

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Навчально-науковий центр заочної форми навчання

(повна назва)

Кафедра Інформаційних управляючих систем

(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів динамічної адаптації рекомендацій в
рекомендаційних системах

(тема)

Виконав:

студент 2 курсу, групи ІУСТЗМ-21-1

Євгенія КАРУННА

(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні
науки

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні управляючі
системи та технології

(повна назва освітньої програми)

Керівник професор каф. ІУС Оксана ЧАЛА

(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри



(підпис)

Костянтин ПЕТРОВ

(власне ім'я, прізвище)

2022 р.

Харківський національний університет радіоелектроніки

Навчально-науковий центр заочної форми навчання

Кафедра Інформаційних управляючих систем

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні управляючі системи та технології

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри



(підпис)

« 21 » листопада 20 22 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Карунній Євгенії Юріївні

(прізвище, ім'я, по батькові)

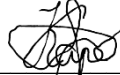
1. Тема роботи Дослідження методів динамічної адаптації рекомендацій в рекомендаційних системах
затверджена наказом університету від 14 листопада 2022 р. № 185Стз
2. Термін подання студентом роботи до екзаменаційної комісії 16.12.2022 р.
3. Вихідні дані до роботи Науково-технічні публікації, дані статей, результати експериментальних досліджень, дані проектів щодо розробки рекомендаційних систем.
4. Перелік питань, що потрібно опрацювати в роботі Аналіз онлайн та офлайн підходів до побудови рекомендаційних систем. Аналіз характеристик рекомендаційних систем. Аналіз процесу побудови рекомендаційних систем. Дослідження методу побудови рекомендаційних систем. Постановка задачі дослідження. Метод динамічної адаптації рекомендацій. Побудова патернів зміни вподобань користувача. Удосконалений метод динамічної адаптації рекомендацій. Розробка технології динамічної адаптації рекомендацій. Експериментальна перевірка удосконаленого методу динамічної адаптації рекомендацій. Програмна реалізація удосконаленого методу. Експериментальна перевірка удосконаленого методу.

КАЛЕНДАРНИЙ ПЛАН

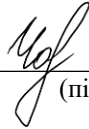
№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз онлайн та офлайн підходів до побудови рекомендаційних систем	10.10.22 – 22.10.22	виконано
2	Метод динамічної адаптації рекомендації	23.10.22 – 4.11.22	виконано
3	Розробка технології динамічної адаптації рекомендацій	5.11.22 – 19.11.22	виконано
4	Експериментальна перевірка	20.11.22 – 26.11.22	виконано
5	Оформлення пояснювальної записки та графічного матеріалу	27.11.22 – 16.12.22	виконано
6	Перевірка на плагіат	17.12.2022	виконано
7	Попередній захист кваліфікаційної роботи	19.12.2022	виконано

Дата видачі завдання 21 листопада 2022 р.

Студент _____


(підпис)

Керівник роботи _____


(підпис)

проф Чала О. В.

(посада, власне ім'я, прізвище)

ЗМІСТ

Скорочення та умовні позначки	7
Вступ.....	8
1 Аналіз онлайн та офлайн підходів до побудови рекомендаційних систем .	10
1.1 Аналіз характеристик рекомендаційних систем	10
1.2 Аналіз процесу побудови рекомендацій	17
1.3 Дослідження методів побудови рекомендацій в рекомендаційних системах.....	19
1.4 Постановка задачі дослідження	33
2 Метод динамічної адаптації рекомендацій.....	35
2.1 Побудова патернів зміни вподобань користувачів	35
2.2 Удосконалений метод динамічної адаптації рекомендацій	38
3 Розробка технології динамічної адаптації рекомендацій.....	41
4 Експериментальна перевірка удосконаленого методу динамічної адаптації рекомендацій.....	43
4.1 Програмна реалізація удосконаленого методу	43
4.2 Експериментальна перевірка удосконаленого методу	48
Висновки	52
Перелік джерел посилання	53
Додаток А. Графічний матеріал.....	58

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 72 с., 4 табл., 19 рис., 1 дод., 35 джерел.

АДАПТАЦІЯ, ГІБРИДНІ МОДЕЛІ, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, ПАТЕРН, РЕКОМЕНДАЦІЇ НА ОСНОВІ КОНТЕНТА, РЕКОМЕНДАЦІЙНІ СИСТЕМИ.

Об'єкт дослідження – процес побудови рекомендацій щодо персоналізованого набору товарів та послуг для користувача.

Предмет дослідження – методи побудови рекомендацій.

Мета дослідження – підвищення ефективності побудови формування рекомендацій з урахуванням змін інтересів користувача.

У роботі виконано дослідження методів побудови рекомендацій в рекомендаційних систем. Проаналізовано існуючі методи побудови рекомендацій, на підставі проведеного аналізу запропоновано удосконалений метод динамічної адаптації рекомендацій шляхом використання патерну.

Результати магістерської роботи можуть бути використані при побудові рекомендаційної системи з динамічною зміною рекомендацій.

ABSTRACT

Explanatory note to the qualification work: 72 pp., 4 tables, 19 figures, 1 appendices, 35 sources.

ADAPTATION, COLLABORATIVE FILTERING, CONTENT-BASED RECOMMENDATIONS, HYBRID MODELS, PATTERN, RECOMMENDER SYSTEMS.

The object of study is the process of building recommendations for personalized equipment of goods and services for users.

The subject of study is methods of building recommendations.

The purpose of study is to increase the effectiveness of building recommendations, taking into account changes in the user's interests.

The research of methods of building recommendations in recommender systems is carried out in the work. The existing methods of building recommendations have been analyzed, based on the analysis, we offer an improved method of dynamic adaptation of recommendations using a pattern.

The results of the master's thesis can be used in the construction of a recommender system with a dynamic change of recommendations.

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

CTR – click-through-rate;

RS – рекомендаційна система;

MAE – mean absolute error;

RMS – root mean squared error;

ROC – receiver operating characteristic;

DCG – дисконтований кумулятивний приріст;

IDEF – integration definition;

СУБД – система управління базами даних;

КФ – колаборативна фільтрація.

ВСТУП

В останні десятиліття Інтернет стрімко розвивався, і щодня в глобальній мережі генеруються та накопичуються величезні обсяги інформації. Користувачам доводиться працювати з цими даними: обробляти, систематизувати, знаходити дані, що відповідають їхнім інформаційним потребам. Людям важко вибрати інформацію, яка їх цікавить, за допомогою звичайного перегляду, оскільки релевантна інформація часто втрачається у великих даних. У зв'язку з цим створюються інструменти, які можуть допомогти людям у їх пошуку, забезпечуючи найбільш підходящий для користувача контент.

Як залучення користувачів до контенту в Інтернеті стало вирішальним елементом у більшості випадків надання контенту мультимедійні платформи – тобто збереження інтересу користувача до наданого вмісту та максимізації їхнього часу на перегляд/читання/прослуховування вмісту, роль рекомендаційних систем неможливо переоцінити у формуванні та покращенні взаємодії з користувачем, коли справа доходить до споживання і взаємодії з цим вмістом, оскільки це сприяє розподілу переважної кількості даних у стислій, цілеспрямованій і цікавій добірці елементів, які користувач, швидше за все, знайде приємним і цікавим.

Традиційно рекомендаційної системи або використовують колаборативну фільтрацію, тобто використання статистики користувачів і їхні неявні/явні відгуки (перегляди, вподобання, час перегляду) щоб знайти продукти для рекомендації (основне припущення полягає в тому, що люди зі схожими інтересами взаємодіють однакові елементи) або надавати рекомендації на основі вмісту, які спираються на вміст самого елемента знаходити подібний вміст без будь-якого введення з боку користувача.

Рекомендаційні системи (RS) допомагають користувачеві в процесі прийняття рішень, і формуються на основі порівняння довгострокових

інтересів користувачів у режимі офлайн, а також короткострокових – у онлайн режимі. Однак недостатньо уваги приділяється зміні інтересів користувачів з часом, що потребує динамічної адаптації рекомендацій. Така проблема виникає, наприклад, при зміні демографічних характеристик користувачів (освіта, зміна місця проживання, тощо). Вирішення цієї проблеми пов'язано з побудовою та використанням патернів, які враховують зміну інтересів користувачів з часом.

Об'єктом дослідження в рамках магістерської кваліфікаційної роботи є процес побудови рекомендацій щодо персоналізованого набору товарів та послуг для користувача.

Предметом дослідження являються методи побудови рекомендацій.

Метою даної роботи є дослідження методів побудови рекомендацій з урахуванням змін інтересів користувача.

Для досягнення мети, необхідно вирішити наступні задачі:

- аналіз характеристик рекомендаційних систем;
- дослідження методів побудови рекомендацій;
- удосконалення методу динамічної адаптації;
- експериментальна перевірка удосконаленого методу.

1 АНАЛІЗ ОНЛАЙН ТА ОФЛАЙН ПІДХОДІВ ДО ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1 Аналіз характеристик рекомендаційних систем

Рекомендаційні системи – це алгоритм штучного інтелекту, зазвичай пов'язаний із машинним навчанням, який використовує великі дані, щоб пропонувати або рекомендувати додаткові продукти споживачам. Вони можуть ґрунтуватися на різних критеріях, зокрема минулих покупках, історії пошуку, демографічній інформації та інших факторах. рекомендаційної системи дуже корисні, оскільки вони допомагають користувачам знаходити продукти та послуги, які вони могли б не знайти самостійно. [2].

Рекомендаційні системи зазвичай складаються з трьох компонентів: дані користувача/клієнта (персональні дані: інтереси, історія покупок, рейтинги та відгуки), дані про товари (специфікації та функції товару) і методи фільтрації, які використовують два вищезгаданих типи даних для фільтрації елементів та рекомендації тих, які близькі до інтересів користувача.

На рисунку 1.1 наведено базовий рівень даних. Цей рівень містить дані профілю користувача, дані про товари, дані про поведінку та дані коментарів. Даними профілю користувача можуть бути зріст і вага користувачів, товари, які вони придбали, їхні вподобання щодо покупок або освіта [5]. Даними про товари є ціни, кольори та походження товарів. Якщо елемент є відео, даними елемента є інформація про відео, наприклад відеовміст і теги. Дані про поведінку стосуються взаємодії між користувачами та елементами. Наприклад, коли користувач переглядає відео, він може поставити лайк до відео, додати відео до вибраного або заплатити за відео. Усі ці дії є даними про поведінку користувача. Дані коментарів можуть включати сторонні дані та

бути доступними не для кожного елемента на кожній платформі. Однак дані користувача, дані про предмет і дані про поведінку є важливими [4].

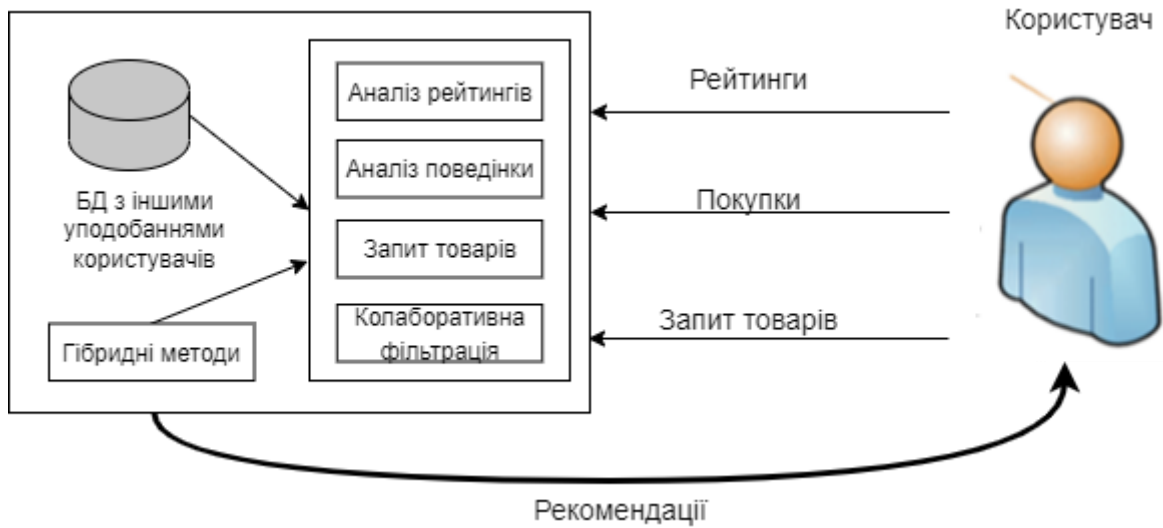


Рисунок 1.1 – Загальна архітектура рекомендаційної системи

В основі будь-якої рекомендаційної системи лежить так звана матриця переваг. Це матриця, яка розміщує всіх клієнтів сервісу по одній осі, а рекомендовані об'єкти – по іншій. На перетині деяких пар (користувач, об'єкт) ця матриця заповнюється рейтингами – відомими показниками інтересу користувача до продукту, вираженими за заданою шкалою (наприклад, від 1 до 5). (рисунок 1.2) [5].

	Об'єкт 1	Об'єкт 2	Об'єкт 3	Об'єкт 4	Об'єкт 5
Користувач 1		3		5	
Користувач 2	1		1	1	
Користувач 3	2			3	2
Користувач 4		4			5
Користувач 5	5		2	3	4

Рисунок 1.2 – Матриця переваг рекомендаційної системи

Користувачі зазвичай оцінюють лише невелику підмножину продуктів у каталозі, а завдання рекомендаційних систем полягає в узагальненні цієї інформації та прогнозуванні ставлення клієнтів до інших невідомих продуктів [3].

Люди мають різні моделі споживання, тому нові продукти не обов'язково рекомендуються. Можна відображати дублікати, наприклад, для поповнення запасів певного товару. Відповідно до цього принципу можна виділити дві групи товарів:

- повторювані, наприклад, шампуні або бритва, які потрібні завжди;
- неповторні, наприклад, книги або фільми.

Якщо продукт не можна явно віднести до одного з класів, то необхідно визначати допустимість повторних покупок індивідуально [6].

В рекомендаційних системах результати можуть бути обчислені онлайн у режимі реального часу, офлайн у пакетному режимі. Кожен підхід має свої переваги та недоліки, які необхідно враховувати для кожного випадку використання. Розглянемо онлайн та офлайн режими (рисунок 1.3).

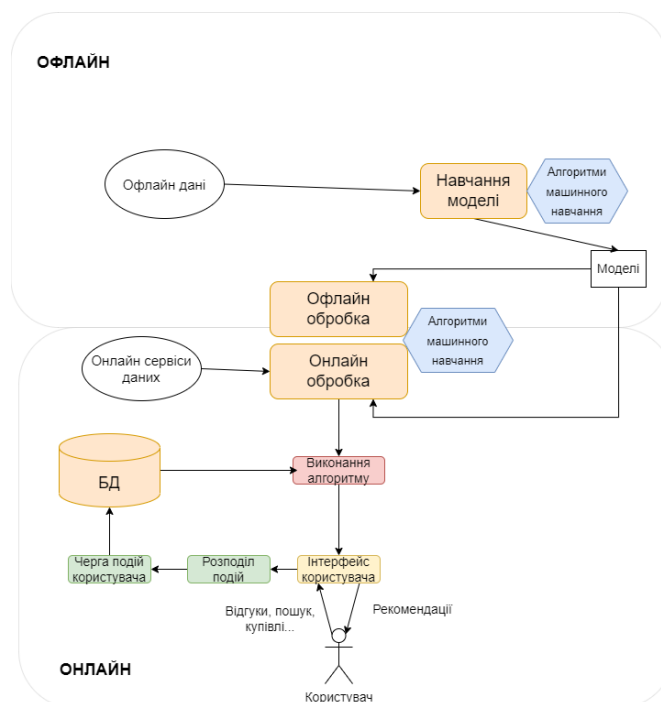


Рисунок 1.3 – Офлайн та онлайн режими роботи рекомендаційної системи

Офлайн обчислення мають менше обмежень щодо обсягу даних і обчислювальної складності алгоритмів, оскільки вони виконуються в пакетному режимі з послабленими вимогами до часу. Однак він може легко застаріти між оновленнями, оскільки останні дані не включаються [7].

В магістерській кваліфікаційній роботі було розглянуто динамічну адаптацію рекомендацій, тобто рекомендації розглядаються у режимі реального часу. В даному режимі дані надходять потоками та обробляються й аналізуються під час створення. Онлайн-обчислення можуть швидко реагувати на останні події та взаємодію користувачів, але мають відповідати на запити в режимі реального часу. Це може обмежити обчислювальну складність використовуваних алгоритмів та збільшити час обробки даних.

Для створення рекомендаційної системи необхідно притримуватись таких характеристик як точність передбачення, корисність, охоплення, впевненість, довіра, інтуїтивність, новизна, різноманітність, надійність, конфіденційність. В таблиці 1.1 наведено докладний опис ключових характеристик [8].

Таблиця 1.1 – Опис ключових характеристик рекомендаційної системи

Назва	Опис
Точність передбачення	В основі переважної більшості рекомендаційних систем лежить механізм прогнозування. Цей механізм може передбачати думку користувачів щодо елементів (наприклад, рейтинги фільмів) або ймовірність використання (наприклад, покупка). Основне припущення в рекомендаційній системі полягає в тому, що користувач віддасть перевагу системі, яка забезпечує точніші прогнози.

Продовження таблиці 1.1

Корисність	<p>Цінність, яку система або користувач отримує від рекомендації. Наприклад, веб-сайти електронної комерції, такі як Wayfair або Amazon, використовують рекомендації, щоб збільшити свій дохід за рахунок покупок клієнтів. Однак користувальницькі вподобання важко охопити та змоделювати.</p> <p>У програмах, де користувачі оцінюють елементи, надана можливість використовувати рейтинги як допоміжну програму</p>
Охоплення	<p>Термін охоплення стосується частки елементів, які може рекомендувати рекомендаційна система. Найпростішою мірою для цього визначення є відсоток усіх елементів, які будь-коли можуть бути рекомендовані. Іншим показником є різноманітність продажів, яка вимірює, наскільки неоднаково різні товари вибирають користувачі, коли використовується певна рекомендаційна система.</p>
Впевненість	<p>Впевненість до рекомендації можна визначити як впевненість системи до її рекомендацій або прогнозів. Він зростає разом із обсягом даних. Наприклад, якщо онлайн-магазин рекомендує один комп'ютер з дуже високою впевненістю, а інший комп'ютер з аналогічними характеристиками, таким же рейтингом, але з нижчою впевненістю, покупець може відразу додати перший комп'ютер до кошика для покупок, але може додатково прочитати опис іншого продукту та, можливо, його опис, перш ніж вирішити його купити.</p>

Продовження таблиці 1.1

Довіра	Оскільки впевненість – це довіра системи до її рейтингів, то в даному випадку розглядається довіра користувача до рекомендацій системи. Наприклад, системі може бути корисно рекомендувати кілька предметів, які користувач уже знає та любить. Таким чином, хоча користувач не отримує жодної цінності від цієї рекомендації, він помічає, що система надає розумні рекомендації, які можуть збільшити її довіру до системних рекомендацій щодо невідомих елементів. Одним з методом оцінки довіри користувачів є опитування користувачів.
Інтуїтивність	Інтуїтивність можна розглядати як кількість актуальної інформації, нової для користувача в рекомендації. Наприклад, дотримуючись успішної рекомендації книги на Scribd, читач дізнається про нового автора, який їй подобається, що можна вважати інтуїтивною випадковістю.
Новизна	Нові рекомендації – це елементи, про які користувач не знав. Очевидним і простим у реалізації підходом є відфільтрування елементів, які користувач уже оцінив або використав. Однак у багатьох випадках користувачі повідомлятимуть не про всі елементи, які вони використовували в минулому. Цього методу недостатньо, щоб відфільтрувати всі елементи, які користувач уже знає.

Кінець таблиці 1.1

Різноманітність	Різноманітність зазвичай визначається як протилежність подібності. У деяких випадках пропозиція набору подібних елементів може бути не такою корисною для користувача, оскільки дослідження діапазону елементів може зайняти більше часу.
Надійність	Стабільність рекомендації за наявності фальшивої інформації, яка зазвичай вводиться спеціально, щоб вплинути на рекомендації. Наприклад, господар Airbnb може підвищити рейтинг своїх оголошень, вставивши підроблені профілі гостей, які позитивно оцінюють оголошення.
Конфіденційність	У більшості систем колаборативної фільтрації користувач охоче розкриває свої вподобання щодо елементів, щоб отримати корисні рекомендації. Однак для більшості користувачів важливо, щоб їхні налаштування залишалися приватними. Для рекомендаційних систем розкриття приватної інформації вважається неприйнятним навіть для одного користувача. З цієї причини аналіз конфіденційності зазвичай зосереджується на найгіршому сценарії, ілюструючи теоретичні випадки, за яких особиста інформація користувачів може бути розкрита.

1.2 Аналіз процесу побудови рекомендацій

Процес побудови містить в собі такі етапи – етап збору інформації, використання явного, неявного або гібридного зворотного зв'язку, етап навчання та етап передбачення/рекомендації.

На етапі збору інформації відбувається збір відповідної інформації про користувачів для створення профілю користувача або моделі для завдань прогнозування, включаючи атрибути користувача, поведінку або вміст ресурсів, до яких користувач отримує доступ. Рекомендаційний агент не може працювати точно, доки профіль/модель користувача не буде добре побудовано. Система повинна знати якомога більше інформації про користувача, щоб надати обґрунтовані рекомендації з самого початку. Рекомендаційні системи покладаються на різні типи вхідних даних, такі як зворотний зв'язок високої якості, який містить в собі дані користувачів щодо їхнього інтересу до елемента або неявний зворотний зв'язок шляхом опосередкованого висновку про переваги користувача через спостереження за поведінкою користувача [8]. Гібридний зворотний зв'язок також можна отримати шляхом поєднання явного та неявного зворотного зв'язку. На платформі онлайн-навчання профіль користувача – це набір персональних даних, пов'язаних із конкретним користувачем. Ця інформація включає когнітивні навички, інтелектуальні здібності, стилі навчання, інтереси, уподобання та взаємодію з системою. Профілі користувачів зазвичай використовуються для отримання інформації, необхідної для створення моделі користувача. Таким чином, профіль користувача описує просту модель користувача. Успіх будь-якої рекомендаційної системи значною мірою залежить від її здатності представляти поточні інтереси користувачів. Точні моделі необхідні для отримання відповідних і точних рекомендацій від будь-якого методу прогнозування.

Як правило, користувачам пропонується оцінити елементи через інтерфейс системи, щоб побудувати та вдосконалити свої моделі. Точність рекомендації залежить від кількості оцінок, наданих користувачем. Єдиним недоліком цього методу є те, що він вимагає зусиль від користувачів, а також користувачі не завжди готові надати достатньо інформації. Незважаючи на те, що явний зворотній зв'язок потребує більше зусиль з боку користувача, він все одно вважається наданням більш надійних даних, оскільки він не передбачає видалення налаштувань із дій, а також забезпечує прозорість у процесі рекомендацій, що призводить до дещо нижчої якості рекомендацій. . Більше довіри до рекомендацій [9].

Система автоматично визначає вподобання користувача, відстежуючи різні дії користувача, такі як історія покупок, історія навігації та час, проведений на певних веб-сторінках, посилання, за якими користувач переходить, вміст електронної пошти та натискання кнопок тощо. Неявний зворотний зв'язок знімає навантаження з користувача на основі його поведінки в системі. Хоча цей метод не вимагає зусиль від користувача, він менш точний. Крім того, також було стверджено, що неявні дані про переваги насправді можуть бути більш об'єктивними, оскільки немає упередженості, що виникає через те, що користувачі реагують соціально бажаним способом [9], і немає проблем із самооцінкою чи будь-якої потреби підтримувати імідж для інших [10].

Сильні сторони як неявного, так і явного зворотного зв'язку можна об'єднати в гібридній системі, щоб мінімізувати їхні слабкі сторони та отримати найкращу продуктивну систему. Цього можна досягти, використовуючи неявні дані як перевірку явного рейтингу або дозволяючи користувачеві надавати явний зворотний зв'язок лише тоді, коли він вирішує висловити явний інтерес.

На етапі навчання відбувається застосування алгоритму навчання для фільтрації та використання функцій користувача на основі відгуків, зібраних на етапі збору інформації.

На етапі рекомендації/передбаченні відбувається рекомендація або передбачення того, якому типу елементів може віддати перевагу користувач. Це може бути зроблено або безпосередньо на основі набору даних, зібраного на етапі збору інформації, яка може базуватися на пам'яті чи на основі моделі, або через спостережувану системою діяльність користувача (рисунок 1.4).



Рисунок 1.4 – Послідовність формування рекомендацій

1.3 Дослідження методів побудови рекомендацій в рекомендаційних системах

Використання ефективних і точних методів рекомендацій дуже важливо для надання необхідних та корисних рекомендацій користувачам системи. Для цього необхідно розуміти особливості і потенціал різних методів рекомендацій, знати анатомію різних методів фільтрації рекомендацій (рисунок 1.5).

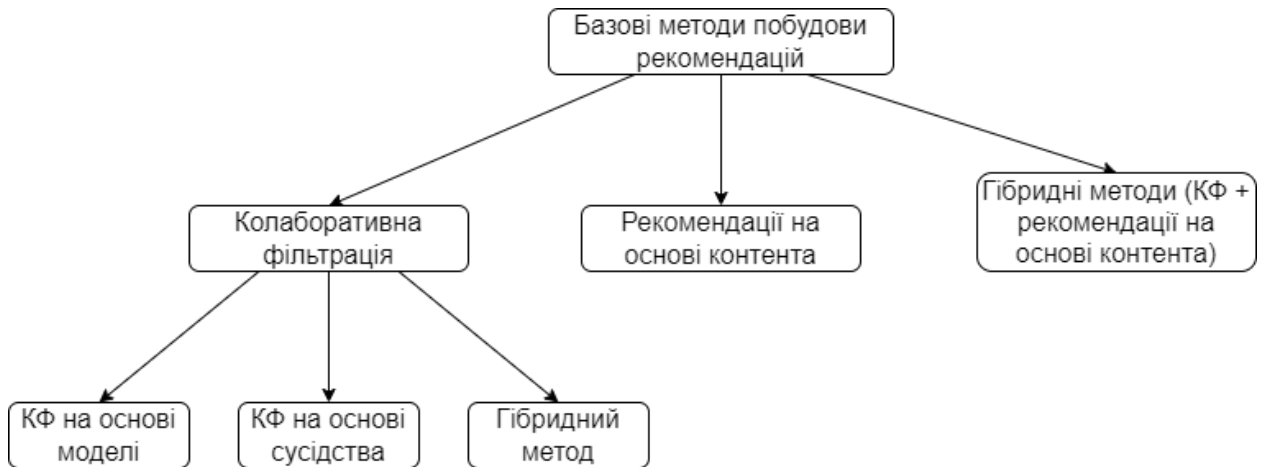


Рисунок 1.5 – Методи побудови рекомендаційних систем

Рекомендаційні системи працюють на двох рівнях:

- глобальні оцінки; особливості та переваги, що не змінюються місяцями або роками; Цікаві сторінки; залежність від характерних рис користувача: стать, місце проживання і так далі;
- короткочасні тренди та швидкі зміни інтересів у часі.

Дані збирають "явним" та/або "неявним" способами. У першому випадку відвідувачу пропонують заповнювати анкети, проходити опитування тощо. визначення його переваг. Метод ефективний, лише користувачі не завжди погоджуються на заповнення анкет та проходження опитувань. Тому вигадують, як зацікавити їх цим (зазвичай пропонують якусь вигоду в обмін).

Другий метод передбачає фіксування поведінки споживача на сайті або у додатку: які сторінки/розділи подивився, що додавав до кошика, які та де залишав коментарі тощо. За правильної організації збору даних та подальшої аналітики метод дає хороші результати. Але і тут є низка складнощів. По-перше, перегляд окремих сторінок чи карток товару не дає відомостей, чи сподобалося зрештою людині пропозицію. По-друге, навіть якщо покупка, складно припустити, що безпосередньо підштовхнуло людини до її вчиненню [6].

Як і у випадку з типами рекомендаційних систем окремо способи використовують рідко. Для отримання кращих результатів їх комбінують.

Правильно налаштований збір інформації дозволяє зробити рекомендації релевантними. З їхньою допомогою скорочується час пошуку необхідних товарів чи послуг, і навіть підвищується можливість здійснення супутніх цільових дій. Наприклад, споживач хотів купити лише смартфон, але в результаті додатково придбав чохол для нього. Коли споживач витрачає мінімум часу для пошуку потрібних речей та отримує бажане, його лояльність до платформи підвищується.

Ще один важливий момент у роботі рекомендаційних систем, який варто враховувати, – дотримання конфіденційності. Вони можуть передбачити такі результати та виявити закономірності, про які користувачі навіть не замислювалися або не хотіли, щоб про них хтось знав.

Існує 4 типи рекомендаційних систем:

- колаборативна фільтрація (collaborative filtering);
- засновані на контенті (content-based);
- засновані на знаннях (knowledge-based);
- гібридні (hybrid).

Колаборативна фільтрація (collaborative filtering)

Рекомендації ґрунтуються на історії оцінок як самого користувача, так і інших. У другому випадку система враховує споживачів, цінності чи інтереси яких схожі з вашими.

Такі системи почали активно розробляти в 90-х роках. У рамках цього підходу рекомендації генеруються на основі інтересів інших подібних користувачів. Такі рекомендації є результатом «співпраці» багатьох користувачів.

Типи колаборативних рекомендаційних систем:

Фільтрація на основі пам'яті: головним чином запам'ятовується матриця взаємодії користувача з елементом і те, як користувач на це реагує, тобто оцінка, яку користувач дає елементу. В основному дві секції:

Фільтрація між користувачами: у такому випадку, якщо характеристики користувача А схожі з деяким іншим користувачем Б, тоді продукти, які сподобалися Б, рекомендуються А. Як твердження, можна сказати: «користувачі, яким подобаються продукти, подібні до вас також сподобалися ці продукти» [3].

Якщо один користувач А поводить себе так само, як інші користувачі В, С і D, то для продукту x оцінка А визначається за формулою:

$$R_{xu} = \frac{(\sum_{i=0}^n R_i)}{n}, \quad (1.1)$$

де R_{xu} – рейтинг, наданий x користувачем u, а $i=0$ до n – це користувачі, які продемонстрували поведінку, подібну до u. Тепер усі n користувачів не є однаковою кількістю, подібною до користувача u. Отже знаходимо зважену суму для визначення рангу.

$$R_{xu} = \frac{(\sum_{i=0}^n R_i W_i)}{\sum_{i=0}^n W_i}, \quad (1.2)$$

Вагові коефіцієнти тут є використаними показниками подібності.

Тепер користувачі демонструють деякі відмінності в поведінці під час оцінювання. Деякі оцінюють кожен продукт на високому рівні, інші – ні, тобто, можливо, один користувач оцінює в діапазоні від 3 до 5, тоді як інші користувачі оцінюють від 1 до 3. Обчислюється середнє значення всіх оцінок, наданих користувачем, і віднімається значення від R_i , щоб нормалізувати оцінки кожного користувача [11].

Фільтрування елементів: якщо користувачеві А подобається елемент x, тоді елементи y і z, подібні до x за властивостями, тоді y і z рекомендуються користувачеві. Можна припустити: «Оскільки вам сподобалося це, вам також можуть сподобатися дані продукти» [12]. Рівняння 1.1-1.2 використовуються тут, де R – це оцінка, яку користувач дає продукту x, а це середнє значення

оцінок, які присутні для таких продуктів, як x . Також береться середньозважена величина, де вага – це подібність між продуктами.

Показники подібності – це математичні показники, які використовуються для визначення того, наскільки вектор схожий на даний вектор.

Найчастіше використовуються такі показники подібності:

- косинус подібності: косинусний кут між векторами;
- скалярний добуток: косинусний кут і величина векторів також мають значення;
- евклідова відстань: поелементно зведена в квадрат відстань між двома векторами;
- Подібність Пірсона: це коефіцієнт, який визначається як:

$$R_{xu} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \quad (1.3)$$

Фільтрація на основі моделі: в даному випадку не потрібно запам'ятовувати матрицю. З матриці необхідно дізнатися, як поводить себе конкретний користувач або предмет. Методом стискання великої матриці взаємодії за допомогою розмірного скорочення або за допомогою алгоритмів кластеризації. У цьому типі використовуються моделі машинного навчання, які намагаються передбачити, скільки оцінок користувач дасть продукту. Є кілька методів [11]:

- алгоритми кластеризації;
- алгоритм на основі матриці факторизації;
- методи глибокого навчання.

У класичній реалізації алгоритму є один явний мінус – він погано застосовується на практиці через квадратичну складність. Як і будь-який метод найближчого сусіда, він вимагає розрахунку всіх попарних відстаней між користувачами (а користувачів можуть бути мільйони). Незавжди

порахувати, що складність розрахунку матриці відстаней буде $O(n^2m)$, де n – число користувачів, а m – число товарів [11].

Персональна рекомендація передбачає максимальне використання інформації про самого користувача, в основному про його попередні покупки. Першими з'являються методи фільтрації на основі вмісту. У цьому підході опис продукту (контент) узгоджується з інтересами користувачів, отриманими з попередніх оцінок. Чим більше продукт відповідає цим інтересам, тим вище оцінка потенційного інтересу користувача. Очевидною вимогою є те, що всі позиції в каталозі повинні мати опис [13].

Історично темами рекомендацій на основі контенту зазвичай були продукти з неструктурованими описами: фільми, книги, статті. Наприклад, такими маркерами можуть бути текстові описи, коментарі, приведення тощо. Однак ніщо не заважає використовувати звичайні цифрові або категоріальні символи.

Неструктуровані ознаки описуються типовим текстовим способом – векторами в просторі слів (модель векторного простору). Кожен елемент такого вектора – ознака, що потенційно характеризує інтерес користувача. Аналогічно, продукт – вектор у тому самому просторі [3].

У міру взаємодії користувача із системою, векторні описи придбаних ним товарів об'єднуються (підсумовуються та нормалізуються) у єдиний вектор i , таким чином, формується вектор його інтересів. Далі досить знайти товар, опис якого найближче до вектора інтересів, тобто. розв'язати задачу пошуку n найближчих сусідів.

Не всі елементи однаково значущі: наприклад, союзні слова не мають жодного корисного навантаження. Тому щодо кількості співпадаючих елементів у двох векторах всі виміри потрібно попередньо зважувати за їх значимістю. Це завдання вирішує перетворення TF-IDF, яке призначає більшу вагу більш рідкісним інтересам. Збіг таких інтересів має більше значення щодо близькості двох векторів, ніж збіг популярних [13].

$$W_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right), \quad (1.4)$$

де вага слова x в описі товару y :

$tf_{x,y}$ = частота слова x в описі товару y

df_x = кількість товарів, що містять слово x

N = загальна кількість товарів

Принцип TF-IDF застосовний і до звичайних номінальних атрибутів, таких, як, наприклад, жанр, режисер, мова. TF – міра значущості атрибута для користувача, IDF – міра «рідкості» атрибута.

Деякі моменти, які можна врахувати при реалізації:

- при формуванні vector-space представлення товару замість окремих слів можна використовувати n -грами (послідовні пари слів, трійки тощо). Це зробить модель більш деталізованою, проте знадобиться більше даних для навчання;

- у різних місцях опису товару вага ключових слів може відрізнитися (наприклад, опис фільму може складатися з заголовка, короткого опису та детального опису);

- описи товару від різних користувачів можна зважувати по-різному. Наприклад, можемо давати більшу вагу активним користувачам, які мають багато оцінок;

- аналогічно можна зважувати і на товар. Чим більший середній рейтинг об'єкта, тим більша його вага;

- якщо опис товару допускає посилання зовнішні джерела, можна аналізувати також всю пов'язану з товаром сторонню інформацію.

Як міра близькості двох векторів найчастіше використовується косинусна відстань.

$$sim(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}, \quad (1.5)$$

При додаванні нової оцінки вектор інтересів оновлюється інкрементально (тільки за тими елементами, що змінилися). При перерахунку є сенс давати новим оцінкам трохи більше ваги, оскільки переваги можуть змінюватися [13].

Також до методів фільтрації рекомендацій відносяться наївний баєсів класифікатор, алгоритм Rocchio та дерева рішень.

Наївний баєсів класифікатор є імовірнісним алгоритмом машинного навчання і належить до загального класу класифікаторів Байєса. Байєсовські класифікатори будують свої моделі на основі попередніх спостережень, які використовуються як навчальні дані. Байєсівський метод оцінює апостеріорну ймовірність належності елементу d до класу c , яка є ймовірністю $P(c|d)$. Теорема Байєса виражається так:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}, \quad (1.6)$$

де апіорна ймовірність $P(c)$ – це ймовірність спостереження за елементом у класі c , ймовірність $P(d|c)$ – це ймовірність спостереження за елементом d , заданим класом c , а ймовірність $P(d)$ – це ймовірність спостереження за екземпляром d і елемент класифікується з використанням класу з найвищою ймовірністю

$$c = \operatorname{argmax}_{c_i} = \frac{P(c_j)P(d|c_j)}{P(d)}, \quad (1.7)$$

Алгоритм Rocchio застосовує техніку зворотного зв'язку релевантності, яка допомагає користувачам поступово уточнювати запити на основі попередніх результатів пошуку. Даний алгоритм представляє елементи як вектори, так що елементи з подібним вмістом мають подібні вектори. Кожен компонент такого вектора відповідає терміну в елементі, як правило, слову. Вага кожного компонента обчислюється за допомогою схеми зважування термінів TF-IDF. Навчання досягається шляхом об'єднання векторів

документів (позитивних і негативних прикладів) у вектор прототипу для кожного класу в наборі класів c . Щоб класифікувати новий документ d , обчислюється подібність між векторами прототипу та відповідним вектором документа, що представляє d . Для кожного класу (наприклад, за допомогою косинусної міри подібності), тоді d призначається класу, вектор елемента якого має найбільше значення подібності [12].

Дерево рішень – це набір вузлів, упорядкованих у вигляді бінарного дерева. Дуги дерева представляють рішення, а вузли містять класифіковані об'єкти. Дуги рішень можуть приймати такі значення, як true або false. Для кожного значення рішення дерево розширюється, що містить нові дочірні вузли, що містять нові рішення, доки не буде досягнуто аркуша. Для випадку дерева рішень істина-хибність, якщо рішення, яке посилається на кореневий вузол, є істинним, тоді слідуйте шляхом до правого дочірнього елемента, тоді як якщо воно хибне, слідуйте шляхом до лівого дочірнього елемента. Це повторюється для кожного дочірнього вузла, доки не буде досягнуто листок, де лист є останнім дочірнім вузлом шляху, і він містить класифікаційне значення елемента [11].

Рекомендаційні системи засновані на знаннях (knowledge-based) працюють на основі знань про якусь предметну область: про користувачів, товари та інші, які можуть допомогти в ранжируванні. Як і у випадку з «content-based», оцінки інших користувачів системи не враховують [6].

Рекомендаційна система базована на знаннях пропонує рекомендації, які вже існують у базах даних або базах знань, на які, таким чином, динамічно не впливають оцінки чи останні налаштування. Рекомендаційна система базована на знаннях поділяється на дві підкатегорії:

- системи на основі випадків базуються на міркуванні на основі випадків, які спираються на подібність між поточним випадком і рішеннями, які вже існують у базі даних. Взаємодія із системою на основі випадків складається з чотирьох етапів циклу: отримання, повторне використання, перегляд і збереження;

- системи з обмеженнями базуються на заданому наборі переваг і надають набір можливих рішень, включаючи пояснення, чому саме ці рішення були обрані. Набір обмежень може бути трьох типів: обмеження сумісності, умови фільтра та обмеження продукту.

Рекомендаційні системи на основі довіри (Trust-based Recommendation Systems) є вдосконаленням класичних методів рекомендацій, спрямованих на підвищення точності результатів рекомендацій, взятих із добре відомих методів колаборативної фільтрації. Логіка даної рекомендаційної системи полягає у використанні графіків, що представляють відношення між користувачами та елементами на основі їхнього зв'язку за певними атрибутами [14]. Дані рекомендаційні системи зазвичай використовуються в соціальних мережах, де в мережі підключено величезну кількість користувачів, і зазвичай користувачі підключаються через причину або ознаку. Покращення методів, заснованих на довірі, порівняно з традиційними методами співпраці передбачають заміну міри подібності, що використовується у формулах, на фактор довіри між парами користувачів. Trust – Based середньозважене значення:

$$r(u, i) = \frac{\sum_{v \in R^T} t_{u,v} r_{v,i}}{\sum_{v \in R^T} t_{u,v}}, \quad (1.8)$$

Trust – Based середньозважене значення представляє довіру між u та v користувачами як значення ваги замість міри подібності між u та v користувачів. Колаборативна фільтрація на основі довіри:

$$r(u, i) = \text{mean}(r_u) + \frac{\sum_{v \in R^T} t_{u,v} (r_{v,i} - \text{mean}(r_u))}{\sum_{v \in R^T} t_{u,v}}, \quad (1.9)$$

Ця формула є удосконаленням формули Резніка, де коефіцієнт кореляції Пірсона замінено фактором довіри t як ваговим фактором рейтингів. Відфільтрований довірою означає:

$$r(u, i) = \frac{i}{|RT|} \sum_{v \in R^T} r_{v,i}, \quad (1.10)$$

Усвідомлення контексту в RS передбачає використання даних, які характеризують об'єкт, щоб використовувати їх як контекстну інформацію для обчислення рекомендацій, де це необхідно. Завдяки загальному огляду RS стало помітно, що основні зусилля у створенні RS зосереджені на добре змодельованій інформації, яку RS використовує для своїх обчислень. Використовуючи останнє як принцип для розробки RS, контекстно-залежні рекомендаційної системи використовуються як засіб для збору цінної інформації, яка характеризує сутності в системі [15].

Гібридні (hybrid)

Різні підходи до створення рекомендаційних систем мають свої переваги і недоліки, в зв'язку з чим розумним видається поєднання кількох підходів в одній системі. Такі рекомендаційні системи, що базуються на комбінації описаних вище методів, називають гібридними. Гібридна система, що поєднує методи А та В, намагається використовувати переваги А, щоб виправити недоліки В [16].

Великі сервіси та інтернет-магазини використовують гібридні варіанти. Універсальної інструкції та рекомендацій щодо реалізації такого інструменту немає. Є кілька поширених типів комбінування:

- реалізація окремо колаборативних та контентних алгоритмів та поєднання їх припущень;
- включення деяких контентних правил до колаборативної методики;

- включення деяких колаборативних правил до контентної методики;

- побудова загальної моделі, що включає правила обох методик.

Зазвичай ці варіанти беруть як основу і доповнюють за власним бажанням та за критеріями сфери діяльності. Як і у випадку з knowledge-based, основним недоліком гібридних систем є складність розробки [16].

Оцінка рекомендаційних систем ділиться на три експериментальні рівні. Поділ оцінки на рівні полегшує оцінку RS шляхом порівняння їх один з одним. Рівні оцінювання залежать від атрибутів і характеристик, які має кожна система. Крім того, типи оцінювання, які можна застосувати до RS, залежать від моделі даних, яку використовує рекомендаційна система, домену, на який посилається RS, і типу очікуваних результатів. Оцінка рекомендаційних систем ділиться на наступні три рівні: офлайн-експерименти, дослідження користувачів і онлайн-оцінка. [16] Для кожного налаштування експерименту оцінки RS важливо дотримуватися наступних основних вказівок:

- перед проведенням експерименту необхідно сформулювати гіпотезу;

- під час порівняння алгоритмів за певною гіпотезою всі змінні, які не перевіряються, залишатимуться фіксованими;

- коли проводяться висновки з експериментів, витягнуті висновки повинні мати можливість узагальнюватися поза межами безпосереднього контексту експериментів.

Оцінка RS є доказом того, що розроблена система та її алгоритм рекомендацій дають бажані рекомендації. Таким чином, процес оцінки необхідно брати до уваги від фази проектування RS до її остаточної розробки [17]. Усвідомлення того, як буде застосовуватися рекомендаційна система, допомагає розробити відповідний експеримент оцінки, який відповідає тому, на чому зосереджено RS. Існують такі метрики оцінки для рекомендаційних систем:

- метрики на основі прогнозів – показники точності.

Показники прогнозування використовуються для порівняння того, яка рекомендаційна система робить менше помилок, коли робить висновок про те, як користувач оцінить запропоновану рекомендацію.

Показники точності вимірюють якість близькості до істини або справжнього значення, досягнутого системою і зазвичай точність вимірюється за допомогою наступного виразу:

$$accuracy = \frac{\text{number of successful recommendation}}{\text{number of recommendation}}, \quad (1.11)$$

Гнучкішим показником, який використовується як показник передбачення, є середня абсолютна помилка (MAE) [17]. MAE вимірює середнє абсолютне стандартне відхилення між кожним прогнозованим рейтингом і реальним вибором кожного користувача:

$$MAE = \frac{\sum_{i=1}^N |p_i - r_i|}{N}, \quad (1.12)$$

де p_i – прогнозований рейтинг, r_i – фактичний рейтинг, а N – загальна кількість прогнозів.

Для випадків із більшими відхиленнями від фактичних рейтингів замість MAE використовується середня квадратична помилка MSE.

$$MSE = \frac{\sum_{i=1}^N (p_i - r_i)^2}{N}, \quad (1.13)$$

- показники, пов'язані з пошуком інформації.

Інформаційний пошук відноситься до рекомендацій елементів на основі їх вмісту. Такими рекомендаціями можуть бути документи, рекомендовані з використанням методів фільтрації на основі тегів [17]. Показники, які використовуються для оцінки таких рекомендацій, не залежать від уподобань

користувача через оцінки, а переважно від корисності рекомендації чи задоволеності користувача. Найбільш часто використовуваними показниками є відгук та точність – це частка рекомендованих пунктів, які є актуальними, і вона виражається як:

$$Precision = \frac{|relevant\ items\ recommended|}{|items\ in\ the\ list|}, \quad (1.14)$$

Відгук визначається як частка релевантних рекомендацій, які надаються користувачеві, і виражається як:

$$Recall = \frac{|relevant\ items\ recommended|}{|relevant\ items|}, \quad (1.15)$$

Необхідність знати відповідні елементи для того, щоб мати можливість обчислити метрику відгуку, призвела до характеристики відгуку як непотрібною для оцінки рекомендаційних систем [17]. Подолання останньої характеристики, відгуку є корисним у випадках, коли оцінка RS виконується з використанням наборів даних, для яких рекомендовані елементи та налаштування користувача вже існують. Дисконтований кумулятивний приріст (DCG) використовується для вимірювання ефективності рекомендаційних елементів на основі їх порядку появи і визначається як:

$$DCG = \sum_i^p \frac{2^{rel_i-1}}{\log_2(1+i)}, \quad (1.16)$$

де p – позиція елемента в ранжованому списку, а rel_i – градуйована релевантність елемента на позиції i .

Користувач вважатиме список рекомендацій успішним, якщо він містить результати, про які він не знає, але в той же час задовольняє його смаки. Крім точності, для RS також дуже важливо показувати корисні та задовільні

рекомендації. Для розрізнення між точністю та задоволенням або точністю та корисністю було введено показники різноманітності, новизни та покриття. Новизна частини інформації, як правило, стосується того, наскільки вона відрізняється від «того, що бачив раніше» конкретним користувачем або спільнотою в цілому. Різноманітність, як правило, стосується набору предметів і пов'язана з тим, наскільки вони відрізняються один від одного. Охоплення відноситься до відсотка елементів, частини проблемної області, які рекомендаційна система може створити для користувача або групи користувачів [18]. Подібність внутрішнього списку для вимірювання різноманітності списку рекомендацій визначається як:

$$ILS(Pw_i) = \frac{\sum_{b_k \in \partial Pw_i} \sum_{b_e \in \partial Pw_i} c_0(b_k, b_e)}{2}, \quad (1.17)$$

де $c_0(b_k, b_e)$ є подібність між двома елементами b_k і b_e ($b_k \neq b_e$) і Pw_i це подібність списку всередині списку.

1.4 Постановка задачі дослідження

Існуючі підходи до побудови рекомендацій базуються на встановленні та подальшому використанні зв'язків між інтересами схожих користувачів, а також схожості характеристик рекомендованих товарів та послуг. Рекомендації формуються на основі порівняння довгострокових інтересів користувачів у режимі офлайн, а також короткострокових – у онлайн режимі. Однак недостатньо уваги приділяється зміні інтересів користувачів з часом, що потребує динамічної адаптації рекомендацій. Така проблема виникає, зокрема, внаслідок зміни демографічних характеристик користувачів (отримання освіти, зміна місця проживання, тощо). Вирішення цієї проблеми

може базуватись на використанні патернів, які враховують зміну інтересів користувачів з часом.

Об'єктом дослідження в рамках магістерської кваліфікаційної роботи є процес побудови рекомендацій щодо персоналізованого набору товарів та послуг для користувача.

Предметом дослідження являються методи побудови рекомендацій.

Метою даної роботи є дослідження методів побудови рекомендацій з урахуванням змін інтересів користувача.

Для досягнення мети, необхідно вирішити наступні задачі:

- аналіз характеристик рекомендаційних систем;
- дослідження методів побудови рекомендацій;
- удосконалення методу динамічної адаптації;
- експериментальна перевірка удосконаленого методу.

2 МЕТОД ДИНАМІЧНОЇ АДАПТАЦІЇ РЕКОМЕНДАЦІЙ

2.1 Побудова патернів зміни вподобань користувачів

Рекомендації для цільових користувачів формуються на основі інформації про вибір товарів і послуг схожими користувачами, а також інформації про рейтинги товарів. Рейтинги відображають ступінь популярності користувачів, але вони можуть бути сфальсифіковані в результаті атак [15, 16] віртуальних користувачів, або шилінгових атак. Рекомендація після шилінгу є спотвореною, оскільки вона відображає вимоги як цільових, так і атакуючих користувачів. Користувач не знає, як працює алгоритм побудови рекомендацій, не має інформації про шилінгову атаку, тому спотворені рекомендації в кінцевому підсумку змушують користувача відмовитися від використання рекомендаційної системи.

Для того, щоб користувач довіряв отриманим рекомендаціям і міг врахувати їх навіть у разі шилінгових атак, рекомендація поєднується зі зміною інтересів користувача [17-19]. Зміна інтересів показує причини і способи отримання рекомендації, встановлює асоціативні зв'язки між рекомендацією і знаннями предметної галузі. Поєднання рекомендації зі зміною інтересів підвищує шанси того, що користувач придбає рекомендований продукт або послугу.

Однак і рекомендації, і інтереси користувача повинні бути актуальними. Зміни інтересів можуть відбуватися циклічно або еволюційно. Тому при побудові рекомендацій важливо враховувати динаміку змін вимог і інтересів користувачів у часі, тобто описувати процес відбору користувачів за допомогою часових знань. Останні задають порядок у часі для кожної пари подій [21], коли користувач вибирає елементи, виставляє оцінки тощо. Набір

цих знань формує закономірності динаміки переваг користувача на обраному часовому інтервалі.

Для побудови рекомендацій з урахуванням змін переваг користувача необхідно виділити типові елементи темпоральних знань, побудови патернів з таких елементів на основі аналізу послідовності дій користувача. Правила встановлюють порядок у часі для пари подій. Імовірність реалізації правила визначається його вагою [18, 19].

На сьогоднішній день існують методи побудови пояснень з використанням темпоральних правил [23]. Перший метод використовує набір тимчасових правил, які визначають зміни в налаштуваннях користувача в заданий проміжок часу, наприклад місяць, рік. Пояснення у вигляді числового показника розраховуються з використанням ваг цих правил. Методика [23] формує опис процесу вибору споживачами конкретного товару у вигляді послідовності часових правил. Пояснення формується як числова оцінка процесу відбору користувача. Загальною рисою цих методів є те, що вони пояснюють користувачеві рекомендаційної системи причину рекомендації за допомогою знаку та значення числового показника пояснення. Знак вказує на збільшення або зменшення попиту на даний товар. Величина пояснювального показника відображає темпи збільшення або зменшення обсягу продажів за певний період часу. Недоліком цих методів є їх залежність від вибору основних часових правил, а також часового інтервалу, який використовується при побудові показника пояснення. Для усвідомленого вибору останнього необхідно виявити типові моделі групової поведінки користувачів. Ці патерни обмежені певним часовим інтервалом і складаються з послідовностей часових залежностей. Визначення таких закономірностей дозволить, по-перше, виділити основні часові залежності для побудови адаптивних рекомендацій методами [24, 25]. По-друге, інформація про тривалість патерна дає змогу визначити інтервали деталізації часу побудови пояснень.

Патерн динаміки споживчого вибору відображає типові зміни уподобань групи користувачів рекомендаційної системи у часі. Він поєднує кілька

послідовностей вибору різними користувачами однієї підмножини елементів та визначається набором подій вибору $A=\{a_i\}$. Такі події реєструються рекомендаційною системою або системою електронної комерції у вигляді журналу продажів, журналу дій користувача тощо.

Для кожної події в журналі вказується час її виникнення t_i , код вибраного товару, код користувача u_n , а також кількість вибраних елементів e_i .

Події впорядковані за часом, котрі можна описати часовими правилами двох типів. Правила наступного типу, або X-правила, визначають часовий порядок для пар подій (a_i, a_{i+1}) , коли друга подія відбувається одразу після першої. Тобто немає проміжних подій між двома виборами рекомендаційної системи або системи електронної комерції, які фіксують інші покупки товарів або послуг.

Такі правила, як Future або F-правила, визначають часовий порядок для пар подій (a_i, a_1) , між якими відбуваються інші події. Це правило визначає, що подія a_i має відбутися після майбутньої події a_1 .

Послідовний вибір користувачами кількох елементів представлений послідовністю X-правил. Якщо один із користувачів вибирає перший елемент, а потім пропускає вибір кількох елементів і вибирає останній елемент, його дії описуються F-правилом.

Оскільки користувачі можуть або послідовно вибирати одні й ті самі елементи в певний проміжок часу, або вибирати різні елементи, їхня поведінка описується моделями паралельно-альтернативних P1 і послідовний альтернативний вибір P2.

Структура таких патернів наведена в таблиці 2.1.

Таблиця 2.1 – Структура патернів динаміки переваг користувача

Патерн	Альтернативи	Представлення за темпоральними правилами
P1 – паралельно-альтернативний вибір	Послідовний вибір набору з кількох предметів або обрання першого і останнього елементів з набору	Правило F-типу або послідовність правил X-типу
P2 – послідовно-альтернативний вибір	Послідовний вибір кількох предметів, для яких реалізується паралельно-альтернативний вибір	Послідовність правил X-типу, за якою слідує кілька альтернатив між правилом F-типу або послідовністю правил X-типу

Патерн паралельно-альтернативного вибору P2 описує локальну ситуацію, коли користувачі реалізують лише альтернативну поведінку. Якщо поведінка користувачів рекомендаційної системи частково збігається, доцільно використовувати патерн P2 послідовного-альтернативного вибору. Комбінація цих закономірностей дозволяє описати вибір кількох альтернатив протягом одного періоду часу.

2.2 Удосконалений метод динамічної адаптації рекомендацій

Удосконалений метод динамічної адаптації рекомендацій передбачає використання патернів, які враховують зміну інтересів користувачів з часом.

Розроблений метод побудови патернів використовує в якості вхідних даних журнал подій рекомендаційної системи.

Метод містить у собі наступні етапи:

Етап 1. Підготовка даних.

Етап 2. Встановлення зв'язків між користувачами в режимі офлайн.

Етап 3. Формування набору рекомендацій в режимі офлайн.

Етап 4. Корегування рекомендацій в режимі онлайн.

Етап 5. Динамічна адаптація рекомендацій з використанням патернів.

Методика побудови патернів складається з наступних кроків.

Крок 1. Відбір інформації про зміну інтересів користувачів до рекомендованої в онлайн режимі групи товарів.

$$E_j = \langle e_{j,i}, e_{j,i+1}, \dots : (\forall i) \tau_i \in \Delta\tau \rangle, \quad (2.1)$$

де $\Delta\tau$ – часовий інтервал деталізації покупок.

Крок 2. Підготовка X залежностей виду $X_{j,i+1} = (e_{j,i}, e_{j,i+1})$ і F залежностей виду $F_{j,i+n} = (e_{j,i}, e_{j,i+n} : n > 1)$ для патернів.

Крок 3. Розрахунок ваги X і F залежностей.

$$w_{j,i+1} = e_{j,i+1} - e_{j,i} \quad (2.2)$$

$$w_{j,i+n} = e_{j,i+n} - e_{j,i} \quad (2.3)$$

Крок 4. Формування патерну у вигляді

$$P = \{(X_{j,i+1}, w_{j,i+1}), (F_{j,i+n}, w_{j,i+n})\} \quad (2.4)$$

Крок 5. Відбір рекомендацій для яких виконується умова:

$$(\sum_i w_{j,i+1} + \sum_{i,n} w_{j,i+n}) > 0 \quad (2.5)$$

Етап 6. Представлення рекомендацій користувачеві.

На рисунку 2.1 візуально представлено етапи удосконаленого методу динамічної адаптації рекомендацій.

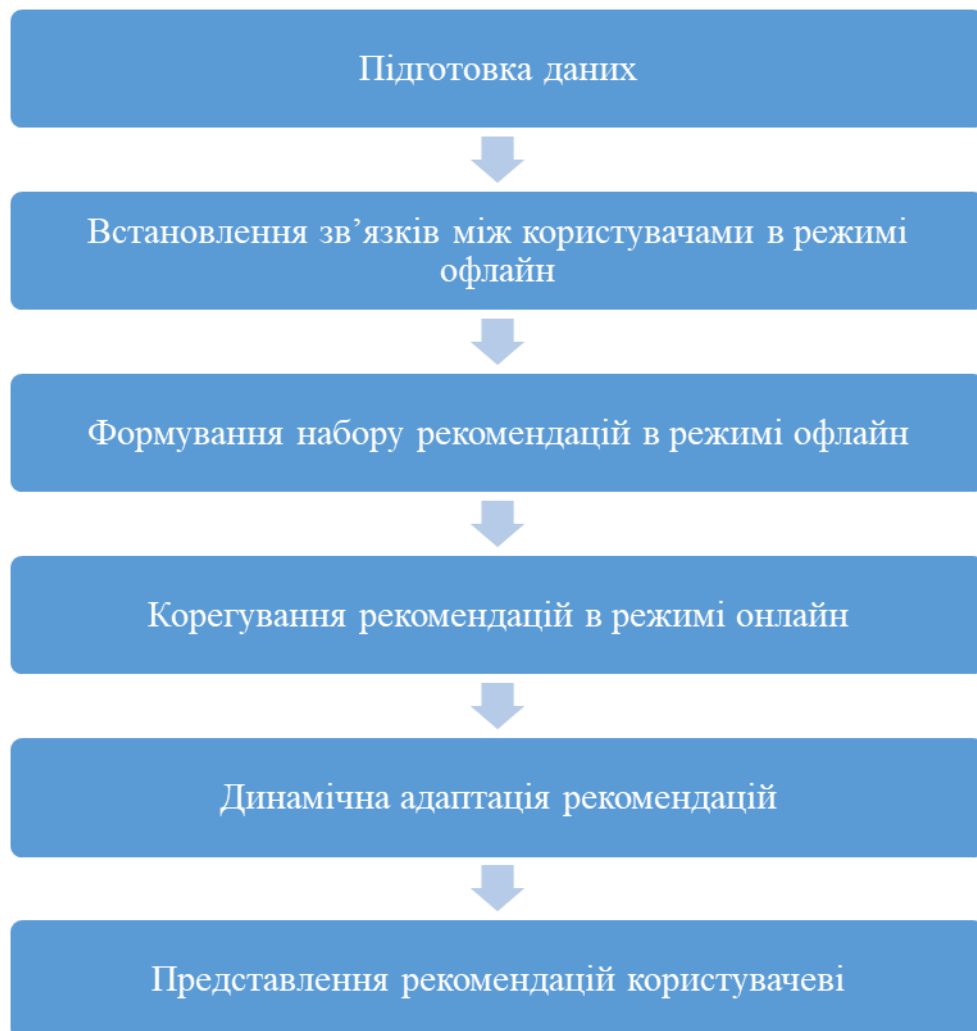


Рисунок 2.1 – Візуальне представлення етапів удосконаленого методу динамічної адаптації рекомендацій

3 РОЗРОБКА ТЕХНОЛОГІЇ ДИНАМІЧНОЇ АДАПТАЦІЇ РЕКОМЕНДАЦІЙ

Для зображення технології використання удосконаленого методу динамічної адаптації рекомендацій була використана методологія IDEF (Integration Definition), котра представляє собою набір стандартизованих методів і сімейство графічних мов, що використовуються для інформаційного моделювання в галузі інженерії програмного забезпечення (ПЗ), бізнес-процесів і об'єктів, а також для вдосконалення бізнес-процесів. Метою цих розроблених методів IDEF є надання можливості фахівцям осмислювати проблеми з різних точок зору та рівнів абстракції. У зв'язку з цим інтегровані методи IDEF представляють базовий інструментарій деяких сучасних стратегій і методологій вдосконалення бізнес-процесів.

Існує потужна програмна підтримка, яка інтегрує методи IDEF та дозволяє поєднати методи IDEF з іншими інструментами, такими як програмне забезпечення для моделювання бізнес-процесів, програмне забезпечення для управління витратами на основі діяльності тощо. Деякими інтегрованими методами IDEF є:

1. IDEF0 для функціонального моделювання;
2. IDEF1 для інформаційного моделювання;
3. IDEF1X для моделювання даних;
4. IDEF2 для імітаційного моделювання;
5. IDEF3 для моделювання процесів;
6. IDEF4 для об'єктно-орієнтованого проектування.

Метою функціонального моделювання є виявлення областей, де система є найслабшою та найбільш вразливою. Аналіз недоліків і «вузьких місць» починається з побудови моделі AS-IS (як є), тобто моделі існуючої системи. Отримана модель AS-IS використовується для виявлення дублювання роботи, роботи без ресурсів, неефективної роботи та інших недоліків у системі.

Виправлення недоліків і перенаправлення потоку інформації та матеріалів призводить до створення моделі, яка наближає ідеальну організацію процесу.

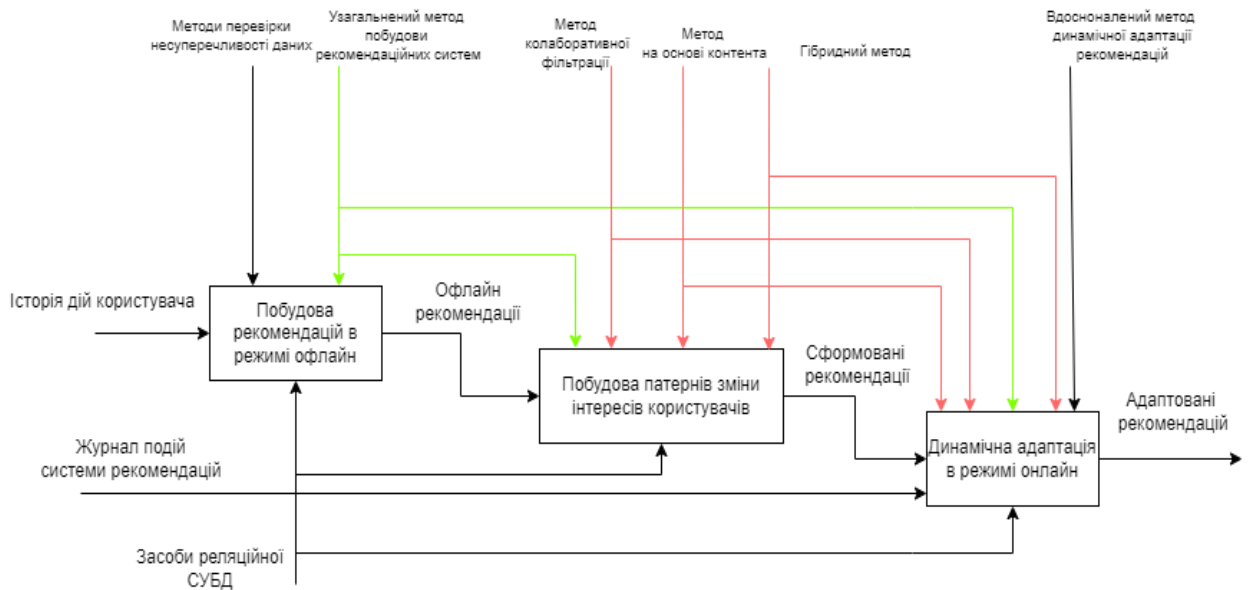


Рисунок 3.1 – Технологія використання удосконаленого методу

Спочатку рекомендації формуються в офлайн режимі, вхідними даними для формування є історія дій користувача. З допомогою методів побудови рекомендаційних систем відбувається процес уточнення рекомендацій, тобто наскільки рекомендації є актуальними або коректними для користувача. На третьому етапі відбувається динамічна адаптація рекомендацій в режимі онлайн з допомогою вдосконаленого методу в основі якого лежить побудова патернів динамічних вподобань користувача.

4 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА УДОСКОНАЛЕНОГО МЕТОДУ ДИНАМІЧНОЇ АДАПТАЦІЇ РЕКОМЕНДАЦІЙ

4.1 Програмна реалізація удосконаленого методу

Для збереження і обробки тестових даних була використана СУБД MySQL, оскільки вибірка даних не буде великою і не матиме складної обробки, а MySQL має унікальну архітектуру механізму зберігання, що робить його швидким. Також дана СУБД більш безпечна, оскільки складається з надійного рівня безпеки даних для захисту конфіденційних даних від зловмисників, а паролі зашифровані.

Для розробки та перевірки удосконаленого методу динамічної адаптації рекомендацій було обрано мову програмування Python, так як дана мова програмування надзвичайно читабельна, що зменшує складність кодування, надаючи чіткий і легко читабельний код. Це допомагає зрозуміти взаємозв'язок між різними об'єктами та спрощує весь життєвий цикл розробки на Python. Дана мова не дотримується стандартної конфігурації і фокусується на універсальності в розробці.

Також однією з найбільш значущих переваг програмування на Python є великі бібліотеки, що пропонуються для розробки. У ній є бібліотеки для всього, включаючи операції з рядками, інструменти сервісів, інтерфейси операційних систем, наукове програмування, статистику та обчислення.

В роботі будуть використовуватися такі пакети як pandas (високорівнева Python бібліотека для аналізу даних), numpy, Surprise – пакет з відкритим вихідним кодом для побудови рекомендаційної системи на основі даних рейтингу. Пакет надає всі необхідні інструменти для побудови рекомендаційної системи – від завантаження набору даних, вибору алгоритму прогнозування, до оцінки моделі.

TensorFlow, котрий містить бібліотеку для побудови рекомендаційної системи під назвою TensorFlow Recommenders. Як і інший пакет, TensorFlow Recommenders містить приклади наборів даних, алгоритми рекомендацій, оцінки моделей та розгортання.

Вся взаємодія користувачем проводиться через графічний інтерфейс, макет якого представлений на рисунках 4.1, 4.2.

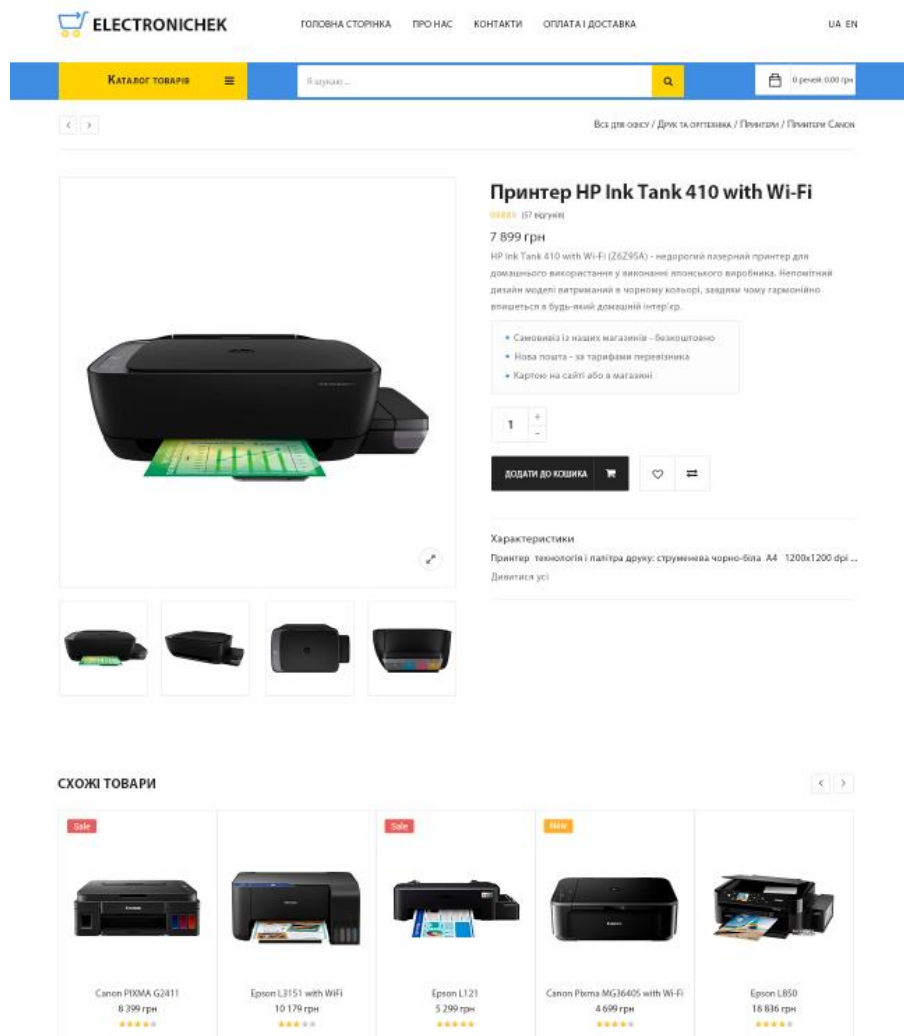


Рисунок 4.1 – Макет рекомендаційної системи

ELECTRONICHEK ГОЛОВНА СТОРІНКА ПРО НАС КОНТАКТИ ОПЛАТА І ДОСТАВКА UA EN

КАТАЛОГ ТОВАРІВ Я шукаю ... 0 реєстр. 0.00 грн

Всі для опису / Друж та оптимізація / Принтери / Принтери Canon

Принтер Canon i-SENSYS LBP6030B

8 811 грн

Canon i-SENSYS LBP6030 (8468B006) - недорогий лазерный принтер для домашнего использования у виконанні японського виробника. Непомітний дизайн моделі витриманий в чорному кольорі, завдяки чому гармонійно впишеться в будь-який домашній інтер'єр.

- Самовивіз із наших магазинів - безкоштовно
- Нова пошта - за тарифами перевізника
- Картою на сайті або в магазині

1

ДОДАТИ ДО КОШИКА

Характеристики
Принтер технологія і палітра друку: лазерна чорно-біла А4 2400x600 dpi ...
Дивитися усі

СХОЖІ ТОВАРИ






<p>Sale</p>  <p>Canon LBP-6030w 9 404 грн</p>	 <p>Xerox Phaser 3020BI 5 346 грн</p>	<p>Sale</p>  <p>Pantum P2207 4 668 грн</p>	<p>Sale</p>  <p>Canon i-Sensys LBP112 8 496 грн</p>	 <p>HP Laser 107a 7 107 грн</p>
--	--	---	---	--

Рисунок 4.2 – Макет рекомендаційної системи

Спочатку користувач переглядає один товар, а потім переходить на інший товар, який його цікавить. Як видно з рисунку перелік схожих товарів відразу генерується та відображається користувачу.

На наступних рисунках будуть відображені елементи коду, написані при виконанні кваліфікаційної роботи.

```

driver = 'DRIVER={SQL Server}'
server = 'SERVER=localhost'
port = 'PORT=1433'
db = 'DATABASE=testdb'
user = 'UID=me'
pw = 'PWD=pass'
conn_str = ';'.join([driver, server, port, db, user, pw])

conn = pyodbc.connect(conn_str)
cursor = conn.cursor()

cursor.execute('select * from table_name')
row = cursor.fetchone()
rest_of_rows = cursor.fetchall()

lensdir = '/glade/collections/cdg/data/CLIVAR_LE/cesm_lens/Amon/ts/'
outputdir = '/glade/work/rwills/python_output/forced_component/'
name = "cesm_lens"
varnam = 'ts'

```

Рисунок 4.3 – Код для підключення бази даних

```

if varnam == 'ts50':
    mask = mask[8:40, :]
    landmask = landmask[8:40, :]
try:
    ts_all = pickle.load(open(outputdir + name + "_" + varnam + "_all.p", "rb"))
    ts_clim_all = pickle.load(open(outputdir + name + "_" + varnam + "_clim_all.p", "rb"))
    lat = ts_all.lat
    lon = ts_all.lon
    time = ts_all.time
except:
    files = sorted(os.listdir(lensdir))
    files = [s for s in files if "rcp85" in s]
    n = len(files)
    ne = np.empty(n)

    filename0 = lensdir + files[0]
    ds0 = xr.open_dataset(filename0)
    lon = ds0.lon
    lat = ds0.lat
    time = ds0.time

```

Рисунок 4.4 – Код для роботи з даними

```

S = np.matmul(pvec[:, 0:neof], np.diag(1 / s[0:neof]))
Sadj = np.matmul(np.diag(s[0:neof]), pvec[:, 0:neof].T)

ensmeanPCs = np.matmul(X_ensmean_flat.values, S) # ensemble-mean principal components

gamma = np.cov(ensmeanPCs.T) # covariance matrix of ensemble-mean principal components

u2, signal_frac, v2 = np.linalg.svd(gamma)

SNP = np.matmul(v2, Sadj)
SNPs_resaped = SNP.reshape(neof, len(lat), len(lon)) / scale.values[None, :, None]

weights = np.matmul(S, v2.T)
weights = weights.reshape(len(lat), len(lon), neof) * scale.values[:, None, None]

weights = weights.reshape(len(lat) * len(lon), neof)

tk = np.matmul(Xt_flat.values, weights) # compute timeseries from full data matrix
tk_emean = np.matmul(Xt_ensmean_flat.values, weights) # compute ensemble-mean timeseries from ensemble-mean data

sign_eof = np.ones((neof, 1))

```

Рисунок 4.5 – Код для побудови патерну

```

print(signal_frac[0:4])
plt.plot(signal_frac, marker='o')
plt.xlim(0, 4)
plt.title('AUC ROC')

# signal_frac_check = np.zeros(5)
# for ii in range(5):
#     signal_frac_check[ii] = np.mean(np.square(tk_emean[:,ii]))/np.mean(np.square(tk[:,ii]))

# print(signal_frac_check)
# f=plt.figure()
# plt.plot(signal_frac_check,marker='o')
# plt.xlim(0,4)
# plt.title('Signal Fraction Check')

```

Рисунок 4.6 – Код для виводу оцінки рекомендацій

```

AUC ROC before [0.776, 0.782, 0.803, 0.827]

AUC ROC after [0.795, 0.804, 0.826, 0.847]

```

Рисунок 4.7 – Результати виконання програми

4.2 Експериментальна перевірка удосконаленого методу

Для оцінки результатів експерименту була використана крива система помилок AUC – ROC.

ROC-крива створюється шляхом побудови графіка істинно-позитивної частоти (TPR) проти частоти хибних спрацьовувань (FPR) при різних порогових значеннях.

ROC-аналіз надає інструменти для вибору можливих оптимальних моделей і відкидання неоптимальних незалежно від контексту витрат або розподілу класів. ROC-аналіз безпосередньо і природно пов'язаний з аналізом витрат/вигод при прийнятті діагностичних рішень.

AUC означає "Площу під кривою ROC". Тобто AUC вимірює всю двовимірну область під усією кривою ROC.

Значення AUC варіюється від 0 до 1. Модель, прогнози якої на 100 % невірні, має значення AUC, що дорівнює 0,0; той, чий прогнози вірні на 100%, має AUC 1,0 (таблиця 4.1).

Таблиця 4.1 – Емпіричне правило для інтерпретації значення AUC

AUC = 0.5	Ніякого зв'язку, наприклад, навмання підкинути монету.
$0.6 \geq \text{AUC} > 0.5$	Слабий зв'язок.
$0.7 \geq \text{AUC} > 0.6$	Прийнятний зв'язок.
$0.8 \geq \text{AUC} > 0.7$	Сильний зв'язок.
AUC > 0.9	Винятковий зв'язок.

Крім того, AUC також можна використовувати, щоб знайти оптимальне порогове значення для певного тесту, а також порівняти ефективність двох або більше альтернативних тестів.

AUC бажаний з наступних двох причин:

- AUC не залежить від масштабу. Він вимірює, наскільки добре ранжуються прогнози, а не їх абсолютні значення.

- AUC не залежить від порога класифікації. Він вимірює якість прогнозів моделі незалежно від вибраного порога класифікації.

Побудуємо ROC-криву для рекомендацій до вдосконалення методу з допомогою даних, отриманих на етапі програмної реалізації (рис.4.8).

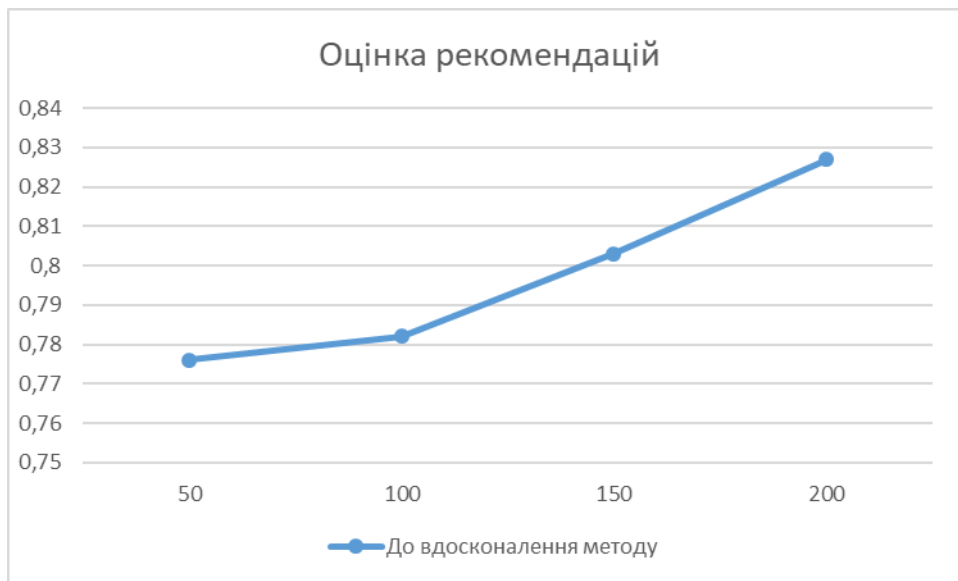


Рисунок 4.8 – Оцінка рекомендацій без динамічної адаптації

Аналогічним чином будемо ROC-криву для рекомендацій після використання вдосконаленого методу з допомогою даних, отриманих на етапі програмної реалізації (рис.4.9).



Рисунок 4.9 – Оцінка рекомендацій з динамічною адаптацією

Як видно з графіків результати вдосконалених рекомендацій більше наближені до вертикальної осі та мають більший емпіричний зв'язок.

Відобразимо дві криві на одному графіку (рис.4.10) та виведемо значення результатів в таблиці 4.2.

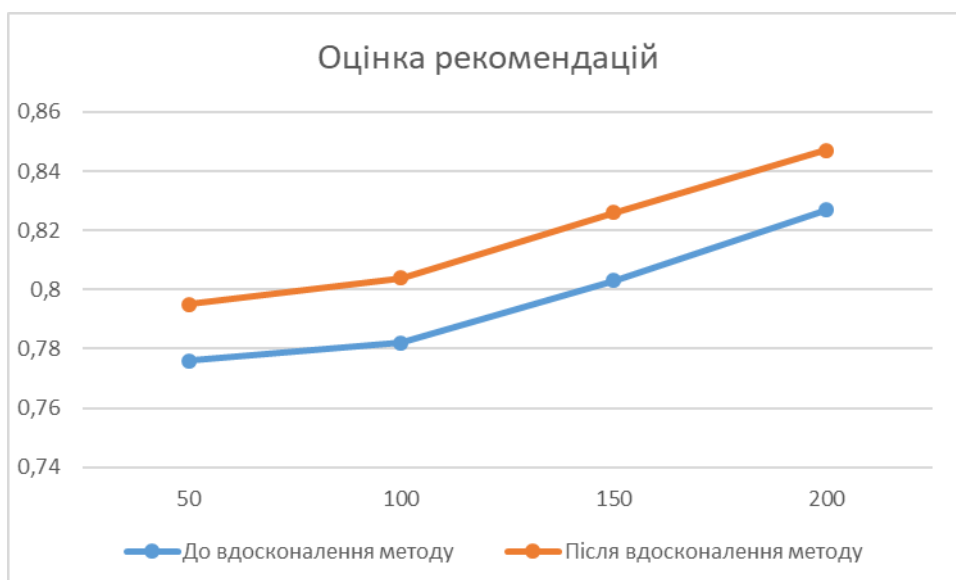


Рисунок 4.10 – Графічне порівняння оцінки рекомендацій двох методів

Таблиця 4.2 – Оцінка рекомендацій на тестових вибірках

Кількість записів	50	100	150	200
AUC ROC до вдосконалення	0,776	0,782	0,803	0,827
AUC ROC після вдосконалення	0,795	0,804	0,826	0,847

Як видно з таблиці вдосконалений метод в середньому працює краще і при збільшенні вибірки елементів збільшується і значення AUC ROC, а отже чим більше користувачів в системі і чим більше запитів вони дають системі тим якісніше виходять результати.

Приклад отриманих рекомендацій та вхідних даних для початкового і вдосконаленого методу наведені на рисунках 4.11 та 4.12.

```

ItemId      ItemName
18          Kyocera MA2000W
207         HP LaserJet Pro M428fdw with Wi-Fi, DADF
114         Canon PIXMA Ink Efficiency E414
92          Xerox WorkCentre 3025BI Wi-Fi
67          Epson L805 with WI-FI
34          HP Laser 107w with Wi-Fi

```

Рисунок 4.11 – Рекомендації користувача без динамічної адаптації

```

ItemId      ItemName
114         Canon PIXMA Ink Efficiency E414
67          Epson L805 with WI-FI
34          HP Laser 107w with Wi-Fi

```

Рисунок 4.12 – Рекомендації користувача з динамічною адаптацією

Аналіз позицій користувача показує наступне. При використанні вдосконаленого методу кількість рекомендованих товарів зменшилася.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було проаналізовано основні характеристики рекомендаційних систем та процес їх побудови. Було досліджено методи побудови рекомендаційних систем та виявлено проблему динамічної адаптації рекомендацій, так як зміні інтересів користувачів з часом приділяється недостатньо уваги. Вирішенням проблеми стало використання патернів, які враховують зміну інтересів користувачів з часом.

При експериментальній перевірці удосконалений метод показав кращі результати, аніж метод без використання адаптації. Також при використанні вдосконаленого методу кількість рекомендованих товарів зменшилася.

Результати даної роботи можуть бути використані практично при вирішенні задач, що потребують динамічної адаптації в рекомендаційних системах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки до передатестаційної практики для студентів усіх форм навчання спеціальності 122 – Комп’ютерні науки, освітньо-професійної програми "Інформаційні управляючі системи та технології" / Упоряд.: Чалий С.Ф., Євланов М. В., Чала О. В. – Харків: ХНУРЕ, 2021
2. Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems / Yehuda Koren, Robert Bell, and Chris Volinsky // Computer. – 2009. – № 8. – pp. 30–37.
3. Recommender Systems: Algorithms and Applications 1st Edition / S. Vairachilai, Sirisha Potluri, P. Pavan Kumar, Sachi Nandan Mohanty., 2021. – 230 с.
4. Gerardus Blokdyk. Recommender system Second Edition / Gerardus Blokdyk., 2021. – 310 с.
5. Recommender Systems: Issues, Challenges, and Research Opportunities [Електронний ресурс] – Режим доступу до ресурсу: <https://fardapaper.ir/mohavaha/uploads/2019/03/Fardapaper-Recommender-Systems-Issues-Challenges-and-Research-Opportunities.pdf>
6. A study of the dynamic features of recommender systems [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/257512777_A_study_of_the_dynamic_features_of_recommender_systems.
7. System Architectures for Personalization and Recommendation [Електронний ресурс] – Режим доступу до ресурсу: <https://netflixtechblog.com/system-architectures-for-personalization-and-recommendation-e081aa94b5d8>
8. Oard DW, Kim J. Implicit feedback for recommender systems. In: Proceedings of 5th DELOS workshop on filtering and collaborative filtering; 1998. p. 31–6.

9. Gadanho SC, Lhuillier N. Addressing uncertainty in implicit preferences. In: Proceedings of the 2007 ACM conference on Recommender Systems (RecSys '07). ACM, New York, NY, USA; 2007. p. 97–104.
10. Buder J, Schwind C. Learning with personalized recommender systems: a psychological view. *Comput Human Behav* 2012;28(1): 207–16.
11. Xiaoyuan Su and Taghi M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques / Xiaoyuan Su and Taghi M. Khoshgoftaar // Hindawi Publishing Corporation. – Advances in Artificial Intelligence archive, USA : 2009. – p. 1-19.
12. Linden G. Amazon.com recommendations: Item-to-item collaborative filtering / G. Linden, B. Smith, J. York // *IEEE Internet Computing*. – 2003. – №7. – pp. 76–80.
13. Recommender Systems: Behind the Scenes of Machine Learning-Based Personalization [Электронный ресурс] – Режим доступа до ресурсу: <https://www.altexsoft.com/blog/recommender-system-personalization/>
14. Recommendation systems: Principles, methods and evaluation [Электронный ресурс] – Режим доступа до ресурсу: <https://reader.elsevier.com/reader/sd/pii/S1110866515000341?token=92C35706F93C23367C9893B4842E3675EAD730946EEF69253261FDA94AFE8C653D118578A0B9260972521F49024727F4&originRegion=eu-west-1&originCreation=20221207161958>
15. Chala O., Novikov, L. and Chernyshova, L. (2019), “Method for detecting shilling attacks in e-commerce systems using weighted temporal rules”, *EUREKA: Physics and Engineering*, vol. 5, pp. 29-36. doi: 10.21303/2461-4262.2019.00983
16. Chala O., Novikov, L. and Chernyshova, L. (2020), “Method for detecting shilling attacks in recommendater systems using objective feedback”, *EUREKA: Physics and Engineering*, vol. 5, pp.. 21-30. doi: 10.21303/2461-4262.2020.001394.

17. Chalyi S., Leshchynskyi V. and Leshchynska I. (2019), "The concept of designing explanations in the recommender systems based on the white box", *Control, navigation and communication systems*, Vol. 3 (55). pp. 156-160. DOI: <https://doi.org/10.26906/SUNZ.2019.3.156>.
18. Adomavicius, G. and Tuzhilin, A. (2005), "Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions", *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, iss. 6, pp. 734-749. <https://doi.org/10.1109/TKDE.2005.99>
19. Zachary C. L. (2016), "The mythos of model interpretability", *Communications of the ACM*, pp. 1-6.
20. Rabiou I., Naomie S., Aminu D., Akram O. Recommender system based on temporal models: A Systematic Review // *Applied Sciences*. – 2020. – Vol. 10(7). – P. 2204. doi: 10.3390/app10072204
21. Chalyi S., Pribylnova I. (2019), "The method of constructing recommendations online on the temporal dynamics of user interests using multilayer graph", *EUREKA: Physics and Engineering*, vol. 3, pp. 13-19. doi: 10.21303/2461-4262.2019.00894
22. Chalyi S., Leshchynskyi V. and Leshchynska I. (2019), Modeling explanations for the recommended list of items based on the temporal dimension of user choice. *Control, navigation and communication systems*, Vol. 6 (58), pp. 97-101. doi:<https://doi.org/10.26906/SUNZ.2019.6.097>.
23. Chu W. and Park S.-T. (2009), "Personalized recommendation on dynamic content using predictive bilinear models," *International Conference on World Wide Web*, pp.691-700. 10.1145/1526709.1526802
24. Chalyi S., Leshchynskyi V. (2020) "Method of constructing explanations for recommender systems based on the temporal dynamics of user preferences", *EUREKA: Physics and Engineering*, vol. 3, pp. 43-50.
25. Chalyi S., Leshchynskyi V., Leshchynska I. (2020), "Detailing explanations in the recommender system based on matching temporal knowledge", *Eastern-European Journal of Enterprise Technologies*, vol 4, no 2, pp. 6-13.

26. Basic Concepts and Architecture of a Recommender System [Электронный ресурс] – Режим доступа до ресурсу: https://www.alibabacloud.com/blog/basic-concepts-and-architecture-of-a-recommender-system_596642
27. Recommendation Systems Explained [Электронный ресурс] – Режим доступа до ресурсу: <https://towardsdatascience.com/recommendation-systems-explained-a42fc60591ed>
28. Introduction To Recommender Systems- 1: Content-Based Filtering And Collaborative Filtering [Электронный ресурс] – Режим доступа до ресурсу: <https://towardsdatascience.com/introduction-to-recommender-systems-1-971bd274f421>
29. Research Problems in Recommender systems [Электронный ресурс] – Режим доступа до ресурсу: <https://iopscience.iop.org/article/10.1088/1742-6596/1717/1/012002/pdf>
30. Recommendation System Algorithms: An Overview [Электронный ресурс] – Режим доступа до ресурсу: <https://www.kdnuggets.com/2017/08/recommendation-system-algorithms-overview.html>
31. Recommendation System Series Part 8: The 14 Properties To Take Into Account When Evaluating Real-World Recommendation Systems [Электронный ресурс] – Режим доступа до ресурсу: <https://towardsdatascience.com/recsys-series-part-8-the-14-properties-to-take-into-account-when-evaluating-real-world-recsys-f71cc6e1f195>
32. A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System [Электронный ресурс] – Режим доступа до ресурсу: <https://iopscience.iop.org/article/10.1088/1742-6596/1000/1/012101/pdf>
33. Recommender Systems Review: Types, Techniques and Applications [Электронный ресурс] – Режим доступа до ресурсу: <http://www2.cs.ucy.ac.cy/~george/files/IGI15.pdf>
34. Functional and Information Modeling of Production Using IDEF

Methods [Електронний ресурс] – Режим доступу до ресурсу: https://www.svjme.eu/?ns_articles_pdf=/ns_articles/files/ojs3/1563/submission/1563-1-1898-1-2-20171103.pdf&id=4931.

35. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи (для студентів усіх форм навчання другого (магістерського) рівня програми "Інформаційні управляючі системи та технології) / Упоряд.:Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. - Харків: ХНУРЕ,2021.- 30с.