

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

"На правах рукопису"
УДК _____

«До захисту допущено»
Завідувач кафедри
Наталія АУШЕВА
“ ” _____ 2025 р.

Магістерська дисертація

на здобуття ступеня другого (магістерського) рівня вищої освіти
за освітньою програмою “Цифрові технології в енергетиці”
зі спеціальності 122 “Комп’ютерні науки”

на тему Інтелектуальні методи персоналізації на основі історії
користувача для рекомендаційних систем

Виконала: студентка 2 курсу, групи ТР-41мп

Гурин Ірина Андріївна

(прізвище, ім’я, по батькові)

(підпис)

Науковий керівник к.т.н., доцент Світлана ШАПОВАЛОВА

(науковий ступінь, вчене звання, ім’я ПРИЗВИЩЕ)

(підпис)

Рецензент доцент кафедри ТАЕ, к.т.н., доцент

Артур РАЧИНСЬКИЙ

(посада, науковий ступінь, вчене звання, ім’я ПРИЗВИЩЕ)

(підпис)

Н.контроль старший викладач Ольга БЕСПАЛА

(посада, ім’я ПРИЗВИЩЕ)

(підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студентка

(підпис)

Київ - 2025

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

Рівень вищої освіти – другий (магістерський)

спеціальність 122 “Комп’ютерні науки”

Освітньо-наукова програма “Цифрові технології в енергетиці”

ЗАТВЕРДЖУЮ

Завідувач кафедри ЦТЕ

Наталія АУШЕВА

«__» _____ 2025 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Гурин Ірині Андріївні

(прізвище, ім’я, по батькові)

1. Тема роботи Інтелектуальні методи персоналізації на основі історії користувача для рекомендаційних систем

керівник роботи Шаповалова Світлана Ігорівна, к.т.н., доцент

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «03» листопада 2025 р. № 4779-с

2. Термін подання студентом дисертації 06.12.2025

3. Об’єкт дослідження : методи та моделі програмної реалізації рекомендаційної системи.

4. Вихідні дані: персоналізовані списки рекомендованих бізнесів для кожного користувача.

5. Перелік завдань:

1) провести аналіз підходів до побудови рекомендаційних систем та методів персоналізації.

- 2) виконати попередню обробку та підготовку даних.
- 3) провести налаштування алгоритмічних моделей та провести порівняльну оцінку їх ефективності на основі метрик Precision@30, Recall@30 та F1-score@30
- 4) розробити програмний застосунок, для представлення отриманих результатів системи.
6. Орієнтований перелік ілюстративного матеріалу : схема роботи системи, схеми концептуальної роботи всіх методів фільтрації, приклади структури JSON-файлів, блок-схеми алгоритмів, ілюстрації отриманих результатів роботи системи.
7. Орієнтований перелік публікацій : тези доповідей на двох міжнародних конференціях

6. Дата видачі завдання «16» вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

	Назва етапів підготовки магістерської дисертації	Термін виконання	Примітка
1	Аналіз рекомендаційних методів та огляд сучасних досліджень	02-05.09.2025	Виконано
2	Обробка та підготовка даних Yelp Dataset	06-21.09.2025	Виконано
3	Програмна реалізація рекомендаційних методів: CF, CB та Hybrid	22-28.09.2025	Виконано
4	Проведення обчислювальних експериментів	29.09-12.10.2025	Виконано
5	Аналіз і порівняння результатів моделей	13-19.10.2025	Виконано
6	Захист створеного програмного забезпечення	21.10.2025	Виконано
7	Оформлення магістерської дисертації	22.10-24.11.2025	Виконано

Студентка

(підпис)

Ірина ГУРИН

(ім'я, ПРИЗВИЩЕ)

Керівник роботи

(підпис)

Світлана ШАПОВАЛОВА

(ім'я, ПРИЗВИЩЕ)

АНОТАЦІЯ

Магістерська дисертація виконана на 92 сторінках, містить 17 ілюстрацій, 3 додатки та 31 джерело в переліку посилань.

Мета роботи - визначення методу рекомендацій, який надає найбільш точні та релевантні рекомендації, та створення програмної системи для демонстрації результатів проведених експериментів.

Методи та засоби: мова програмування Python, середовища PyCharm та Jupyter Notebook, бібліотеки NumPy, Pandas, Scikit-learn, Implicit, фреймворк FastAPI, веб-технології HTML/CSS/JavaScript, система контролю версій Git/GitHub.

Результат – веб-застосунок, що демонструє роботу рекомендаційної системи та дозволяє переглядати персоналізовані рекомендації, сформовані на основі обчислених моделей.

Роботу було апробовано на двох конференціях: XXII Міжнародній науково-практичній конференції молодих вчених та студентів “Сучасні проблеми наукового забезпечення енергетики” [1] та II Міжнародній науково-практичній конференції “Progressive Approaches in Science and Engineering” [2].

Результати магістерської дисертації були використані під час розробки концепції стартапу Motiv AI – цифрового помічника для підвищення продуктивності студентів [3]. Зокрема, підходи до персоналізації та методи обробки користувацьких даних стали основою механізмів рекомендацій у даному проєкті.

Ключові слова: рекомендаційна система, колаборативна фільтрація, контентно-орієнтована фільтрація, гібридна фільтрація, персоналізація, Python, REST API, веб-застосунок.

ABSTRACT

The master thesis consists of 92 pages, contains 17 illustrations, 3 appendix, and 31 sources in the list of references.

The purpose of the work is to determine the recommendation method that provides the most accurate and relevant recommendations and to create a software system to demonstrate the results of the experiments.

Methods and tools: Python programming language, PyCharm and Jupyter Notebook environments, NumPy, Pandas, Scikit-learn, Implicit libraries, FastAPI framework, HTML/CSS/JavaScript web technologies, Git/GitHub version control system.

The result is a web application that demonstrates the operation of the recommendation system and allows viewing personalized recommendations based on calculated models.

The work was approbated at two conferences: the XXII International Scientific and Practical Conference of Young Scientists and Students “Modern Problems of Scientific Support for Energy” [1] and the II International Scientific and Practical Conference “Progressive Approaches in Science and Engineering” [2].

The results of the master's thesis were used in the development of the Motiv AI startup concept – a digital assistant for improving student productivity [3]. In particular, approaches to personalization and methods of processing user data became the basis for recommendation mechanisms in this project.

Keywords: recommendation system, collaborative filtering, content-based filtering, hybrid filtering, personalization, Python, REST API, web application.

ЗМІСТ

ВСТУП.....	8
1 МЕТОДИ ФОРМУВАННЯ ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ.....	11
1.1 Задача визначення рекомендацій	11
1.2 Огляд та класифікація статистичних методів рекомендацій	13
1.3 Колаборативна фільтрація	15
1.4 Контентно-орієнтована фільтрація	17
1.5 Гібридна фільтрація	19
Висновки до розділу 1	21
2 ПІДГОТОВКА ТА ОБРОБКА ДАНИХ	24
2.1 Джерела даних для побудови рекомендаційної системи	24
2.1.1 Дані про бізнес.....	24
2.1.2 Дані про відгуки користувачів.....	26
2.2 Підготовка текстових даних для аналізу	28
2.3 Побудова профілів користувачів і бізнесів.....	30
2.3.1 Профілі в колаборативній фільтрації	30
2.3.2 Профілі в контентно-орієнтованій фільтрації.....	32
Висновки до розділу 2.....	35
3 АНАЛІЗ ТА НАЛАШТУВАННЯ МОДЕЛЕЙ.....	36
3.1 Вибір та характеристика тестових груп користувачів	36
3.2 Налаштування моделі колаборативної фільтрації.....	37
3.3 Налаштування контентно-орієнтована фільтрація	38
3.4 Налаштування гібридної фільтрації	39
3.5 Аналіз експериментальних результатів	40
3.5.1 Результати колаборативної фільтрації	41

3.5.2	Результати контентно-орієнтованої фільтрації.....	43
3.5.3	Результати гібридної фільтрації	47
	Висновки до розділу 3	50
4	ПРОГРАМНА РЕАЛІЗАЦІЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ	51
4.1	Загальна архітектура системи.....	51
4.1.1	Структура файлів та модулів	52
4.1.2	Архітектура серверної частини та API.....	57
4.2	Інструменти розробки та середовище виконання	59
4.3	Програмне середовище	61
	Висновки до розділу 4.....	65
5	ОСНОВНІ ПОЛОЖЕННЯ СТАРТАП-ПРОЕКТУ	66
5.1	Формування ідеї продукту та її ринковий контекст.....	66
5.2	Загальні дані стартап-проєкту	67
5.3	Аналіз конкурентів та унікальні характеристики продукту	68
5.4	Технологічна здійсненність проєкту.....	71
5.5	Ринкові можливості, загрози та потенціал комерціалізації	72
5.6	Напрями подальшого розвитку продукту	73
	Висновки до розділу 5	75
	ВИСНОВКИ	76
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	77
	ДОДАТОК А	81
	ДОДАТОК Б	85
	ДОДАТОК В.....	92

ВСТУП

Впровадження персоналізованих інтелектуальних систем обумовлено стрімким розвитком цифрового середовища та активною інтеграцією онлайн-платформ для пошуку товарів, послуг та місцевих бізнесів. Сьогодні користувачі хочуть отримувати рекомендації, які найкраще відповідають їхнім уподобанням, витрачаючи на пошук якомога менше часу. Водночас для компаній важливо підвищити зацікавленість клієнтів та побудувати довгострокові відносини. В результаті системи рекомендацій стали фундаментальною технологією, яка дозволяє цифровим платформам працювати в таких сферах, як електронна комерція, ресторани, послуги бронювання, мобільні додатки та соціальні платформи.

Однак впровадження персоналізованих систем стикається з низкою проблем:

- багато платформ не мають достатньо даних для формування якісних рекомендацій;
- традиційні алгоритми часто демонструють нестабільну роботу, коли поведінка користувачів є нерівномірною або неактивною;
- комерційні рішення (такі як AWS Personalize або Google Vertex AI) часто є занадто дорогими або технологічно складними для малих компаній.

Аналіз сучасних підходів показує, що більшість методів зосереджуються на поведінковій взаємодії або властивостях об'єктів. Проте жоден з них окремо не гарантує стабільно високих результатів у реальному використанні. Все це визначає необхідність розробки персоналізованої системи рекомендацій, яка може адаптуватися до різних обсягів даних, поєднувати статистичні підходи та забезпечувати високий рівень персоналізації для широкої аудиторії. Нерівномірність даних користувачів є особливою проблемою, оскільки профілі варіюються від надзвичайно активних до майже пасивних. Це ускладнює процес холодного старту і значно знижує точність стандартних моделей. Тому завдання полягає в дослідженні алгоритмів в контексті їх роботи в різних конфігураціях моделей і з різним рівнем деталізації даних профілів користувачів.

Мета магістерської роботи полягає у визначенні методу рекомендацій, який забезпечує найвищу точність та релевантність рекомендацій, шляхом порівняння статистичних підходів та створення програмної системи для демонстрації результатів проведених експериментів.

Для досягнення поставленої мети визначено такі **завдання**.

1. Провести аналіз підходів до побудови рекомендаційних систем та методів персоналізації.
2. Виконати попередню обробку та підготовку даних.
3. Провести налаштування алгоритмічних моделей та провести порівняльну оцінку їх ефективності на основі метрик Precision@30, Recall@30 та F1-score@30
4. Розробити програмне середовище, для представлення отриманих результатів системи.

Об'єкт дослідження – методи та моделі програмної реалізації рекомендаційної системи.

Предмет дослідження – методи колаборативної, контентно-орієнтованої та гібридної фільтрації, способи їх комбінування та параметризації для підвищення якості персоналізації.

У роботі використано такі **методи дослідження**:

- методи обробки даних для підготовки структурованих даних системи;
- методи статистичного моделювання для побудови рекомендаційних моделей;
- методи машинного навчання: для обчислення рейтингових списків;
- метрики класифікації: для оцінки якості рекомендацій;
- методи програмної інженерії: для створення застосунку;
- методи порівняльного аналізу: для співставлення результатів моделей між різними групами користувачів.

Вибір методів обумовлений необхідністю оцінити ефективність моделей на нерівномірних даних та забезпечити відтворюваність результатів.

Отримані результати мають значну практичну вагу та можуть безпосередньо застосовуватися у створенні інформаційних систем персоналізації. У межах

проведених експериментів розроблено рекомендаційний застосунок, що функціонує через REST API та може бути інтегрований у будь-яку веб-платформу. Оптимальні конфігурації моделей для різних груп користувачів дозволяють адаптувати алгоритми до реальних умов використання, включаючи потреби активних, нерегулярних або майже неактивних користувачів. Створена архітектура системи є гнучкою та легко масштабованою: модулі обробки даних можна швидко переналаштувати для роботи з іншими наборами даних, а саму систему підлаштувати під локальні сервіси відгуків, маркетплейси, платформи місцевих бізнесів чи мобільні додатки. Це забезпечує основу для подальшого впровадження результатів дослідження у практичні рішення.

Результати роботи було обговорено на конференціях XXII Міжнародна науково-практична конференція молодих вчених та студентів “Сучасні проблеми наукового забезпечення енергетики” та II Міжнародна науково-практична конференція “Progressive Approaches in Science and Engineering”.

Структура кваліфікаційної роботи охоплює вступну частину та п'ять основних розділів. Дані розділи включають постановку задачі розробки рекомендаційної системи, підготовку та обробку даних, аналіз та налаштування моделей, програмну реалізацію рекомендаційної системи та основні положення стартап-проекту. Завершують роботу висновки та список використаних джерел. Загальний обсяг роботи складає 92 сторінок і включає 17 ілюстрацій, 3 додатки та 31 джерело у списку літератури.

1 МЕТОДИ ФОРМУВАННЯ ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ

У даному розділі наведено теоретичний матеріал, на якому було засновано побудову системи рекомендацій. Розділ містить аналіз основних статистичних підходів та принципів їх роботи. Окрему увагу приділено трьом ключовим, що були обрані для подальшої реалізації, а саме: колаборативна, контентно-орієнтована та гібридна фільтрації.

Продемонстровані методи є основою рекомендаційної системи, тому важливим етапом були їх аналіз та порівняння для розуміння їх особливостей.

1.1 Задача визначення рекомендацій

Розроблене програмне середовище призначене для дослідження та демонстрації роботи інтелектуальних методів персоналізації в рекомендаційних системах. Його основна мета – формування релевантних рекомендацій для користувача на основі його історії взаємодій – попередніх оцінок, переглядів та відгуків.

Система реалізує три підходи до побудови рекомендацій: колаборативний, контентно-орієнтований та гібридний метод. Таким чином, програмне забезпечення дає змогу проводити експерименти з різними алгоритмами, параметрами та оцінювати їх ефективність за обраними метриками.

Програмне забезпечення орієнтоване на три основні групи користувачів: дослідників, аналітиків даних та студентів, які вивчають принципи роботи сучасних систем персоналізації. Його можна використовувати як навчальну або дослідницьку платформу, яка дає змогу не лише оцінювати точність реалізованих моделей, а й розширювати систему новими методами фільтрації, удосконалювати та адаптувати наявні під власні потреби, а також інтегрувати їх з іншими алгоритмами машинного навчання.

У процесі реалізації використовується Yelp Open Dataset [4], який містить дані про бізнеси, користувачів та тексти їх відгуків. У процесі роботи різноманітні дані в числових, текстових і словникових форматах було очищено, нормалізовано та уніфіковано. Числові параметри були нормалізовані за типом, а категоріальні дані були перевірені та узгоджені.

Для аналізу текстових відгуків користувачів застосовано TF-IDF та LDA моделі, які дали змогу представити тексти у вигляді числових векторів і тематичних просторів. Це дозволило створити векторні профілі користувачів і бізнесів, що надалі використовувалися в методах системи рекомендацій.

У системі реалізовано три інтелектуальні методи фільтрації:

- колаборативний метод, визначає схожість між користувачами за їхніми оцінками і формує рекомендації на основі вподобань користувачів зі схожою поведінкою;
- контентно-орієнтований метод, базується на аналізі характеристик об'єктів (категорії бізнесу, ключові слова з відгуків, TF-IDF-вектори) та порівнює їх зі смаками користувача;
- гібридний метод, поєднує результати обох підходів, використовуючи ваговий коефіцієнт α для балансування між контентною та колаборативною складовими.

На вхід система отримує основні файли у форматі JSON:

- `yelp_academic_dataset_business.json` – дані про бізнеси (назва, категорії, геолокація, рейтинг, атрибути);
- `yelp_academic_dataset_review.json` – тексти відгуків користувачів, їхні оцінки та ідентифікатори бізнесів.

Після обробки даних з цих файлів і розрахунку показників схожості формуються персоналізовані списки рекомендованих бізнесів. На виході користувач отримує впорядкований перелік об'єктів, який відображає індивідуальні інтереси на основі попереднього досвіду взаємодії.

Для оцінювання ефективності наданих системою рекомендацій використано метрики Precision, Recall та F1-score [5], які дозволяють кількісно порівняти

результати різних методів. Під час експериментів досліджувалося, як зміна різноманітних гіперпараметрів та інших особливостей методів впливають на кінцевий результат точності рекомендацій.

Для представлення результатів роботи та перевірки функціональності розроблених методів створено програмне середовище. Воно дозволяє візуалізувати процес формування рекомендацій, аналізувати результати для різних груп та індивідуальних користувачів, а також оцінювати ефективність алгоритмів за обраними показниками. Додаток виконує роль зручного інструменту для аналізу та демонстрації роботи системи, забезпечуючи інтуїтивну взаємодію з отриманими результатами без потреби ручного запуску коду.

Розроблена система має ряд обмежень, зумовлених специфікою вхідних даних і форматом проведення експериментів. Зокрема, вона базується на готовому наборі даних Yelp Open Dataset, який містить уже зібрану інформацію. Це означає, що система функціонує виключно в офлайн-режимі, без можливості динамічного оновлення чи збору нових даних у реальному часі.

Крім того, серед обмежень варто відзначити відсутність інтеграції з веб-платформами та зовнішніми API, через що рекомендації створюються виключно на підставі наявних історичних даних. Однак такий підхід дозволяє забезпечити стабільність та відтворюваність результатів експериментів, що є ключовим для дослідницької та освітньої діяльності.

Отже, у результаті було створено програмне середовище, яке поєднує кілька підходів до побудови рекомендаційних систем і дозволяє проводити експерименти з їх параметрами.

1.2 Огляд та класифікація статистичних методів рекомендацій

Рекомендаційні системи відіграють ключову роль у сучасних інформаційних технологіях, надаючи можливість персоналізувати взаємодію користувачів із

цифровим контентом. Їх активно використовують у сферах електронної комерції, потокових сервісів, соціальних мереж і багатьох інших, де виникає потреба підлаштувати потік інформації відповідно до індивідуальних вподобань користувача.

Методи формування рекомендацій умовно поділяються на статистичні (традиційні) та методи глибокого навчання.

Методи глибокого навчання [6] базуються на застосуванні багатосарових нейронних мереж, які здатні автоматично виявляти складні латентні зв'язки між користувачами та об'єктами. Дані методи демонструють високу точність за умови наявності великих обсягів даних. Проте їхнє використання супроводжується певними недоліками: вони вимагають значних обчислювальних ресурсів, тривалого процесу навчання, великої кількості даних і складних налаштувань. Незважаючи на те, що такі підходи широко застосовують у промислових системах, для дослідницьких чи навчальних цілей вони часто є надлишковими через недостатню прозорість і складність інтерпретації результатів.

У свою чергу, статистичні методи рекомендацій базуються на аналізі існуючих зв'язків між користувачами, їх оцінками та характеристиками об'єктів. Вони застосовують математичні моделі, методи схожості векторів, факторизацію матриць або кластеризацію для виявлення закономірностей. Основна перевага таких методів полягає в їхній інтерпретованості, стабільності та більш низьких вимогах до обчислювальних ресурсів. Статистичні методи дозволяють моделювати поведінку користувачів більш гнучко, без необхідності залучення великих даних чи складних моделей. Вони часто є базовим шаром для сучасних систем персоналізації, створюючи основу для розробки більш складних гібридних або нейромережових методів. Окрім цього, статистичний підхід дає змогу проводити окремий аналіз ефективності кожного компонента системи, досліджуючи вплив гіперпараметрів, методів векторизації або способів нормалізації даних.

Серед статистичних підходів можна виділити три основні, які лежать в основі більшості класичних рекомендаційних систем: колаборативна, контентно-

орієнтована та гібридна фільтрація. Кожен із цих підходів має свою логіку формування рекомендацій, характерні переваги та недоліки.

1.3 Колаборативна фільтрація

Колаборативна фільтрація (Collaborative Filtering, CF) є одним із найбільш поширених статистичних методів, що застосовуються у побудові рекомендаційних систем. Її основна ідея полягає в тому, що користувачу пропонуються лише ті об'єкти, які раніше сподобались людям зі схожими вподобаннями [7]. Іншими словами, система визначає закономірності у поведінці користувачів, які взаємодіють між собою або оцінюють схожий контент (рис. 1.1).

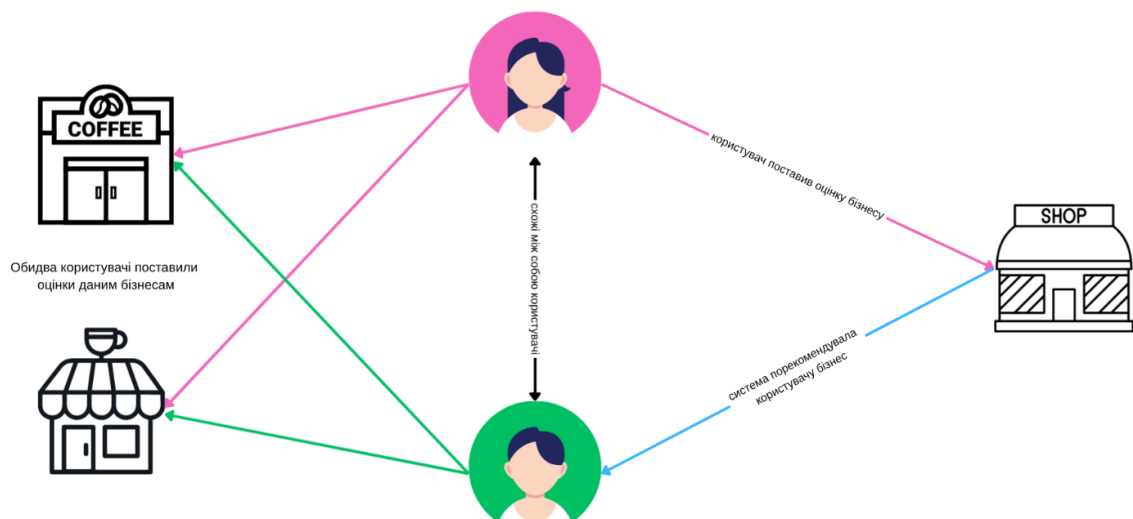


Рисунок 1.1 – Концепт роботи колаборативної фільтрації

Даний метод фільтрації реалізується на основі матриці взаємодій "користувач-об'єкт", де кожен рядок відповідає окремому користувачу, а стовбець – об'єкту. Значення у комірках цієї матриці це оцінки, які користувачі залишили відповідним об'єктами. У разі відсутності оцінки в матриці система пробує

передбачити її значення та формує перелік рекомендацій на основі цих передбачень. Колаборативна фільтрація поділяється на два основні типи в залежності від способу оцінки схожості.

1. User-based: фокусується на схожості між користувачами. Якщо двоє користувачів мають схожі уподобання і однаково оцінили певні об'єкти, то система рекомендує одному з них ті елементи, які він ще не бачив, але інший користувач дуже високо оцінив.

2. Item-based: аналізує схожість між об'єктами. Якщо користувачу сподобався певний об'єкт, модель знаходить інших користувачів зі схожими оцінками та рекомендує бізнеси, які вони також високо оцінили.

У колаборативної фільтрації для визначення схожості між об'єктами найчастіше використовують одну з таких метрик: косинусна схожість (Cosine Similarity), коефіцієнт кореляції Пірсона або евклідова відстань. У випадках коли матриця взаємодій має багато пропущених значень, точність даних методів суттєво зменшується. Для вирішення даної проблеми застосовуються моделі латентних факторів, що можна досягнути за допомогою алгоритму ALS (Alternating Least Squares). Його принцип полягає у розкладанні початкової матриці на дві компактні: матрицю користувачів U , де кожен рядок є набором прихованих характеристик певного користувача, та матрицю об'єктів V , що містить латентні властивості кожного об'єкта.

Під час навчання модель поступово уточнює значення векторів матриць U та V так, щоб різниця між реальними оцінками користувачів і прогнозами (що отримуються як добуток $U \times V^T$) була якомога меншою. У результаті отримуємо вектор користувача, який демонструє його основні вподобання у прихованому багатовимірному просторі, та вектор бізнесу, який показує, які характеристики об'єкту є важливими для моделі та яким чином він співвідноситься з іншими бізнесами за даними ознаками.

Головною особливістю даного підходу є можливість виявлення прихованих зв'язки між користувачами та об'єктами навіть у тих випадках коли вони не оцінювали їх. Наприклад, якщо користувач ніколи не оцінював конкретну

категорію бізнесів, система все рівно може рекомендувати йому відповідні варіанти, тому що його латентний профіль схожий до профілю іншого користувача, який високо оцінив об'єкти даної категорії.

1.4 Контентно-орієнтована фільтрація

Іншим важливим статистичним методом побудови рекомендацій є контентно-орієнтована фільтрація (Content-Based Filtering, CBF). На відміну від колаборативної, даний метод формує рекомендації не на основі поведінки інших користувачів, а на аналізі характеристик самих об'єктів, що було оцінено поточним споживачем.

Основна ідея підходу полягає у тому, що якщо користувачу сподобався певний об'єкт, то інші об'єкти зі схожими характеристиками також, імовірно, будуть для нього цікавими [8]. Наприклад, якщо користувач залишив високі оцінки декільком бізнесам з категорії “coffee” і позитивно відгукувався про атмосферу та ціни, то система може порекомендувати інші бізнеси зі схожими ознаками (рис. 1.2).

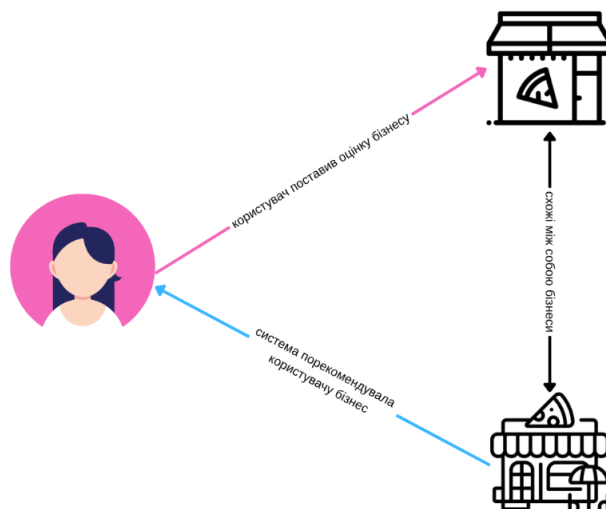


Рисунок 1.2 – Концепт роботи контентно-орієнтованої фільтрації

Саме тому для використання даного методу фільтрації кожен об'єкт описується набором своїх атрибутів. Це можуть бути як числові чи категоріальні так і текстові дані. У системі дані характеристики перетворюються у векторні представлення з використання різних методів обробки природної мови та статистичного аналізу.

У випадку текстових даних, а саме описів чи відгуків користувачів, застосовуються такі моделі:

- TF-IDF (Term Frequency – Inverse Document Frequency): метод, який визначає вагу слова, в залежності від того, як часто воно з'являється у всьому документі (в усіх текстах відгуків);
- LDA (Latent Dirichlet Allocation): це метод тематичного моделювання, який знаходить приховані теми у текстах всіх відгуків; після обробки кожен відгук представляється як набір ймовірностей того, наскільки він належить до кожної з тем.

Після того, як було побудовано векторні представлення усіх бізнесів, вони використовуються для побудови профіля користувача, що представлено як усереднений вектор характеристик тих об'єктів, які він оцінив. Таким чином, профіль користувача буде повністю описувати його індивідуальні вподобання на основі оцінених бізнесів.

Завдяки отриманим профілям користувачів з використанням косинусна схожість визначаються бізнеси, які найбільш схожі до вподобань користувача. Об'єкти з найвищою схожістю і сформують топ рекомендацій для системи.

Основною перевагою даного методу фільтрації є повна незалежність поточного користувача від інших. На відміну від колаборативної фільтрації, де інші користувачі відігравали головну роль у формуванні рекомендацій, в контентно-орієнтованій фільтрації система передбачає рекомендації просто на наборі бізнесів, що містяться в системі. Даний підхід надає можливість передбачити рекомендації навіть для ситуацій, де в системі є лише дані одержного користувача. Окрім цього, даний алгоритм високу інтерпретованості: легко пояснити, чому конкретний об'єкт

рекомендовано, адже все побудовано на схожості характеристик бізнесів, а не на прихованих зв'язках.

Проте метод також має свої недоліки, як наприклад схильність пропонувати лише дуже схожі між собою об'єкти без надавання диверсійних рекомендацій. Окрім цього, результати рекомендацій системи суттєво залежить від якості та повноти описових атрибутів. Якщо об'єкт не має достатньо інформації (опису, категорій або відгуків), алгоритм не зможе коректно оцінити його схожість.

Попри всі ці недоліки, контентно-орієнтовані методи залишаються надзвичайно важливими в системах, що працюють із великими текстовими або категоріальними даними, оскільки вони дозволяють враховувати змістову природу інформації, а не лише поведінкові взаємозв'язки.

1.5 Гібридна фільтрація

Останнім методом рекомендацій є гібридна фільтрація (Hybrid Filtering), що поєднує в собі елементи колаборативного та контентно-орієнтованого підходів. Це робиться для того, щоб компенсувати обмеження кожного з них і забезпечити більш збалансовану роботу системи рекомендацій [9].

Основна ідея алгоритму полягає в тому, що об'єднати інтеграції поведінкових і контентних характеристик користувача для підвищення точності та стійкості результатів (рис. 1.3).

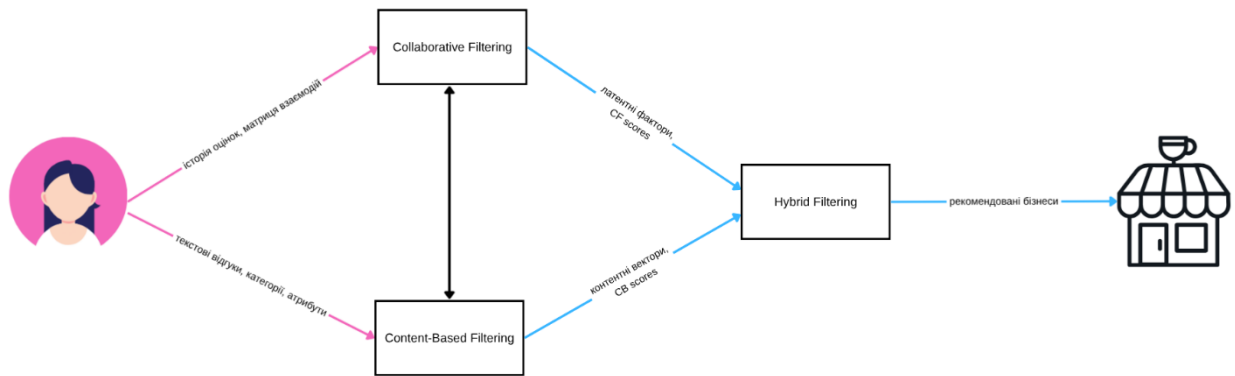


Рисунок 1.3 – Концепт роботи гібридної фільтрації

Колаборативна фільтрація добре працює у випадках, коли система має велику кількість активних користувачів, проте втрачає ефективність у випадку нових або малопопулярних (ефект “cold start” [10]). На відміну від цього, контентно-орієнтована фільтрація може надавати рекомендації навіть при мінімальній кількості користувачів та історій взаємодій, проте схильна часто пропонувати занадто схожі між собою об’єкти, що знижує різноманітність рекомендацій. Саме тому комбінація цих двох підходів дозволяє знайти баланс між двома типами аналізу інформації, що робить гібридну модель більш універсальною.

Існує декілька стратегій побудови гібридних систем:

- послідовне поєднання (Sequential Hybridization): результати одного методу використовуються як вхід для іншого;
- паралельне поєднання (Parallel Hybridization): обидва методи працюють незалежно, а результати комбінуються на фінальному етапі;
- зважене поєднання (Weighted Hybridization): кінцева оцінка формується як лінійна комбінація двох часткових результатів за допомогою вагового коефіцієнта α .

У розробленій системі використано саме зважене поєднання, рекомендаційна оцінка якого для кожного бізнесу формується за формулою:

$$R_{\{hybrid\}} = \alpha \cdot R_{\{content\}} + (1 - \alpha) \cdot R_{\{collab\}},$$

де $R_{\{content\}}$ – оцінка, отримана з контентно-орієнтованої моделі,

$R_{\{collab\}}$ – оцінка з моделі колаборативної фільтрації,

α – коефіцієнт, який визначає вагу впливу кожного з методів (у межах 0–1).

Завдяки даному параметру можна регулювати, який із двох підходів має більший вплив. Наприклад, при $\alpha = 0.9$ система більше покладається на контентні ознаки, а при $\alpha = 0.1$ – на поведінкові дані користувачів. Даний метод, адаптує систему модель під різні сценарії: роботи, то з новими користувачами, то з старими профілями.

Перевага гібридної фільтрації полягає у її високій гнучкості та стійкості до проблем, пов'язаних із браком даних. Вона ефективно функціонує навіть у випадку часткової відсутності оцінок чи описів та забезпечує при цьому належний рівень точності. Окрім цього, гібридний підхід сприяє збільшенню різноманітності рекомендацій, що робить їх менш передбачуваними та більш корисними для користувача.

Гібридна фільтрація поєднує ключові переваги двох підходів: високу точність колаборативних моделей і змістову чутливість контентних. Завдяки цьому вона є одним із найдієвіших статистичних методів рекомендацій та зберігає свою актуальність навіть у сучасних системах.

Висновки до розділу 1

У даному розділі було розглянуто статистичні методи рекомендацій та представлено узагальнену схему роботи кожного з них. Обрані методи фільтрації суттєво відрізняються принципами роботи та результатами, які вони надають. У таблиці 1.1 наведено їх ключові характеристики, що були враховані під час розробки рекомендаційної системи.

Таблиця 1.1 – Порівняльна характеристика основних статистичних методів рекомендацій

Метод	Основний принцип	Переваги	Недоліки
Колаборативна фільтрація	Рекомендації формуються на основі схожості користувачів між собою	- Не потребує інформації про характеристики об'єктів - Виявляє приховані зв'язки між користувачами - Легко масштабувати на великі набори даних	- Проблема «cold start» - Залежить від кількості активних користувачів системи
Контентно-орієнтована фільтрація	Рекомендації формуються на основі схожості об'єктів між собою	- Може працювати навіть у системах з одним користувачем - Не має проблеми з «cold start» - Легко пояснити причину вибору об'єктів	- Схильність до надавання дуже схожих рекомендацій - Вимагає детального опису об'єктів - Не виявляє поведінкові схожості між користувачами

Продовження таблиці 1.1

Гібридна фільтрація	Поєднує обидва попередні методи і формує рекомендації на основі схожих характеристик	- Компенсує недоліки обох методів - Підвищує точність та різноманітність рекомендацій	- Складність налаштування вагових коефіцієнтів - Підвищувала обчислювальна складність - Потреба в нормалізації обох моделей
------------------------	--	---	--

Ефективність даних методів варіюється залежно від доступності даних, їх структури та конкретних цілей системи. Отже, детальний аналіз сильних і слабких сторін кожного методу є важливим етапом, що допомагає зробити обґрунтований вибір оптимального підходу або комбінувати кілька методів в рамках гібридної фільтрації.

2 ПІДГОТОВКА ТА ОБРОБКА ДАНИХ

У даному розділі представлені дані, що були використанні для побудови системи та проведення експериментів. Описано вміст кожного з використаних файлів та його призначення в системі. Окрім цього наведений опис обробки даних та їх підготовки до використання.

2.1 Джерела даних для побудови рекомендаційної системи

Вхідними даними для розробленої системи було обрано відкритий набір Yelp Open Dataset, через його структурованість та велику кількість взаємопов'язаних даних. Цей набір містить кілька JSON-файлів [11], що охоплюють дані про користувачів, бізнеси, їх оцінки та відгуки.

Для реалізації рекомендаційної системи було використано такі основні файли:

- `yelp_academic_dataset_business.json` – містить дані про бізнеси назви, категорії, рейтинги, місцезнаходження, тощо;
- `yelp_academic_dataset_review.json` – містить інформацію про користувачів, їхні оцінки та текстові відгуки на різні бізнеси.

2.1.1 Дані про бізнес

Бізнес дані формують основу контентних характеристик об'єктів, що використовуються при побудові профілів у контентно-орієнтованій та гібридній фільтрації.

Першим джерелом даних для розробки став JSON-файл `yelp_academic_dataset_business.json` (рис. 2.1). У ньому зібрана інформація про бізнеси, представлені на платформі Yelp. Файл містить 150 тисяч унікальних

записів про бізнеси різних типів категорії, що забезпечує велике різноманіття контентних ознак. Кожен запис у файлі відповідає конкретному бізнесу і включає такі ключові параметри:

- **business_id**: унікальний ідентифікатор бізнесу;
- **name**: назва закладу чи компанії;
- **address**: адрес закладу;
- **categories**: список категорій, до яких належить бізнес;
- **stars**: середня оцінка від користувачів;
- **review_count**: кількість залишених користувачами відгуків;
- **attributes**: додаткові характеристики, як-от наявність доставки, тип кухні, ціновий діапазон тощо.

business_id	name	address	stars	review_count	categories	attributes
0 Pns2L4eNsF08kk83dixA6A	Abby Rappoport, LAC, C...	1616 Chapala St, Ste 2	5.0	7	Doctors, Traditional Chin...	{\ByAppointmentOnly\': \True\}
1 mpf3x-BjTdTEA3yCZrAYPw	The UPS Store	87 Grasso Plaza Shopping...	3.0	15	Shipping Centers, Local S...	{\BusinessAcceptsCreditCards\': ...
2 tUFRWirKiKi_TAnsVWINQQ	Target	5255 E Broadway Blvd	3.5	22	Department Stores, Shoppi...	{\BikeParking\': \True\', \Bu...
3 MTSW4McQd7CbVtyjqoe9mw	St Honore Pastries	935 Race St	4.0	80	Restaurants, Food, Bubble...	{\RestaurantsDelivery\': \Fals...
4 mWMc6_wTdE0EUBKIGXDVfA	Perkiomen Valley Brew...	101 Walnut St	4.5	13	Brewpubs, Breweries, Food	{\BusinessAcceptsCreditCards\': ...
5 CF33F8-E6oudUQ46HnavjQ	Sonic Drive-In	615 S Main St	2.0	6	Burgers, Fast Food, Sandw...	{\BusinessParking\': \None\', ...
6 n_0UpQx1hsNbnPUSlodU8w	Famous Footwear	8522 Eager Road, Dierber...	2.5	13	Sporting Goods, Fashion, ...	{\BusinessAcceptsCreditCards\': ...
7 qkRM_2X51Yqk3btLwAQIg	Temple Beth-El	400 Pasadena Ave S	3.5	5	Synagogues, Religious Org...	None
8 k0h1BqXX-Bt0vf1op7Jr1w	Tsevi\'s Pub And Grill	8025 Mackenzie Rd	3.0	19	Pubs, Restaurants, Italia...	{\Caters\': \True\', \Alcohol...
9 bBDEgkFA10tx9Lfe7BZUQ	Sonic Drive-In	2312 Dickerson Pike	1.5	10	Ice Cream & Frozen Yogurt...	{\RestaurantsAttire\': \casu...
10 UJsufbvfyf0NHwEdvAHKJA	Marshalls	21705 Village Lakes Sc Dr	3.5	6	Department Stores, Shoppi...	{\RestaurantsPriceRange2\': \2...
11 eEOYSgkmpB90uNA7L0DMRA	Vietnamese Food Truck		4.0	10	Vietnamese, Food, Restaur...	{\Alcohol\': \none\', \Ou...
12 il_Ro8jwPLHresjw9EGmBg	Denny\'s	8901 US 31 S	2.5	28	American (Traditional), R...	{\RestaurantsReservations\': \...
13 jaxMSoInw8Poo3XeMt8LQ	Adams Dental	15 N Missouri Ave	5.0	10	General Dentistry, Dentis...	{\ByAppointmentOnly\': \True\}
14 0bPLkL0qhP05kt1_EXmNQ	Zio\'s Italian Market	2575 E Bay Dr	4.5	100	Food, Delis, Italian, Bak...	{\OutdoorSeating\': \False\', ...
15 MUTTqe8uqyMdB1B6RmNeA	Tuna Bar	205 Race St	4.0	245	Sushi Bars, Restaurants, ...	{\RestaurantsReservations\': \...
16 rBmPy_Y1UbBx8ggHlyb7hA	Arizona Truck Outfitte...	625 N Stone Ave	4.5	10	Automotive, Auto Parts & ...	{\DriveThru\': \False\', \Bus...
17 M0XSSHqrAS0nhgbWDJpQA	Herb Import Co	712 Adams St	4.0	5	Vape Shops, Tobacco Shops...	{\BusinessParking\': \garag...
18 8w6ISVjYkE2tSqn3CDMu8A	Nifty Car Rental	1241 Airline Dr	3.5	14	Automotive, Car Rental, H...	None
19 R0eacJQwBeh05Rqg7F6TCg	BAP	1224 South St	4.5	205	Korean, Restaurants	{\NoiseLevel\': \u\'quiet\', ...
20 WKMJwqnfZKsAae7SRMP6jA	Roast Coffeehouse and ...	10359 104 Street NW	4.0	40	Coffee & Tea, Food, Cafes...	{\OutdoorSeating\': \False\', ...
21 qh0dDeI3K4jy2KyzwFN53w	Barnes & Noble Bookse...	8675 River Crossing Blvd	4.0	38	Shopping, Books, Mags, Mu...	{\BusinessAcceptsCreditCards\': ...
22 kFmV-JZpuN6TVNS06hHdkw	Hibachi Express	6625 E 82nd St	4.0	20	Steakhouses, Asian Fusio...	{\RestaurantsDelivery\': \True...

Рисунок 2.1 – Вміст JSON-файлу `yelp_academic_dataset_business.json`

Під час попереднього аналізу вхідних даних було виявлено, що набір містить ознаки різних типів даних: числові, текстові та словникові (dictionary). Числові поля описували параметри, як-от рейтинги та кількість відгуків. Текстові включали найменування назви, адреси й категорій, а словникові представляли вкладені характеристики закладів у форматі ключ-значення, таких як `attributes`.

Через неоднорідність структури даних у деяких записах були зафіксовані помилки у форматуванні: некоректно закриті дужки, подвійні лапки або текстові значення, які не розпізнавалися як JSON-об'єкти. Для усунення цих недоліків записи пройшли парсинг [12] та були приведені до правильного словникового формату [13].

У процесі аналізу також було виявлено пропущені значення (NaN) та неповні записи. Щоб уникнути помилок у подальшій обробці, такі значення замінювалися на нулі або порожні списки відповідно до типу даних. Логічні змінні у форматі True / False було конвертовано в числовий вигляд (1 та 0), що зробило їх придатними для математичних операцій та використання в алгоритмах машинного навчання.

Для уніфікації шкал усі числові дані були піддані нормалізації. Поле з великим розкидом значень, як-от кількість відгуків, пройшло логарифмічну нормалізацію, тоді як показник рейтингу (stars), був лінійно нормалізований до діапазону [0, 1].

2.1.2 Дані про відгуки користувачів

Другим, не менш важливим джерелом даних став JSON-файл `yelp_academic_dataset_review.json`, який містить повну інформацію про відгуки на бізнеси. Набір включає більше 6 мільйонів текстових відгуків, що дає змогу проводити широкий аналіз та використовувати алгоритми тематичного моделювання для виявлення тенденцій у вподобаннях користувачів.

У даному наборі зберігається дані про те, хто саме, коли і якому бізнесу залишив відгук, а також текстові оцінки їх взаємодії (рис. 2.2). Ключові атрибути набору даних:

- `review_id`: унікальний ідентифікатор відгуку;
- `user_id`: унікальний ідентифікатор користувача;
- `business_id`: унікальний ідентифікатор бізнесу, до якого належить відгук;
- `stars`: оцінка бізнесу;

- text: текстовий вміст відгуку.

review_id	user_id	business_id	stars	text
1yzkzMBuDc0oIATmm-YT6A	batsML880B8-BMV1YBh0VQ	NLV0ppshTiJk6JVdFSxVAQ	5	Pleasantly intrigued by th...
p1He-R5tvsxamxNF0Bf-cg	T9n6TKhkhqwh7u3a700-pQ	c_xAbBTiIHqPm1zM0vQcg	5	I am not sure what I was e...
f00ZEC8D2f1hI6bs08X3gQ	4zUi7exGanxHz_zXJuHjMw	Sv1MEZP-mMfp8SmE0hwYEA	5	Mmm...still thinking abou...
ZoeZ_aC_CiWSuqvfe38nAQ	bHNxtcIKeU5Ifnd4BwBoWw	KB63GV4WHU-iLLSWiapbJw	4	Very good even by Nashvill...
ygAtxZ16aimqj8IZGpobRA	W9zN7u6b-e3bkZ69W92fig	BX3LLonrLLLfVX-tNrE7yA	5	New location is beautiful!...
BpsN31jf_e0B29_SBfQjbg	Fu8D01UpP5DNnaI7dP9xEw	S8ZFYEgMejpChID8tzKo9A	5	Oh look, another person ra...
wdVvwaPmYgA17bIsILv1LQ	zwNDsl7oCCNv3iNI-53NoQ	j-qtdD550LfSqfsWuQTDJg	1	Went on a Friday night. Go...
PLrmW2F7dksNL9JH2t03-Q	i5xSc8B5LrOW7MFNGSFVuw	J1bG9ezYHMkzqV1yRsVtQQ	4	Came here at the recommend...
J29BCSFmUs7NPzVRyRstIg	VBjAZ4JxPmaqRqBVbhjRGQ	aSkAe5w4CEAXdUrLVEygA	1	The worst Domino\'s I\'ve ...
ThiT1UQNvdZ26lo6VmM8Kg	prnfXddY0ffxyyCKfWf72w	zupI44qitH6T8Bo90jA6Yw	3	There steak bagel is delis...
lyD5iMWEuAM3sWQCHRL0ig	KwI-RhKF_5EIK2-jRdWNfg	N4TImiSncqw-FTXT2a-sVw	5	Jared was able to fit me i...
Q1s8Qer9A58DPxRpBmDgDg	CMRTvF6kjcUrcZssmnWABw	-U1xpWbKkMTQ3YCELPsw6A	5	My husband and I moved to ...
gIF9khk9J0nnR8Li6LNgQ	B3ASjMLRgsQCCAvL9PzPKw	mCEj1o5tHc-IMXtvGXd0rg	5	Best tanning salon in the ...
rwUvvaUmQzYU4Ca7i6z5nA	ObZnTiWMA-4gaTqa1_mB6A	akPsFyMcb2D90nCsSXBZfQ	3	We drove up to the Hilltop...
DE9TfCE0fNa7SjijAJkbZQ	Tds6Tu-6fp6-0W2ALZD5Yw	o5UrU0Kcmj5GFXmB60sb6w	5	Wow!! Best food of our ent...
GArBRE5xvx1j1gNmFu2RmA	UJ-peKY9Q8i9Q-8IF65Qag	LgWLFdHfW2ZD8M-2vaMULg	3	Kababeque was the first pl...
6_IazxcSon09_pxedfIaCA	4nHcdEhLjf2428tsiJUyYQ	mdSjI3ig48WS_jLJ6Vxq0A	5	Great Place! Wonderful! If...
9m4xLKL9AQ6Dw2gmV6xbYw	CcqpZKc5ngs88akzQ0rFHg	s_RStWu76oxfAPHmqmUMMW	5	I love the yachtsman, it\'...
-zYEqIYpWNB08-uIRR0LQ	Am_G2m0pwVrEg9AoUp5tow	wLYV0aohTpwfKyIM5uXPeA	3	This location of Five Guys...
_H4ijuduLxE9tUTMmhyKvA	fWBXv9HPxo0SMr4d1Zn2yA	a32M9H9nkkPGTKB5W27NA	4	If you go to this place, y...
jmhUJT3vMsztfqG_SJEZ_w	lm56NZaqG90rCIq7mdZYgA	N-EQ9UDuFHT9RyC1JFX5pQ	2	Not much to be said that h...

Рисунок 2.2 – Вміст JSON-файлу yelp_academic_dataset_review.json

Даний набір даних відіграв важливу роль у створенні системи рекомендаційної системи. Оцінки користувачів (stars) слугували основою для реалізації колаборативної фільтрації, яка базується на аналізі подібності вподобань між користувачами. Текстові частини відгуків (text) використовувалися для контентно-орієнтованої фільтрації, де проводився аналіз змісту відгуків з метою визначення тематичної схожості бізнесів.

Під час попередньої обробки було виконано токенизацію, очищення текстів від пунктуації, чисел і службових слів, а також приведення їх до нижнього регістру [14]. Після цього оброблений текст був використаний в алгоритмах обробки тексту TF-IDF та LDA.

2.2 Підготовка текстових даних для аналізу

У системах контентно-орієнтованої фільтрації текстові дані служать основним джерелом інформації про об'єкти, які неможливо описати лише за допомогою структурованих характеристик. Опрацювання текстів відгуків є важливою частиною розробленої моделі, оскільки це основа для створення профілів користувачів і ознак об'єктів. Для реалізації було використано кілька методів, включаючи векторизацію TF-IDF [15] та тематичне моделювання LDA [16]. Обидва методи обробки тексту мають різний підхід до змісту відгуків, тому рекомендації можна розглядати з більшою кількістю семантичних характеристик, коли вони використовуються разом.

Обробка тексту починається з налаштування вхідних даних. Кожен відгук очищається від непотрібних елементів, які не мають важливого смислового навантаження. Це виконується за допомогою спеціально розробленої функції попередньої обробки, яка переводить текст у нижній регістр, видаляє будь-які числа та конструкції у форматі «10pm», «3rd» або подібних, видаляє знаки пунктуації та нормалізує загальний текст. Після цього слова з менше трьох символів відкидаються, оскільки вони не містять багато корисної інформації. Завершальним етапом є стемінг [17], тобто зведення слів до їхньої основи (наприклад, «gunning» до «gun»), що дозволяє скоротити різноманітність слів і зменшити обсяг словника. Таким чином створюється однорідний, очищений текстовий корпус, який готовий до векторизації.

Після очищення всі відгуки, що належать одному бізнесу, об'єднуються в один документ. Після цього один бізнес був представлений як сукупність усіх доступних відгуків. Отриманий набір документів насливався на вхід моделі TF-IDF. У даній роботі використовується масштабний TF-IDF-векторизатор із налаштуваннями, спрямованими на максимальне зменшення шуму, водночас зберігаючи інформативні фрази. Зокрема, максимальна кількість ознак обмежена 5000, як уніграми, так і біграми, а слова, які зустрічаються надто часто або надто рідко, фільтруються. У текстах відгуків TF-IDF створює велику розріджену

матрицю, у якій кожен рядок представляє бізнес, а кожен стовпець представляє важливе слово або фразу. Базові «контентні» характеристики компанії створюються за допомогою цієї матриці, яка потім поєднується з іншими наборами даних.

Тим не менш, TF-IDF не здатний виявити структурні елементи тексту, особливо тематичні закономірності. Через це було вирішено використати тематичне моделювання LDA як додатковий метод. З допомогою даного методу кожен документ було перетворено у розподіл тем. Таким чином було визначено, наскільки текст відгуків відноситься до певних тематичних напрямків (наприклад, «мексиканська кухня», «кава та десерти», «нічне життя» або «скарги на сервіс»).

У системі було натреновано три окремі LDA-моделі з різною кількістю тем (20, 40 та 60), що дозволяє оцінити, як зміна розмірності тематичного простору впливає на точність рекомендацій. Для кожної моделі створена матриця тем, де кожен бізнес представлений як набір ваг, які показують, наскільки сильно він належить до кожної з тем.

В експериментальній частині системи зберігаються TF-IDF-матриці, LDA-моделі та матриці тем. У цьому процесі вони використовуються разом з іншими бізнес-ознаками, такими як категорії та атрибути об'єктів. Наприклад, розглядаються варіанти, у яких використовуються лише тематичні ознаки (LDA), лише категорії або їх поєднання з атрибутами бізнесу.

Таким чином, для створення якісних контентних ознак об'єктів аналіз текстових даних є життєво важливим. Без нього контентно-орієнтована фільтрація була б значно менш інформативною. На відміну від LDA, TF-IDF дозволяє оцінювати важливість ключових фраз, тоді як LDA відкриває тематичну структуру текстів і дозволяє моделям зрозуміти глибші семантичні зв'язки між компаніями. У підсумку, ці методи складають основу контентного блоку рекомендаційної системи та безпосередньо впливають на точність отриманих рекомендацій.

2.3 Побудова профілів користувачів і бізнесів

Формування профілів користувачів та бізнесів є одним з ключових етапів в побудові рекомендаційної системи, адже саме на основі цих даних і формуються самі рекомендації [18]. Їх побудова відрізняється в залежності від методу, адже кожен з них має свій особливий підхід, який потребує правильного представлення, підготовки та оцінки даних.

2.3.1 Профілі в колаборативній фільтрації

Формування профілів користувачів і бізнесів у даному методі здійснюється на основі історії їхніх взаємодій, збережених у відгуках. Кожен користувач має власний набір оцінок для різних бізнесів, що утворює матрицю взаємозв'язків “користувач – об’єкт”. У даній матриці рядки відповідають користувачам, стовпці – бізнесам, а значення в комірках відображають оцінки, які користувач надав бізнесу. Якщо користувач не взаємодіяв із певним бізнесом, відповідна комірка залишається порожньою (табл. 2.1).

Таблиця 2.1 – Приклад матриці взаємозв'язків “користувач-об’єкт”

	Business1	Business2	Business3	Business4
User1	1	3	5	1
User2	Nan	4	5	1
User3	2	4	Nan	2
User4	3	5	3	2
User5	4	2	2	3

На рисунку 2.1 представлено загальну структуру процесу колаборативної фільтрації, яка демонструє основні етапи формування рекомендацій – від побудови матриці взаємодій до отримання кінцевого списку рекомендованих бізнесів [19].

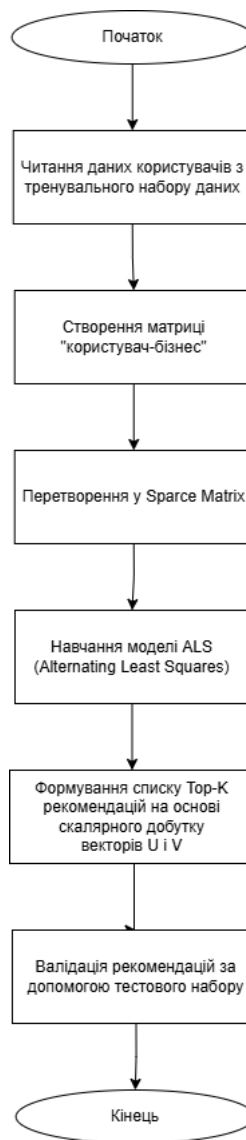


Рисунок 2.1 – Принцип роботи алгоритму колаборативної фільтрації

Для подальшого навчання матриця перетворюється у розріджений формат (Sparse Matrix), що дозволяє ефективно зберігати дані з великою кількістю пропусків.

Перед побудовою профілів було проведено нормалізацію оцінок, щоб зменшити індивідуальні відмінності у способах оцінювання користувачів. Деякі споживачі схильні завжди ставити високі бали, а інші, навпаки, занижені, що може викривляти результати. Для вирівнювання цього ефекту з кожної оцінки було спершу віднято середній бал конкретного користувача (щоб врахувати його стиль оцінювання) та середній бал бізнесу (щоб компенсувати вплив популярності чи

непопулярності самого об'єкта). Після цього до них було додано середнє значення всіх оцінок у вибірці. Такий чином оцінки перестали відображати абсолютне значення "добре/погано" і почали показувати, наскільки конкретна оцінка відхиляється від типового рівня цього користувача і бізнесу.

Процес створення профілів здійснюється за допомогою методу факторизації матриць ALS (Alternating Least Squares) [20]. Алгоритм розділяє початкову матрицю взаємодій на дві менші:

1. матрицю U (фактори користувачів), де кожен рядок представляє векторну характеристику користувача;
2. матрицю V (фактори об'єктів), де кожен рядок відповідає профілю бізнесу.

У результаті кожен користувач і бізнес описується латентним вектором заданої довжини, що визначається параметром *factors* і відображає приховані вподобання. ALS навчається так, щоб передбачені оцінки, що обчислюються добуток векторів, якомога менше відрізнялися від реальних оцінок у таблиці користувачів і бізнесів.

Після завершення навчання отримані профілі використовуються для прогнозування невідомих оцінок. Це досягається за допомогою обчислення скалярного добутку між вектором користувача та вектором бізнесу. Бізнеси з найвищими передбаченими оцінками формують індивідуальний список рекомендацій.

Окрім цього, щоб перевірити здатність моделі працювати на нових даних, початковий набір було розділено на навчальну (80 %) і тестову (20 %) вибірки. Навчальна частина використовується для побудови профілів, а тестова служить оцінкою точності рекомендацій за показниками $Precision@30$, $Recall@30$ та $F1@30$.

2.3.2 Профілі в контентно-орієнтованій фільтрації

У даному методі для кожного користувача формується його індивідуальний профіль, який являє собою середній вектор усіх бізнесів, що були ним оцінені.

Таким чином, цей вектор відображає загальну картину уподобань користувача та узагальнює його історію взаємодій [21].

На рисунку 2.2 представлено загальну структуру процесу контентно-орієнтованого методу фільтрації.

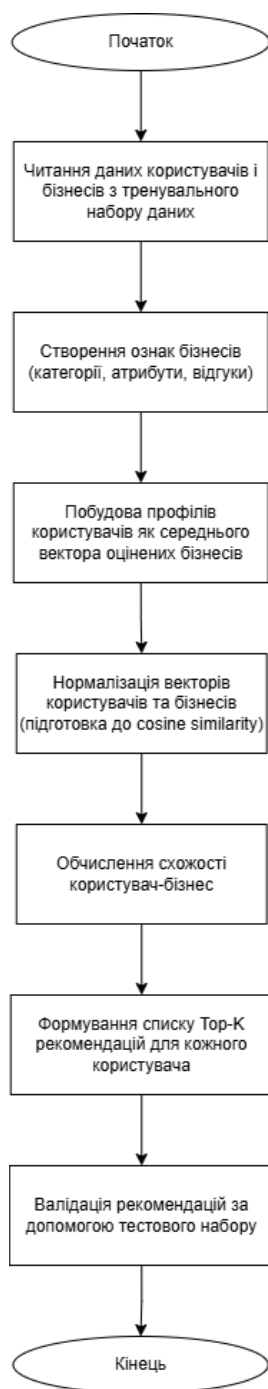


Рисунок 2.2 – Принцип роботи алгоритму контентно-орієнтованої фільтрації

Ключовим етапом контентно-орієнтованої фільтрації є побудова векторів бізнесів, які формуються на основі їхніх характеристик. Оскільки дані про бізнеси містять категорії, атрибути та текстові відгуки користувачів, для їхнього представлення було застосовано дві моделі аналізу текстів – TF-IDF та тематичну модель LDA. Таким чином було отримано числові векторні представлення текстів, що відображають частотність і семантичну структуру слів.

Для розширення експериментального аналізу було реалізовано кілька конфігурацій бізнес-векторів, у яких змінювався набір характеристик – наприклад, окреме використання лише тематичних ознак, лише категорій або їх комбінацій. Це було зроблено для оцінки впливу різних типів контентних ознак на точність рекомендацій.

Після побудови векторів бізнесів створювалися вектори користувачів – як середні значення векторів усіх бізнесів, яким користувач залишав оцінки. Для забезпечення коректності обчислень усі вектори було нормалізовано та перетворено у розріджені матриці, аналогічно до колаборативного методу.

Пошук найбільш релевантних бізнесів здійснювався за допомогою методу Nearest Neighbors [22] із використанням метрики косинусної схожості [23]. Для кожного користувача визначалися відстані до всіх бізнесів, після чого відбиралися топ-30 найближчих об'єктів.

Як і в колаборативному підході, навчання відбувалося на тренувальній вибірці, а оцінювання якості рекомендацій – на тестовій, що дозволяло об'єктивно перевірити здатність системи передбачати нові бізнеси, які користувач раніше не бачив.

У результаті система формує впорядкований список рекомендованих бізнесів для кожного користувача – тобто ті об'єкти, чия схожість із його профілем є найвищою.

Висновки до розділу 2

У даному розділі було представлено вхідні дані, які стали базою для створення рекомендаційної системи. Детально описано процес аналізу та підготовки текстових даних, а також підходи до формування профілів користувачів і бізнесів. Обробка даних дозволила досягти узгодженості, нормалізації та забезпечити готовність інформації до наступного етапу – налаштування моделей і тестування різних методів фільтрації.

3 АНАЛІЗ ТА НАЛАШТУВАННЯ МОДЕЛЕЙ

У даному розділі представлено аналіз та налаштування статистичних методів, застосованих у створеній рекомендаційній системі. Розглянуто детальний опис побудови та принципів дії трьох основних підходів до фільтрації: колаборативного, контентно-орієнтованого та гібридного. Для кожного методу представлено налаштування його моделей та ключові результати експериментів, які ілюструють їхню ефективність у представленні персоналізованих рекомендацій.

3.1 Вибір та характеристика тестових груп користувачів

Для проведення експериментів та перевірки ефективності розроблених рекомендаційних алгоритмів було створено три різні типи груп користувачів, що відрізняються між собою рівнем активності.

1. Група Top-30 (найактивніші користувачі). До даної категорії входять користувачі, що залишили найбільшу оцінок бізнесам. Вони мають найбільш інформативні та детальні профілі, що надає змогу алгоритмам формувати рекомендації з найбільшою точністю. Тестування на даній групі демонструє найбільшу продуктивність алгоритмів, адже вони отримують достатню кількість даних про вподобання споживача.

2. Група Least-30 (пасивні користувачі). У початкових даних було виявлено велику кількість споживачів з мінімальною активністю, з лише одним відгуком чи оцінкою бізнесу. Такі записи є малокорисними для навчання моделей, адже за відсутності відгуків чи їх малої кількості неможливо адекватно оцінити ефективність тої чи іншої рекомендаційної моделі. Саме тому для усунення нерепрезентативних даних було застосовано метод “elbow” [24], який допоміг визначити оптимальну межу між неактивними та помірно активними користувачами. У результаті даного аналізу з набору даних було вилучено

користувачів з дуже низькою активністю. Таким чином до групи Least-30 увійшли користувачі, які мають мінімально допустимий рівень активності та з допомогою яких можна буде оцінити продуктивність моделей за умов часткового браку інформації.

3. Група Random-30 (випадкові користувачі). Ця група містить користувачів з різним рівнем активності та виступає як контрольний набір користувачів. Вона надає можливість продемонструвати загальний баланс та універсальність методів фільтрації.

Для усіх груп користувачів було застосовано однакові налаштування моделей, що гарантує коректне порівняно отриманих результатів ефективності.

3.2 Налаштування моделі колаборативної фільтрації

У розробленій системі метод колаборативної фільтрації реалізований з використанням методу ALS, що входить до складу бібліотеки `implicit`.

Основними параметрами моделі є кількість латентних факторів (`factors`), коефіцієнт регуляризації (`regularization`), кількість ітерацій навчання (`iterations`) та початкове випадкове значення (`random_state`). Дані параметри впливають на швидкість навчання моделі та її точність під час роботи з новими користувачами.

Саме тому, з метою пошуку найбільш оптимальних налаштувань було проведено серію експериментів, в межах яких модель тестувалась при трьох різних комбінаціях параметрів. Було перевірено такі конфігурації:

- `factors`: [50, 100, 200] – визначає розмірність латентного простору, який використовується для опису прихованих вподобань користувачів;
- `regularization`: [0.001, 0.01, 0.1] – регулює рівень згладжуваності моделі, що допомагає запобігти її перенавчанню;
- `iterations`: [10, 50, 100] – встановлює кількість циклів оновлення латентний факторів у процесі навчання.

Кожна з утворених комбінацій налаштувань дозволяла оцінити баланс між точністю прогнозів та стабільністю результатів.

Збільшення кількості факторів підвищує здатність моделі враховувати складні взаємозв'язки між користувачами та бізнесами, проте це вимагає багато пам'яті та часу на обчислення. Підвищення коефіцієнта регуляризації знижує ризик перенавчання, але може суттєво негативно вплинути на точність рекомендацій. Збільшення кількості ітерацій покращує збіжність моделі, проте це може призвести до марного використання ресурсів без покращення якості результатів.

3.3 Налаштування контентно-орієнтована фільтрація

Для проведення експериментів з контентно-орієнтованою фільтрацією було створено декілька конфігурацій моделей, які відрізняються між собою набором ознак, що використовуються для опису бізнесу. Основною метою даного підходу було визначити, яким чином різні поєднання текстових та категоріальних ознак впливають на ефективність рекомендацій.

Основними джерелами ознак стали:

- тексти відгуків користувачів, оброблені за допомогою TF-IDF та тематичної моделі LDA;
- категорії бізнесів також оброблені за допомогою LDA;
- атрибути бізнесів, наприклад наявність доставки, тип атмосфери тощо.

Для тестування було створено такі типи конфігурацій:

- `lda_only`: використання лише тематичних ознак із моделі LDA;
- `cat_only`: застосування лише категоріальних ознак бізнесів;
- `lda+cat`: поєднання тематичних ознак бізнесів.

З допомогою даного підходу було оцінено вплив окремих груп ознак на якість рекомендацій та визначено, які параметри найбільше сприяли їх точності.

Налаштування параметрів тематичної моделі LDA включало варіювання кількості тем ($n_topics = [20, 40, 60]$), для дослідження того як деталізація тематичного простору впливає на точність рекомендацій.

3.4 Налаштування гібридної фільтрації

У впровадженій рекомендаційній системі гібридний метод фільтрації комбінує переваги двох статистичних підходів – колаборативної та контентно-орієнтованої фільтрації. Основна мета цього підходу полягає в розробці більш збалансованих рекомендацій, що враховують як поведінкові вподобання користувачів, так і контентну схожість бізнесів.

Для створення моделі було попередньо сформовано дві матриці оцінок:

1. матриця передбачуваних оцінок, побудована на основі ALS-моделі (CF);
2. матриця тематичної схожості бізнесів, отримана за допомогою контентно-орієнтованої фільтрації (CB).

Під час розробки гібридної моделі було виявлено, що набір бізнесів, з якими працює гібридна система, відрізняється від тих, що використовуються в окремих підходах: колаборативній і контентно-орієнтованій фільтрації. У колаборативній частині кожна модель ALS оперує власним фіксованим списком бізнесів, а в контентній частині бізнеси визначаються за структурою ознак (LDA та категорії). Натомість у гібридному підході об'єднується обидва списки, тому гібридний метод функціонує на об'єднанні двох множин бізнесів, а не на їхньому перетині. Через це результати гібридної моделі не можуть бути напряду порівняні з початковими метриками CF і CB, оскільки вони базувалися на іншому наборі бізнесів.

Для забезпечення коректності порівняння метрики гібридного підходу були перераховані з використанням тих самих користувачів і тестового набору, але з врахуванням розширеного списку бізнесів, доступного у гібридній моделі.

У результаті всі підходи (CF, CB і Hybrid) мають узгоджені такі аспекти:

- набір користувачів;

- тестовий набір реальних бізнесів для цих користувачів;
- методи розрахунку метрик.

Відмінність полягає лише у просторі бізнесів, з якого кожна модель генерує рекомендації.

У гібридному методі було використано два механізми комбінування результатів.

1. Лінійне зважування: у даному випадку підсумкова оцінка бізнесу для конкретного користувача обчислюється як лінійна комбінація результатів двох моделей.

2. Метод вагового ранжування Борда: бізнеси отримують бали залежно від їх позиції в кожному із двох рекомендованих списків, після чого вони комбінуються з урахуванням вагового коефіцієнта α .

Для дослідження впливу різних балансів між двома джерелами даних було проведено експерименти зі значеннями $\alpha = [0.5, 0.7, 0.8, 0.9]$. Такі значення були обрані для поступового посилювання впливу колаборативної моделі, що зазвичай краще працює для активних користувачів, водночас не відкидаючи контентну частину, яка особливо важлива для тих користувачів, чия історія взаємодій є обмеженою.

На основі об'єднаних оцінок для кожного користувача формувався список з топ-30 найближчих бізнесів. Після цього всі рекомендації проходили фільтрацію для виключення об'єктів, які користувач уже оцінював раніше, що дозволяє уникнути повторних рекомендацій.

3.5 Аналіз експериментальних результатів

Усі методи рекомендацій, що були реалізовані в даній магістерській дисертації, надають кінцевий список рекомендацій, які можуть бути найбільш підходящими для користувача. Незважаючи на те, що контентно-орієнтовані,

колаборативні та гібридні підходи використовують різні механізми, кінцева мета залишається одною -: якісно визначити Top-30 рекомендації.

Основною метою експериментальної частини є визначення методу, який надає найвищу релевантність, тобто найкраще поєднання покриття та точності. У роботі використовується формат Top-N рекомендацій, тому частина класичних метрик є непридатною.

Для такого підходу найбільш точними є метрики Precision@30, Recall@30 і F1@30, які безпосередньо оцінюють поверненого списку рекомендацій. Precision показує кількість правильних рекомендацій, а Recall – охоплення релевантних об'єктів. Тим не менш, окремі характеристики цих показників можуть бути оманливими: висока точність може супроводжуватися низьким покриттям або навпаки.

Саме тому, основною метрикою аналізу в даній роботі є F1-оцінка, оскільки вона збалансовує Precision і Recall та дозволяє об'єктивно порівнювати різні моделі в однакових умовах.

3.5.1 Результати колаборативної фільтрації

У моделі ALS було протестовано різні значення параметрів:

- factors: [50, 100, 200];
- regularization: [0.001, 0.01, 0.1];
- iterations: [10, 50, 100];

У процесі експериментів було протестовано усі можливі унікальних конфігурації моделі ALS (3 значення factors $\times$ 3 значення regularization $\times$ 3 значення iterations $\times$ 3 групи користувачів). У даному підрозділі наведено лише узагальнені підсумкові результати: конфігурації, що показали найкращі значення метрик та середні показники для різних груп користувачів. Підсумки роботи методу фільтрації представлено у таблиці 3.1.

Таблиця 3.1 – Найкращі результати F1@30 для колаборативної фільтрації

User group	factors	regularization	iterations	Precision@30	Recall@30	F1@30
Top users	200	0.1	100	0.145	0.043	0.066
Random users	100	0.1	50	0.012	0.121	0.022
Least users	50	0.01	10	0.001	0.033	0.002

На рисунку 3.1 наведено порівняння середніх значень Precision@30, Recall@30 та F1-score@30 для трьох груп користувачів. Графік надає інформацію про поведінку ALS-моделі у різних інформаційних умовах.

У групи найактивніших користувачів спостерігається найвищий Precision@30 (≈ 0.145), що є очікуваним результатом, оскільки модель отримує достатній обсяг історичних оцінок для побудови точних рекомендацій. Однак Recall@30 залишається відносно низьким (≈ 0.043), що означає, що модель повертає лише частину всіх потенційно релевантних об'єктів. В результаті F1-score становить 0.066, що є найвищим серед усіх груп.

Група випадкових користувачів демонструє найвищий Recall@30 (≈ 0.121), що пояснюється різноманітністю їх профілів: модель отримує помірний обсяг інформації без надлишку шуму. Проте Precision@30 є значно нижчим (≈ 0.012), що вказує на те, що розсіяність історій користувачів знижує точність передбачень. Підсумковий F1-score дорівнює 0.022, що хоч і нижче, ніж у top-users, але все одно є значно вищим, ніж у least-users.

У найменш активних користувачів модель має мінімальні значення Precision@30 (≈ 0.001) та F1-score (≈ 0.002). Відносно вищий Recall@30 (≈ 0.033) зумовлений тим, що для майже порожніх профілів модель рекомендує універсальні бізнеси, які співпадають з невеликим числом релевантних об'єктів. В цілому ця

група підтверджує фундаментальну проблему колаборативної фільтрації – ALS практично не працює за умов браку даних.

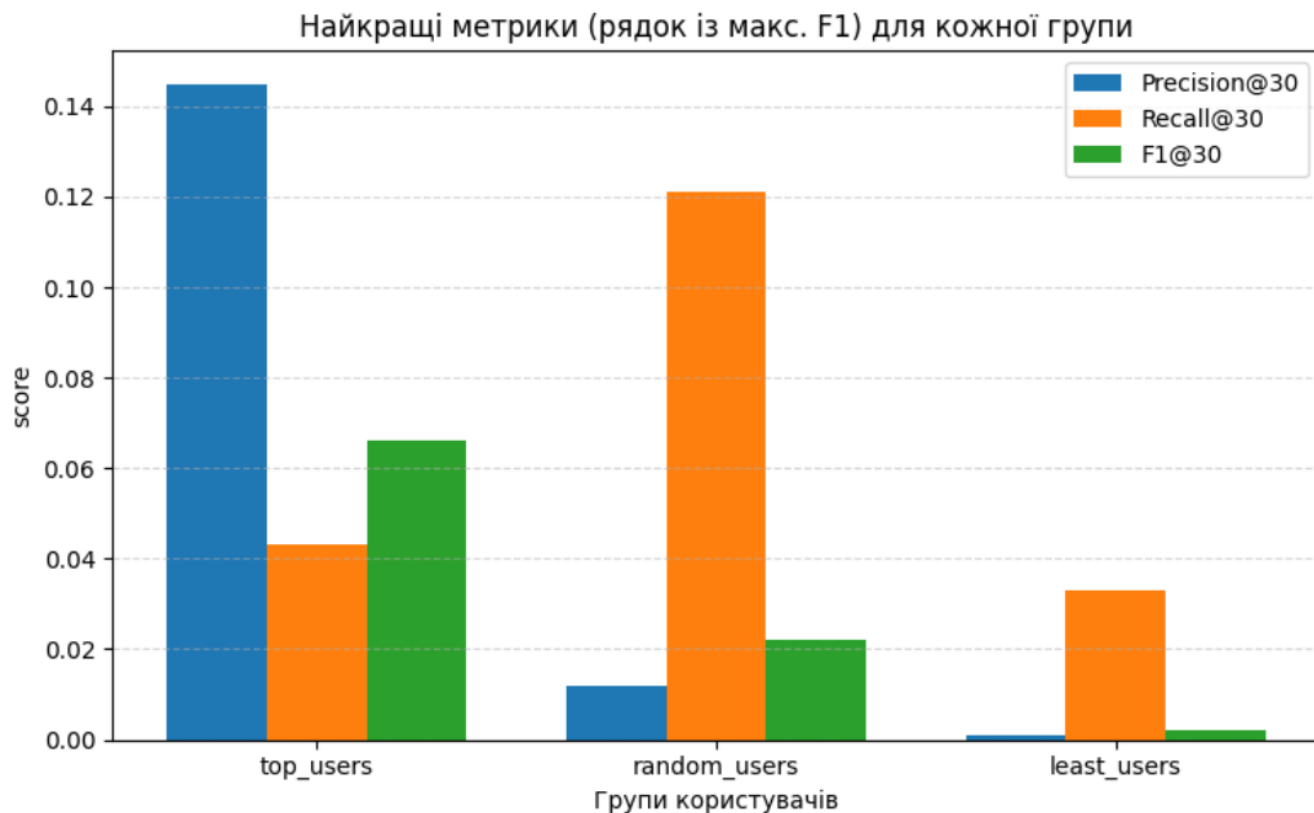


Рисунок 3.1 – Порівняння найкращих значень метрик Precision, Recall та F1-score для різних груп користувачів у колаборативній фільтрації

3.5.2 Результати контентно-орієнтованої фільтрації

У процесі тестування контентно-орієнтованої фільтрації було перевірено різні комбінації ознак, що включали кількість тематичних компонентів LDA та розмір категоріального простору. Для кожної групи користувачів було відібрано конфігурацію, яка забезпечила найвище значення F1-score. Зведені результати наведені в таблиці 3.2. Аналіз результатів контентно-орієнтованої фільтрації показав суттєві відмінності між трьома групами користувачів, які різняться рівнем активності та обсягом доступної історії взаємодій.

Таблиця 3.2 – Найкращі результати F1@30 для контентно-орієнтованої фільтрації за різних конфігурацій ознак

User group	Lda topics number	Categories number	Precision@30	Recall@30	F1@30
Top users	20	20	0.016	0.002	0.004
Top users	60	-	0.016	0.002	0.004
Top users	-	60	0.013	0.001	0.003
Random users	40	60	0.005	0.017	0.008
Random users	20	-	0.004	0.009	0.006
Random users	-	100	0.003	0.010	0.005
Least users	20	100	0.002	0.013	0.003
Least users	40	-	0.002	0.015	0.003
Least users	-	100	0.001	0.008	0.001

Для кращої демонстрації та аналізу даних, на рисунку 3.2 надане графічне представлення отриманих результатів.

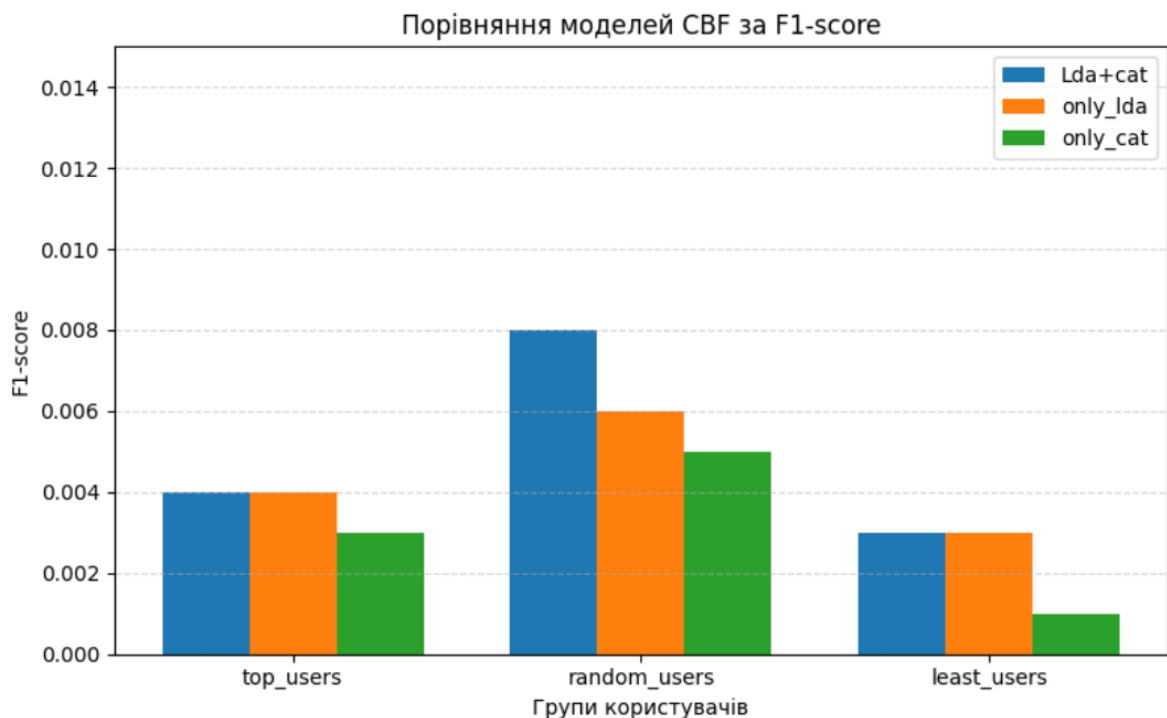


Рисунок 3.2 – Порівняння різних налаштувань моделей CBF за F1-score

Для групи `top_users`, що має найбільший обсяг оцінок та відгуків, найкращий результат склав $F1 = 0.004$. Він був отриманий одразу для двох конфігурацій:

- 20 тематичних компонент LDA та 20 категорій бізнесів;
- 60 тематичних компонент LDA та без застосування категоріальних ознак.

Значення $F1$ залишилися майже однаковими, незважаючи на різницю між параметрами. Це означає, що контентно-орієнтована фільтрація не має суттєвих переваг для дуже активних користувачів. CBF не враховує реальну історію поведінки, тому модель не може повністю використовувати велику кількість оцінок, які вже є. В результаті навіть зміна параметрів LDA майже не впливає на точність рекомендацій.

Контентно-орієнтована модель дала найкращі результати для групи `random_users`, де $F1$ досяг значення 0,008. Конфігурація з 40 темами LDA та 40 категоріями бізнесу була найкращою. Така поведінка є закономірною, оскільки користувачі з випадковим рівнем активності мають достатній обсяг інформації, щоб модель могла визначити відповідні патерни. Однак цей обсяг інформації не є настільки великим, щоб викликати надмірне зміщення чи шум у даних. Застосування тематичних і категоріальних ознак забезпечило найкращий баланс між змістовною деталізацією та узагальненістю. Конфігурації, які використовували лише категорії або LDA, показали значно нижчі результати ($F1 = 0.006$ та $F1 = 0.005$), що підкреслює перевагу комбінованих моделей.

Модель показала найнижче значення $F1$ для користувачів з найменшою активністю `least_users`. Найкращий результат становив 0.003 у двох випадках: коли використовувалися 20 тем LDA з 100 категоріями і 40 тем LDA, але без категорій. Низька якість у цій групі пояснюється недостатньою кількістю даних для інтерпретації тематичних і категоріальних ознак. Незалежно від вибраної конфігурації LDA або кількості категорій, якість рекомендацій значно падає, коли користувача майже немає історії взаємодій.

Окрім аналізу того, як різні конфігурації моделей впливають на якість рекомендацій, важливо також розглянути поведінку самих метрик оцінювання.

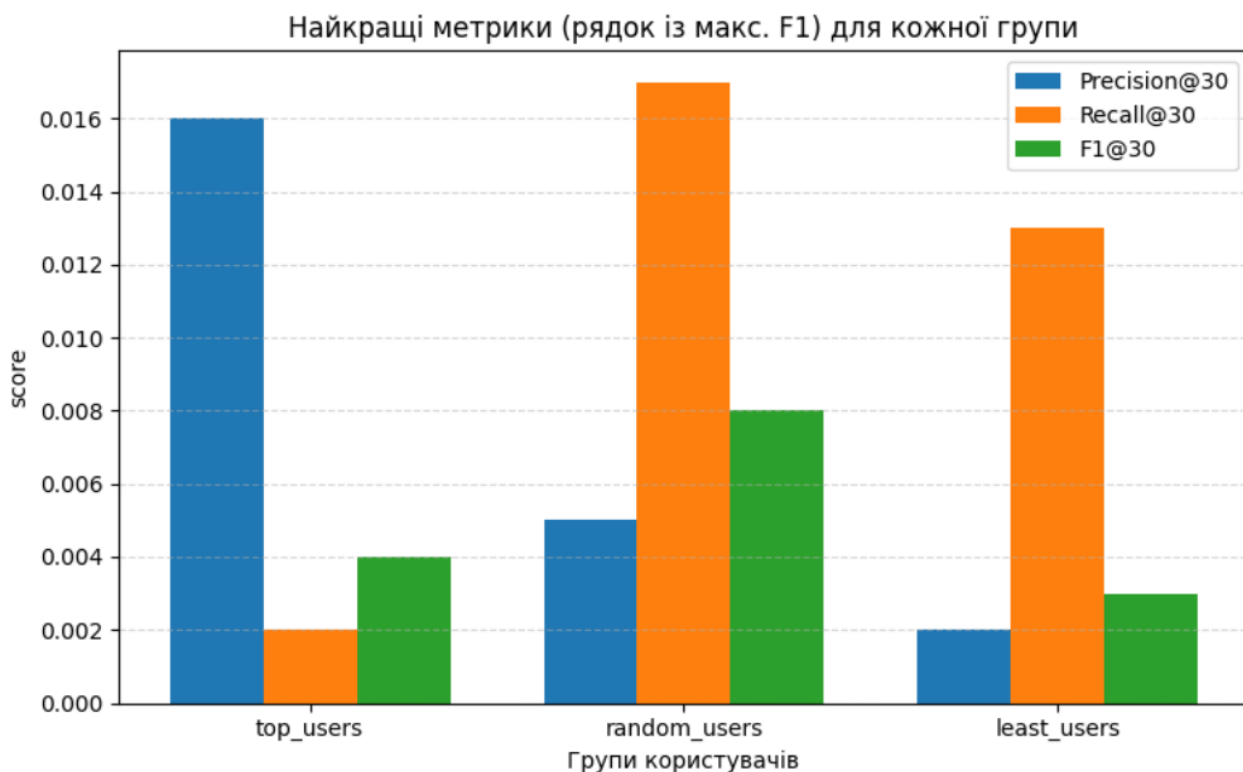


Рисунок 3.3 – Порівняння найкращих значень метрик Precision, Recall та F1-score для різних груп користувачів у контентно-орієнтованій фільтрації

На рисунку 3.3 наведено графічне порівняння значень Precision@30, Recall@30 та F1@30 для найкращих конфігурацій моделей у кожній групі користувачів.

У групі найактивніших користувачів спостерігається найвищий Precision@30 – 0.016, що пояснюється наявністю великої кількості інформації про вподобання користувача, яка добре відображена в структурі даних. Однак Recall@30 залишається дуже низьким (0.002). Це свідчить про те, що контентна модель знаходить лише невелику частину релевантних бізнесів, незважаючи на високий рівень деталізації профілів. Зведена метрика $F1@30 = 0.004$ також показує, що контентний метод не здатен повністю використати переваги великої історії взаємодій.

Група випадкових користувачів демонструє найбільш збалансовані та найвищі результати $F1@30 = 0.008$, як видно на графіку. Окрім цього, серед усіх

груп $\text{Recall}@30 = 0.017$ в даної групи найвищий, що свідчить про здатність моделі ідентифікувати більше схожих об'єктів.

Остання група (найменш активних користувачів) демонструє найгірші результати за F1 (0.003). Такий результат є очікуваним, оскільки модель практично не містить даних про вподобання цих користувачів, оскільки лише невелика кількість відгуків дозволяє системі визначити, що їм подобається.

3.5.3 Результати гібридної фільтрації

Як уже було зазначено у підрозділі, присвяченому налаштуванню гібридної фільтрації, попередні результати колаборативної та контентно-орієнтованої моделей не можуть бути прямо порівняні з результатами гібридної моделі. Це обумовлено тим, що гібридний підхід працює з іншим набором бізнесів, ніж ті методи, що використовувалися в окремих моделях. Колаборативна фільтрація формує рекомендації на основі бізнесів, що зафіксовані у взаємодіях користувачів, в той час як контентно-орієнтована – на основі бізнесів, що відібрані для побудови тематичних і категоріальних описів.

У випадку гібридного методу ці множини були об'єднані, що призвело до змін у просторі кандидатів для рекомендацій і вплинуло на значення оцінюваних метрик. З метою забезпечення коректного порівняння було повторно розраховано результати колаборативної та контентно-орієнтованої фільтрації для тих самих користувачів, з тим самим тестовим набором і з використанням об'єднаного простору бізнесів, який застосовується в гібридному підході. Це дало змогу отримати узгоджені значення метрик для всіх трьох алгоритмів.

Підсумкові результати представлені в таблиці 3.3. Можна помітити, що гібридний метод забезпечив найвищий F1-score у групі `random_users`, де поєднання колаборативної та контентної інформації виявилось найбільш ефективним. Для групи `top_users` гібридний підхід показав лише незначне покращення точності порівняно з колаборативною моделлю, що пояснюється високою активністю цих

користувачів і насиченістю їхніх профілів. Водночас для `least_users` результати гібридної та колаборативної моделей практично збіглися, що пояснюється недостатньою кількістю даних і обмеженою корисністю їхнього комбінування.

Таблиця 3.3 – Результати роботи всіх розроблених методів фільтрації на одних і тих самих даних

User group	Method	Precision@30	Recall@30	F1-score@30
Top users	Hybrid	0.268	0.041	0.071
Top users	CF	0.234	0.037	0.063
Top users	CB	0.019	0.003	0.005
Random users	Hybrid	0.178	0.496	0.262
Random users	CF	0.169	0.491	0.254
Random users	CB	0.007	0.022	0.011
Least users	Hybrid	0.092	0.502	0.156
Least users	CF	0.092	0.502	0.156
Least users	CB	0.002	0.014	0.004

Гібридна фільтрація продемонструвала трохи вищі показники F1 у порівнянні з чистою колаборативною фільтрацією. Зокрема, для випадкової групи користувачів значення F1 зросло з 0.251 до 0.261. Однак ця різниця виявилася незначною, що ілюструє рисунок 3.4. Основною причиною стала низька якість прогнозування контентно-орієнтованої фільтрації, яка показала F1 на рівні 0.003–0.01. Це обмежувало можливість значного покращення вже міцних результатів колаборативної моделі. Хоча під час експерименту було випробувано різні конфігурації контентних ознак, такі як кількість тем LDA, розмір категоріального словника та їх комбінації, найкращі параметри все одно не дозволили досягти високої точності через обмеження самого датасету. Yelp Dataset має багато шуму, містить неструктуровані тексти, неоднорідні категорії та істотно варіативну поведінку користувачів, що ускладнює формування якісних бізнес-профілів на основі текстових характеристик.

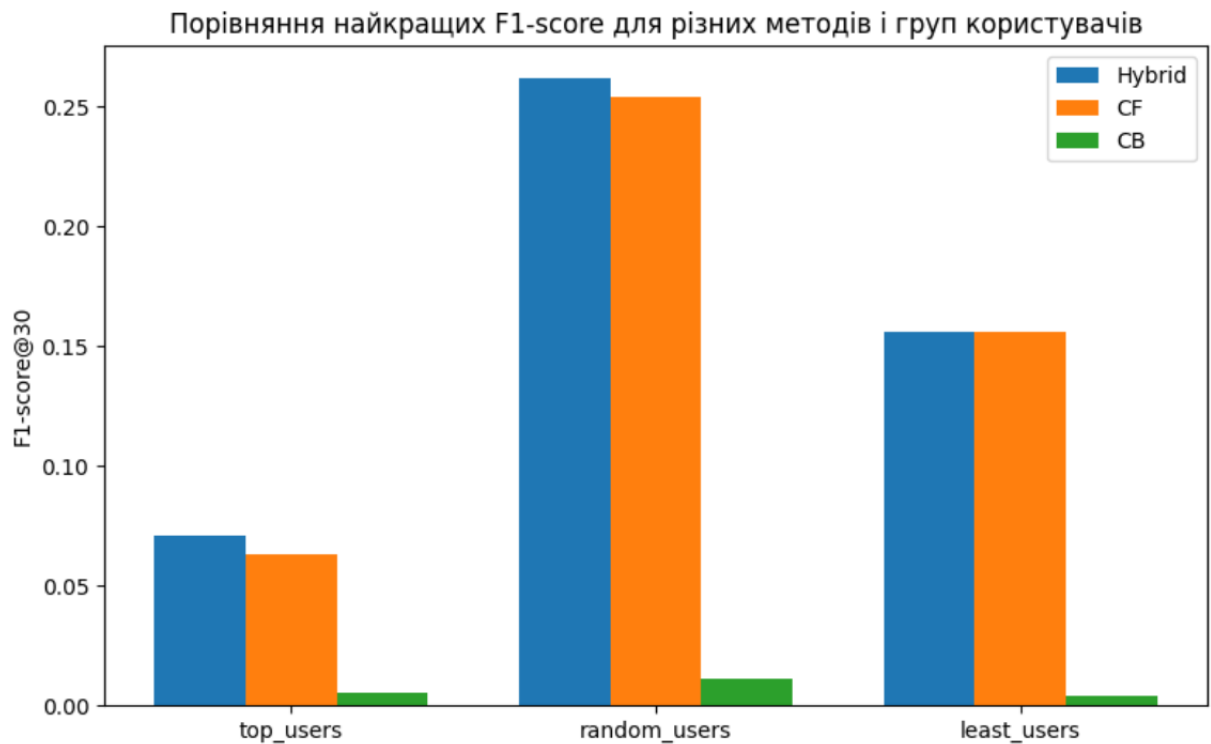


Рисунок 3.4 – Найкращі значення F1-score@30 для усіх методів фільтрацій на спільному наборі та окремих групах

У майбутньому якість контентно-орієнтованої частини можна покращити за рахунок застосування сучасніших методів, таких як sentence embeddings (SBERT), глибокі мовні моделі, мультимодальні описи або чистіша категоризація бізнесів. Проте для поточної вибірки саме колаборативна фільтрація забезпечила стабільно високі результати і слугувала основним чинником підвищення якості рекомендацій. Саме через це гібридний підхід виявився лише трохи ефективнішим за чисту колаборативну модель замість того, щоб демонструвати значну перевагу. Він покращив прогнози для всіх груп користувачів, але рівень покращення обмежувався слабкими показниками контентної компоненти.

Висновки до розділу 3

У даному розділі розглянуто яким чином було налаштовано три рекомендаційних підходи: колаборативна фільтрація, контентно-орієнтована фільтрація та їхня гібридна комбінація. Для кожного методу було використано окремі типи вхідних даних та параметрів: у моделі колаборативної фільтрації проводився підбір гіперпараметрів ALS, а контентно-орієнтована фільтрація потребувала оптимізації кількості LDA-тем і розмірів категоріальних векторів. До того ж було створено три групи користувачів, за допомогою чого алгоритми були протестовані за різних умов активності.

Експериментальне дослідження дало змогу оцінити ефективність кожного підходу на основі метрик Precision@30, Recall@30 та F1-score@30. Порівняльний аналіз показав, що гібридна фільтрація забезпечує найкращі результати серед розглянутих моделей, хоча її перевага над колаборативним методом виявилася незначною через низьку якість контентних ознак. Проте отримані дані підтверджують доцільність комбінування різних методів, а також наголошують на критичному значенні якості вхідних даних для вдосконалення точності рекомендацій.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

У даному розділі представлено повну програмну реалізацію створеної рекомендаційної системи. Розглянуто ключові модулі, взаємодію між компонентами системи, а також інструменти, що були використані для розробки моделі та створення інтерфейсу користувача. Окремо представлено архітектуру REST-орієнтованого API, структуру розробленого середовища та технічні засоби, що забезпечують виконання та відтворення результатів роботи системи.

4.1 Загальна архітектура системи

Розроблена система має дворівневу структуру, яка складається з обчислювального конвеєра формування рекомендацій і веб-орієнтованого середовища. Рисунок 4.1 показує, як ці рівні співпрацюють один з одним.

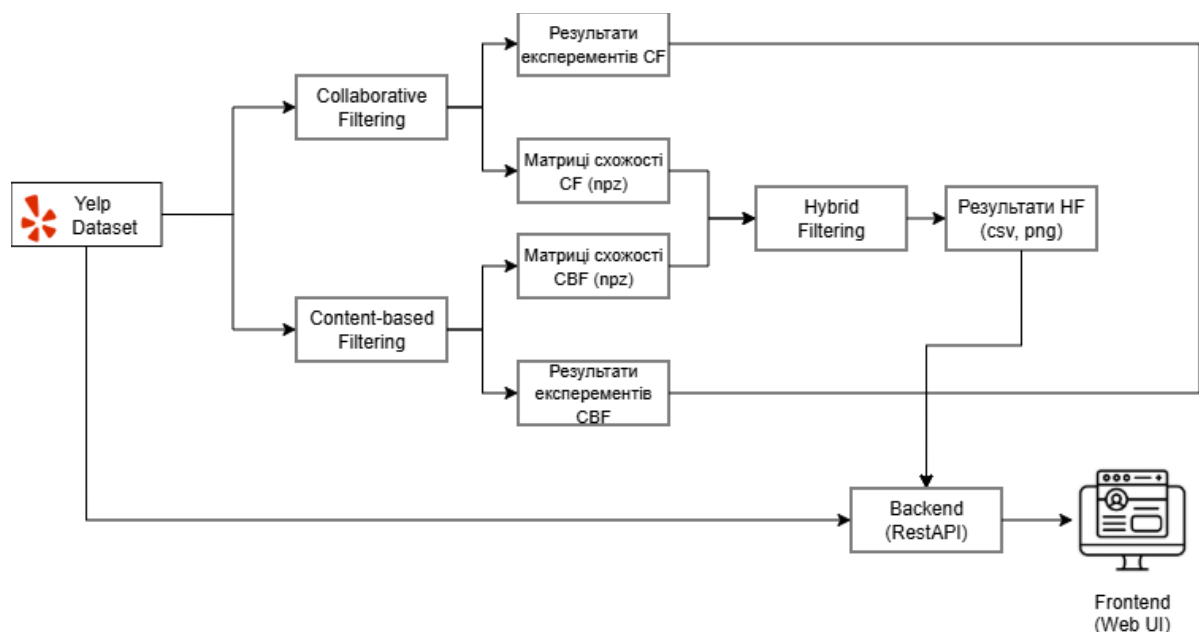


Рисунок 4.1 – Загальна схема роботи рекомендаційної системи

Розроблена система працює з відкритим набором даних Yelp Dataset, який містить структуровану інформацію про користувачів, бізнеси та відгуки. Після того, як ці дані нормалізуються та перетворюються у внутрішні формати (CSV або PKL), вони передаються до модулів колаборативної (CF) та контентно-орієнтованої (CBF) фільтрації.

Обидва модулі функціонують незалежно й генерують власні результати: матриці схожості у форматі NPZ [25], експериментальні показники (CSV) та візуалізації (PNG). На основі цих даних створюються вхідні набори для гібридного модуля, який об'єднує результати CF і CBF і створює підсумкові метрики та графічні матеріали.

Після цього функціонує веб-середовище, яке отримує доступ до всіх сформованих результатів: файлів Yelp Dataset, проміжних даних CF і CBF, а також підсумкових результатів гібридної фільтрації. Серверна частина реалізована як REST-орієнтований API, що забезпечує централізований доступ до обчислених даних та надає їх у структурованому вигляді фронтенду. Веб-інтерфейс дозволяє користувачу переглядати показники роботи моделі, результати експериментів та порівнювати різні конфігурації системи.

4.1.1 Структура файлів та модулів

Проект організовано у вигляді багаторівневої структури папок. Даний підхід забезпечує чітке логічне розділення реалізованих методів, вхідних даних, результатів експериментів та файлів веб-інтерфейсу.

У кореневій директорії розміщено такі основні підпапки:

- `collaborative_filtering`: реалізація методу колаборативної фільтрації;
- `contentbased_filtering`: реалізація контентно-орієнтованого підходу;
- `hybrid_filtering`: модуль гібридної фільтрації;
- `lda`: папка, що містить дані різних LDA конфігурацій;
- `yelp_dataset`: вхідні дані у форматі JSON;

- app: реалізація веб-інтерфейсу програмного середовища.

Папка **CollaborativeFiltering** містить програмний код, підготовлені дані та результати обчислень, що пов'язані з реалізацією методу колаборативної фільтрації. Основний файл даного модуля - `collaborative_filtering_code.ipynb`, у якому реалізований повний процес формування рекомендацій на основі історії оцінок користувачів.

У даному модулі виконуються наступні важливі етапи. Спочатку відбувається завантаження та очищення даних про користувачів, бізнеси й надані оцінки. Після цього будується матриця взаємодій «користувач–бізнес». Наступним кроком було проведення нормалізації даних та опрацювання пропущених значень, що надало можливість зменшити вплив індивідуальних особливостей користувачів під час оцінювання. Після підготовки даних обчислюється міра схожості між користувачами на основі косинусної відстані, для визначення групи користувачів із подібними вподобаннями та сформувані рекомендації на основі поведінки найближчих сусідів. На фінальному етапі модуль обчислює метрики якості: Precision, Recall та F1-score, а також формує графічні візуалізації для аналізу структури даних і результатів.

Для візуального аналізу розподілу даних створено низку графіків: `distribution_review_user_id.png`, `distribution_review_business_id.png`, `distribution_review_year.png`. Вони демонструють активність користувачів, популярність бізнесів і часову динаміку відгуків. Підсумкові значення метрик та результати експериментів зберігаються у файлах `experiment_results.csv`, `best_metrics_per_group.png`, `precision_comparison.png`, `recall_comparison.png` та `f1_comparison.png`.

Папка також містить піддиректорію `hybrid_data`, де розміщені сформовані матриці схожості та інші проміжні результати, необхідні для роботи гібридного модуля. З цього модуля гібридна фільтрація отримує всі підготовлені дані без потреби повторного обчислення.

Папка **ContentBasedFiltering** містить програмний код та результати обчислень, що пов'язані з контентно-орієнтованим методом рекомендацій.

Основний файл `content_based_filtering_code.ipynb` реалізує повний цикл контентно-орієнтованої фільтрації: опрацювання даних, побудова профілів, створення рекомендацій та оцінка точності моделі.

У межах даного модуля виконується очищення та стандартизація даних про бізнеси, що включає перетворення текстових, категоріальних і числових ознак у придатний формат для подальшої обробки. На основі цих даних формується векторне представлення бізнесів, після чого обчислюється міра схожості між користувачами та бізнесами. З використанням цих результатів формується топ-N список рекомендацій для кожного користувача, а отримані результати оцінюються за метриками Precision, Recall та F1-score. Окрім цього, для зручної оцінки результатів також створюються їх графічні візуалізації, які порівнюють ефективність різних конфігурацій моделі.

Окремим компонентом модуля є файл `CB_reviews_lda.ipynb`, у якому реалізовано тематичне моделювання текстів відгуків за допомогою алгоритму LDA (Latent Dirichlet Allocation). Ця частина дає змогу побудувати тематичні векторизації бізнесів та врахувати змістову подібність між ними, що особливо важливо для контентно-орієнтованих методів.

Файл `clean_business_data.ipynb` відповідає за попереднє детальне очищення та стандартизацію даних бізнесів. У даній папці також зберігаються проміжні результати у форматах PKL і CSV, що включають підготовлені ознаки, матриці категорій, результати тематичного моделювання, текстові та графічні представлення результатів експериментів. Зокрема, файли `categories_20_df.pkl`, `categories_60_df.pkl` та `categories_100_df.pkl`, що відповідають різним масштабам категоріальних представлень, що використовувалися під час експериментів.

Так само як і модуль колаборативної фільтрації, папка `ContentBasedFiltering` містить проміжні файли, необхідні для роботи гібридної системи. Завдяки цьому гібридний метод використовує вже сформовані ознаки та матриці схожості, що значно скорочує час виконання та усуває потребу в повторному опрацюванні даних.

Папка **HybridFiltering** містить програмний код та результати обчислень, пов'язані з реалізацією гібридного методу рекомендацій. Даний модуль є завершальним компонентом системи рекомендації та об'єднує два попередні підходи (контентно-орієнтований та колаборативний) у єдину узгоджену модель персоналізації. Такий підхід дозволяє отримати більш збалансовані та точні рекомендації, враховуючи як схожість користувачів між собою, так і змістову подібність об'єктів.

Спершу завантажуються попередньо підготовлені матриці схожості, сформовані на основі двох методів. Для забезпечення коректності подальших обчислень ці значення нормалізуються, що дозволяє усунути різницю в масштабах між контентними та колаборативними векторами.

Після цього модель виконує об'єднання результатів фільтрацій за допомогою різних механізмів зважування. В експериментальній частині було протестовано декілька значень вагового коефіцієнта α , який контролює внесок кожного з методів у фінальну рекомендацію. Для кожної конфігурації обчислюються метрики точності: Precision, Recall та F1-score.

Папка також містить проміжні та фінальні результати експериментів у форматах CSV і NPZ, включно з матрицями гібридних оцінок і наборами рекомендацій. Крім того, у директорії зберігаються графічні матеріали, а саме файл `method_results_comparison.png`, який демонструє порівняння ефективності гібридного методу з контентно-орієнтованим і колаборативним підходами.

Папка **lda** зберігає в собі дані тематичного моделювання LDA. У дослідженні розглядалося кілька LDA-конфігурацій із різними кількостями тем (20, 40 та 60), тому необхідно було збереження як самої моделі, так і відповідної матриці тем для кожного бізнесу. Після завершення тренування кожна LDA-модель зберігається у файлі `lda_model_{n_topics}_topics.pkl`, а матриця тем зберігається у файлі `topic_matr_lda_{n_topics}_topics.pkl`. Така структура гарантує повну відокремленість даних і дозволяє легко підключати їх до експериментів без необхідності повторного навчання.

Крім того, у директорії **lda/** містяться файли, що відносяться до TF-IDF-векторизації, включно з матрицею ознак (`tfidf_matrix_5000.pkl`) та словником термінів (`feature_names.pkl`). Вони також зберігаються у стиснутому форматі та підключаються до системи лише під час формування комбінованих наборів ознак. Такий підхід дозволяє суттєво зменшити час на завантаження та обробку даних у подальших етапах роботи. Усі проміжні результати виступають як частина внутрішньої бази даних рекомендаційної системи та є ключовими компонентами при формуванні різних конфігурацій контентно-орієнтованих моделей.

Папка **yelp_dataset** містить вхідні дані, отримані з відкритого набору Yelp Open Dataset, який використовується для навчання та тестування рекомендаційних алгоритмів. Основними файлами є:

- `yelp_academic_dataset_business.json`: дані про бізнеси (назви, категорії, середні рейтинги, атрибути);
- `yelp_academic_dataset_review.json`: текстові відгуки користувачів, їхні оцінки та посилання на відповідні бізнеси.

Кожен JSON-файл містить тисячі записів із вкладеними структурами (словниками), що потребували перетворення та нормалізації перед використанням. Під час роботи дані було перетворено у формат CSV і PKL, що надало можливість зручніше опрацьовувати їх у середовищі Python.

Папка **app** містить вебсередовище, яке дозволяє взаємодіяти з усіма реалізованими методами в інтерактивному форматі. Структура директорії включає клієнтську та серверну частину, що забезпечує повний цикл роботи інтерфейсу, від отримання даних до їх відображення.

Основні елементи папки:

- `backend/`: серверна частина на python (FastAPI), що містить REST-ендпоінти для отримання необхідних даних;
- `css/`: стилі інтерфейсу;
- `js/`: клієнтська логіка взаємодії з backend-ендпоінтами;
- `images/`: допоміжні графічні елементи інтерфейсу;

- `index.html`: основна вебсторінка, що містить структуру інтерфейсу, підключення скриптів і стилів, а також панель вибору методів фільтрації, параметрів та відображення результатів.

4.1.2 Архітектура серверної частини та API

Серверна частина розробленої рекомендаційної системи побудована за принципами модульності та відокремлення рівнів обробки даних. Її основним завданням є забезпечення доступу програми до файлів з даними, списків рекомендацій та результатів експериментів. Серверна архітектура ґрунтується на REST-орієнтованого API, що дозволяє клієнтській частині звертатися до різних функцій системи через чітко визначені HTTP-маршрути.

У процесі роботи сервер взаємодіє з кількома категоріями даних. Першою групою є JSON-файли Yelp Dataset, які використовувалися під час створення моделей та формування набору даних для тестування. Друга категорія – результати обчислення колаборативної, контентно-орієнтованою, гібридної моделей та їх налаштування для кожного з експериментів, а також сформовані рекомендаційні списки для кожного користувача. Сервер не виконує повторного навчання моделей, а працює виключно зі збереженими результатами.

API реалізовано у підпапці `backend/`, де зосереджено основні модулі обробки запитів, функції читання даних і маршрути. У середині цієї частини визначено набір REST-ендпоінтів, через які вебінтерфейс отримує інформацію про доступні моделі, групи користувачів, результати тестування, списки рекомендацій і параметри їх формування (табл. 4.1).

Таблиця 4.1 – REST-ендпоінти програмного середовища

Endpoint	Метод	Призначення	Параметри	Формат відповіді
/users	GET	Отримання списку доступних користувачів	—	JSON список користувачів (ids)
/recommendations/cf	GET	Повертає готові CF-рекомендації для вибраного користувача	user_id, group, factors, regularization, iterations	JSON список бізнесів (ids)
/recommendations/cb	GET	Повертає рекомендації контентно-орієнтованої моделі	user_id, group, combination	JSON
/recommendations/hybrid	GET	Повертає результати гібридної моделі	user_id, group, alpha	JSON
/business/{id}	GET	Повернення детальної інформації про бізнес	{business_id}	JSON об'єкта бізнесу
/analytics/best_result	GET	Повертає найкращі результати моделі та її налаштування	group, method	JSON (precision, recall, f1, параметри моделі)

Продовження таблиці 4.1

/analytics/plot	GET	Повертає графік	group, method	PNG
-----------------	-----	--------------------	---------------	-----

Взаємодія між API та фронтендом відбувається за класичною схемою: вебінтерфейс надсилає HTTP-запит, сервер читає відповідні файли з диску, виконує необхідні перетворення (наприклад, фільтрацію за групою або конфігурацією моделі), після чого формує структуровану відповідь. Завдяки такому підходу серверна частина виконує роль узагальнюючого рівня доступу до результатів моделювання, де кожна функція має чітко визначену відповідальність, а структура API залишається прозорою і легко розширюваною.

4.2 Інструменти розробки та середовище виконання

Розробка системи рекомендацій включала використання сучасного набору інструментів і бібліотек Python, які забезпечили ефективну обробку великих обсягів даних, побудову моделей машинного навчання, аналіз текстів і візуалізацію отриманих результатів. Основним середовищем розробки була IDE PyCharm, яка використовувалася для роботи з інтерактивними середовищами та звичайними Python-модулями. Jupyter Notebook [26] був використаний для проведення значної кількості експериментів; він надає зручні інструменти для покрокової розробки, налагодження коду та миттєве відображення проміжних результатів.

Окрім роботи з початковими JSON-файлами з Yelp Open Dataset, у процесі розробки активно використовувалися інші проміжні файли, які забезпечують різні рівні структуризації даних. Серед них були такі формати як: CSV, PKL та NPZ, що надавали можливість зручного зберігання даних для повторного їх використання, без необхідності повторного запуску обчислювальних блоків, що значно скоротило час проведення експериментів із моделями рекомендацій.

Основні використані бібліотеки:

- numpy – числові обчислення, вектори, матриці;
- pandas – табличні структури даних, профілі користувачів і бізнесів;
- scikit-learn – нормалізація даних, cosine similarity, TF-IDF, метрики точності;
- implicit – реалізація ALS-моделі для колаборативної фільтрації;
- nltk – очищення й підготовка текстових даних;
- scipy.sparse – створення й оптимізація розріджених матриць;
- matplotlib – побудова графіків та аналітичних діаграм;
- joblib – кешування проміжних результатів і збереження моделей;
- uvicorn + FastAPI – запуск серверної частини програмного середовища, обробка HTTP-запитів і повернення JSON-результатів.

Для створення програмного середовища були використані технології FastAPI та Uvicorn. Дані технології дозволяють швидко запустити локальний REST-сервер та ефективно обслуговувати HTTP-запити. FastAPI було обрано за його здатність добре працювати та спрощувати оголошення кінцевих точок. Унікальна архітектура ASGI сервера Uvicorn дозволяє швидко виконувати логіку бекенду та забезпечує стабільну взаємодію між обчислювальними модулями та клієнтською стороною.

Клієнтська частина застосунку була реалізована за допомогою звичайних веб-технологій HTML, CSS та JavaScript. Їх використання зробило інтерфейс простим і швидким у завантаженні та незалежним від сторонніх бібліотек. Такий підхід робить інтерфейс більш універсальним і дозволяє використовувати на будь-якій платформі. HTML відповідав за структуру сторінок, CSS відповідав за їх візуальне оформлення, а JavaScript відповідав за інтерактивність і обмін даними з бекендом через REST-запити.

Контроль версій програми здійснювався за допомогою Git з віддаленим репозиторієм на платформі GitHub, що дозволило нам вести структуровану історію змін, забезпечити повторюваність експериментів, організувати розподіл роботи між окремими функціональними блоками та надати резервні копії коду. Крім того,

GitHub використовувався для зберігання допоміжних матеріалів, проміжних наборів даних, графічних результатів та версій окремих модулів системи.

Для візуалізації результатів тестування та проміжних етапів роботи системи, було використано формати файлу PNG та SVG (наприклад, `distribution_review_user_id.png`, `best_metrics_per_group.png`). Найчастіше вони використовувались для порівняння результатів метрик, отриманих після проведення експериментів.

4.3 Програмне середовище

У рамках виконаної роботи було створено програмне середовище, яке надає можливість зручного перегляду результатів, отриманих під час експериментів.

На відміну від повноцінної рекомендаційної системи, яка працює в режимі реального часу і весь час здійснює складні обчислення, інтерфейс не генерує нових рекомендацій. Його функція полягає у відображенні вже попередньо розрахованих результатів для різних моделей і налаштувань. Таке рішення було зумовлене значними обсягами даних та високою обчислювальною складністю залучених методів. Моделі колаборативної та контентно-орієнтованої фільтрації, опрацьовують масиви даних із десятків тисяч бізнесів та потребують значних ресурсів для повторного навчання або обчислення показників схожості. Виконання таких операцій у реальному часі потребує багато ресурсів та може тривати від кількох хвилин до кількох годин. Саме з цієї причини всі моделі були попередньо розраховані під час виконання експериментів, а результати збережено у вигляді окремих файлів.

Протягом експериментів були сформовані рекомендації для певних груп користувачів та збережені разом із такими метриками як `Precision@30`, `Recall@30`, `F1@30`. Дана інформація міститься в системі й виступає базою даних для середовища, що дозволяє швидко переглядати рекомендації для кожного протестованого користувача.

Функціональність інтерфейсу передбачає:

- вибір типу рекомендаційної моделі (колаборативна, контентно-орієнтована або гібридна);
- налаштування параметрів моделі (наприклад, цільової групи або параметрів конкретної моделі);
- введення ідентифікатора користувача та отримання списку його рекомендацій з детальною інформацією про бізнес;
- перегляд аналітичної частини системи, де можна переглянути отримані результати системи у вигляді результатів метрик та їх графічного представлення.

Таким чином, створена програма служить інструментом для візуалізації та аналізу результатів експериментів, надаючи можливість порівняти ефективність різних алгоритмів. Відсутність необхідності повторних обчислень робить її оптимальним засобом для представлення результатів дослідницької роботи. Програма відрізняється простотою у використанні, швидким доступом до даних та чітким представленням інформації у зручному форматі.

На рисунку 4.2 представлено різні варіанти меню вибору типу та налаштувань рекомендаційної системи.

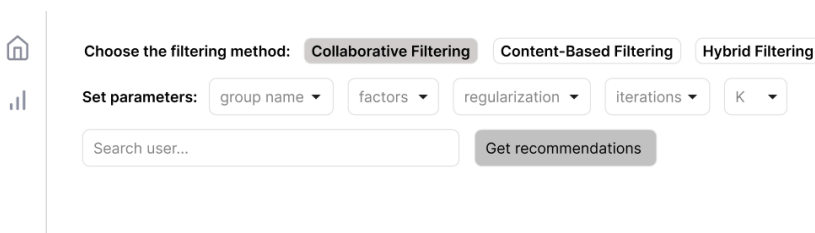
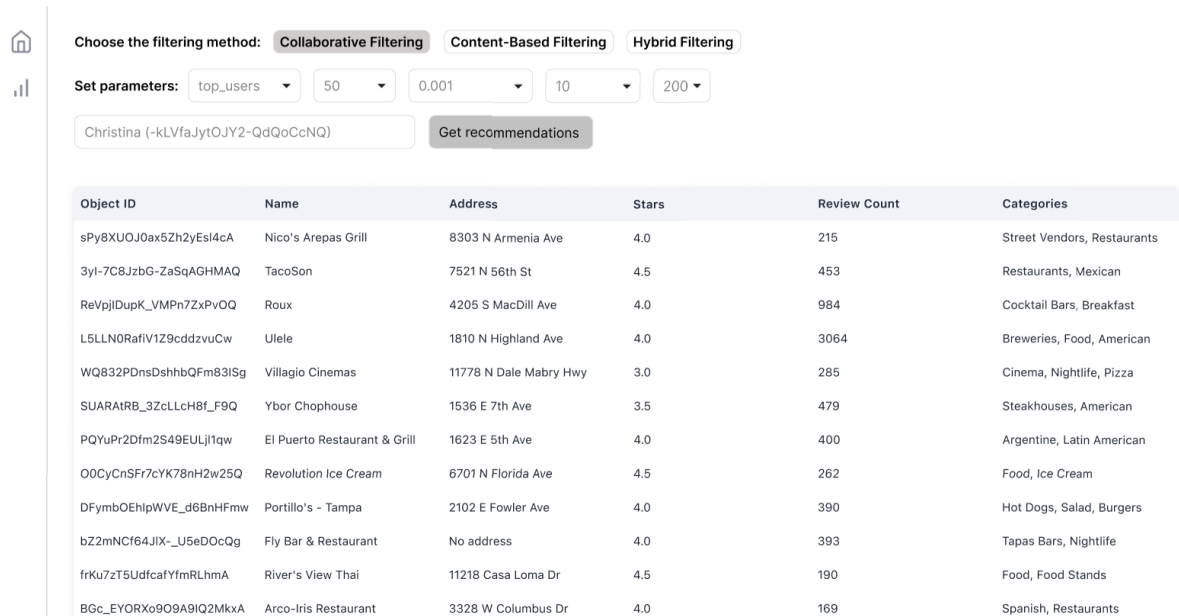


Рисунок 4.2 – Меню налаштування типу фільтрації та параметрів моделі

Спершу користувач обирає потрібний алгоритм фільтрації та відповідну групу користувачів. Після цього інтерфейс автоматично адаптується, динамічно показуючи набір параметрів, характерних для обраного методу, які налаштовуються за допомогою випадаючих списків. Далі потрібно ввести ім'я користувача. Завдяки функції автозаповнення процес пошуку значно спрощується.

Після введення даних необхідно натиснути кнопку «Отримати рекомендації». У відповідь система завантажує і відображає перелік бізнесів, рекомендованих для зазначеного користувача згідно з обраною моделлю. Даний список містить основну детальну інформацію про кожний рекомендований бізнес (рис. 4.3).



Object ID	Name	Address	Stars	Review Count	Categories
sPy8XUQJ0ax5Zh2yEsl4cA	Nico's Arepas Grill	8303 N Armenia Ave	4.0	215	Street Vendors, Restaurants
3yl-7C8JzbG-ZaSqAGHMAQ	TacoSon	7521 N 56th St	4.5	453	Restaurants, Mexican
ReVpjlDupK_VMPn7ZxPvOQ	Roux	4205 S MacDill Ave	4.0	984	Cocktail Bars, Breakfast
L5LLNORafivI29cddzvuCw	Ulele	1810 N Highland Ave	4.0	3064	Breweries, Food, American
WQ832PDnsDshhbQFm83ISg	Villagio Cinemas	11778 N Dale Mabry Hwy	3.0	285	Cinema, Nightlife, Pizza
SUARARB_3ZcLLCH8f_F9Q	Ybor Chophouse	1536 E 7th Ave	3.5	479	Steakhouses, American
PQYuPr2Dfm2S49EULj1qw	El Puerto Restaurant & Grill	1623 E 5th Ave	4.0	400	Argentine, Latin American
OOCyCnSfr7cYK78nH2w25Q	Revolution Ice Cream	6701 N Florida Ave	4.5	262	Food, Ice Cream
DFymbOEhlpWVE_d6BnHFmw	Portillo's - Tampa	2102 E Fowler Ave	4.0	390	Hot Dogs, Salad, Burgers
bZ2mNCf64JIX_U5eDOcQg	Fly Bar & Restaurant	No address	4.0	393	Tapas Bars, Nightlife
frKu7zT5UdfcafYfmlhmA	River's View Thai	11218 Casa Loma Dr	4.5	190	Food, Food Stands
BGc_EYORXo9O9A9IQ2MkxA	Arco-Iris Restaurant	3328 W Columbus Dr	4.0	169	Spanish, Restaurants

Рисунок 4.3 – Список рекомендацій для обраного користувача

Окрім цього, у розробленій програмі передбачено окрему секцію інтерфейсу, що відповідає за перегляд підсумкових результатів експериментів із рекомендаційними алгоритмами. Доступ до цієї секції здійснюється через бокове меню.

У даній секції користувач може ознайомитися з найефективнішими метрики, отриманими під час експериментів, зокрема $Precision@30$, $Recall@30$ та $F1@30$. Вони представлені для кожної групи користувачів, а також супроводжуються параметрами моделей, за яких було досягнуто найкращих результатів.

Інтерфейс надає можливість:

- обрати метод фільтрації;
- визначити цільову групу користувачів;
- переглянути найкращі результати для кожної групи;

- ознайомитися з параметрами моделей, які забезпечили найвищу якість;
- проаналізувати графічне порівняння метрик між різними групами.

Для отримання результатів експериментів, користувач повинен обрати бажаний метод фільтрації чи поєднання їх усіх, після цього за допомогою випадального списку обрати групу користувачів, яка його цікавить, або ж усі наявні групи. Для отримання результатів експериментів, потрібно натиснути на кнопку «Get analytics» (рис. 4.4).

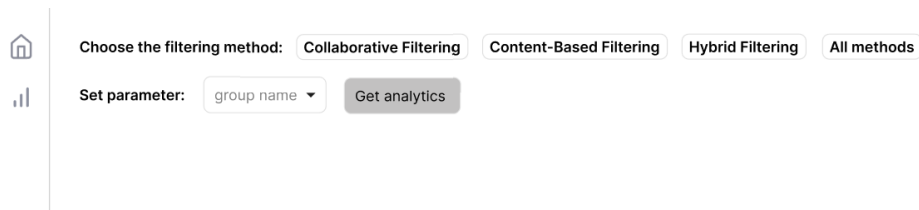


Рисунок 4.4 - Меню налаштування для отримання аналітичних даних

Після натискання кнопки, на екрані з'явиться інформація про налаштування моделей та їх результати (рис. 4.5). Дана частина інтерфейсу надає можливість порівняти отримані результати рекомендаційної системи. Важливо зазначити, що система не виконує кожен раз нові обчислення, а лише відображає вже раніше підготовлені дані.

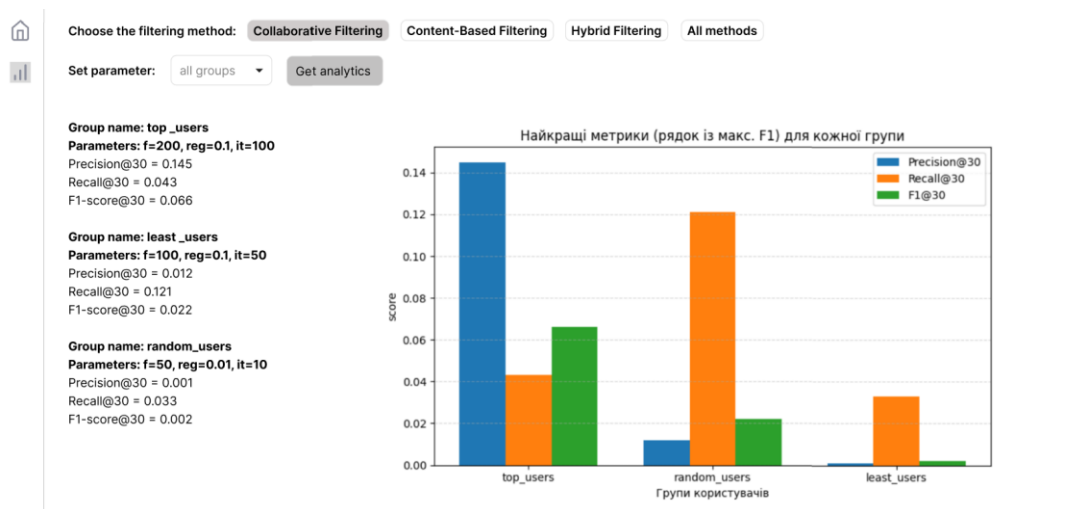


Рисунок 4.5 – Інтерфейс перегляду результатів проведених експериментів

Висновки до розділу 4

У даному розділі представлено та розглянуто програмну реалізацію рекомендованої системи. Представлена архітектура показала, як організовано внутрішню логіку системи та взаємодію між її модулями, ключові компоненти проекту та те, як вони пов'язані один з одним.

Було розглянуто структуру файлів і модулів, що відповідають за реалізацію методів фільтрації, обробку даних та представлення результатів. Розглядається також архітектура серверної частини та REST API, які поєднують обчислювальні модулі з веб-інтерфейсом, надаючи простий доступ до результатів моделей.

Крім того, показано роботу веб-застосунку, який дозволяє переглядати рекомендації, взаємодіяти з даними та отримувати візуальну аналітику. Реалізоване середовище підтвердило працездатність інтегрованих моделей і показало, що в подальшому система може бути використана для вирішення реальних завдань персоналізації.

5 ОСНОВНІ ПОЛОЖЕННЯ СТАРТАП-ПРОЕКТУ

Програмний модуль персоналізованих рекомендацій, який було розроблено в рамках магістерської роботи, служить основою для створення стартап-проєкту. Це рішення спрямоване на впровадження інструменту для аналізу поведінки користувачів та автоматизації процесу генерування рекомендацій у сферах локального бізнесу, сервісів відгуків або цифрових маркетплейсів. У даному розділі представлено оцінку ринкових можливостей, аналіз цільової аудиторії, опис конкурентного середовища та обґрунтування технологічної та економічної реалістичності ідеї.

5.1 Формування ідеї продукту та її ринковий контекст

У результаті дослідження методів персоналізації та аналізу проблеми інформаційного перенавантаження розроблено концепцію стартап-проєкту. Йдеться про інтелектуальну рекомендаційну систему, призначену для платформ із великим обсягом контенту або бізнес-об'єктів. Ця система автоматично створює персоналізовані рекомендаційні списки, базуючись на поведінкових даних користувачів, аналізі змісту їхніх відгуків і структурних характеристик самих об'єктів.

Ідея ґрунтується на кількох важливих принципах:

- сучасні інформаційні сервіси, такі як мобільні додатки місцевих бізнесів та онлайн-маркетплейси, накопичують великий обсяг контенту, що ускладнює процес вибору контенту для користувачів;
- персоналізовані рекомендації збільшують залученість користувачів, збільшують частоту повторних відвідувань і збільшують середній час користування сервісом;

- більшість локальних компаній або невеликих платформ стикаються з труднощами впровадження систем персоналізації через їхню складність і високу вартість розробки;

- створений модуль рекомендаційної системи легко інтегрується в будь-який веб-сервіс за допомогою REST API і працює з різними джерелами даних; однак для того, щоб працювати з новими наборами даних, модуль попередньої обробки має бути мінімально адаптований, а це означає, що його потрібно налаштувати під специфіку конкретного датасету.

Таким чином, технологічна платформа дозволяє створити ринковий продукт, який допоможе компаніям підвищити якість взаємодії з користувачами.

5.2 Загальні дані стартап-проєкту

Запропонований продукт розрахований на підприємства, які намагаються покращити свій цифровий сервіс, а також на великих операторів онлайн-платформ, яким потрібні точні та масштабовані модулі персоналізації.

Основна ідея проєкту полягає у впровадженні універсального інтелектуального модуля персоналізації, що може бути інтегрованим в уже існуючі веб-платформи та функціонувати незалежно від їхньої технічної інфраструктури. Даний модуль може працювати з різноманітними джерелами даних, адаптуватися до специфіки конкретної предметної області та генерувати персоналізовані рекомендації. Даний підхід спрямований на оптимізацію пошуку інформації, зменшення навантаження на користувача та покращення якості прийнятих ним рішень. В таблиці 5.1 наведена загальна характеристика ідеї стартап-проєкту, її зміст, застосування та вигода для користувача.

Таблиця 5.1 – Загальна характеристика ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Інтелектуальний модуль персоналізації рекомендацій, який аналізує історію взаємодій користувача, зміст відгуків і характеристики об'єктів, формуючи релевантні рекомендації на основі статистичних методів фільтрації.	1. Платформи відгуків (аналог Yelp).	– Менше часу на пошук потрібних варіантів.
	2. Локальні бізнес-каталоги та мобільні додатки ресторанів.	– Індивідуальні рекомендації.
	3. Маркетплейси та системи бронювання.	– Зменшення інформаційного перевантаження.
	4. Сервіси, що надають персоналізовані послуги.	– Підвищення зручності та якості взаємодії.

5.3 Аналіз конкурентів та унікальні характеристики продукту

На ринку вже представлено чимало рішень, які частково або повністю реалізують механізми персоналізації. Це охоплює як вбудовані рекомендаційні модулі, такі як Amazon, Yelp [27] і Google Maps, так і окремі платформи машинного навчання, такі як AWS Personalize [28] або Google Vertex AI Recommendations [29].

Втім, незважаючи на те, що ці рішення різноманітні, більшість з них спрямовані на великі компанії. Вони потребують значних фінансових вкладень, оскільки базуються на «закритих» моделях, які складно адаптувати до конкретних завдань у вузьких галузях. Крім того, їх розгортання вимагає великого обсягу даних і особливої інфраструктури. Це створює значні проблеми для малих платформ, стартапів і освітніх проєктів. Опції персоналізації стають або надто дорогими, або технічно складними для впровадження.

Завдяки більш гнучкій і доступній архітектурі дані обмеження були враховані та вирішені в проєкті. Основні переваги цього методу можна сформулювати таким чином:

- підтримка одразу трьох класів моделей – колаборативної, контентної та гібридної фільтрації;
- зручна робота з малими платформами та локальними наборами даних, що робить систему доступною навіть без великої інфраструктури;
- веб-інтерфейс для демонстрації роботи алгоритмів у режимі реального часу;
- проста інтеграція через REST API;
- можливість розширення функціональності системи шляхом додавання нових типів ознак (LDA-топіки, змінені TF-IDF матриці, розширені категорії тощо) без необхідності змінювати базову архітектуру чи логіку роботи моделей.

Характеристики продукту порівнювалися з аналогами за критеріями вартості, технологічної складності, точності рекомендацій та доступності. Отримані результати продемонстровано в таблиці 5.2.

Таблиця 5.2 – Порівняння ключових характеристик ідеї проєкту з конкурентами

№ з/п	Економічні характеристики	Концепція (стратегія) конкурентів			
		Власний проєкт	Конкурент 1 (AWS Personalize)	Конкурент 2 (Vertex AI)	Конкурент 3 (Yelp Internal AI)
1	Вартість	Низька	Висока	Висока	Висока
2	Гнучкість моделей	Висока	Середня	Висока	Низька
3	Поріг обсягу даних	Низький	Високий	Високий	Дуже Високий
4	Можливість локального розгортання	Так	Ні	Ні	Ні

Продовження таблиці 5.2

5	Інтеграція через API	Так	Так	Так	Ні
6	Адаптація малих бізнесів	Висока	Низька	Низька	Низька
7	Прозорість алгоритмів	Повна	Часткова	Часткова	Немає

Для отримання повного розуміння конкурентного середовища кожна платформа пройшла додатковий аналіз сильних, нейтральних і слабких сторін. Результати порівняння представлені в таблиці 5.3.

Таблиця 5.3 – Порівняння слабких, нейтральних та сильних сторін конкурентів

	Власний проєкт	Конкурент 1 (AWS Personalize)	Конкурент 2 (Vertex AI)	Конкурент 3 (Yelp Internal AI)
Слабкі сторони	-потребує адаптації під нові дані; -обмежене масштабування.	-висока ціна; -залежність від AWS.	-високий поріг входу; -складна інтеграція.	-недоступна зовнішнім користувачам; -закритий код.
Нейтральні сторони	-класичні ML-методи; -базова інфраструктура.	-висока масштабованість, але “чорний ящик”.	-сильні ML-інструменти, але потребують досвіду.	-оптимізація під власний набір даних.
Сильні сторони	-низька вартість; -локальне розгортання.	-автоматизація; -стабільність; -промисловий рівень.	-гнучкі моделі; -висока масштабованість.	-найбільший обсяг даних; -вузькі моделі.

Дані демонструють, що запропонований модуль може зайняти нішу рішень для малого та середнього бізнесу, а також платформ, які потребують адаптивної персоналізації та ресурсоефективності.

5.4 Технологічна здійсненність проєкту

У рамках оцінки технологічної здійсненності продукту було проведено аудит технологій, необхідних для його розробки та запуску. Використані інструменти охоплюють повний цикл розробки системи, технології для обробки даних, побудови моделей, розробки API тощо. Саме дані інструменти дозволяють визначити, наскільки легко продукт можна адаптувати до реального середовища та які частини потребують покращення.

Технології створення програмного продукту:

- Python, Pandas, NumPy – основні бібліотеки для обробки даних;
- Scikit-learn, Implicit – статистичні моделі для CF і текстового аналізу;
- LDA (Latent Dirichlet Allocation) – тематичне моделювання;
- FastAPI – створення REST API;
- HTML/CSS/JS – фронтенд веб-інтерфейсу;
- Docker – контейнеризація;
- GitHub – контроль версій.

Крім того, оцінка технологій містить в собі оцінку їх масштабованості, простоти інтеграції та можливої залежності від зовнішніх сервісів. Окремі компоненти системи можна оновлювати чи замінювати без суттєвого впливу на інші частини архітектури. Даний підхід дозволяє розширювати функції, такі як оптимізація векторизації, перенесення backend-частини в хмарні середовища або заміна класичних моделей на нейромережеві.

Щодо рівня технологічного досягнення (TRL) [30], то на основі проведених експериментів система може бути оцінена як TRL 5–6, що є прототипом, який

працює в реальних умовах. Після оптимізації продуктивності можна досягти рівня TRL 7–8, що означає її готовність до впровадження комерційних послуг.

5.5 Ринкові можливості, загрози та потенціал комерціалізації

Для розробки стартап-проєкту важливим етапом є оцінка ринку, оскільки це надає можливість визначити, наскільки розроблений модуль може бути конкурентоспроможним. Окрім цього це також відкриває нові можливості для його впровадження. Аналіз можливостей та ризиків має на меті обґрунтувати доцільність виходу на ринок і реалістично оцінити шанси продукту на комерціалізацію.

Зі сторони ринкових можливостей, розроблений модуль персоналізації може бути інтегрований у багатьох цифрових сервісах. Потреба в персоналізованих рекомендаціях зростає, а рішення, що доступні на локальних платформах, є або дорогими, або складними для впровадження. Завдяки своїй структурі модуль може бути вбудований у вже існуючі системи і при цьому не потрібно їх повністю перебудовувати. Окремою можливістю є моделі монетизації, такі як SaaS або B2B-ліцензування, що пропонують можливість розширювати продукт без значних витрат на підтримку.

Під час планування впровадження системи також необхідно враховувати ринкові загрози. Одним із найважливіших з них є конкуренція з великими технологічними компаніями, які активно пропонують інструменти персоналізації AI. Швидкий розвиток нейромережевих моделей є ще однією загрозою, адже вона може вимагати постійного технологічного оновлення проєкту. Крім того, ефективність системи значною мірою залежить від якості вхідних даних, якщо відгуків недостатньо або атрибути неструктуровані, якість рекомендацій може знизитися.

Незважаючи на дані ризики, розроблене рішення має значний потенціал для продажу. Модуль можна використовувати на локальних платформах відгуків,

мобільних додатках ресторанів і послуг, системах бронювання та маркетплейсах. Крім того, є можливість запропонувати API-доступ або спростити мобільну версію. Модуль може бути адаптований до багатьох форматів даних завдяки своїй гнучкій архітектурі. Для цього достатньо лише внести незначні зміни до попередньої обробки даних, щоб підлаштувати систему під нові датасети. Дана особливість робить продукт придатним для широкого кола замовників.

Таким чином, ринкова ситуація створює сприятливі умови для впровадження розробленого рішення, оскільки існує багато можливостей для його застосування та реальні сценарії комерціалізації, незважаючи на певні технологічні та конкурентні перешкоди.

5.6 Напрями подальшого розвитку продукту

Для того, щоб розроблена система змогла працювати в промислових умовах і використовуватися в комерційних умовах, її потрібно суттєво вдосконалити. Це зумовлено технічними обмеженнями дослідницької версії, а також вимогами сучасного ринку до масштабованих, високопродуктивних сервісів персоналізації.

Оптимізація використання пам'яті та швидкодії модулів контентно-орієнтованої та колаборативної фільтрації є ключовими напрямками розвитку. Алгоритми на цьому етапі дослідження працюють з великими матрицями ознак, створеними TF-IDF, LDA та іншими методами побудови профілів. Це може призвести до надмірного навантаження на сервер і затримки відповіді. Таким чином, необхідно запровадити механізми стискування даних, оптимізацію матричних операцій або використання простих моделей векторизації, які є адаптованими для реального часу.

Ще одним напрямом є спрощення або заміна ресурсомістких текстових моделей, зокрема TF-IDF та LDA. У поточній версії системи вони забезпечують якісну контентну фільтрацію, проте потребують значних обчислювальних ресурсів та тривалого оновлення моделей при зміні даних. Для ринкового застосування

важливо розглянути перехід на інкрементальні або lightweight-алгоритми [31], які дозволяють оновлювати ознаки без повного перерахунку.

Удосконалення веб-інтерфейсу та розширення функціональності є важливими з точки зору продуктового використання. На даному етапі інтерфейс користувача забезпечує перегляд рекомендацій, однак для комерційної версії необхідні механізми авторизації, ізоляція сесій, система управління доступом і можливість кастомізації профілів. Дані рішення визначають, наскільки продукт відповідає стандартам безпеки та приватності даних.

Оскільки якість рекомендацій безпосередньо залежить від актуальності даних, розробка модуля автоматичного оновлення моделей є важливою частиною процесу. На відміну від реального сервісу, у прототипі оновлення здійснюється вручну. Завдяки переходу до автоматичного планового оновлення або потокової обробки даних, система зможе адаптуватися до змін у поведінці користувачів та буде гарантувати високу якість персоналізації.

Наступним пріоритетом є створення десктопної або мобільної версії системи. Це може бути корисним для компаній, які хочуть використовувати рекомендаційну систему як окремий продукт або всередині корпоративної мережі.

Також важливим елементом масштабування є впровадження кешування результатів. За допомогою цього можна зменшити кількість обчислень, зменшити навантаження на сервер і забезпечити швидку реакцію системи, що є життєво важливим для реальних умов для користувачів.

У сукупності дані напрями розвитку дозволять створити повноцінний продукт, здатний працювати в умовах високих навантажень, надавати стабільні рекомендації та відповідати ринковим вимогам до сучасних інтелектуальних сервісів персоналізації.

Висновки до розділу 5

Розроблений стартап-проект демонструє технологічну та ринкову здійсненність запропонованого рішення. Аналіз показав, що інтелектуальний модуль персоналізації має переваги порівняно з іншими; він може працювати з невеликими кількостями даних, легко інтегруватися в існуючі веб-платформи та має потенціал масштабуватися до комерційного рівня. Подальший розвиток продукту дозволяє створити повноцінну екосистему персоналізованого контенту та вийти на ринок SaaS-рішень для цифрових сервісів.

ВИСНОВКИ

1. У результаті проведеного дослідження було проведено детальний аналіз сучасних підходів побудови рекомендаційних систем. Це дозволило визначити три найбільш релевантні методи фільтрації: колаборативну, контентно-орієнтовану та гібридну.

2. Під час роботи з Yelp Open Dataset було виконано повний цикл підготовки даних, включаючи первинне зчитування великих JSON-файлів, очищення, нормалізацію, виділення ознак та створення окремих наборів даних для кожного методу фільтрації. Даний підхід створив надійний фундамент для тренування моделей і забезпечив коректність подальших обчислень.

3. У ході обчислювальних експериментів виконано порівняльне тестування реалізованих алгоритмів, що дало змогу оцінити їх ефективність за різними метриками – Precision@30, Recall@30 та F1@30. Отримані результати продемонстрували, що гібридний підхід виявився найкращим, оскільки він поєднує сильні сторони колаборативної та контентно-орієнтованої фільтрації.

4. На основі реалізованих методів рекомендацій створено програмне середовище. Розроблений застосунок дозволяє завантажувати результати моделювання, обирати метод рекомендацій, взаємодіяти з підготовленими CSV/NPZ-файлами та відображати підсумкові рекомендації у зручному вигляді для користувача. Оскільки система розширювана, модульна та адаптована для інтеграції нових методів і джерел даних, вона придатна для подальших наукових експериментів і практичного застосування.

Результати, отримані в магістерській дисертації, були використані під час створення стартап-проекту Motiv AI, який спрямований на допомогу студентам в плануванні навчання та підвищенні їх мотивації. Підходи, викладені в роботі, до обробки даних користувачів, створення профілів та персоналізації рекомендацій були інтегровані в механізми взаємодії та рекомендації цього сервісу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Huryn I.A., Shapovalova S. I. Intelligent personalization methods based on the user's history for recommendation system. Сучасні проблеми наукового забезпечення енергетики: матеріали XXII міжнар. наук.-практ. конф. молод. вчених і студентів у 2 т. 22–25 квіт. 2025, Київ. КПІ ім. Ігоря Сікорського. Вид-во «Політехніка». Т.2. С.182-183. ISBN 978-966-990-154-5. https://iate.kpi.ua/uploads/p_164_79639699.pdf

2. Shapovalova S., Huryn I. Recommendations based on the processing of text data and user behavior patterns. Progressive Approaches in Science and Engineering: Collection of Scientific Papers with Proceedings of the 2nd International Scientific and Practical Conference. International Scientific Unity. November 26-28, 2025. Copenhagen. Pp. 329-332. ISBN 979-8-89704-979-0. <https://10.70286/ISU-26.11.2025.006>

3. Гурин І., Шакіров А., Шостак Р. Motiv – цифровий AI-помічник для підвищення продуктивності студентів. Startup Incubator Odessa Polytechnic National University. Сертифікат №OP-STI-2025-004», 2025. URL: <https://startup-incubator.vercel.app/>

4. Yelp Dataset. Kaggle: веб-сайт. URL: <https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset> (дата звернення: 01.11.2025).

5. Precision and recall at K in ranking and recommendations. Evidently AI: веб-сайт. URL: <https://www.evidentlyai.com/ranking-metrics/precision-recall-at-k> (дата звернення: 10.11.2025).

6. Як працює Deep Learning. IT School FoxmindEd: веб-сайт. URL: <https://foxminded.ua/deep-learning/> (дата звернення: 19.11.2025).

7. Collaborative Filtering. VPN Unlimited: веб-сайт. URL: <https://www.vpnunlimited.com/help/cybersecurity/collaborative-filtering> (дата звернення: 10.11.2025).

8. What is content-based filtering? IBM: веб-сайт. URL: <https://www.ibm.com/think/topics/content-based-filtering> (дата звернення: 19.11.2025).

9. What is hybrid filtering in recommender systems? Milvus: веб-сайт.
URL: <https://milvus.io/ai-quick-reference/what-is-hybrid-filtering-in-recommender-systems> (дата звернення: 10.11.2025).

10. The Cold Start Problem for Recommender Systems. Medium: веб-сайт.
URL: <https://medium.com/@markmilankovich/the-cold-start-problem-for-recommender-systems-89a76505a7> (дата звернення: 19.11.2025).

11. Working With JSON Data in Python. Real Python: веб-сайт.
URL: <https://realpython.com/python-json/> (дата звернення: 19.11.2025).

12. Pandas – Parsing JSON Dataset. GeeksforGeeks: веб-сайт.
URL: <https://www.geeksforgeeks.org/python/pandas-parsing-json-dataset/> (дата звернення: 10.11.2025).

13. A Comprehensive Guide: Convert String to Dictionary Python. Analytics Vidhya: веб-сайт. URL: <https://www.analyticsvidhya.com/blog/2024/01/a-comprehensive-guide-convert-string-to-dictionary-python/> (дата звернення: 01.11.2025).

14. Text Preprocessing in NLP with Python Codes. Analytics Vidhya: веб-сайт.
URL: <https://www.analyticsvidhya.com/blog/2021/06/text-preprocessing-in-nlp-with-python-codes/> (дата звернення: 01.11.2025).

15. TF-IDF Vectorizer Explained. Medium: веб-сайт.
URL: <https://medium.com/@gusainanurag58/tf-idf-vectorizer-explained-373b3f07d23b> (дата звернення: 10.11.2025).

16. What is latent Dirichlet allocation. IBM: веб-сайт.
URL: <https://www.ibm.com/think/topics/latent-dirichlet-allocation> (дата звернення: 10.11.2025).

17. What is stemming? IBM: веб-сайт.
URL: <https://www.ibm.com/think/topics/stemming> (дата звернення: 20.11.2025).

18. How do recommender systems incorporate user profiles? Milvus: веб-сайт.
URL: <https://milvus.io/ai-quick-reference/how-do-recommender-systems-incorporate-user-profiles> (дата звернення: 10.11.2025).

19. Collaborative filtering: How to build a recommender system. Redis: веб-сайт. URL: <https://redis.io/blog/collaborative-filtering-how-to-build-a-recommender-system/> (дата звернення: 10.11.2025).

20. Alternating Least Squares: A Cornerstone in Modern Data Analysis and Recommendation Systems. Medium: веб-сайт. URL: <https://medium.com/aimonks/alternating-least-squares-a-cornerstone-in-modern-data-analysis-and-recommendation-systems-ceb7310313f9> (дата звернення: 10.11.2025).

21. Step-by-Step Guide to Building Content-Based Filtering. StrataScratch: веб-сайт. URL: <https://www.stratascratch.com/blog/step-by-step-guide-to-building-content-based-filtering/> (дата звернення: 10.11.2025).

22. Nearest Neighbors. scikit-learn: веб-сайт. URL: <https://scikit-learn.org/stable/modules/neighbors.html> (дата звернення: 10.11.2025).

23. The cosine similarity and its use in recommendation systems. Medium: веб-сайт. URL: <https://naomy-gomes.medium.com/the-cosine-similarity-and-its-use-in-recommendation-systems-cb2ebd811ce1> (дата звернення: 10.11.2025).

24. The Elbow Method: Finding the Optimal Number of Clusters. Medium: веб-сайт. URL: <https://medium.com/@zalarushirajsinh07/the-elbow-method-finding-the-optimal-number-of-clusters-d297f5aeb189> (дата звернення: 12.11.2025).

25. Saving NumPy Arrays to .npz Files: A Complete Guide. SparkCodehub: веб-сайт. URL: <https://www.sparkcodehub.com/numpy/data-export/save-npz> (дата звернення: 10.11.2025).

26. Mastering Jupyter Notebooks: Best Practices for Data Science. CoderPad: веб-сайт. URL: <https://coderpad.io/blog/data-science/mastering-jupyter-notebooks-best-practices-for-data-science/> (дата звернення: 10.11.2025).

27. Transforming Yelp with AI: Smarter search, seamless connections, instant answers and a more conversational experience. Yelp Blog: веб-сайт. URL: <https://blog.yelp.com/news/fall-product-release-2025/> (дата звернення: 20.11.2025).

28. Amazon Personalize features. Amazon Web Services: веб-сайт. URL: <https://aws.amazon.com/de/personalize/features/> (дата звернення: 20.11.2025).

29. About recommendation models. Google Cloud: веб-сайт. URL: <https://docs.cloud.google.com/retail/docs/models> (дата звернення: 20.11.2025).

30. Рівень технологічної готовності (TRL). Rasph: веб-сайт. URL: <https://www.rasph.com/uk/рівень-готовності-технології-трл> (дата звернення: 20.11.2025).

31. Recommendation Systems: Applications and Examples. AIMultiple: веб-сайт. URL: <https://research.aimultiple.com/recommendation-system/#tutorial-building-a-recommendation-system-with-lightfm> (дата звернення: 20.11.2025).

ДОДАТОК А

Тези XXII Міжнародної науково-практичної конференції молодих
вчених та студентів “Сучасні проблеми наукового забезпечення
енергетики”

УКР.НТУУ “КПІ ім. Ігоря Сікорського”_ІАТЕ_ЦТЕ_ТР-41мп

Аркушів 3

Київ – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

У 2-х томах

Том 2

Матеріали XXII Міжнародної
науково-практичної конференції
молодих вчених і студентів
м. Київ, 22–25 квітня 2025 року



Київ – 2025

UDC 004.8

MS's programme 1st year Huryn I.A.

¹ Assoc.prof., cand.eng.sc. Shapovalova S.I.

<https://scholar.google.com.ua/citations?user=bigIE98AAAAJ&hl=en>

¹ Igor Sikorsky Kyiv Polytechnic Institute

INTELLIGENT PERSONALIZATION METHODS BASED ON THE USER'S HISTORY FOR RECOMMENDATION SYSTEM

Statement of the problem and its relevance. Modern online services are actively using recommendation systems in order to improve the personalization of content and to attract new audiences. However, existing approaches often face challenges such as the lack of customer history and the difficulty of taking into account all the preferences of other users. That's why it's important to optimize algorithms on a case-by-case basis to improve the accuracy and relevance of recommendations. This ensures an individual approach to each customer and can also take into account changes in user preferences in real time.

Analysis of the latest research. Recommendation systems are an important part of many online platforms, enabling users to find relevant content or products based on their preferences and behavior. For example, Netflix [1] uses collaborative filtering, which generates recommendations based on other users' views and ratings. However, this method has limitations in the context of a cold start, when a new user does not have enough information to make accurate recommendations.

As for Spotify [2], it combines collaborative filtering with a content-oriented approach that takes into account not only user behavior but also the characteristics of music tracks, such as genre, tempo and mood. This approach does not always work effectively with new or rare content elements. When there is insufficient user interaction, the system struggles to generate relevant recommendations.

The online platform Amazon [3] also uses a hybrid approach to help it provide high-quality recommendations to its users. However, scaling data can present several challenges, including increased computational complexity, difficulties in maintaining data consistency and sparsity issues, which can significantly impact the accuracy and effectiveness of recommendations.

Each of the reviewed services uses its own unique approaches, which significantly depend on the scale of the data, the specifics of the platform, and its requirements. Therefore, when implementing recommendation systems, it is important to analyze each approach in detail to choose the most effective and optimal one for a particular system.

The goal is to create a testing environment based on collaborative, context-oriented and hybrid approaches to experimentally select the most effective approach.

Main part. As part of the work, the main methods of recommendation systems are implemented. Each of them has its own peculiarities and is applied to achieve maximum accuracy of personalized recommendations based on the user's history.

1. Collaborative filtering is an information retrieval method that provides recommendations to users based on how other users with similar preferences and behaviors have interacted with particular items. In other words, collaborative filtering algorithms group users based on similarities in their behavior and use the common characteristics of the group to recommend items to other users. Collaborative recommendation systems assume that users with similar preferences have similar interests and tastes [4].

2. The content-based approach in recommendation systems allows personalization of the user's experience by offering content similar to what they have already interacted with. The algorithm analyses content characteristics, such as keywords, topics or genres and recommends

other content with similar characteristics. As a result, user A will receive different recommendations than user B. This method often uses cosine similarity, which estimates the similarity between the user's preference vector and the item profile [5].

3. A hybrid method is an algorithm that combines collaborative filtering and a content-based approach to achieve more accurate performance while minimizing the disadvantages of individual methods. This technology allows to get recommendations that take into account not only the similarities between users or their behavior, but also the characteristics of the items themselves. The hybrid approach allows for more accurate and relevant recommendations, reducing the risk of errors that can occur when using only one method.

To develop an effective recommendation system, it is essential to determine the method that ensures the highest accuracy of recommendations. For this purpose, the Yelp dataset is used, which contains detailed information about various business establishments, including their categories, user ratings, textual reviews and additional attributes. This dataset enabled the modeling of real-world scenarios in which recommendation systems operate, allowing for an in-depth analysis of user preferences based on their past interactions. By examining user ratings and reviews, the system can identify trends and preferences, while incorporating business attributes like categories and pricing to provide more accurate and personalized recommendations.

During the study, three primary recommendation methods are implemented: collaborative, content-based and hybrid. Each of these techniques is tested on the selected dataset to evaluate its accuracy in generating relevant recommendations. Given the diverse nature of the data, the algorithms underwent optimization to enhance their performance and better accommodate the dataset's specific characteristics. This step ensured that the recommendations provided are more precise and aligned with user preferences. The optimization process included fine-tuning model parameters and adapting the methods to handle potential biases and data sparsity issues. Additionally, continuous evaluation and iteration are conducted to identify areas of improvement and refine the methods further. This thorough approach allows for the development of a robust and reliable recommendation system that can effectively meet the needs of its users.

Through a series of experiments, the most effective recommendation method, the one that delivered the highest accuracy in recommendations, is identified. The insights gained from this research can be served as a foundation for further algorithmic improvements and practical applications in real-world recommendation systems. Future enhancements may include the refinement of existing models, the incorporation of additional contextual factors such as time trends and user demographics, or the exploration of deep learning techniques to further improve recommendation accuracy.

Conclusions. The collaborative filtering, content-based approach and hybrid methods for recommendation systems are implemented and tested. The accuracy of the recommendations of each algorithm is evaluated and their strengths and weaknesses are identified by means of a comparative analysis of performance on the selected data set. Based on the tests and optimizations, the most effective recommendation method for this dataset is determined – the hybrid method.

References:

1. Netflix. URL: <https://www.netflix.com/browse/m/trending-now> (access date 28.02.2025).
2. Made For You | Spotify. *Spotify*. URL: <https://open.spotify.com/genre/0JQ5DA0t0bjZptfcdMSK13> (access date 28.02.2025)
3. Amazon.com. URL: <https://www.amazon.com/gp/yourstore/recs> (access date 28.02.2025).
4. Jacob Murel, Eda Kavlakoglu. What is collaborative filtering? *IBM*. URL: <https://www.ibm.com/think/topics/collaborative-filtering> (access date 28.02.2025).
5. Hashim S., Waden J. Content-based filtering algorithm in social media. *Wasit Journal of Computer and Mathematics Science*. 2023. Vol. 2, no. 1. P. 21–26. URL: <https://doi.org/10.31185/wjcm.112> (access date 28.02.2025).

ДОДАТОК Б

Тези II Міжнародної науково-практичної конференції “Progressive
Approaches in Science and Engineering”

УКР.НТУУ “КПІ ім. Ігоря Сікорського”_ІАТЕ_ЦТЕ_ТР-41мп

Аркушів 6

Київ – 2025

isu-conference.com



 **ISU**
INTERNATIONAL SCIENTIFIC UNITY

COLLECTION OF SCIENTIFIC PAPERS



**ISSUE
№47**

**2ND INTERNATIONAL SCIENTIFIC
AND PRACTICAL CONFERENCE**

**PROGRESSIVE
APPROACHES
IN SCIENCE
AND ENGINEERING**

**NOVEMBER 26-28, 2025
COPENHAGEN, DENMARK**



стабільне зниження кількості помилок і значне зростання точності у порівнянні з традиційними одноагентними системами.

Список використаних джерел

1. Ji Z., Lee N., Frieske R. et al. Survey of Hallucination in Natural Language Generation. 2023.
2. Rawte V., Guo W., Shridhar K. Reducing LLM hallucinations using verifier models. 2023.
3. OpenAI. GPT-4 Technical Report. 2023.
4. Anthropic. Claude 3 Model Card. 2024.
5. Google DeepMind. Gemini 1.5 Research Overview. 2024.

DOI 10.70286/ISU-26.11.2025.006

PERSONALIZED RECOMMENDATIONS BASED ON THE PROCESSING OF TEXT DATA AND USER BEHAVIOR PATTERNS

Shapovalova Svitlana

Cand. Eng. Sc., Associate Professor

Hurny Iryna

Postgraduate student

Department of Digital Technologies in Energy

Igor Sikorsky Kyiv Polytechnic Institute, Ukraine

In modern information systems, personalized recommendations play an important role in improving the user experience when interacting with large amounts of digital content. Recommendation models are widely used in various services, which leads to a growing need for adaptive approaches to processing diverse information. This paper presents a comprehensive approach to creating a recommendation system based on the analysis of user interaction history and combining content and collaborative filtering methods. The Yelp Open Dataset, which includes both structured data about businesses and their interactions with users, as well as text reviews, was used to develop the system.

The main goal is to create a model that can generate relevant recommendations based on the user's previous experience. Three main approaches are used to achieve this goal: collaborative, content-based and hybrid filtering.

Collaborative filtering is based on identifying hidden factors in the matrix of interactions between users and businesses, which makes it possible to predict preferences based on analogies with other users [1]. The content approach uses semantic representations of businesses formed from text descriptions, categories, and attributes [2]. The hybrid approach combines content and collaborative models, complementing their strengths [3]. At the same time, its effectiveness directly depends

Progressive Approaches in Science and Engineering

on the quality of each component: if one of the methods is unstable, it can reduce the overall result of the recommendations.

Two key JSON files from the Yelp Open Dataset [4] were used to implement the system:

- `yelp_academic_dataset_business.json` – contains information about businesses, their categories, rating, reviews count, and attributes;
- `yelp_academic_dataset_review.json` – contains text reviews and user ratings.

Data preprocessing consisted of several stages aimed at cleaning and structuring information before creating models. Text reviews underwent a standard processing procedure, which included removing extra characters, converting text to lowercase, normalizing, and lemmatizing to reduce word form diversity. The texts were then presented in TF-IDF format, which allowed the relevance of terms within each document to be preserved. Based on the TF-IDF vectors obtained, thematic modeling was performed using the LDA algorithm. This made it possible to identify hidden themes in the reviews and use them as additional characteristics in the content profile of businesses. Given the large number and variety of categories, the text descriptions of the categories were also processed using LDA, which made it possible to create compact thematic vectors for this group of parameters.

In addition to working with text data, numerical characteristics of businesses were prepared. Due to the different nature and scale of indicators such as the number of reviews and the rating of an establishment, they were normalized separately. Logarithmic scaling was applied to the number of reviews, which reduced the impact of values with high variability. The rating of establishments, being limited to a fixed interval, was normalized using a linear approach. This ensured the correct combination of textual, categorical, and numerical characteristics into a single content profile.

User and business profiles were created separately for content-based and collaborative approaches. In content-oriented filtering, the business profile was represented as a vector created based on text descriptions, categories, and key business characteristics. The user profile was determined through an averaged vector representation of the businesses with which they interacted, reflecting their general preferences. In collaborative filtering, the formation of profiles for both parties was based on latent factors obtained after training the ALS model [5], which allowed the discovery of hidden connection structures in the “user-business” matrix.

To evaluate the effectiveness of the system, three groups of users were formed: active users with a large number of interactions, those whose actions were minimal, and a random sample. This made it possible to analyze changes in the accuracy of recommendations depending on the amount of available data.

During the process of setting up collaborative filtering for the ALS model, key parameters were determined, such as the number of latent factors, the level of regularization, and the number of iterations. The results obtained indicate that the model performs quite competitively in a random sample of users, where the F1-score was 0.254. Interestingly, even in the category of users with low activity levels (least users), the model showed unexpectedly strong results, with an F1-score of 0.156. This demonstrates the algorithm's ability to generate relevant recommendations even with a

Progressive Approaches in Science and Engineering

minimum of available data. However, in the group of the most active users, the metric value dropped significantly (F1-score = 0.063). This result is likely due to the high diversity of their preferences and the absence of clearly defined repetitive behavior patterns.

Content-based filtering showed significantly lower accuracy across all user categories. The TF-IDF and LDA methods for topic modeling did not provide a sufficient level of recommendation relevance: the F1-score for the random group was only 0.011, for the most active users – 0.005, and for the least active – 0.004. The main problem was that the text descriptions and categorical information for many businesses were either incomplete or too general, which prevented the model from forming clear user profiles.

The best results were achieved thanks to a hybrid approach that combines latent factors of ALS with content features. In the random group, the F1-score was 0.262, which is the highest value among all tested models. A similar trend was observed in the least users group, where the hybrid method provided a high level of accuracy with an F1-score of 0.156. At the same time, in the group of the most active users, the results remained quite low (F1-score = 0.071). This indicates the dependence of the hybrid model on the stability of its components: if collaborative filtering demonstrates instability, such limitations are transferred to the hybrid approach.

Accuracy was evaluated using Precision@30, Recall@30, and F1-score@30 metrics. The summary results for all three models are shown in Table 1, which allows for a clear comparison of their effectiveness among different user groups.

Table 1. Comparison of model results by Precision@30, Recall@30, and F1-score@30 metrics in different user groups.

	Method	Precision@30	Recall@30	F1-score@30
random	Hybrid	0.178	0.496	0.262
random	CF	0.169	0.491	0.254
random	CB	0.007	0.022	0.011
top	Hybrid	0.268	0.041	0.071
top	CF	0.234	0.037	0.063
top	CB	0.019	0.003	0.005
least	Hybrid	0.092	0.502	0.156
least	CF	0.092	0.502	0.156
least	CB	0.002	0.014	0.004

To present the system's operation and interact with the user, a separate software application was created that demonstrates the results of experiments and provides access to recommendations. Communication between the external environment and the server side is carried out via REST-API, implemented using FastAPI. This component allows you to transfer user parameters and receive ready-made recommendations.

The file structure of the project is organized according to the sequence of processing steps rather than strictly separated modules. Data preparation, text preprocessing, TF-IDF vectorization and normalization are implemented in separate Jupyter notebooks dedicated to early-stage processing. Topic modeling using LDA is performed in an independent notebook, which allows experimenting with different

Progressive Approaches in Science and Engineering

topic configurations. The generation of recommendations, creation of user and business profiles, and calculation of evaluation metrics are implemented in another notebook responsible for the modeling stage. This workflow-based structure reflects the practical nature of the system, where each notebook corresponds to a distinct processing phase and produces intermediate files that are later used by the backend.

A separate interface for customers provides a convenient way to visualize the system's operation: the user enters the necessary parameters, the application transmits them to the API, and receives the generated recommendations and accompanying metrics on the quality of the results in response.

The diagram of the developed system is shown in Figure 1.

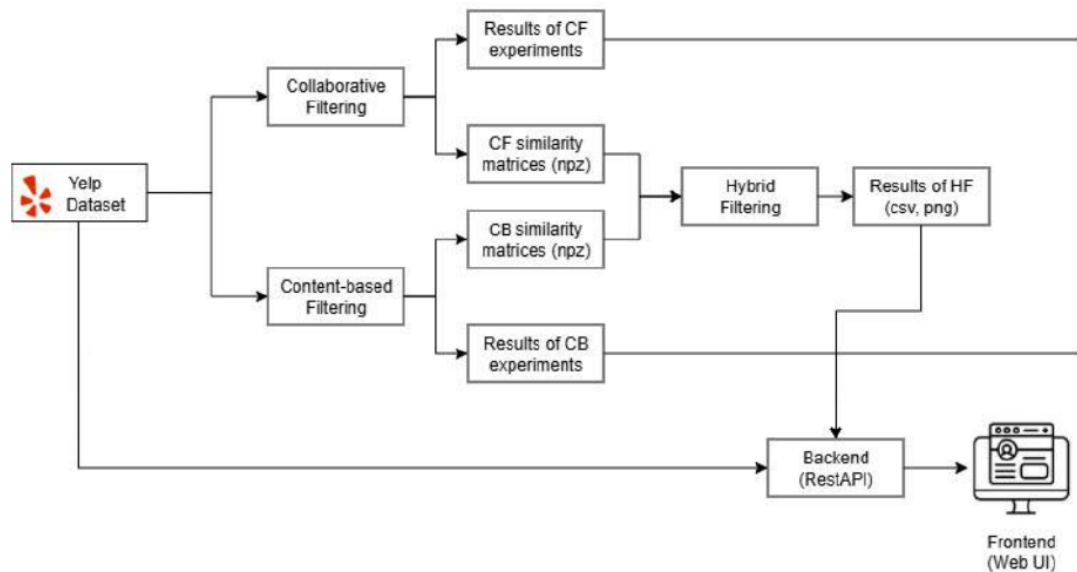


Figure 1. System Architecture

Conclusions: a comprehensive approach to creating a recommendation system based on the analysis of user interaction history and the integration of collaborative and content-based methods is presented. Systematic model tuning and experiments have confirmed the effectiveness of the hybrid approach, which provides high accuracy and stable results under conditions of varying data availability. The proposed system design and implementation clearly demonstrate its practical suitability for processing large volumes of information and diverse data sources.

References

1. VPN Unlimited. (n.d.). What is collaborative filtering? VPN Unlimited. <https://www.vpnunlimited.com/help/cybersecurity/collaborative-filtering>
2. Murel, J., & Kavlakoglu, E. (n.d.). What is content-based filtering? IBM. <https://www.ibm.com/think/topics/content-based-filtering>
3. Milvus. (n.d.). What is hybrid filtering in recommender systems? Milvus. <https://milvus.io/ai-quick-reference/what-is-hybrid-filtering-in-recommender-systems>

4. Yelp Inc. (n.d.). Yelp dataset. Kaggle. <https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset>
5. Gomed, E. (2024, January 24). Alternating least squares: A cornerstone in modern data analysis and recommendation systems. Medium. <https://medium.com/aimonks/alternating-least-squares-a-cornerstone-in-modern-data-analysis-and-recommendation-systems-ceb7310313f9>

ІЄРАРХІЯ КЕШІВ І ПОНЯТТЯ FALSE SHARING

Кіш Віктор

здобувач вищої освіти магістерського рівня
Кафедра системного аналізу і теорії оптимізації
Факультет математики та цифрових технологій
ДВНЗ «УжНУ», Україна

Ієрархія кешів сучасних процесорів є ключовим компонентом оптимізації продуктивності систем, що виконують інтенсивні обчислення та потребують швидкого доступу до оперативних даних [1]. Типова архітектура включає три рівні кеш-пам'яті: L1, L2 та L3. Кеш L1 має мінімальну затримку доступу та найменший об'єм, забезпечуючи обслуговування найбільш частих запитів. L2 є більшим і повільнішим, проте усе ще значно швидшим за оперативну пам'ять. L3 кеш займає проміжне місце в ієрархії, будучи спільним для декількох ядер у межах одного фізичного процесора. Така структура забезпечує компроміс між швидкодією та об'ємом, дозволяючи мінімізувати середній час доступу до даних у більшості сценаріїв [1; 2].

Однією з фундаментальних властивостей кешів є використання фіксованого розміру блоків, так званих *cache lines*, які зазвичай складають 64 байти [1;4]. Коли процесор звертається до певної змінної, у кеш завантажується вся лінія, що містить цю змінну, незалежно від того, чи потрібні інші дані з цієї лінії. Цей механізм ефективний для просторової локальності, однак він створює низку проблем у багатопотокових програмах, зокрема явище *false sharing* [3;4]. Псевдоспільність виникає тоді, коли декілька потоків використовують різні змінні, які розташовані в межах однієї кеш-лінії. Незважаючи на те, що логічно між цими змінними немає зв'язку, апаратний протокол когерентності сприймає будь-який запис у частину цієї лінії як модифікацію всієї лінії [4].

Протоколи когерентності, такі як MESI та його розширення, намагаються гарантувати узгодженість даних між ядрами [1;4]. Коли один потік записує у свою змінну в межах спільної *cache line*, процесор змушений інвалідувати ту саму лінію у кешах інших ядер. Якщо робота виконується на різних сокетів системи, ця операція стає ще дорожчою, оскільки кеш-лінія передається фізично через міжсокетну шину [2]. Таким чином, навіть якщо потоки не працюють із загальними даними, вони вступають у прихований конфлікт за одну й ту саму кеш-лінію, що може призвести до істотної деградації продуктивності [3]. У

ДОДАТОК В

Сертифікат участі в Startup Incubator



Рисунок В1 – Сертифікат участі в Startup Incubator