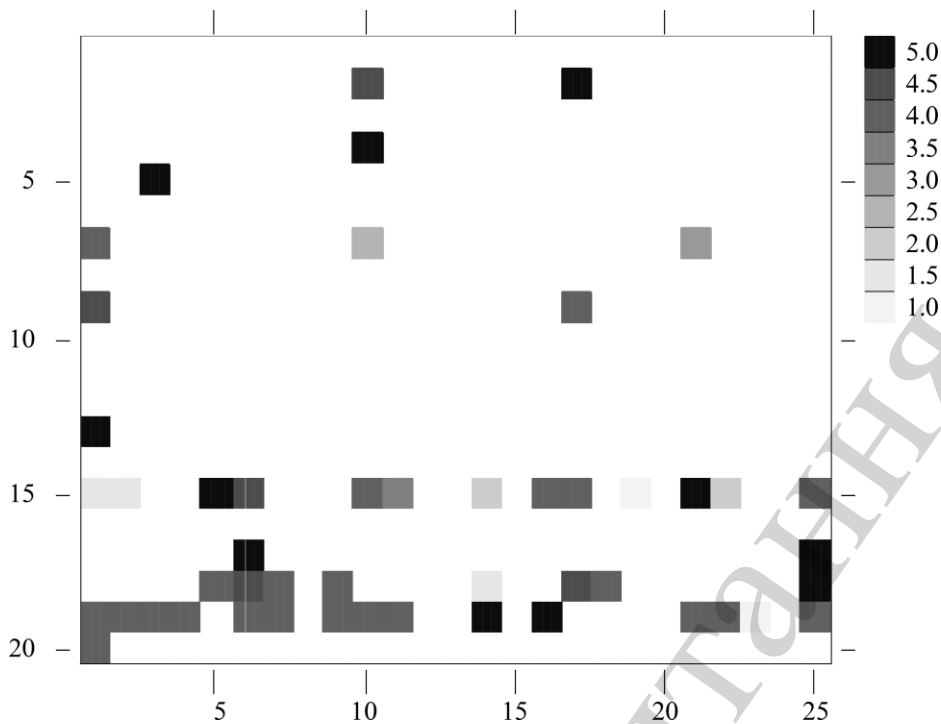


Рис. 24. Теплокарта рейтингів перших 20 користувачів та перших 25 фільмів згідно з MovieLens (ml-latest-small.zip)

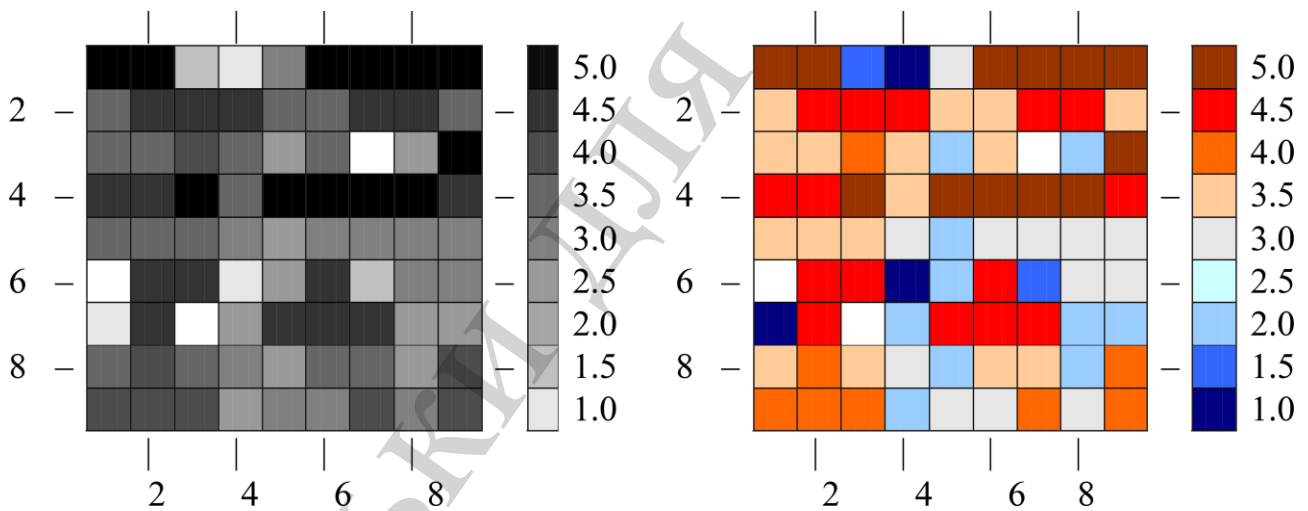


Рис. 25. Теплокарта найкращих користувачів та фільмів згідно з MovieLens

Варто взяти до уваги користувачів, які бачили найбільше фільмів. Більшість з них переглянули всі найкращі фільми. Деякі стовпці теплової карти темніші, ніж інші, а це означає, що ці стовпці зображають найвищий рейтинг фільмів. І навпаки, темні ряди зображають користувачів, що дають найбільш високі оцінки. Тому доречним є нормалізування даних.

Процес підготовки даних складається із наступних кроків [25]:

- обрати релевантні дані;
- нормалізувати дані;
- перетворити дані в двійковий формат.

Для обрання найбільш релевантних даних визначена мінімальна кількість користувачів/фільмів, які оцінили фільм /оцінені як 5.0:

421×444 rating matrix of class 'realRatingMatrix' with 37915 ratings.

Така вибірка найбільш релевантних даних містить 421-го користувача та 444-ри фільми (у порівнянні з попередніми 671-им користувачем та 9066-ма фільмами в загальному наборі даних). Використовуючи той же підхід, що і раніше, візуалізовано топ 2 % користувачів і фільмів у новій матриці найбільш актуальних даних (рис. 26). На тепловій карті деякі рядки темніші, ніж інші, так як окремі користувачі дають вищі рейтинги по всіх фільмах (рис. 26).

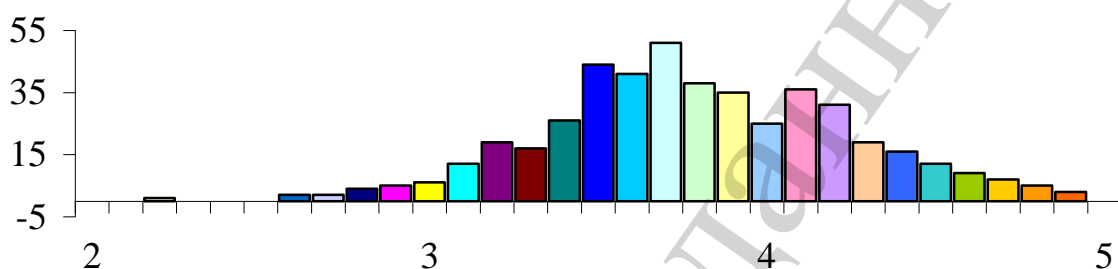


Рис. 26. Розподіл середнього рейтингу для кожного активного користувача згідно з MovieLens (ml-latest-small.zip)

Користувачі, які дають лише високі (або лише низькі) рейтинги всім переглянутим фільмам можуть зіпсувати результати. Щоб уникнути цього, дані було нормалізовано таким чином, щоб середня оцінка кожного користувача відповідала 0. Для швидкої перевірки було розраховано середню оцінку кожного користувача, яка дорівнює 0 (нормалізація даних):

```
> ratings_movies_norm <- normalize(ratings_movies)
> sum(rowMeans(ratings_movies_norm) > 0.00001)
[1] 0
```

Деякі моделі рекомендацій працюють на двійкових даних, тому буде корисно перетворити дані в двійковий формат, тобто визначити таблицю, яка містить тільки 0 та 1. Пропущені значення або погані оцінки можна вважати за 0. У цьому випадку доцільно:

- Визначити матрицю, що має значення або 1, якщо користувач оцінив фільм, або 0, в іншому випадку. В останньому випадку інформація про рейтинги втрачається.

- Визначити матрицю, що має 1, якщо оцінка вища або дорівнює певному пороговому значенню (наприклад, 3), і 0 в іншому випадку. У цьому випадку, дати поганий рейтинг фільму означає взагалі не оцінити його.

Доцільність вибору того чи іншого методу залежить від контексту. Використовуючи два різних підходи, визначено дві матриці, візуалізовано 5-

відсоткову частину кожної з бінарних матриць (рис. 27). У першому варіанті визначено матрицю, клітинки якої рівні 1, якщо фільм переглянутий (рис. 27, *a*). У другому варіанті визначено матрицю, клітинки якої рівні 1, якщо клітинка має рейтинг вище порогового значення (рис. 27, *б*). На другій теплокарті більше білих клітинок, ніж на першій, це означає, що є більше фільмів з відсутнім або поганим рейтингом, ніж тих, яких не бачили користувачі.

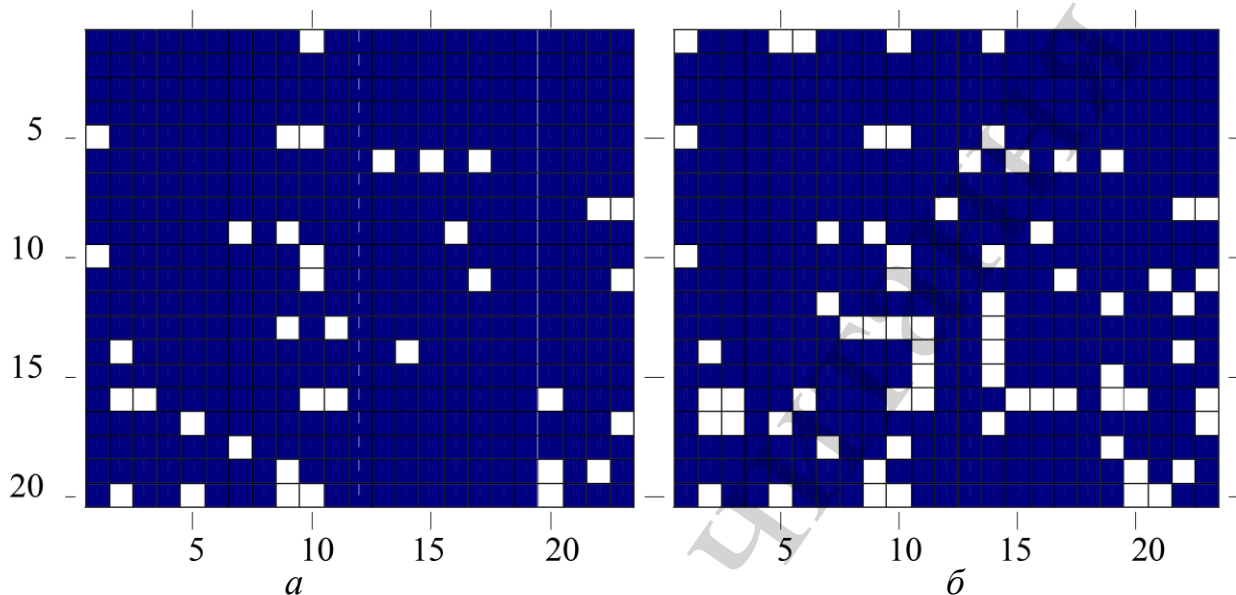


Рис. 27. Теплокарта згідно з MovieLens (ml-latest-small.zip) : *a* – 1, якщо фільм переглянутий; *б* – 1, клітинка має рейтинг вище порогового значення

Відповідно до цього підходу, новому користувачеві шукають подібних на нього користувачів. А фільми, які ці користувачі оцінили високим рейтингом, рекомендуються новому користувачу [20].

Для кожного нового користувача виконуються наступні кроки:

1. Визначення міри подібності кожного користувача-сусіда до нового. Як і в IBCF, застосовують косинусний коефіцієнт або коефіцієнт кореляції Пірсона.

2. Визначення найбільш схожих користувачів. Варіанти:

– Використовуючи топ k користувачів (k найближчих сусідів).

– Використовуючи користувачів, у яких схожість вища за певне порогове значення.

3. Оцінка фільмів згідно з рейтингом, який поставили найбільш схожі користувачі. Кінцевий рейтинг є середнім рейтингом серед подібних користувачів. Використовують такі підходи:

– Середній рейтинг.

– Середньозважений рейтинг, використовує подібність в якості ваг.

4. Отримання найбільш релевантних фільмів.

Спочатку варто перевірити параметри за замовчуванням моделі UBCF. Тут np – це число подібних користувачів [24]. Функцією подібності за замовчуванням використовується косинусний коефіцієнт – cosine. А відтак, варто збудува-

ти модель з використанням стандартних параметрів та з використанням навчального набору даних:

```

$method
[1] "cosine"
$nn
[1] 25
$sample
[1] FALSE
$normalize
[1] "center"
Recommender of type 'UBCF' for 'realRatingMatrix'
Learned using 330 users.
330 x 444 rating matrix of class 'realRatingMatrix' with 29917 ratings.
Normalized using center on rows.

```

Визначено десять найкращих рекомендацій для кожного нового користувача: Recommendations as 'topNList' with n=10 for 91 users.

Матриця з табл. 9 містить movieId кожного рекомендованого фільму (рядки) для перших чотирьох користувачів (стовпці) всього набору даних () синим колборм позначено найбільш релевантні фільми за UBCF).

Таблиця 9

Матриця з рекомендаціями для перших чотирьох користувачів (UBCF)

	[,1]	[,2]	[,3]	[,4]
[1,]	1089	608	858	1090
[2,]	1206	318	260	595
[3,]	293	527	318	1080
[4,]	858	47	899	2804
[5,]	47	593	50	1278
[6,]	1732	2959	246	1674
[7,]	1213	50	111	2529
[8,]	6016	2571	750	596
[9,]	924	223	1198	661
[10,]	111	923	923	1285

Також вираховано, скільки разів рекомендується кожен фільм і побудовано відповідну гістограму частот (рис. 28):

У порівнянні з IBCF, розподіл має більш довгий хвіст. Це означає, що деякі фільми рекомендуються набагато частіше, ніж інші [26]. Максимум становить понад 30, у порівнянні з 10 для IBCF (табл. 10).

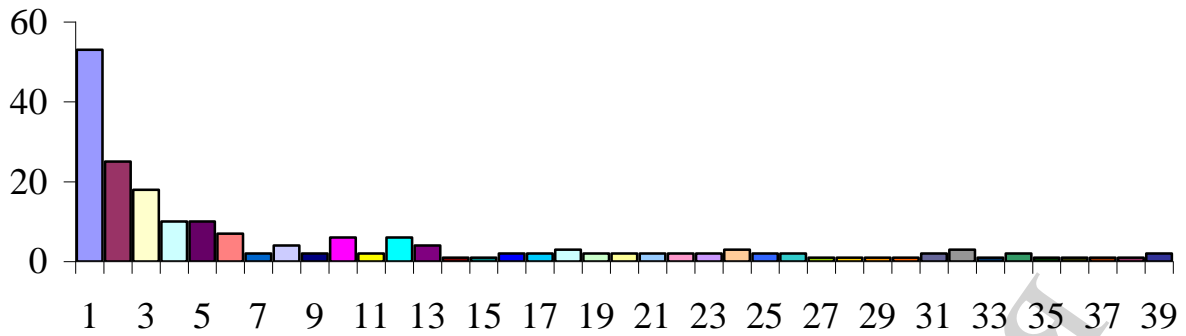


Рис. 28. Розподіл кількості фільмів для UBCF згідно з MovieLens

Таблиця 10

Найбільш релевантні фільми (UBCF) згідно з MovieLens (ml-latest-small.zip)

	Назва фільму	Кількість елементів
50	Usual Suspects, The (1995)	39
858	Godfather, The (1972)	34
318	Shawshank Redemption, The (1994)	32
527	Schindler's List (1993)	32

UBCF повинен мати доступ до початкових даних і тримати весь набір інформації в пам'яті, що часто ускладнює роботу при великій матриці рейтингів. Крім того, побудова матриці подібності вимагає багато обчислювальної потужності, а також часу. Однак точність UBCF є кращою, ніж IBCF, а UBCF є хорошим вибором, якщо набір даних є не дуже великим [21].

В залежності від аналізу історії конкретного користувача (кількості переглянутих фільмів, його оцінок для конкретних фільмів, кількість переглянутих фільмів конкретного жанру, кількість обраних жанрів тощо) можна скорегувати навчання системи для формування рекомендацій цьому користувачу. Також можна вивести критерії для оцінювання методів для їх подальшого порівняння. Візьмемо самий поширений параметр – кількість переглянутих фільмів з їх подальшим оцінюванням від користувача. Логічно – чим більше переглянув користувач фільмів і чим більше поставив оцінок цих фільмам – тим краще в подальшому ГРС має сформував релевантний список фільмів для користувачів. Але кожний попередньо розглянутий метод має ряд недоліків, які впливають на оцінювання релевантності фільмів в рекомендованому списку для конкретного користувача. Оберемо три однакові за кількістю групи користувачів A–C, які відповідно переглянули та виставили відповідні оцінки для фільмів. В групу A входять ті, що переглянули до 100 фільмів та виставили відповідні оцінки, в групу B – між 100 та 150 включно, а в групу C – понад 150 фільмів (рис. 29). Експериментальні результати показують, що показники роботи запропонованого ГРС на основі технології CF+CBF+ML кращі, ніж у двох одиничних моделей, CF та CBF, та їх комбінацій як CF+CBF, CF+ML та CBF+ML.

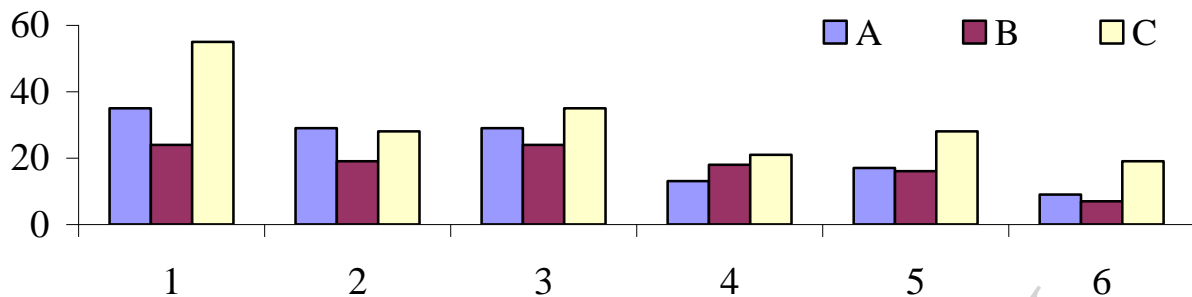


Рис. 29. Порівняльна характеристика у відсотках зростання помилки в рекомендаціях при використанні відповідних методів, де 1 – CF, 2 – CBF, 3 – CF+CBF, 4 – CF+ML, 5 – CBF+ML, 6 – CF+CBF+ML

Нажаль порівняти результати хоча б з одною із відомих систем неможливо, бо це комерційні проекти, і ніхто не дасть доступ ні до статистики функціонування системи, ні до результатів застосування методів формування рекомендацій та ML в цих системах. Але опишемо ці системи з боку користувача на прикладі найвідоміших як IMDb, Netflix та Rotten Tomatoes.

1. IMDb робить персональні рекомендації, які допомагають користувачам знайти нові фільми та телепередачі. Рекомендовані фільми відображаються у розділі «Recommended For You» на домашній сторінці IMDb [14]. IMDb приймає всі фільми та телепередачі, які користувач або оцінив або додав до свого списку спостереження, а потім порівнює ці дані з рейтингами інших користувачів IMDb. Потім IMDb може знайти фільми і телепередачі у людей з подібними смаками. Для кожної рекомендації користувач може побачити список фільмів або телепередач, на яких базується ця рекомендація [14]. Коли користувач дає фільму позитивну оцінку або додає фільм у свій список спостереження, IMDb відстежує це як фільм, який цікавить користувача. Якщо у IMDb немає жодних рекомендацій для користувача (або через те, що користувач вийшов з системи, або тому, що не оцінив багато елементів), IMDb покаже користувачу список елементів, які багато людей бачили, щоб користувач міг вводити рейтинги для отримання рекомендацій. Щоб покращити персоналізовані рекомендації, користувачу потрібно знайти і оцінити більше фільмів. Нові рейтинги мають безпосередній вплив на рекомендації. Оцінивши та додавши назви до списку спостереження, потрібно перезавантажити сторінку, щоб оновити рекомендації IMDb [14].

2. Netflix розбиває глядачів на більш ніж дві тисячі смакових груп. Більше 80 % телевізійних програм, які люди дивляться на Netflix, надаються через систему рекомендацій. Це означає, що більшість того, що користувачі вирішили подивитися на Netflix, є результатом рішень, прийнятих рекомендаційною системою. Netflix використовує машинне навчання і алгоритми, які допомагають порушити попередньо сприйняті уявлення глядачів і виявити, що вони, можливо, спочатку не вибрали. Для цього він дивиться на аномалії потоку в контенті, а не спирається на широкі жанри, щоб зробити свої прогнози. Це пояснює, як, наприклад, один з восьми користувачів, які дивляться одне з шоу Marvel Netflix, є абсолютно новим. Незважаючи на те, що Netflix має понад

100 мільйонів користувачів по всьому світу, якщо врахувати кілька профілів користувачів для кожного абонента, це призведе до загального обсягу 250 мільйонів активних профілів. Те, що Netflix бере з цих профілів – це дані – що люди дивляться, що спостерігають, що вони спостерігали раніше, що вони дивилися рік тому, що вони спостерігали останнім часом і у який час дня. Ці дані утворюють першу частину алгоритму [15]. Потім ця інформація поєднується з більшою кількістю даних, спрямованих на розуміння змісту шоу. Друга частина алгоритму зібрана з десятків власних і позаштатних співробітників, які дивляться кожне шоу на Netflix і позначають його тегами. Netflix бере всі ці теги і дані про поведінку користувачів, а потім використовує складні алгоритми машинного навчання, які визначають, що саме важливо. Скільки повинно мати значення, якщо користувач дивився щось вчора? Чи повинен він рахувати вдвічі більше або в десять разів більше, ніж те, що користувачі дивилися рік тому? Як приблизно місяць тому? Як щодо того, як вони дивилися десять хвилин і відмовилися дивитися далі, або вони передивилися через дві ночі? У цьому Netflix допомагає машинне навчання. Глядачі вписуються в декілька груп смаку – з яких є «кілька тисяч» – і це ті, які впливають на те, які рекомендації з'являються у верхній частині екранного інтерфейсу, які рядки жанру відображаються, і як кожен рядок упорядковано для кожного окремого користувача [15]. Теги, які використовуються для алгоритмів машинного навчання, однакові по всьому світу. Тим не менш, менші підмножини тегів використовуються в більш зовнішньому вигляді, безпосередньо в інтерфейсі користувача і відрізняються залежно від країни, мови та культурного контексту.

3. Rotten Tomatoes це один із найбільших сайтів відгуків та рецензій на фільми. Вони зберігають сукупний рейтинг критиків [16]. Rotten Tomatoes враховує всі рецензії критиків в індустрії розваг. Вони оцінюють фільми не в масштабах 1–10, а в 1–100, але тільки по двох аспектах – “Rotten” або “Fresh” (“Rotten” не подобається, тоді як “Fresh” – подобається). Отже, якщо фільм оцінюється вище 60 %, це означає, що критикам це подобається, і аудиторії також повинно сподобатися. Більше 80 % з певною кількістю «сертифікованих критиків» фільм вважається “Fresh”, який варто рекомендувати. Середній рейтинг – це середній показник усіх оцінок (з 10), які професійні критики дали фільму. Рейтинг аудиторії – це середня оцінка, яку надають користувачі Rotten Tomatoes (з 5). Статус «Certified Fresh» – це особлива відзнака, присуджена найкращим фільмам та телевізійним шоу [16]. Для отримання кваліфікації фільми або телепередачі повинні відповідати наступним вимогам:

- стійкий показник Tomatometer 75 % або вище;
- принаймні п'ять відгуків від Top Critics;
- фільми у широкому випуску повинні мати не менше 80 відгуків;
- фільми з обмеженим випуском повинні мати не менше 40 відгуків;
- допускаються лише окремі сезони телевізійних шоу, і кожен з них повинен мати не менше 20 відгуків.

Фільм або ТВ-шоу, які відповідають вимогам «Certified Fresh», не одразу отримують нагороду. Натомість, автоматично відзначаються для розгляду персоналом Rotten Tomatoes. Після того, як команда зможе визначити, що навряд

чи оцінка опуститься нижче мінімальних вимог, вони відзначатимуть його як «Certified Fresh». Якщо оцінка Tomatometer знижується нижче 70 %, то фільм або телевізійне шоу втрачає статус «Certified Fresh». Оцінка аудиторії, позначена попкорном, – це відсоток користувачів, які позитивно оцінили фільм або телешоу. Якщо принаймні 60 % користувачів оцінили фільм або телепередачу з рейтингом 3,5 зірочок або вище, відображається повне відро з попкорном, що вказує на його популярний статус («Fresh status») [16]. Коли менш ніж 60 % користувачів оцінили фільм або телепередачу рейтингом 3,5 зірочок або вище, на дисплеї відображається перекинуте відро з попкорном, яке вказує на його мало популярний статус («Rotten status»). Знак «плюс» з'являється для телепрограм, які ще не мають оцінок аудиторії («Audience Scores»). Відсоток, який ви бачите, пов'язаний з цією піктограмою, – це відсоток тих користувачів, які вказали, що вони хочуть бачити ці телепередачі, на відміну від користувачів, які вказали, що вони не хочуть.

Детально проаналізувавши проблеми методів генерації рекомендацій у рекомендаційних системах, такі як холодний старт, розрідженість даних, точність, масштабованість та різноманітність, та розглянувши відомі засоби вирішення цих проблем можна зробити висновок, що потрібно використовувати саме гібридні системи рекомендацій щоб модельована система була ефективною. Застосування РС досі поширене для простих і недорогих продуктів, таких як фільми, музика, новини та книги. Хоча існують системи, що керують більш складними типами елементів, наприклад, фінансові інвестиції або подорожі, ці категорії товарів розглядаються як типові випадки.

Набагато більше досліджень потребують алгоритми та інтерфейси користувачів для створення узгоджених послідовностей рекомендацій. Зокрема, слід моделювати вплив на користувача декількох контекстних умов, таких як спосіб, у який могли б бути показані вже показані раніше рекомендації, та як це буде впливати на оцінку користувачем майбутніх рекомендацій. Звичайно, боротьба з людськими смаками та уподобаннями є надзвичайно складною проблемою, особливо, якщо у багатьох випадках користувач може вирішити скористатися системою, не знаючи, для якого саме жанру у нього є настрої. Кожна людина відрізняється у тому, що приносить їй радість і як це впливає на вибір фільму для перегляду. Крім того, люди змінюються з плином часу, тому алгоритми повинні допомогти передбачити ці зміни. Щоб врахувати цей людський фактор, необхідно доповнити алгоритмічні підходи ідеями, які можна отримати від досліджень подібних систем та їх метаданих.

9. Висновки

1. Розглянута проблема проектування інтелектуальної системи комерційного поширення інформаційних продуктів із застосуванням персоналізованого підходу до відвідувачів на основі категорій та тегів цікавого відвідувачам контенту. Застосування систем рекомендацій досі поширене для простих і недорогих продуктів, таких як фільми, музика, новини та книги. Хоча існують системи, що керують більш складними типами елементів, наприклад, фінансові інвестиції або подорожі, ці категорії товарів розглядаються як типові випадки. На-

багато більше досліджень потребують алгоритми та інтерфейси користувачів для створення узгоджених послідовностей рекомендацій. Зокрема, слід моделювати вплив на користувача декількох контекстних умов, таких як спосіб, у який могли б бути показані вже показані раніше рекомендації, та як це буде впливати на оцінку користувачем майбутніх рекомендацій. Звичайно, боротьба з людськими смаками та уподобаннями є надзвичайно складною проблемою, особливо, якщо у багатьох випадках користувач може вирішити скористатися системою, не знаючи, для якого саме жанру у нього є настрої. Кожна людина відрізняється у тому, що приносить їй радість і як це впливає на вибір фільму для перегляду. Крім того, люди змінюються з плином часу, тому алгоритми повинні допомогти передбачити ці зміни. Щоб врахувати цей людський фактор, необхідно доповнити алгоритмічні підходи ідеями, які можна отримати від досліджень подібних систем та їх метаданих. Розглянуто методи контентної фільтрації, колаборативної фільтрації та гібридні методи. Проведено аналіз алгоритмів колаборативної фільтрації. Обґрунтовано вибір гібридних алгоритмів Machine Learning, включаючи алгоритм User-based колаборативної фільтрації для формування рекомендацій фільмів для користувачів системи. Обґрунтовано доцільність розробки системи підтримання прийняття рішень надання релевантних рекомендацій фільмів з врахуванням особистих потреб користувача.

2. Розроблений метод персоналізації комерційного контенту згідно потреб користувача на основі колаборативної фільтрації та Machine Learning. Через зростання обсягу інформації та даних, виникають різноманітні проблеми у методах фільтрації, включаючи щільність і масштабованість. Дані для Machine Learning розробленої системи взято із *MovieLens* [30]. Тут фільмам наданий рейтинг за 5-бальною шкалою. Із-за неможливості мати доступ до потужних серверів для обчислень у проєкті використано найменший доступний варіант (*ml-latest-small.zip*), який на час завантаження містив 100000 рейтингів та 3600 ключових слів (тегів) для 9000 фільмів. Ці дані були створені 600 користувачами у період з січня 1995 року до вересня 2018 року [22]. Усі вибрані користувачі вже оцінили не менше, як 20 фільмів. Створена ГРС використовує файли *movies.csv* (табл. 4) і *ratings.csv* (табл. 5) з *MovieLens* для генерування рекомендацій. Але Machine Learning провадилось на англійських даних у зв'язку з відсутності на сьогоднішній аналогічних статистичних даних, зібраних з вподобань українського населення. Незважаючи на це, розроблений метод можна використовувати в подальшому для будь-яких ключових слів (тегів) різних мов. Вбудований в систему парсер не залежить від мови ключових слів пошуку від користувачів системи. Лише залежить від наповнення словників та бази даних фільмів ГРС. Новітні тенденції в Machine Learning використовують різні шари обробки, що допомагає в навчанні за допомогою складних контекстних особливостей. На думку експертів, встановлено, що Machine Learning здатне визначати чинники, що впливають на підбір релевантних фільмів, що сприяє поліпшенню надання рекомендацій конкретному користувачу. Для вирішення цих завдань пропонується новий удосконалений метод навчання, на відміну від наявних систем рекомендацій, в основі якого лежать гібридні методи, а базується на Machine Learning. Основним призначенням системи є надання релевантних рекомендацій.

цій фільмів з врахуванням особистих потреб користувача. Створену систему рекомендується використовувати для збору даних про вподобання різних людей у виборі фільмів та надання відповідних щоденних рекомендацій. Також систему можна легко адаптувати для рекомендацій музики, книг, криптовалют чи навіть біржових ринків. Гібридний алгоритм буде приймати вхідну інформацію у різному вигляді, нормалізувати її, та видавати відповідні рекомендації. Зважаючи на це, систему можна застосовувати як у повсякденному житті для вибору фільмів, так і у великих корпораціях у робочих процесах. Розроблено структуру системи підтримання прийняття рішень надання релевантних рекомендацій фільмів з врахуванням особистих потреб користувача. Проведено системний аналіз розроблюваної системи. Виконано проектування архітектури системи. Проведено структурний аналіз і проектування програмних модулів системи релевантності фільмів до потреб користувача. Виконано аналіз мети функціонування системи, конкретизовано її до окремих аспектів і критеріїв якості. Побудовано дерево цілей системи та проведено аналіз ГРС за допомогою системного методу аналітичної ієрархії. Проаналізовано основні варіанти досягнення мети та наявні ресурси. Також побудовані та описані UML діаграми. Крім того, чітко визначені та описані такі питання як мета системи, її призначення, місце застосування, обґрунтування розроблення, очікувані ефекти від її впровадження та була створена концептуальна модель системи.

3. Розроблена інтелектуальна система поширення комерційного контенту в Інтернет-прострі на основі SEO-технологій, нейронних мереж та Machine Learning. Також сформульовані загальні функціональні вимоги до системи поширення комерційного контенту в Інтернет-прострі. Розроблена система на основі сучасних методів SEO-технологій з врахування метрик оцінювання роботи інформаційно-пошукового модуля системи дозволяє підбирати релевантний контент згідно інтересів персоналізованого користувача. Система володіє класами та підкласами, до яких належать реальні комерційні інформаційні продукти, між якими побудовані логічні зв'язки за допомогою яких відбувається інтелектуальна подача контенту на основі персоналізації потреб та зацікавлень користувача. Також на основі сучасних методів Machine Learning розроблена система навчається уточнювати результати пошуку затребуваного контенту згідно персоналізації вподобань користувача. Розроблена система підтримання прийняття рішень надання релевантних рекомендацій фільмів з врахуванням особистих потреб користувача є вільною у доступі, та розміщена на безкоштовному сервері. Проте, щоб успішно розвивати та покращувати її, а також перенести систему на платний стабільніший сервер, доцільно було б залучити кошти інвесторів, чи звичайні пожертви від вдячних користувачів. Тому, на мою думку, ця система є доцільною з точки зору її актуальності та можливої окупності. Ця система є дуже корисна для кінцевих користувачів, оскільки заощаджує значний час для вибору релевантних фільмів для кожного. Тому, є потрібною з точки зору прийняття споживачами на ринку інформаційних технологій. Сьогодні на ринку програмного забезпечення рекомендацій фільмів є такі гіганти як Netflix, Rotten Tomatoes, Movielens та IMDb. Проте всі є іноземними та англомовними. На українському ж ринку аналогів немає. Тому, щоб виправити цю ситуацію,

доцільно створити подану систему. У перспективі розвитку цього ринку передбачається створення та покращення україномовних систем рекомендацій фільмів. Оскільки в Україні немає створених аналогів систем рекомендацій фільмів, то є необхідність у розробленні такого програмного забезпечення.

4. Здійснено практичну реалізацію системи підтримання прийняття рішень надання релевантних рекомендацій фільмів з врахуванням особистих потреб користувача на основі розробленого алгоритму. Проаналізовано роботу системи підтримання прийняття рішень надання релевантних рекомендацій фільмів з врахуванням особистих потреб користувача. Удосконалено алгоритм формування рекомендацій системи, який базується на гібридних алгоритмах Machine Learning, включаючи алгоритм User-based колаборативної фільтрації. У результаті виконання практичної частини кваліфікаційної роботи розроблено систему у вигляді веб-додатка «VikToFilm» – рекомендаційну систему фільмів на основі гібридних алгоритмів Machine Learning, включаючи алгоритм User-based колаборативної фільтрації. Наведено опис програмного засобу, загальні відомості, функціональне призначення, опис логічної структури, використовувані технічні засоби, виклик і завантаження, вхідні та вихідні дані системи. Наведено аналіз контрольного прикладу роботи програми. Здійснено аналіз результатів експериментальної апробації запропонованого методу персоналізації комерційного контенту згідно потреб користувача. Проаналізовано роботу системи підтримання прийняття рішень надання релевантних рекомендацій фільмів з врахуванням особистих потреб користувача. Розроблено алгоритм роботи системи, який базується на гібридних алгоритмах Machine Learning, включаючи алгоритм User-based колаборативної фільтрації. У результаті виконання практичної частини кваліфікаційної роботи розроблено систему у вигляді веб-додатка «VikToFilm» – рекомендаційну систему фільмів на основі гібридних алгоритмів Machine Learning. Наведено опис програмного засобу, загальні відомості, функціональне призначення, опис логічної структури, використовувані технічні засоби, виклик і завантаження, вхідні та вихідні дані системи. Наведено аналіз контрольного прикладу роботи програми. Завершено практичне виконання розробки системи підтримання прийняття рішень надання релевантних рекомендацій фільмів з врахуванням особистих потреб користувача.

Систему рекомендується використовувати для збору даних про вподобання різних людей у виборі фільмів та надання відповідних релевантних рекомендацій. У подальших дослідженнях у цій області доцільно вдосконалити спосіб збереження даних про фільми та замінити алгоритм ранжування фільмів на основі відгуків, взявши за основу гібридний алгоритм, що враховує якісні та кількісні характеристики відгуків.

Література

1. Melville, P., Mooney, R., Nagarajan, R. (2016). Content-Boosted Collaborative Filtering for Improved Recommendations. National Conference on Artificial Intelligence: «AAAI-2002», 187–192.
2. Lytvyn, V., Vysotska, V., Demchuk, A., Demkiv, I., Ukhanska, O., Hladun, V. et. al. (2019). Design of the architecture of an intelligent system for distributing commercial content in the internet space based on SEO-technologies, neural networks,

and Machine Learning. *Eastern-European Journal of Enterprise Technologies*, 2 (2 (98)), 15–34. doi: <https://doi.org/10.15587/1729-4061.2019.164441>

3. Jones, M. T. (2013). Recommender systems, Part 1. Introduction to approaches and algorithms. URL: <https://www.ibm.com/developerworks/opensource/library/os-recommender1>

4. Su, X., Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 2009, 1–19. doi: <https://doi.org/10.1155/2009/421425>

5. Burov, Y., Vysotska, V., Kravets, P. (2019). Ontological approach to plot analysis and modeling. *Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Systems (COLINS-2019)*. Volume I: Main Conference, 2362, 22–31.

6. Sarwar, B., Karypis, G., Konstan, J., Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the Tenth International Conference on World Wide Web - WWW '01*. doi: <https://doi.org/10.1145/371920.372071>

7. Schafer, J. B., Konstan, J., Riedl, J. (1999). Recommender systems in e-commerce. *Proceedings of the 1st ACM Conference on Electronic Commerce - EC '99*. doi: <https://doi.org/10.1145/336992.337035>

8. Gope, J., Jain, S. K. (2017). A survey on solving cold start problem in recommender systems. *2017 International Conference on Computing, Communication and Automation (ICCCA)*. doi: <https://doi.org/10.1109/ccaa.2017.8229786>

9. Ge, M., Delgado-Battenfeld, C., Jannach, D. (2010). Beyond accuracy: Evaluating recommender systems by coverage and serendipity. *Proceedings of the fourth ACM conference on Recommender systems - RecSys '10*, 257–260. doi: <https://doi.org/10.1145/1864708.1864761>

10. Bobadilla, J., Ortega, F., Hernando, A., Bernal, J. (2012). A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, 26, 225–238. doi: <https://doi.org/10.1016/j.knosys.2011.07.021>

11. Nambiar, R., Bhardwaj, R., Sethi, A., Vargheese, R. (2013). A look at challenges and opportunities of Big Data analytics in healthcare. *2013 IEEE International Conference on Big Data*. doi: <https://doi.org/10.1109/bigdata.2013.6691753>

12. Calero Valdez, A., Ziefle, M., Verbert, K. (2016). HCI for recommender systems: The past, the present and the future. *RecSys '16 Proceedings of the 10th ACM Conference on Recommender System*, 123–126. doi: <https://doi.org/10.1145/2959100.2959158>

13. Kotsiantis, S. B., Zaharakis, I., Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Frontiers in Artificial Intelligence and Applications*, Volume 160: Emerging Artificial Intelligence Applications in Computer Engineering, 3–24.

14. Recommended For You FAQ. URL: <https://help.imdb.com/article/imdb/discover-watch/recommended-for-you-faq/GPZ2RSPB3CPVL86Z/>

15. Netflix Prize. URL: <https://www.netflixprize.com/>

16. About Rotten Tomatoes. URL: <https://www.rottentomatoes.com/about>
17. Lytvyn, V., Vysotska, V., Rzheuskyi, A. (2019). Technology for the Psychological Portraits Formation of Social Networks Users for the IT Specialists Recruitment Based on Big Five, NLP and Big Data Analysis. Proceedings of the 1st International Workshop on Control, Optimisation and Analytical Processing of Social Networks (COAPSN-2019), 2392, 147–171.
18. Lytvyn, V., Vysotska, V., Rusyn, B., Pohreliuk, L., Berezin, P., Naum, O. (2019). Textual Content Categorizing Technology Development Based on Ontology. Workshop Proceedings of the 8th International Conference on “Mathematics. Information Technologies. Education”, 2386, 234–254.
19. Lytvyn, V., Kuchkovskiy, V., Vysotska, V., Markiv, O., Pabyrivskyy, V. (2018). Architecture of System for Content Integration and Formation Based on Cryptographic Consumer Needs. 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). doi: <https://doi.org/10.1109/stc-csit.2018.8526669>
20. Rubens, N., Elahi, M., Sugiyama, M., Kaplan, D. (2015). Active Learning in Recommender Systems. Recommender Systems Handbook, 809–846. doi: https://doi.org/10.1007/978-1-4899-7637-6_24
21. Ms. Ashwini A. Chirde, Ms. Urmila K. (2015). Combination of a Cluster-Based and Content-Based Collaborative Filtering Approach for Recommender System. International Journal on Recent and Innovation Trends in Computing and Communication, 3 (7), 4770–4774.
22. Harper, F. M., Konstan, J. A. (2015). The MovieLens Datasets. ACM Transactions on Interactive Intelligent Systems, 5 (4), 1–19. doi: <https://doi.org/10.1145/2827872>
23. Golemund, G. (2015). Hands-On Programming with R: Write Your Own Functions and Simulations. Sebastopol, United States.
24. McLeod, D., Chen, A.-Y. (2009). Collaborative Filtering for Information Recommendation Systems. Research Reports.
25. Ricci, F., Rokach, L., Shapira, B. (Eds.) (2015). Recommender Systems Handbook. Springer. doi: <https://doi.org/10.1007/978-1-4899-7637-6>
26. Linden, G., Smith, B., York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Computing, 7 (1), 76–80. doi: <https://doi.org/10.1109/mic.2003.1167344>
27. The Comprehensive R Archive Network. URL: <https://cran.r-project.org>
28. RStudio. URL: <https://www.rstudio.com/products>
29. Chapter 2 Getting Started. URL: <https://docs.rstudio.com/shinyapps.io/getting-started.html>
30. MovieLens Latest Datasets. URL: <https://grouplens.org/datasets/movielens/latest>
31. Sitecore Documentation: Access all the latest Sitecore documentation. URL: <https://doc.sitecore.com>
32. Nouh, R., Lee, H.-H., Lee, W.-J., Lee, J.-D. (2019). A Smart Recommender Based on Hybrid Learning Methods for Personal Well-Being Services. Sensors, 19 (2), 431. doi: <https://doi.org/10.3390/s19020431>

33. Mobasher, B. (2007). Data Mining for Web Personalization. *Lecture Notes in Computer Science*, 90–135. doi: https://doi.org/10.1007/978-3-540-72079-9_3
34. Berko, A., Alieksieiev, V. (2018). A Method to Solve Uncertainty Problem for Big Data Sources. 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP). doi: <https://doi.org/10.1109/dsmp.2018.8478460>
35. Xu, G., Zhang, Y., Li, L. (2010). Web Content Mining. *Web Mining and Social Networking*, 71–87. doi: https://doi.org/10.1007/978-1-4419-7735-9_4
36. Lytvyn, V., Vysotska, V., Pukach, P., Nytrebych, Z., Demkiv, I., Senyk, A. et. al. (2018). Analysis of the developed quantitative method for automatic attribution of scientific and technical text content written in Ukrainian. *Eastern-European Journal of Enterprise Technologies*, 6 (2 (96)), 19–31. doi: <https://doi.org/10.15587/1729-4061.2018.149596>
37. Gozhyj, A., Kalinina, I., Vysotska, V., Gozhyj, V. (2018). The Method of Web-Resources Management Under Conditions of Uncertainty Based on Fuzzy Logic. 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). doi: <https://doi.org/10.1109/stc-csit.2018.8526761>
38. Lytvyn, V., Vysotska, V., Dosyn, D., Burov, Y. (2018). Method for ontology content and structure optimization, provided by a weighted conceptual graph. *Webology*, 15 (2), 66–85.
39. Khomytska, I., Teslyuk, V. (2016). Specifics of phonostatistical structure of the scientific style in English style system. 2016 XIth International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT). doi: <https://doi.org/10.1109/stc-csit.2016.7589887>
40. Khomytska, I., Teslyuk, V. (2016). The Method of Statistical Analysis of the Scientific, Colloquial, Belles-Lettres and Newspaper Styles on the Phonological Level. *Advances in Intelligent Systems and Computing*, 149–163. doi: https://doi.org/10.1007/978-3-319-45991-2_10
41. Nytrebych, Z. M., Malanchuk, O. M., Il'kiv, V. S., Pukach, P. Ya. (2017). Homogeneous problem with two-point conditions in time for some equations of mathematical physics. *Azerbaijan Journal of Mathematics*, 7 (2), 180–196.
42. Nytrebych, Z., Il'kiv, V., Pukach, P., Malanchuk, O. (2018). On nontrivial solutions of homogeneous Dirichlet problem for partial differential equations in a layer. *Kragujevac Journal of Mathematics*, 42 (2), 193–207. doi: <https://doi.org/10.5937/kgjmath1802193n>
43. Nytrebych, Z., Malanchuk, O., Il'kiv, V., Pukach, P. (2017). On the solvability of two-point in time problem for PDE. *Italian Journal of Pure and Applied Mathematics*, 38, 715–726.
44. Pukach, P. Ya., Kuzio, I. V., Nytrebych, Z. M., Ilkiv, V. S. (2017). Analytical methods for determining the effect of the dynamic process on the nonlinear flexural vibrations and the strength of compressed shaft. *Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu*, 5, 69–76.
45. Pukach, P. Y., Kuzio, I. V., Nytrebych, Z. M., Il'kiv, V. S. (2018). Asymptotic method for investigating resonant regimes of nonlinear bending

vibrations of elastic shaft. *Scientific Bulletin of National Mining University*, 1, 68–73. doi: <https://doi.org/10.29202/nvngu/2018-1/9>

46. Nytrebych, Z., Ilkiv, V., Pukach, P., Malanchuk, O., Kohut, I., Senyk, A. (2019). Analytical method to study a mathematical model of wave processes under twopoint time conditions. *Eastern-European Journal of Enterprise Technologies*, 1 (7 (97)), 74–83. doi: <https://doi.org/10.15587/1729-4061.2019.155148>

47. Pukach, P., Il'kiv, V., Nytrebych, Z., Vovk, M., Pukach, P. (2017). On the Asymptotic Methods of the Mathematical Models of Strongly Nonlinear Physical Systems. *Advances in Intelligent Systems and Computing*, 421–433. doi: https://doi.org/10.1007/978-3-319-70581-1_30

48. Lavrenyuk, S. P., Pukach, P. Y. (2007). Mixed problem for a nonlinear hyperbolic equation in a domain unbounded with respect to space variables. *Ukrainian Mathematical Journal*, 59 (11), 1708–1718. doi: <https://doi.org/10.1007/s11253-008-0020-0>

49. Pukach, P. Y. (2016). Investigation of Bending Vibrations in Voigt–Kelvin Bars with Regard for Nonlinear Resistance Forces. *Journal of Mathematical Sciences*, 215 (1), 71–78. doi: <https://doi.org/10.1007/s10958-016-2823-0>

50. Pukach, P., Il'kiv, V., Nytrebych, Z., Vovk, M. (2017). On nonexistence of global in time solution for a mixed problem for a nonlinear evolution equation with memory generalizing the Voigt-Kelvin rheological model. *Opuscula Mathematica*, 37 (45), 735. doi: <https://doi.org/10.7494/opmath.2017.37.5.735>

51. Pukach, P. Y. (2012). On the unboundedness of a solution of the mixed problem for a nonlinear evolution equation at a finite time. *Nonlinear Oscillations*, 14 (3), 369–378. doi: <https://doi.org/10.1007/s11072-012-0164-6>

52. Pukach, P. Y. (2014). Qualitative Methods for the Investigation of a Mathematical Model of Nonlinear Vibrations of a Conveyer Belt. *Journal of Mathematical Sciences*, 198 (1), 31–38. doi: <https://doi.org/10.1007/s10958-014-1770-x>

53. Bezobrazov, S., Sachenko, A., Komar, M., Rubanau, V. (2016). The Methods of Artificial Intelligence for Malicious Applications Detection in Android OS. *International Journal of Computing*, 15 (3), 184–190.

54. Dunets, O., Wolff, C., Sachenko, A., Hladiy, G., Dobrotvor, I. (2017). Multi-agent system of IT project planning. 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). doi: <https://doi.org/10.1109/idaacs.2017.8095141>

55. Lytvyn, V., Vysotska, V., Pukach, P., Nytrebych, Z., Demkiv, I., Kovalchuk, R., Huzyk, N. (2018). Development of the linguometric method for automatic identification of the author of text content based on statistical analysis of language diversity coefficients. *Eastern-European Journal of Enterprise Technologies*, 5 (2 (95)), 16–28. doi: <https://doi.org/10.15587/1729-4061.2018.142451>

56. Vysotska, V., Lytvyn, V., Burov, Y., Berezin, P., Emmerich, M., Basto Fernandes, V. (2019). Development of Information System for Textual Content Categorizing Based on Ontology. *CEUR Workshop Proceedings*, 53–70.

57. Vysotska, V., Lytvyn, V., Burov, Y., Gozhyj, A., Makara, S. (2018). The consolidated information web-resource about pharmacy networks in city. *Proceedings*

of the 1st International Workshop on Informatics & Data-Driven Medicine (IDDM 2018), 2255, 239–255. URL: <http://ceur-ws.org/Vol-2255/paper22.pdf>

58. Rusyn, B., Vysotska, V., Pohreliuk, L. (2018). Model and Architecture for Virtual Library Information System. 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). doi: <https://doi.org/10.1109/stc-csit.2018.8526679>

59. Lytvyn, V., Vysotska, V., Dosyn, D., Lozynska, O., Oborska, O. (2018). Methods of Building Intelligent Decision Support Systems Based on Adaptive Ontology. 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP). doi: <https://doi.org/10.1109/dsmp.2018.8478500>

60. Lytvyn, V., Vysotska, V., Burov, Y., Bobyk, I., Ohirko, O. (2018). The Linguometric Approach for Co-authoring Author's Style Definition. 2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS). doi: <https://doi.org/10.1109/idaacs-sws.2018.8525741>

61. Zdebskyi, P., Vysotska, V., Peleshchak, R., Peleshchak, I., Demchuk, A., Krylyshyn, M. (2019). An Application Development for Recognizing of View in Order to Control the Mouse Pointer. Workshop Proceedings of the 8th International Conference on “Mathematics. Information Technologies. Education”, 55–74.

62. Veres, O., Rusyn, B., Sachenko, A., Rishnyak, I. (2018). Choosing the method of finding similar images in the reverse search system. Proceedings of the 2nd International Conference on Computational Linguistics and Intelligent Systems. Volume I: Main Conference (COLINS 2018), 2136, 99–107.

63. Rashkevych, Y., Peleshko, D., Vynokurova, O., Izonin, I., Lotoshynska, N. (2017). Single-frame image super-resolution based on singular square matrix operator. 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON). doi: <https://doi.org/10.1109/ukrcon.2017.8100390>

64. Vysotska, V., Lytvyn, V., Hrendus, M., Kubinska, S., Brodyak, O. (2018). Method of Textual Information Authorship Analysis Based on Stylometry. 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). doi: <https://doi.org/10.1109/stc-csit.2018.8526608>

65. Gozhyj, A., Chyrun, L., Kowalska-Styczen, A., Lozynska, O. (2018). Uniform Method of Operative Content Management in Web Systems. Proceedings of the 2nd International Conference on Computational Linguistics and Intelligent Systems. Volume I: Main Conference (COLINS 2018), 2136. P. 62–77. URL: <http://ceur-ws.org/Vol-2136/10000062.pdf>

66. Vysotska, V., Burov, Y., Lytvyn, V., Demchuk, A. (2018). Defining Author's Style for Plagiarism Detection in Academic Environment. 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), 128–133. doi: <https://doi.org/10.1109/dsmp.2018.8478574>

67. Chyrun, L., Vysotska, V., Kis, I., Chyrun, L. (2018). Content Analysis Method for Cut Formation of Human Psychological State. 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP). doi: <https://doi.org/10.1109/dsmp.2018.8478619>

68. Gozhyj, A., Vysotska, V., Yevseyeva, I., Kalinina, I., Gozhyj, V. (2018). Web Resources Management Method Based on Intelligent Technologies. *Advances in Intelligent Systems and Computing III*, 206–221. doi: https://doi.org/10.1007/978-3-030-01069-0_15
69. Chyrun, L., Kis, I., Vysotska, V., Chyrun, L. (2018). Content Monitoring Method for Cut Formation of Person Psychological State in Social Scoring. 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). doi: <https://doi.org/10.1109/stc-csit.2018.8526624>
70. Demchuk, A., Lytvyn, V., Vysotska, V., Dilai, M. (2019). Methods and Means of Web Content Personalization for Commercial Information Products Distribution. *Lecture Notes in Computational Intelligence and Decision Making*, 332–347. doi: https://doi.org/10.1007/978-3-030-26474-1_24
71. Lytvyn, V., Vysotska, V., Kuchkovskiy, V., Bobyk, I., Malanchuk, O., Ryshkovets, Y. et. al. (2019). Development of the system to integrate and generate content considering the cryptocurrent needs of users. *Eastern-European Journal of Enterprise Technologies*, 1 (2 (97)), 18–39. doi: <https://doi.org/10.15587/1729-4061.2019.154709>
72. Vysotska, V., Fernandes, V. B., Lytvyn, V., Emmerich, M., Hrendus, M. (2018). Method for Determining Linguometric Coefficient Dynamics of Ukrainian Text Content Authorship. *Advances in Intelligent Systems and Computing III*, 132–151. doi: https://doi.org/10.1007/978-3-030-01069-0_10
73. Kravets, P. (2010). The control agent with fuzzy logic. *Perspective Technologies and Methods in MEMS Design*, 40–41.
74. Kravets, P. (2007). The Game Method for Orthonormal Systems Construction. 2007 9th International Conference - The Experience of Designing and Applications of CAD Systems in Microelectronics. doi: <https://doi.org/10.1109/cadsm.2007.4297555>
75. Kravets, P. (2016). Game Model of Dragonfly Animat Self-Learning. *Perspective Technologies and Methods in MEMS Design*, 195–201.
76. Bazylyk, O., Taradaha, P., Nadobko, O., Chyrun, L., Shestakevych, T. (2012). The results of software complex OPTAN use for modeling and optimization of standard engineering processes of printed circuit boards manufacturing. 2012 11th International Conference on "Modern Problems of Radio Engineering, Telecommunications and Computer Science" (TCSET), 107–108.
77. Bondariiev, A., Kiselychnyk, M., Nadobko, O., Nedostup, L., Chyrun, L., Shestakevych, T. (2012). The software complex development for modeling and optimizing of processes of radio-engineering equipment quality providing at the stage of manufacture. *TCSET'2012*, 159.
78. Teslyuk, V., Beregovskiy, V., Denysyuk, P., Teslyuk, T., Lozynskiy, A. (2018). Development and Implementation of the Technical Accident Prevention Subsystem for the Smart Home System. *International Journal of Intelligent Systems and Applications*, 10 (1), 1–8. doi: <https://doi.org/10.5815/ijisa.2018.01.01>
79. Basyuk, T. (2015). The main reasons of attendance falling of internet resource. 2015 Xth International Scientific and Technical Conference "Computer

Sciences and Information Technologies” (CSIT). doi: <https://doi.org/10.1109/stc-csit.2015.7325440>

80. Chernukha, O., Bilushchak, Y. (2016). Mathematical modeling of random concentration field and its second moments in a semispace with erlangian distribution of layered inclusions. *Task Quarterly*, 20 (3), 295–334.

81. Chyrun, L., Kowalska-Styczen, A., Burov, Y., Berko, A., Vasevych, A., Pelekh, I., Ryshkovets, Y. (2019). Heterogeneous Data with Agreed Content Aggregation System Development. *Workshop Proceedings of the 8th International Conference on “Mathematics. Information Technologies. Education”*, 2386, 35–54.

82. Chyrun, L., Burov, Y., Rusyn, B., Pohreliuk, L., Oleshek, O. et. al. (2019). Web Resource Changes Monitoring System Development. *Workshop Proceedings of the 8th International Conference on “Mathematics. Information Technologies. Education”*, 2386, 255–273.

83. Vysotska, V., Burov, Y., Lytvyn, V., Oleshek, O. (2019). Automated Monitoring of Changes in Web Resources. *Lecture Notes in Computational Intelligence and Decision Making*, 348–363. doi: https://doi.org/10.1007/978-3-030-26474-1_25

84. Chyrun, L., Gozhyj, A., Yevseyeva, I., Dosyn, D., Tyhonov, V., Zakharchuk, M. (2019). Web Content Monitoring System Development. *Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Systems (COLINS-2019). Volume I: Main Conference*, 2362, 126–142.

85. Rzhеuskyi, A., Gozhyj, A., Stefanchuk, A., Oborska, O., Chyrun, L., Lozynska, O. et. al. (2019). Development of Mobile Application for Choreographic Productions Creation and Visualization. *Workshop Proceedings of the 8th International Conference on “Mathematics. Information Technologies. Education”*, 2386, 340–358.

86. Lytvynenko, V., Savina, N., Krejci, J., Voronenko, M., Yakobchuk, M., Kryvoruchko, O. (2019). Bayesian Networks' Development Based on Noisy-MAX Nodes for Modeling Investment Processes in Transport. *Workshop Proceedings of the 8th International Conference on “Mathematics. Information Technologies. Education”*, 2386, 1–10.

87. Lytvynenko, V., Lurie, I., Krejci, J., Voronenko, M., Savina, N., Taif, M. A. (2019). Two Step Density-Based Object-Inductive Clustering Algorithm. *Workshop Proceedings of the 8th International Conference on “Mathematics. Information Technologies. Education”*, 2386, 117–135.