Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

СИСТЕМА ДОПОМОГИ В ВИБОРІ ВИБІРКОВИХ ДИСЦИПЛІН

Текстова частина до кваліфікаційної випускної роботи

за спеціальністю «Інженерія програмного забезпечення» - 121

Керівник кваліфікаційної роботи,
доцент Афонін А.О.
(прізвище та ініціали)

_____
(підпис)

“_____” _____2023 р.

Виконав студент ІПЗ-4
Ткаченко А.Т.
(прізвище та ініціали)

“_____” _____2023 р.

Київ 2023
Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Доцент

Афонін А.О.

(прізвище та ініціали)

(підпис)

“_____” _____2023 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на кваліфікаційну випускну роботу

студенту Ткаченку Андрію Тарасовичу факультету інформатики 4-гокурсу
ТЕМА Система допомоги в виборі вибіркових дисциплін

Зміст ТЧ до курсової роботи:

Індивідуальне завдання
Календарний план виконання роботи
Анотації
Вступ
1. Огляд сучасних сервісів, що використовують рекомендаційні алгоритми
2. Огляд існуючих алгоритмів для створення рекомендацій
3. Опис практичної частини та аналіз результатів
Висновки
Список літератури
Додатки

Дата видачі “_____” _____2023 р. Афонін А.О. _____
(підпис)

Завдання отримав _____
(підпис)

Тема: Система допомоги в виборі вибіркових дисциплін

Календарний план виконання роботи:

№	Назва етапу	Термін виконання	Примітка
1.	Отримання теми кваліфікаційної роботи	30.09.22	
2.	Аналіз сучасних сервісів створення рекомендацій	31.10.22	
3.	Аналіз популярних методів та підходів створення рекомендацій	05.12.22	
4.	Пошук та ознайомлення з літературою	27.12.22	
5.	Розробка та тестування алгоритму рекомендаційної системи	01.02.23	
6.	Програмна реалізація алгоритму	10.03.23	
7.	Створення серверної частини прототипу додатку	31.03.23	
8.	Створення інтерфейсу прототипу додатку	15.04.23	
9.	Створення презентації	30.04.23	
11.	Перегляд змісту роботи керівником та попередній захист	10.05.23	
12.	Внесення змін до роботи відповідно до зауважень наукового керівника	20.05.23	
13.	Захист кваліфікаційної роботи	01.06.23	

Студент Ткаченко А.Т.
Керівник Афонін А.О.
“_____”_____ 2023 р

Зміст

Анотація

В роботі представлено алгоритм для рекомендацій вибіркових дисциплін, розроблений з використанням колаборативного фільтрування та було реалізовано прототип веб-додатку, який застосовує цей алгоритм. Також у роботі проведено аналіз сучасних рекомендаційних систем, які використовують у популярних продуктах, а також розглянуті існуючі методи створення рекомендацій за допомогою колаборативного фільтрування та фільтрування за змістом. Деталі алгоритму та реалізації прототипу додатку наведені у відповідних розділах роботи.

Ключові слова: системи рекомендацій, рекомендації, Коефіцієнт кореляції Пірсона, колаборативна фільтрація, веб-додаток, Java, Spring Boot, ReactJS.

Вступ

Кожного року до «Києво-Могилянської академії» вступають сотні абітурієнтів, водночас близько 3 500 студентів вже навчаються у самому університеті [1]. Майже усі вони щороку стикаються зі спільною проблемою: вибором вибіркових та професійно-орієнтовних дисциплін на наступний навчальний період. Вирішення цього питання є актуальним для кожного студента, адже від нього залежить вибір обов’язкових та вибіркових дисциплін, зокрема загального та професіонального спрямування, можливість навчатися у обраного викладача, обсяг та баланс навчального навантаження за короткий та тривалий періоди, який зрештою впливає на успішне засвоєння знань. Студенти приймають рішення, аналізуючи численні спільні чинники, зважаючи на індивідуальні критерії, збираючи відгуки від попередніх курсів, аналізуючи досвід інших та порівнюючи з власними потребами та інтересами навчання [2].

Метою цієї роботи є розробка алгоритму системи рекомендації навчальних дисциплін для студентів, які визначають набір вибіркових дисциплін на певний навчальний період, задля спрощення процесу прийняття такого рішення та полегшення вибору з урахуванням особистих критеріїв і характеристик, а також зважаючи на вибір інших студентів. Такий алгоритм допоможе студентам обирати набір дисциплін, які задовольняють їхні потреби та запобігти проблемам невдалого вибору.

Об'єктом дослідження є вивчення наявних підходів до створення рекомендацій, аналіз найбільш популярних систем та алгоритмів, а також їхнє практичне застосування шляхом створення алгоритму рекомендацій, оптимального для вирішення встановленого завдання.

Робота складається із трьох розділів. У першому розділі проаналізовано найбільш популярні рекомендаційні системи, засновані на них сервіси, розглянуто їхні переваги та досліджена специфіка їхньої адаптації до використання у визначеній предметній області.

У другому розділі розглянуто базові принципи, алгоритми та типові підходи до створення системи рекомендацій та їхні переваги, недоліки і особливості застосування, а також описано ключові інструменти, використані при реалізації власної рекомендаційної системи.

У третьому розділі наведено опис та проаналізовано результати розробленої рекомендаційної системи в цілому, та деталі формування і взаємодії алгоритмів, обґрунтовано обраний підхід та послідовність кроків його створення. Також у розділі пояснені етапи інтеграції моделі до додатку-прототипу, описано створену програму та застосовані для цього технології.

Джерелами дослідження були інтернет ресурси про розробку сучасних стандартних алгоритмів рекомендаційних систем та про діючі системи рекомендацій популярних продуктів.

Теоретичний аналіз, проведений у роботі, мав метою порівняння та узагальнення підходів та методів створення подібних сучасних рекомендаційних систем. На базі цього аналізу розроблено власний алгоритм та запропоновано їхню практичну реалізацію у вигляді додатку-прототипу з добору рекомендацій - дисциплін для студентів НаУКМА. Розроблена рекомендаційна система може використовуватися у подальшому для наступних досліджень та слугувати основою для вдосконалення методик та подальших розробок у цій сфері.

Практичне значення розробленої рекомендаційної системи полягає в тому, що запропонований додаток-прототип, та алгоритми, покладені у його основу, можуть допомагати студентам НаУКМА оптимізувати та спростити процес прийняття рішення про обрання вибіркових дисциплін. Крім того, враховані при розробці реалії університетської електронної системи [3] забезпечили алгоритмам додатку адаптивні можливості для його подальшої інтеграції до існуючих систем НаУКМА.

РОЗДІЛ 1: Огляд сучасних сервісів, що використовують рекомендаційні алгоритми

Сучасні рекомендаційні алгоритми є основою багатьох та різноманітних сервісів, метою яких є високоефективна взаємодія з користувачами, забезпечення їхніх потреб, відповідно до їхніх індивідуальних вимог та особистих критеріїв вибору. Серед найвідоміших сервісів, які активно використовують рекомендаційні алгоритми, принципи та підходи проаналізовано у цій роботі, є Amazon, Netflix та Spotify. Попри те, що реалізація алгоритмів добору комбінацій, а також особливості та деталі їхнього функціоналу, є прихованими та захищеними комерційною таємницею, розробники діляться певними аспектами їхньої діяльності, що дозволяє робити висновки щодо принципів роботи алгоритмів [4, 5].

1.1. Spotify

1.1.1. Вплив рекомендаційної системи на бізнес модель Spotify

Рекомендаційна система є ключовим елементом бізнес-моделі компанії Spotify [6]. Саме вона допомагає привернути увагу користувачів та утримувати їх на платформі. Так, Spotify використовує свою рекомендаційну систему і для персоналізації запропонованої користувачам музики, і для покращення якості плейлістів, згенерованих автоматично. Ця система прямо впливає на лояльність слухачів до музичної платформи, покращуючи їх досвід успішного користування, та, відповідно, утримує на сервісі, збільшуючи кількість годин користування, і водночас мінімізує ймовірність відмови від підписки чи переходу до іншого сервісу.

1.1.2. Підхід до створення рекомендацій Spotify

Система рекомендацій Spotify формує пропозицію - «потік» схожих мелодій, які можуть сподобатися користувачу на підставі його попереднього вибору [7]. Для цього система використовує різні методи, які допомагають проаналізувати кожен вподобаний користувачем музичний трек, розібрати його на маленькі частини, визначаючи, що саме привернуло увагу користувача. Отже, алгоритм аналізу системи можна умовно поділити на три частини. Перша - відповідає за порівняння музичного змісту треків, щоб виокремити «особливості» або «типовість» кожного з них. Друга - відповідає за аналіз мелодій, тобто за пошук зв'язків та мелодичний аналогій між вподобаними треками. Третя – відповідає за збір зворотного зв'язку та аналіз реакції користувача.

Аналіз мелодій системою рекомендацій Spotify

Другу умовну частину (аналіз мелодій) алгоритму Spotify можна теж, у свою чергу, умовно поділити на три компоненти: (1) аналіз даних від автора мелодії, (2) аналіз аудіо файлу, (3) аналіз змісту та опису мелодії із застосуванням алгоритмів обробки природньої мови [8].
Завдання першого компоненту - дослідження даних, наданих автором треку назви, авторів, виконавців, країни походження, дати виходу, жанру, музичних інструментів, мови, тегів, які описують стиль та настрій композиції тощо. Другий компонент визначає «особливості» самого аудіофайлу. Наразі відомо про 12 критеріїв [7] за якими алгоритм оцінює мелодію: співвідношення вокалу до інструментальної частини, тривалість треку, середня гучність, темп тощо [6]. На підставі цих даних алгоритм Spotify генерує інші «особливості»

треку, лише про три з яких відомо публічно: танцювальність, енергія та валентність.

- Танцювальність – придатність треку для танців, наприклад, за критеріями темпу, стабільності, сили ритму тощо.
- Енергія – наскільки трек є активним - за критеріями динаміки, гучності та тембру.
- Валентність – наскільки мелодія є «позитивною», або «емоція» треку за шкалою, де високий показник - щаслива та ейфорійна мелодія, а низький - сумна та пригнічена.

Після цього алгоритм системи Spotify «нарізає» мелодію на частини, оцінюючи її часову структуру з різним рівнем деталізації на різних частинах (див. приклад візуалізації аналізу наведено на Рисунку 1.1 [9]). Ці частини визначаються зрушеннями в тембрі та ритмі, за якими аналізуються переходи між частинами пісні: куплет, приспів, соло тощо.

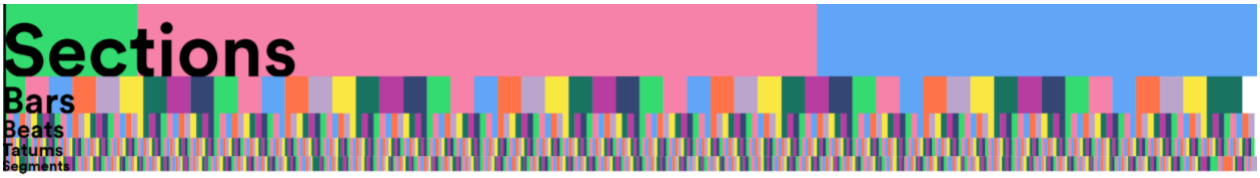

Рисунок. 1.1 Приклад аналізу часової структури мелодії системою Spotify Audio Analysis для треку «Пес Патрон», групи «Карта світу»

У третьому компоненті аналізу мелодії, алгоритм визначає «особливості» треку з іншої точки зору, застосовуючи алгоритми обробки природної мови [10]. При цьому, алгоритм аналізує не лише сам текст пісні. Він допомагає віднайти в інтернеті статті зі згадками обраного треку та з його описом від інших слухачів. Алгоритм також порівнює склад назв та опис

інших плейлістів (наборів рекомендацій), визначаючи категорії, до яких трек віднесли інші слухачі. Наприклад, трек, який інші користувачі часто додавали до плейлістів, у назвах яких зустрічалося слово «сум», алгоритм віднесе до категорії «сумний».

У такий спосіб три складові алгоритму музичного аналізу рекомендаційної системи Spotify формують детальний опис мелодії, комбінують дані з різних джерел для оцінки його «особливостей», типізують його та обирають користувачів, яким його слід порекомендувати.

Колаборативне фільтрування

Spotify, разом із Netflix, були одними із перших компаній стрімінгових сервісів, які застосували алгоритми та підходи колаборативного фільтрування для аналізу вподобань їхніх користувачів, щоб запропонувати їм найкращі рекомендації [4].

Слід зазначити, що на практиці добір музичних рекомендацій (Spotify) суттєво відрізняється від добору рекомендацій для фільмів (Netflix). А пошук схожих слухачів за матрицею, яка описує всіх користувачів та весь каталог музики, займає суттєві програмні ресурси та погано масштабується [11].

Отже, Spotify відійшов від класичного використання користувацько-предметної матриці і ділить музику у інший спосіб. Його колаборативне фільтрування сфокусоване на схожості треків і реалізується різними способами [12]. Найбільш примітивний підхід – «схожа музика зібрана в одному плейлісті». Складніші підходи аналізують час прослуховування користувачем певного набору мелодій за один раз.

Завдяки такому детальному фільтруванню алгоритм Spotify визначає, яку музику та коли слід рекомендувати користувачам, а також формує зв’язки між користувачами та вподобаннями.

Генерація профілю смаків користувача Spotify

Головна мета цього етапу алгоритму Spotify - розуміти смаки користувача [6]. Для цього система збирає дані та відгуки про взаємодію слухачів та музики, а також визначає мету, з якою користувач увімкнув музику.

Так, наприклад, алгоритм розуміє, що коли користувач переглядає автоматично створений плейліст «Що нового?», його наміром є швидке ознайомлення з новими треками. Відповідно, алгоритм визначає, що швидкий перехід користувача від одного треку до іншого не завжди означає, що користувач не вподобав ці мелодії. Водночас, якщо користувач швидко «гортає» певні треки із плейліста «Глибокий фокус», то алгоритм робить висновок, що, скоріше за все, ці мелодії не підходять до цієї категорії, а швидкий перехід користувача є сигналом його незадоволення, і, відповідно, наступного разу цю мелодію краще прибрати з рекомендацій.

Отже, алгоритм Spotify сприймає кожен контакт користувача з музикою як сигнал про вподобання або не вподобання. Такі взаємодії алгоритм поділяє на дві категорії: активні та пасивні [13].

- Активна взаємодія з треком передбачає конкретні дії користувача, як натяк-інструкцію алгоритмові про свої вподобання. Наприклад, додавання до плейлісту «Вподоабання» або блокування мелодії.

- Пасивна взаємодія, про яку користувач часто може не здогадуватися, передбачає аналіз поведінки слухача. Наприклад: тривалість прослухування мелодії, кількість її повторень, частота повернень до неї користувачем.

Оцінюючи ці види взаємодії, алгоритм не надає їм рівної ваги. При цьому система надає перевагу активним взаємодіям, які обробляються та відображаються у профілі користувача. Наприклад, оскільки користувач може слухати музику фоном, тривалість прослуховування не може бути завжди надійним показником міри вподобання треку. (див. Рисунок 1.2 [13])

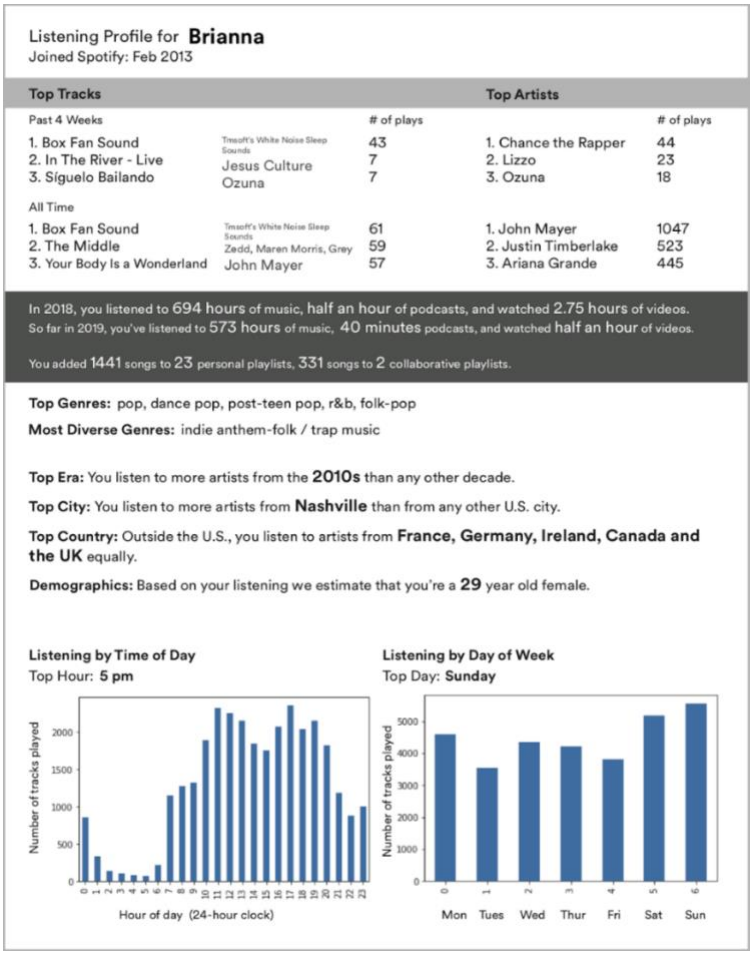

Рисунок. 1.2. Приклад профілю користувача Spotify

Таким чином алгоритм рекомендаційної системи Spotify формує профіль користувача за кількома ключовими категоріями та багатьма детальними підкатегоріями.

Далі, у результаті нової взаємодії користувача з музикою, алгоритм безперервно деталізує, уточнює та змінює цей профіль. Відповідно, показники нещодавньої діяльності отримують більшої ваги, ніж загальні показники історії прослуховування. Так, наприклад, якщо користувач відкриває новий жанр і «позитивно» взаємодіє із алгоритмом , то система рекомендацій намагатиметься пропонувати більше музики в такому жанрі, навіть якщо це дещо протирічить даним історії прослуховування.

1.2. Amazon

1.2.1. Вплив рекомендаційної системи на бізнес модель Amazon

Рекомендаційна система, яку використовує Amazon є невід'ємною складовою бізнес-моделі компанії, формуючи її продажі та доходи [14]. Алгоритм добору рекомендацій спрощує покупцям пошук, допомагає швидше приймати рішення про товари та підтримує лояльність клієнтів, збільшуючи обсяги продажів. Так, нещодавні дослідження [15] свідчать, що 91% клієнтів веб-магазинів готові надати перевагу веб-магазинам, які персоналізують пропозиції та надають рекомендації, допомагаючи з вибором, ніж тим, у яких такий сервіс відсутній. Так само, 98% власників веб-магазинів стверджують, що алгоритми персоналізації покращують їхні стосунки із клієнтами та бізнес вцілому. За даними Amazon, уже з 2013 компанія завдячувала кожній третій своїй продажі власній системі рекомендацій, причому кожен другий з таких покупців ставав її постійним клієнтом [14].

1.2.2. Підхід до створення алгоритму рекомендацій Amazon

Для створення рекомендацій Amazon використовує свою систему під назвою Amazon Personalize, яка працює майже із початку створення компанії і пройшла багато етапів еволюції [16].

Ранні версії системи рекомендацій Amazon ґрунтувалася на алгоритмі колаборитвної фільтрації «Предмет-Предмет», зумовленим великим асортиментом товарів та низьким показником «перетину» між покупками користувачів. Тоді системі було складно знайти клієнтів із схожими історіями покупок та сформувати з них рекомендації новим користувачам. Щоб вирішити це питання Amazon створив алгоритм рекомендацій користувачам на базі схожості товарів із історії покупок та асортименту загального каталогу. Щоб визначити схожі риси у товарів зі загального асортименту та товарів, придбаних користувачами, будувалася матриця схожості товарів, зокрема за критеріями схожості з товарами, придбаними клієнтами разом [18].

Принцип дії алгоритму створення рекомендацій цієї матриці працює наступним чином:

(1) для кожного товару з каталогу аналізуються закупи покупців, які його придбали – «супутні» товари;
(2) інформація про всі «супутні» товари, придбані разом з цим товаром, зберігається для створення відповідних векторів
(3) за допомогою цих векторів алгоритм визначає схожість між певним товаром та рештою «супутніх» товарів каталогу
(4) отримані у такий спосіб показники подібності формують таблицю «супутніх» товарів, подібних до певного товару

Сучасна версія системи рекомендацій Amazon [17] є набагато складнішою, гнучкою та адаптивною. Вона ґрунтується на не лише на численних алгоритмах створення рекомендацій, а і активно використовує машинне навчання. Така система рекомендацій Amazon може бути інтегрованою у різних варіантах до безлічі форматів електронної комерції, навіть надається як окрема послуга користувачам AWS задля створення власних систем рекомендацій.

1.3. Netflix

1.3.1. Вплив рекомендаційної системи на бізнес модель Netflix

Рекомендаційна система Netflix є основою бізнесу компанії. За даними Netflix 80% часу перегляду її користувачів завдячують власне створеним рекомендаціям [19]. Створюючи глядачеві безперервний потік актуальних та точних рекомендацій, Netflix істотно спрощує своїм користувачам процес вибору фільму для перегляду. Стабільно якісні рекомендації Netflix стимулюють глядачів постійно звертатися до платформи, купуючи нову підписку чи продовжуючи її на наступні періоди та поширюючи на додаткові послуги. Отже ретельно продумана рекомендаційна система Netflix, яка забезпечує клієнтам компанії найкращий користувацький досвід, є запорукою утримання постійних клієнтів, що стабілізує рівень доходів та допомагає прогнозувати прибутки, а також економить на витратах із залучення нових клієнтів. Це, у свою чергу, формує міцний фундамент бізнесу вцілому та надає перспективи розвитку компанії, зокрема через інвестиції до цього ключового елементу бізнес-моделі [20].

1.3.2. Підхід до створення рекомендацій Netflix

Як і у Spotify та Amazon, рекомендаційна система Netflix базується на взаємодії численних алгоритмів. Проте, на відміну від інших сервісів, система Netflix сама обирає які алгоритми є найбільш ефективними у прогнозі та впливі на вибір користувача [21].

Результати роботи різних алгоритмів створення рекомендацій та різні категорії контенту, підібрані системою за різними характеристиками можна побачити на «Домашній сторінці» (Homepage) користувача, згенерованій системою. Сторінка має вигляд горизонтальних «рядків» з контентом (фільмами та серіалами), відсортованими у порядку зниження вірогідності їхнього перегляду користувачем. Водночас самі рядки з контентом на екрані відсортовані вертикально за ефективністю їхнього впливу на вибір користувача (див. Рисунок. 1.3. [22])

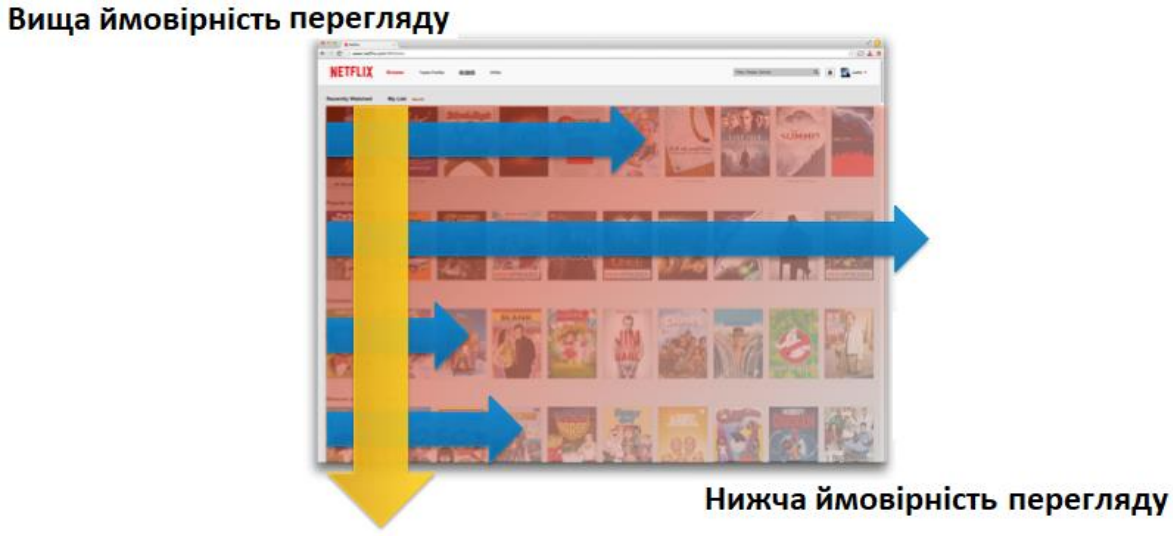

Рисунок 1.3 Приклад візуалізації відсортованих рекомендацій алгоритму Netflix за ймовірністю їхнього перегляду користувачем

Створення рядків перегляду алгоритмом Netflix

Для створення рядків «Домашньої сторінки» застосовуються різні алгоритми Netflix, серед яких виділяють чотири головних:

1. Персоналізований рейтинг відео (PVR) – алгоритм призначений для знаходження категорій контенту за зв'язками «особливостей» цих категорій та «особливостей» користувача.
2. Top-N рейтинг відео – подібний алгоритм, який не фільтрує фільми за категоріями, а просто винаходить контент, який може сподобатися користувачу.
3. Рейтинг популярних тенденцій – алгоритм, що визначає короткострокові тенденції в переглядах фільмів, орієнтуючись на дати. Наприклад на Різдво, люди частіше переглядають різдвяні комедії, а на День Святого Валентина романтичні фільми.
4. Рейтинг «Продовжити перегляд» - алгоритм, який пропонує фільми та телешоу, які користувач починав дивитися, але не закінчив перегляд [23].

Особливості генерації «Домашньої сторінки» Netflix

Для розробки ефективної рекомендації просте використання різних алгоритмів не є достатнім. Важливою є також правильна комбінація результатів, яка дозволяє запропонувати користувачеві не просто набір продуманих рекомендацій але й забезпечити його різноманітність.

Щоб вирішити проблему різноманітності вибору рекомендаційна система Netflix широко застосовує машинне навчання, зокрема для функції оцінки рядків. Нейронна модель навчається на основі інформації про наявні

«домашні сторінки» з рекомендаціями та про конкретну взаємодію користувачів з ними, зокрема йдеться про перегляди, «гортання», конкретний дієвий інтерес у вигляду відкриття запропонованого контенту, тривалість перегляду тощо. На Рисунку 1.4 наведено приклад «домашніх сторінок» з процесу навчання рекомендаційної системи. На кожному з варіантів показник ліворуч означає перегляд користувачем запропонованої рекомендації: відповідно 1 перегляд із 3 запропонованих у першому варіанті, 2 із 3 у другому (з яких один – повтор з першого варіанту) і 3 з 3 відповідно – у третьому (з яких 2 повтори з попередніх варіантів. [23])

Рисунок. 1.4. Варіанти рекомендацій на «домашній сторінці» користувача, згенеровані алгоритмами системи Netflix

Таким чином, розробка алгоритмів рекомендацій, їхня комбінація та взаємодія у поєднанні з безперервним вдосконаленням процесу на базі машинного навчання систем, дозволяє Netflix, Amazon та Spotify, а також та іншим провідним компаніям, які працюють у цій сфері, забезпечити високий показник персоналізації контенту та ефективних рекомендацій та для своїх користувачів.

РОЗДІЛ 2: Огляд існуючих алгоритмів для створення рекомендацій

2.1. Колаборативні методи

Колаборативні методи, які застосовуються багатьма рекомендаційними системами, засновані на використанні існуючих взаємодій між користувачами та предметами рекомендацій. Ці взаємодії зазвичай зберігаються у формі матриці, яка отримала відповідну назву «предметно-користувацька матриця взаємодій».

Особливістю методу є необхідність достатньої історії взаємодій між користувачами та предметами рекомендацій, щоб забезпечити ефективність, точність його прогнозів та дієвість рекомендацій. Зазвичай такі рекомендаційні системи поділяють на дві категорії: засновані на пам’яті та засновані на моделі [24, 25].

2.1.1. Колаборативна фільтрація на основі пам’яті

Методи колаборативної фільтрації, засновані на пам’яті зазвичай націлені на простий пошук схожості («сусідства») [24]. Для розробки рекомендацій чи прогнозів застосовують послідовне обчислення подібності елементів. Такі процеси не передбачають побудову статистичних чи інших моделей, а для визначення подібностей між показниками взаємодій користувачів з предметами зазвичай використовують коефіцієнт кореляції Пірсона, який описує схожість взаємодій користувачів від усередненого показника взаємодій для всіх користувачів предмета [33]. (див. 2.4. Опис інструментів та методів, використаних у роботі)

Способи застосування алгоритму пошуку «сусідів» для рекомендаційних систем можна умовно розділити на дві основні групи «користувач-користувач» та «предмет-предмет» (див. Рисунки А.1 і А.2):

- **Користувач-користувач**. Цей підхід допомагає визначити користувачів («сусідів»), найбільш схожих на користувача, для якого розробляється рекомендація, на базі їхнього досвіду взаємодії із предметами. Результатом стає рекомендація саме предметів, з якими позитивно взаємодіяли «найближчі сусіди» користувача, але з якими сам користувач ще не мав досвіду взаємодії.

- **Предмет-предмет**. На відміну від попереднього підходу (який пошукає «сусідів» - схожих користувачів), цей підхід визначає предмети, найбільш подібні до тих, з якими користувачі вже взаємодіяли. Відповідно, результатом стає рекомендація предметів, подібних до тих, які він вже обирав [24, 26].

Перевагами методу колаборативної фільтрації на основі пам’яті є простота розробки алгоритму, відносно високі. Водночас, проблемою запровадження методу часто стає брак даних на старті («холодний старт»): коли нові користувачі ще не контактували з предметами або коли показників таких взаємодій системі не достатньо для обрахунків, щоб сформулювати дієву рекомендацію чи достовірний прогноз. На практиці, для вирішення цієї проблеми застосовують наступні стратегії:

- Random strategy (стратегія випадковостей). Алгоритм рекомендує випадкові предмети поки триває збір достатнього обсягу даних для формування обґрунтованих рекомендацій

- Maximum expectation strategy (стратегія максимальних очікувань). Алгоритм рекомендує новим користувачам найбільш популярні предмети, або навпаки, пропонує нові предмети найактивнішим користувачам.

- Exploratory strategy (стратегія екстраполяції). Алгоритм рекомендує конкретні набори предметів новим користувачам.
- Стратегія використання неколаборативних методів для формування рекомендацій на ранньому етапі взаємодії з користувачем або предметом [27].

2.1.2. Колаборативна фільтрація на основі моделі

Такий метод створює рекомендації з використанням статистичних моделей, націлених на пошук зв'язків та закономірностей та розроблених на базі алгоритмів машинного навчання.

Цей метод дає найбільш точні результати, оскільки допомагає визначати приховані чинники, закономірності взаємодій, які стають основою для рекомендацій або використовуються для спостережень за користувачами та оцінок предметів. Також, цей метод ефективніше (ніж базовий пошук «сусідів») працює з базами неповних, незакінчених чи розріджених даних.

Недоліком колаборативного методу, заснованого моделі, є складність, тривалий час та трудомісткість розробки, і, як наслідок, висока вартість розробки. Зокрема, реалізація алгоритму потребує машинного навчання та тестування на наборі «навчальних» даних, який може бути недоступним, особливо на початковому етапі створення рекомендаційної системи [25, 28].

2.2. Методи фільтрації на основі вмісту

Відмінність методів фільтрації на основі вмісту (від попередньо розглянутих колаборативних методів) полягає у тому, що для розробки рекомендацій використовують не лише аналіз історії взаємодій між

користувачами та предметами, але й додаткової інформації про користувачів та предмети [25].

Йдеться про метод аналізу, заснованого на визначенні певних «особливостей», як у користувачів, так і у предметів рекомендацій. Зазвичай «особливостями» вважають набори певних характеристик та атрибутів, за якими рекомендаційна система класифікує елементи. Потім, за допомогою моделі досліджують зв’язки, причини, чинники та закономірності та інші форми впливу визначених «особливостей» на взаємодію між користувачами та предметами рекомендацій.

Методи фільтрації, засновані на вмісті, набагато менше залежать від проблем «холодного старту», ніж колаборативні методи. Зокрема, система не потребує історії взаємодій користувачів з предметами рекомендацій, окрім, наприклад початкових ситуацій з користувачами, які мають унікальні «особливості», що їх модель неспроможна обрахувати на старті, проте із накопиченням даних, ефективність методу буде покращуватися [24, 30].

2.3. Гібридні методи

На практиці різні методи фільтрації успішно комбінують, щоб нівелювати їхні недоліки та забезпечити рекомендаційній системі точність, дієвість та результативність. Виділяють сім найбільш ефективних та популярних способів комбінації методів:

1. **Зважений** (Weighted). Додавання, поєднання усіх результатів та оцінок від різних методів шляхом зважування.

2. **Перемикання** (Switching). Динамічне обрання методу рекомендацій залежно від умов шляхом запиту (використання чутливого критерію для

перемикання між методами: якщо один не дає результатів, відбувається перехід до іншого).

3. **Змішаний** (Mixed). Адаптація результатів (створення набору даних на базі результатів від інших методів) задля об’єднаного результату.

4. **Комбінація особливостей** (Feature combination). Різні особливості збираються із різних джерел для використання у рекомендаційному алгоритмі (додана віртуальна рекомендаційна модель, яка збирає такі особливості, наприклад поєднання контенту та результатів спільної роботи).

5. **Доповнення особливостей** (Feature augmentation). Розширення набору особливостей завдяки допоміжному рекомендаційному алгоритму (наприклад, рейтиингу) та використання його для наступної обробки і отримання кінцевого результату у наступному методі

6. **Каскад** (Cascade). Використовують декілька рекомендаційним систем, кожній із яких присвоюється певний пріоритет (ієрархічна структура, коли, наприклад, одна модель дає первинний результат, який обробляється вторинною моделлю).

7. **Мета-рівень** (Meta-level). Поєднання двох рекомендаційних систем, коли вихідні дані однієї стають вхідними даними іншої. Від гібридної системи доповнення особливостей відрізняється тим, що вхідні дані генерує вся модель (не лише перший алгоритм). [31, 32]

2.4. Ключові інструменти, застосовані у роботі

2.4.1. Коефіцієнт Пірсона

Коефіцієнт кореляції Пірсона (ККП) – це статистичний показник, який використовується для вимірювання ступеня лінійної залежності між вдома змінними і позначається як r [33]. Коефіцієнт кореляції Пірсона (для вибірки) обчислюється за формулою

$$r_{xy} = \frac{\sum_{i=1}^{m}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{m}(x_i-\bar{x})^2 \sum_{i=1}^{m}(y_i-\bar{y})^2}} \qquad (2.1)$$

Де $\boldsymbol{r_{xy}}$ - значення ККП

x та **y** – зміні, між якими треба визначити ККП

$\bar{\boldsymbol{x}}$ та $\bar{\boldsymbol{y}}$ - вибіркові середні змінних між якими визначають ККП

ККП описує кореляції між набором змінних у вибірці, значення яких ранжується від -1 до +1, демонструючи наскільки ці змінні залежать одна від одної та які напрямок, межі та ступінь має ця залежність. Тобто, значення ККП більше 0 свідчать про ступінь кореляції (позитивної кореляції), а значення менше 0 – про ступінь негативної кореляції (див. приклад трактування у Таблиці 2.1) [34].

Таблиця 2.1. Приклад: трактування меж показників ККП, ступеня і виду кореляції

Значення ККП	**Ступінь кореляції**	**Вид кореляції**
[0.9, 1.0]	Дуже сильна кореляція	позитивна
[0.7, 0.9]	Сильна кореляція	позитивна
[0.5, 0.7]	Середня кореляція	позитивна
[0.3, 0.5]	Слабка кореляція	позитивна

Значення ККП	Ступінь кореляції	Вид кореляції
[0, 0.3]	Дуже слабка кореляція	позитивна/відсутня
[-0.3, 0]	Дуже слабка кореляція	негативна/ відсутня
[-0.5, -0.3]	Слабка кореляція	негативна
[-0.7, -0.5]	Середня кореляція	негативна
[-0.9, -0.7]	Сильна кореляція	негативна
[-1, -0.9]	Дуже сильна кореляція	негативна

На практиці ККП часто застосовують в алгоритмах колаборативної фільтрації на основі пам’яті для пошуку та оцінки подібності взаємодій користувачів відносно усередненого показника.

У рамках цієї роботи ККП було використано для рекомендаційної системи в алгоритмі пошуку «сусідів» –найбільш схожих за характеристиками користувачів (студентів). Зокрема, ККП було застосовано для оцінки схожості шляхом аналізу оцінок характеристик студента, якому надається рекомендація, з усередненим показником оцінки таких характеристик для інших студентів певної групи (див. деталі практичної реалізації у підрозділі 3.1.3. Ранжування студентів за характеристиками)

2.4.2. Min-max нормалізація значень

Min-max нормалізація (або min-max масштабування) – статистичний метод для масштабування набору даних, який перетворює значення змінних таким чином, щоб вони потрапляли у заданий діапазон значень (зазвичай від 0 до 1 включно). [35] Цей процес полягає у перетворенні кожного значення змінних x у значення $\boldsymbol{x_{i,norm}}$ за формулою

$$x_{i,norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \qquad (2.2)$$

Де $\boldsymbol{x_{i,norm}}$ – нормалізоване значення елементу, $\boldsymbol{x_i}$ – ненормалізоване значення елементу, $\boldsymbol{x_{min}}$ – найменший елемент, $\boldsymbol{x_{max}}$ – найбільший елемент.

За результатом даної формули значення елементів зміної x будуть знаходитися у діапазоні від 0 до 1, причому мінімальне значення змінної буде відповідати 0, а максимальне 1. Такий метод є доволі простим та ефективним і широко використовується для масштабування даних у визначеному діапазоні, дозволяючи забезпечити більш точну роботу машинного навчання.

У цій роботі метод min-max нормалізації застосовано до рекомендаційної системи для (1) масштабування значень перетинів навчальних дисциплін між двома студентами, а також (2) нормалізації значень характеристик студентів.

Метою першого застосування методу min-max нормалізації було збільшення вибірки студентів, які мають набір обраних дисциплін, який є найбільш схожим, з рекомендованим набором для студента, який потребує рекомендації.

Метою другого застосування була нормалізація значень характеристик студентів до єдиного діапазону. Це було необхідно, оскільки кожен студент по-різному оцінює свої характеристики (здібності, потреби у навчанні), відповідно його оцінка власних характеристик буде суб’єктивною, а значення оцінок можуть суттєво варіюватись. Наприклад, одні студенти оцінюють своїх характеристики від 4 до 8, а оцінки інших можуть бути «розкиданими» на максимумах шкали (від 1 до 10). Водночас, для ефективне порівняння потребувало приведення значень до єдиного діапазону, отже застосування min-max нормалізації було необхідним.

РОЗДІЛ 3: Опис практичної частини та аналіз результатів

3.1. Модель рекомендаційної системи

Головним завданням практичної частини цієї роботи була розробка моделі рекомендаційної системи для студентів певного факультету НаУКМА (наприклад, факультету інформаційних технологій), яка б допомагала їм спростити процес обрання вибіркових дисциплін. Ідея полягала у тому, щоб завдяки рекомендаційній системі студенти отримували рекомендований перелік вибіркових дисциплін, проранжований за їхніми побажаннями та з урахуванням попереднього вибору дисциплін іншими студентами. Реалізувати рекомендаційну систему передбачалося у вигляду прототипу додатку, спроможного швидко та ефективно опрацьовувати масиви даних щодо значної кількості користувачів (студентів) і предметів (вибіркових дисциплін). Також, очікувалося, що прототип додатку матиме технічні можливості для адаптації до можливої інтеграції з загально університетською системою.

Для побудови моделі даної рекомендаційної системи було обрано гібридний метод колаборативної фільтрації, оскільки від алгоритмів аналізу потребувалися (1) аналіз поведінки користувачів (пошук студентів, які обрали ті самі дисципліни), а також (2) визначення та сортування тих, які мають подібні характеристики. Відповідно, головним підходом до створення рекомендаційної моделі стало поєднання алгоритмів пошуку «схожості» поведінки (студентів, які обирали подібні набори дисциплін) та ранжування за їхніми характеристиками (на базі оцінок студентів їхніх побажань щодо навчання).

У рекомендаційній моделі дані про студентів збиралися за результатами їхнього анкетування при реєстрації, звідки отримувалася інформація про характеристики студентів, зокрема їхня оцінка власних побажань щодо навчання за визначеними характеристиками по шкалі від 0 до 10. Модель також передбачала наявність переліку вибіркових дисциплін, які мали стати предметом аналізу та рекомендацій. Очікуваним результатом роботи моделі мав стати перелік рекомендацій (вибіркових дисциплін) для зареєстрованого користувача (будь-якого конкретного студента із бази), відфільтрований на підставі історії взаємодій інших користувачів (вибору інших студентів), проранжований за ознакою його релевантності характеристик схожих студентів.

У реалізованій рекомендаційній моделі студент отримував роль користувача, а вибіркові дисципліни (на які він «записався» або міг «записатися») - роль предметів (див. детальніше розділ 3.2. Аналіз технічного завдання). Першим етапом була ідентифікація групи студентів, які мали схожий набір дисциплін. Наступним – ранжування дисциплін на базі характеристик цих студентів.

Тобто, кожна дисципліна отримувала ранг, який дорівнював сумі значень ККП характеристик всіх студентів, які зареєструвалися на цю дисципліну та мають схожий набір обраних дисциплін. Відповідно, чим більше схожих за характеристиками студентів записалося на певну дисципліну, тим вищу позицію вона посідала в рекомендаційному переліку. Дисципліна, частіше обрана студентами, з відмінними характеристиками, потрапляла у кінець рейтингу. Дисципліни, помірної чи відносної цікавості/не цікавості для обох груп студентів отримували місце всередині ранжування.

Таким чином, побудова рекомендованої моделі на базі гібридного методу колаборативної фільтрації забезпечила поєднання алгоритмів пошуку та ранжування отриманих результатів (з використанням ККП та min-max нормалізації – детальніше у розділі 2.4. Ключові інструменти, застосовані у роботі).

3.1.1. Пошук дисциплін, які найчастіше перетинаються

На першому етапі реалізованого алгоритму, для аналізу необхідно було спиратися на об’єктивні дані, у даному випадку – це були дані про дисципліни, обрані студентами (а дані про характеристики студентів були суб’єктивними). Тобто, першим етапом алгоритму було групування студентів за ознакою однорідності по критерію схожості вибору, а саме - за високим показником співпадіння обраних дисциплін – «перетином». Взято до уваги, що у групі студентів із високим перетином дисциплін, проста кількість спільних дисциплін буде різною для студентів різних спеціальностей та років навчання.

Так, у студентів старших курсів, які вже встигли обрати багато предметів, кількість спільних дисциплін буде низькою, а для молодших - навпаки. Тому якщо це значення буде занизьким певні групи студентів можуть отримати помилково велику вибірку, в той час інші можуть отримати замалу. Через це важливо визначити, яке конкретне значення «перетину» буде вважатися достатнім (див. Таблицю 3.1 з ілюстрацією процесу перетину дисциплін для конкретного студента). Так на прикладі зображено п'ятьох студентів, які записані на деякі з восьми дисциплін. У відповідній колонці представлено значення перетину вибору дисциплін (1 означає факт вибору дисципліни студентом) між студентом №1 та рештою студентів, а також значення нормалізації цього показника. Очевидно, що вибір студентів №2 і

№3 є найближчими до вибору студента №1 (значення перетину дорівнює 4 і 3 відповідно, а нормалізоване значення 1 і 0.75).

Таблиця 3.1. Перетин дисциплін для студента №1 та решти студентів

Студент	**Дисципліна**								**Перетин**	**Норм**
	№1	**№2**	**№3**	**№4**	**№5**	**№6**	**№7**	**№8**		
№1	**0**	**1**	**1**	**0**	**0**	**0**	**1**	**1**	--------	--------
№2	0	1	1	0	0	0	1	1	4	**1**
№3	0	0	1	1	0	0	1	1	3	**0.75**
№4	0	0	0	0	1	1	1	0	1	**0.25**
№5	1	0	0	0	1	1	0	0	0	**0**

3.1.2. Визначення константної межі перетину дисциплін

Через потенційно різні значення кількості спільних предметів для різних студентів буде правильним нормалізувати це значення, користуючись формулою min-max нормалізації (формула 2.2) і привести це значення до єдиного масштабу: у цьому випадку - значень від 0 до 1.

Після цього системі треба відсіяти частину студентів, вибір яких не буде враховуватися, для чого необхідно застосувати певну константну межу. Щоб обрати правильне значення константної межі, треба розуміти, яке найбільше значення нормалізованого перетину гарантовано забезпечить результати. За документами на сайті УКМА для студентів факультету інформатики мінімум 50% кредитів має йти на нормативні дисципліни. Тому логічним буде поставити це значення трохи вище 50%, наприклад 70% (**0.7**).

Таке значення було підтверджене як достатнє під час тестів із згенерованими даними, які імітують студентів факультету інформатики з різних курсів.

3.1.3. Ранжування студентів за характеристиками

Після того, як було визначено студентів, яких можна вважати «схожими», та обрані ними дисципліни (певну частину з яких студент, ще не обрав), потрібно проаналізувати, які з цих студентів є схожими (більшою чи меншою мірою) на студента-отримувача рекомендації. Для цього необхідно проранжувати цих студентів.

За критерії цього ранжування було взято «характеристики» студентів, надані ними при реєстрації, коли вони оцінювали свої здатності, навички та побажання за шкалою від 0 до 10 (наприклад, здатність до вивчення певних дисциплін, академічна успішність, бажання отримати теоретичні знання чи практичні навички тощо).

Отже, у групі студентів, «однорідних» за високим показником перетину дисциплін, потрібно знайти «сусідів», чиї характеристики є із максимально схожими. Оптимальним для цього рішенням буде застосування алгоритму пошуку «сусідів» із використанням коефіцієнту кореляції Пірсона (ККП, формула 2.1). Використовуючи його, треба порівняти оцінки характеристик студента-отримувача рекомендації, із з усередненим показником характеристик для групи студентів з високим перетином дисциплін.

Використання ККП дає змогу знайти для кожного студента не лише студентів із схожими характеристиками, а і з протилежними. «Протилежність» як характеристика є також важливою, оскільки дає можливість визначити дисципліни, обрані студентами із протилежними

інтересами, та які студент не хотів би бачити на початку рекомендованого списку. Так, у Таблиці 3.2 наведено порівняння «схожості» характеристик студента №1 із характеристиками решти студентів через визначення коефіцієнту лінійної кореляції між ними.

Таблиця 3.2. Кореляція між характеристиками студента №1 та рештою студентів

Студент	**Характеристика**							***ККП***
	№1	**№2**	**№3**	**№4**	**№5**	**№6**	**№7**	
№1	**8**	**4**	**3**	**0**	**6**	**8**	**2**	**--**
№2	9	9	8	1	8	5	0	0.61
№3	10	7	1	7	9	7	5	0.51
№4	6	5	3	3	1	1	4	-0.08
№5	6	8	2	9	9	2	8	-0.4

3.1.4. Ранжування дисциплін використовуючи надані дані

Наступним кроком створення рекомендацій було визначення оцінки дисциплін для рекомендацій студенту. У цій оцінці потрібно давати перевагу дисциплінам, які найчастіше обирали студенти із схожими обраними дисциплінами та найближчими характеристиками.

Для ранжування дисциплін обрано спосіб, у якому факт вибору дисципліни отримує відповідне значення ККП характеристик відповідного студента, а значення рейтингу кожної дисципліни буде дорівнювати сумі значень цих «фактів» вибору (Таблиця 3.3). Таким чином, дисципліни, популярні серед студентів зі схожими характеристиками, отримають високий рейтинг з позитивним значенням, а дисципліни, популярні серед студентів із протилежними характеристиками - рейтинг з негативним значенням.

Дисципліни, непопулярні серед студентів із схожим набором дисциплін, отримають значення рейтингу близьке до 0.

Таблиця 3.3. Ранжування дисциплін за значенням ККП характеристик та фактом вибору дисципліни рештою студентів

студент	***ККП***	**Дисципліна**						
		№1	**№2**	**№3**	**№4**	**№5**	**№6**	**№7**
№2	0.61	0.61	0.61	0	0.61	0.61	0.61	0.61
№3	0.51	0	0.51	0	0.51	0	0.51	0
№4	0.49	0	0.49	0	0	0.49	0	0
№5	-0.33	-0.33	-0.33	0	-0.33	-0.33	-0.33	0
№6	-0.33	0	-0.33	0	-0.33	-0.33	0	-0.33
№7	-0.34	-0.34	0	0	-0.34	-0.34	-0.34	0
№8	-0.40	0	0	0	0	0	0	-0.40
Рейтинг дисциплін		**-0.05**	**0.96**	**0**	**0.13**	**0.11**	**0.46**	**-0.12**

3.2. Аналіз технічного завдання

Прототип веб-застосунку, розроблений у практичній частині цієї роботі складається із двох частин – системи пропозиції рекомендацій для конкретних студентів та системи адміністрування дисциплін та характеристик студентів.

Такі передумови потребують поділу користувачів на три категорії:

1. незареєстровані користувачі
2. зареєстровані користувачі (студенти)
3. адміністратори

1. **Незареєстровані користувачі** можуть:

- Авторизуватися

- Зареєструватися

2. **Зареєстровані користувачі (студенти)** можуть:
   - Отримати список рекомендованих дисциплін
   - Переглядати дані свого профілю
   - Переглядати та шукати дисципліни за різними параметрами
   - Виписуватися на записуватися на дисципліни
3. **Адміністратори** можуть:
   - Переглядати дані свого профілю;
   - Переглядати та шукати дисципліни за різними параметрами;
   - Створювати нові дисципліни;
   - Редагувати та видаляти створені дисципліни;

Різні ролі та права доступу користувачів вимагають різного відображення того самого додатку.

Так, незареєстрованим користувачам будуть доступними та видимими лише дві сторінки: авторизація та реєстрація.

Для зареєстрованих користувачів (студентів) сторінки неавторизованого користувача будуть недоступними, проте вони матимуть доступ до сторінки свого профілю, сторінки пошуку дисциплін та сторінки рекомендацій.

Адміністратори також матимуть доступ до сторінок профілю користувачів та пошуку дисциплін, але на сторінці пошуку будуть присутні додаткові поля для створення, зміни та видалення дисциплін.

3.3. Обгрунтування вибору архітектури та технологій

Архітектура прототипу є багаторівневою з інтегрованим набором технологій, що забезпечують ефективну роботу додатку. Серед головних

завдань, які мала вирішувати архітектура були (1) швидкість процесів обробки даних та (2) забезпечення можливостей легкої інтеграції додатку до інших систем.

Формування переліку рекомендованих дисциплін потребує аналізу великої кількості даних про вибір та комбінацію дисциплін користувачів, а також порівняння характеристик студентів. Висока ресурсоємність алгоритму створення рекомендацій формує потребу у швидкості обробки. Для вирішення цієї проблеми з одного боку, рекомендаційний алгоритм було покращено шляхом обмеження списку студентів, з якими порівнюються характеристики. З іншого боку, щоб запобігти проблемі зі швидкістю обробки у майбутньому, запропоновано рішення не переносити відповідальність відображення користувацького інтерфейсу до клієнтської частини. У результаті, сторона клієнта отримує єдину сторінку, яка буде постійно «перемальовуватися» та робити запити до серверної частини. Такий підхід називається Single Page Application [36] та покращує адаптивність серверної частини додатку.

В той час, сервер може сконцентрувати свої обмеженні ресурси на опрацюванні REST запитів від клієнта. У такий спосіб формується спроможність економити ресурси серверної частини на рендерінгу та ререндерингу сторінки користувача, і, перекладаючи цю відповідальність на клієнта, системі забезпечуються можливість фокусуватися на виконані ресурсоємних операцій.

Окрім розподілення навантаження, такий підхід дає можливість зробити серверну частину незалежною від реалізації користувацького інтерфейсу і таким чином додаток стає ізоморфним [37].

Серверна частина прототипу додатку створена з використанням мови програмування Java та фреймворку Spring Boot. Можливість автоконфігурації та використання принципів інверсії контролю та ін’єкції залежностей забезпечили прототипу швидкість розробки та гнучкі можливості для подальшого масштабування. Також на фреймворк Spring Boot покладено відповідальність обробки аутентифікації запитів, обробки бізнес логіки та доступу до бази даних. Алгоритм рекомендацій реалізований як окремий сервіс додатку [38, 39].

Інтерфейс додатку реалізований з використанням мови програмування JavaScript та фреймворку ReactJs. Велика популярність та широке застосування фреймворку відкривають доступ до великої бібліотеки модулів, інтегрованих із ним та забезпечують потенційну адаптабельність та можливості інтеграції до інших систем. Фреймворк також відомий своїми швидкістю та ефективністю, завдяки використанню віртуального DOM-дерева, оновлення якого відбуваються в лише необхідних частинах при зміні вхідних даних.

Таким чином, побудова описаної вище багаторівневої архітектури, та поєднання її переліченим вище набором технологій, допомогло вирішити проблеми швидкості обробки процесів, спростило та прискорило розробку прототипу, та загалом забезпечило кращу адаптивність додатку до можливої інтеграції в іншу систему.

3.4. Опис серверної частини

3.4.1. Опис структури серверної частини

Клієнт-серверну частину застосунку реалізовано за допомогою фреймворку Spring Boot, відповідні файли якого зберігаються разом у одному

проекті, структуру якого наведено на Рисунку Б.1. Роботу сервера реалізує модуль org.example.reactspring, сформований пакетами, що відповідають роботу різних аспектів серверної частини.

- **Пакет config** відповідає за налаштування конфігурації проекту. Передусім йдеться про класи, які відповідають за безпеку додатку та обмеження доступу до нього. Тут налаштовується логіка автентифікації запитів та формування мапи доступу до конкретних енд-поінтів.
- **Пакет controller** відповідає за створення енд-поінтів та передачу запитів до сервісів додатку. Підпакет dto містить представлення запитів у вигляді об’єктів Java класів
- **Пакет entity** містить представлення сутностей, які зберігаються та дістаються із бази даних.
- **Пакет repository** містить класи, які забезпечують контакт з базою даних
- **Пакет service** містить реалізацію бізнес логіки самого додатку. Підпакети recommendation та generation відповідають за реалізацію самого алгоритму додатку та генерацію демонстраційних даних.

3.4.2. Модель бази даних

Модель реалізованої бази даних наведено на Рисунку 3.1 у вигляді шести таблиць-даних та зв’язків між ними.

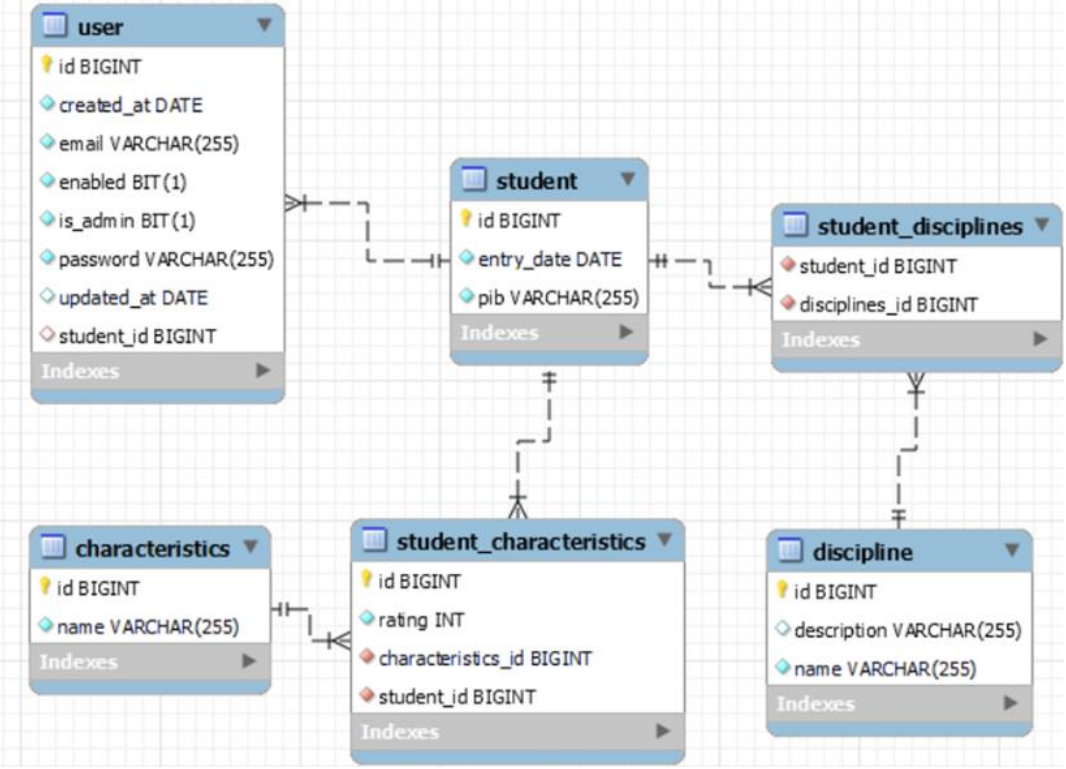

Рисунок 3.1 Схематичне зображення бази даних реалізованої рекомендаційної системи

Опис таблиць:

1. **«user»** – таблиця, що містить дані авторизації про адміністраторів та студентів сайту

   Атрибути:

   - id – унікальний ідентифікатор користувача у базі даних
   - Email – електрона адреса користувача
   - Password – пароль користувача
   - Is_admin – чи має користувач привілеї адміністратора
   - Enabled – чи активував користувач доступ до свого акаунту
   - Student_id – ідентифікатор студента, до якого під’єднаний акаунт
   - Created_at – час створення акаунту користувача
   - Updated_at – час останньої модифікації даних користувача

2. **«student»** – таблиця, що містить загальні дані про студента

   Атрибути:

   - id – унікальний ідентифікатор студента у базі даних
   - Pib – ПІБ студента
   - Entry_date – дата вступу студента

3. «**characteristics»** – таблиця, що описує характеристики студентів

   Атрибути:

   - id – унікальний ідентифікатор характеристики студента у базі даних
   - name – назва характеристки студента

4. **«student_characteristics»** – таблиця, що містить значення оцінок характеристик студентів

   Атрибути:

   - id – унікальний ідентифікатор оцінки характеристики студента у базі даних
   - rating –оціночне значення характеристики студента за шкалою від 0 до 10
   - student_id – ідентифікатор студента
   - characteristics_id – ідентифікатор характеристики студента

5. **«disciplines»** – таблиця, що описує дисципліни університету

   Атрибути:

   - id – унікальний ідентифікатор дисципліни у базі даних
   - name – назва дисципліни
   - description – опис дисципліни

6. **«student_disciplines»** – таблиця, що описує обрані студентом дисципліни

Атрибути:

- student_id – ідентифікатор студента
- discipline_id – ідентифікатор характеристики дисципліни

3.5. Опис інтерфейсу

Для входу на сайт користувач обирає сторінку стандартної авторизації, де форма запропонує йому ввести адресу власної електронної пошти і пароль (Рисунок 3.2)

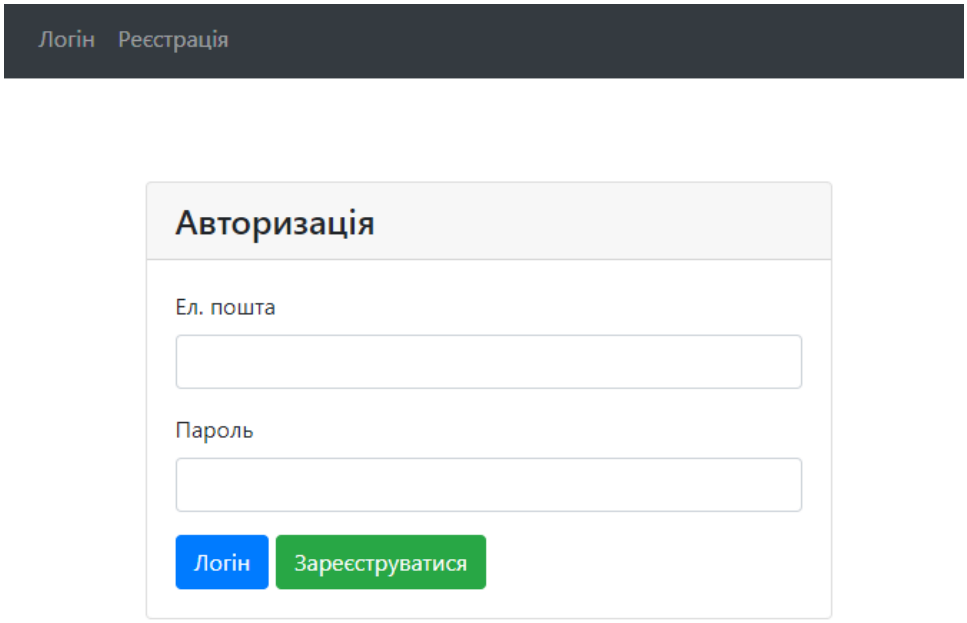

Рисунок 3.2. Інтерфейс сторінки авторизації рекомендаційної системи

Якщо користувач ще не має акаунту, він може перейти на сторінку реєстрації, де реєстраційна форма запропонує ввести ПІБ, дату вступу, базові особисті дані (вік, стать), а також придумати унікальний пароль та підтвердити його.

Перед підтвердженням реєстрації на цій сторінці, студент також має також «описати себе», заповнюючи спрощену «анкету», тобто оцінюючи себе

за визначеними характеристиками по шкалі від 1 до 10. Завдання цих характеристик – перевести у цифровий формат суб’єктивну оцінку студента власних навичок, потреб, вподобань та побажань щодо напрямків розвитку і навчання (Рисунок 3.3)

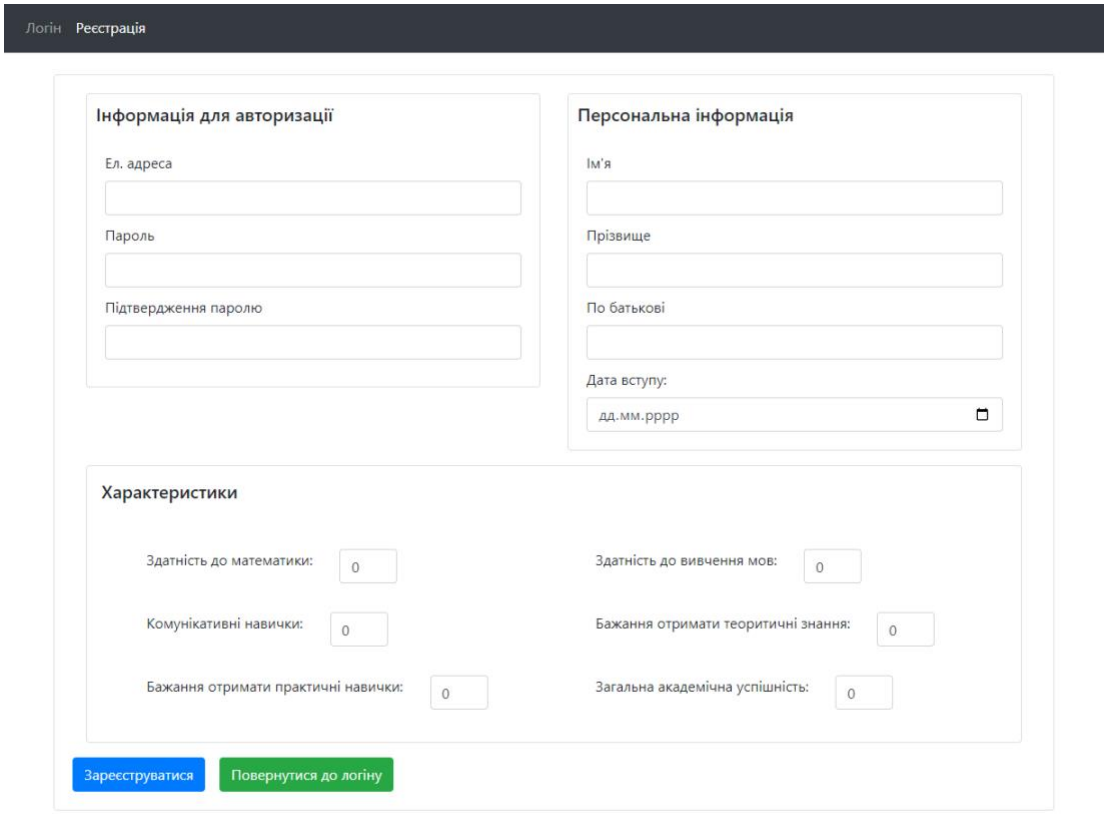

Рисунок 3.3.

Інтерфейс сторінки реєстрації користувача рекомендаційної системи

Якщо система визначила дані користувача валідними, відбувається автоматична авторизація користувача в системі, і система переводить користувача на сторінку його профілю. Також, після успішної авторизації, змінюються опції на навігаційному меню у верхній частині екрану: опції «Логін» та «Реєстрація» змінюються на «Рекомендації», «Усі дисципліни», «Профіль» та «Вихід» (Рисунок 3.4.)

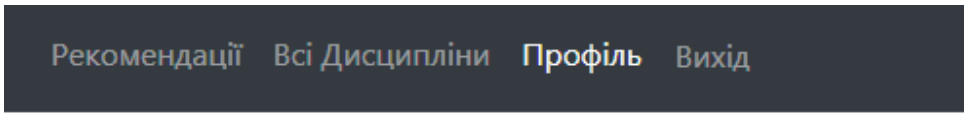

Рисунок 3.4.

Опції навігаційного меню після успішної реєстрації користувача

Сторінка профілю користувача демонструє всі його дані: ПІБ, дата вступу, перелік його характеристик з оцінками, обрані дисципліни.

Користувач має можливість змінювати свій профіль. Наприклад, щоб відмовитися від обраного курсу необхідно просто «натиснути» на червону кнопку «Виписатися» навпроти відповідної дисципліни (Рисунок 3.5)

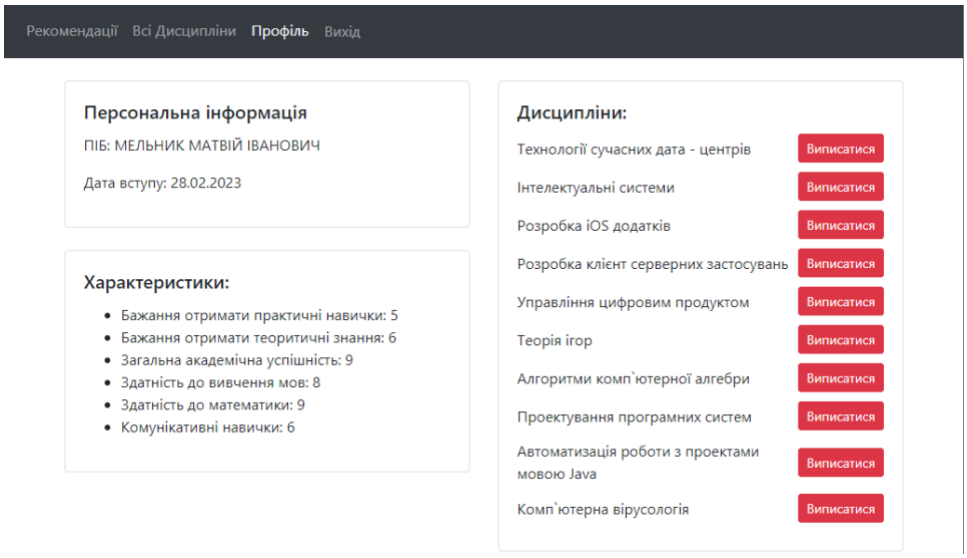

Рисунок 3.5.

Інтерфейс сторінки профілю користувача рекомендаційної системи

Для переходу до меню пошуку та запису на дисципліни користувачу необхідно у навігаційному меню обрати опцію «Усі дисципліни».

На сторінці «Усі дисципліни» користувач має можливість шукати необхідні йому дисципліни або відфільтрувати їх за певними параметрами,

визначати черговість їхнього відображення. Для цього у верхній частині розташована картка параметрів пошуку для запиту.

Фільтрація знайдених дисциплін може виконуватися за трьома параметрами: «Усі», «Тільки записані» та «Тільки не записані». Результати можна сортувати двома способами: “Asc”(у порядку зростання) та “Desc” (у порядку спадання). Кнопка «Пошук» дозволяє оновити результати пошуку після зміни параметру фільтру або створення нового запиту.

Для «гортання» сторінок з результатами пошуку в нижній частині екрану розташовані відповідні інструменти управління: «кнопки» з номерами сторінок та переходів - «Назад» і «Далі». У центральній частині екрану, навпроти кожної дисципліни є також «кнопка» управління, яка дозволяє користувачеві змінити своє рішення щодо кожної дисципліни. Ця «кнопка» пропонує дію «Записатися» чи «Виписатися» залежно від статусу користувача (чи записався вже студент на дисципліну чи ні) (Рисунок 3.6).

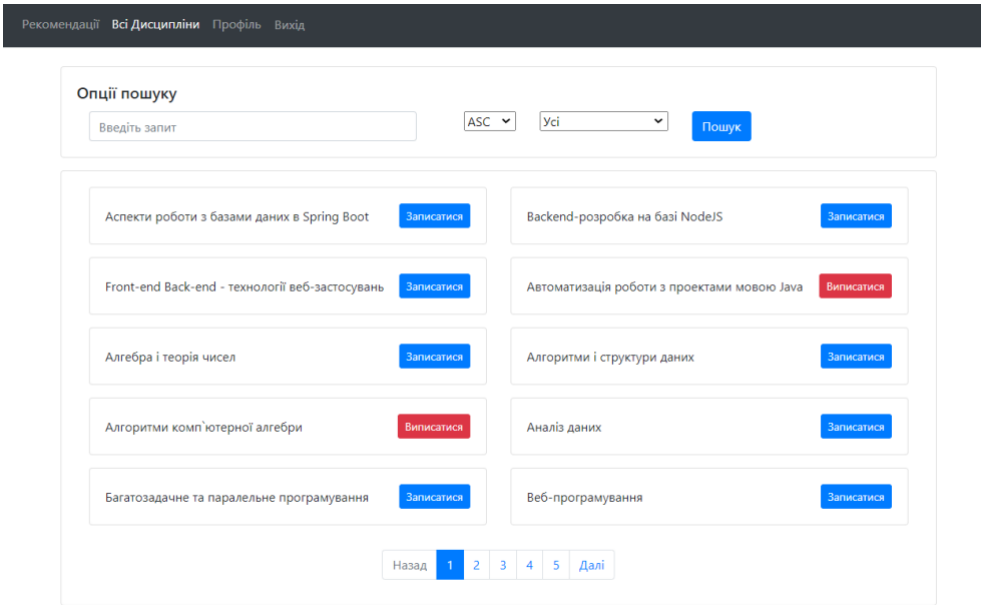

Рисунок 3.6. Інтерфейс сторінки «Всі дисципліни» рекомендаційної системи

Навігаційне меню системи має опцію «Рекомендації», яка переводить користувача на відповідну сторінку. В меню системи ця сторінка оформлена головною. Вона демонструє всі дисципліни (на які користувач ще не записаний) в певній послідовності, створеній алгоритмом рекомендацій системи, індивідуально для конкретного користувача (Рисунок 3.7)

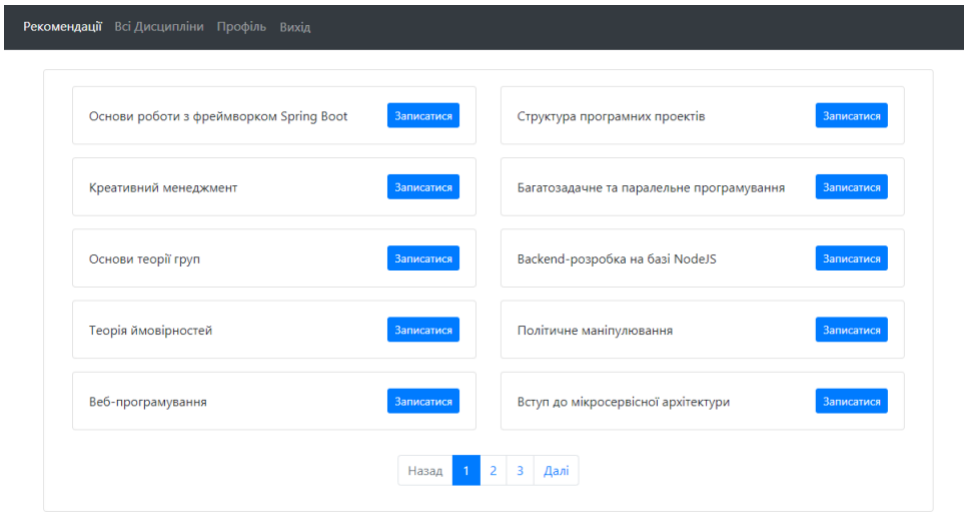

Рисунок 3.7.

Інтерфейс головної сторінки «Рекомендації» з індивідуальною послідовністю запропонованих дисциплін конкретному користувачеві

Висновки по роботі та рекомендації для подальших досліджень

У межах цієї роботи було розглянуто сучасні системи, методи та інструменти створення рекомендацій. Зокрема, досліджено досвід та запоруки успішного використання алгоритмів трьома провідними та найвідомішими сервісами (Spotify, Amazon, Netflix) щодо збору, обробки, аналізу даних для створення персоналізованих рекомендацій користувачам.

В окремому розділі було проаналізовано принципи дії ключових алгоритмів створення рекомендацій а також описано інструменти та методи, застосовані у практичній частині цієї роботи.

На базі проведеного аналізу та використовуючи стандартні алгоритми створення рекомендацій, було розроблено та реалізовано рекомендаційну систему, метою якої була автоматизація підбору вибіркових навчальних дисциплін для студентів НаУКМА.

На основі алгоритму розробленої рекомендаційної системи було створено веб-додаток, серверна частина якого виконана мовою програмування Java з фреймворком Spring Boot, а інтерфейс користувача - з фреймворком ReactJS та допоміжними бібліотеками React Redux та React Bootstrap. Кожний крок виконання практичної частини описано у відповідному розділі роботи.

У подальших дослідженнях може бути розглянуто вдосконалення розробленої системи рекомендацій, зокрема шляхом автоматичної генерації характеристик студентів на підставі реальних даних. Також алгоритми системи можуть бути скомбінованими з іншими підходами до створення рекомендацій для більш якісних та точних рекомендацій студентам.

Практичним застосуванням роботи є можливість майбутньої інтеграції розробленої рекомендаційної системи із сервісам НаУКМА та її використання студентами при виборі навчальних дисциплін.

Список використаних джерел

1. НаУКМА в рейтингах ВНЗ України. *НаУКМА.*
2. Кузьменко Г. ОК, Google: що робити з записом на вибіркові дисципліни, якщо ти фреш/фрешка. URL: https://telegra.ph/OK-Google-shcho-robiti-z-zapisom-na-vibirkovi-disciplini-yakshcho-ti-freshfreshka-03-08.
3. Електронні сервіси ІОЦ. *НаУКМА*. URL: https://www.ukma.edu.ua/index.php/resursi/icc.
4. Dilmegan C. Recommendation systems: applications and examples in 2023. *Marketing recommendation system*. URL: https://research.aimultiple.com/recommendation-system/.
5. Muralidhara G. Decoding amazon's recommendation system. argoid. URL. *A complete study of Amazon's recommendation system. Argoid*. URL: https://www.argoid.ai/blog/decoding-amazons-recommendation-system.
6. Inside Spotify's recommender system: A complete guide to Spotify recommendation algorithms. *Music tomorrow*. URL: https://www.music-tomorrow.com/blog/how-spotify-recommendation-system-works-a-complete-guide-2022.
7. Web API: documentation, track's audio features. *Spotify for developers. Retrieve metadata from Spotify content, control playback, get recommendations*. URL: https://developer.spotify.com/documentation/web-api/reference/get-audio-features.
8. Ciocca S. The science behind personalized music recommendations. *How Does Spotify Know You So Well?*. URL: https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe.
9. Spotify audio analysis. *Spotify*. URL: https://spotify-audio-analysis.glitch.me/.

10. Nagpal M. How does spotify use machine learning?. *DEV: Developers community*. URL: https://dev.to/mage_ai/how-does-spotify-use-machine-learning-4pdg.

11. Schedl M., Zamani H., Chen C.-W. Current challenges and visions in music recommender systems research (abstract). *Springer Link: International journal of multimedia information retrieval 7, 95-116 (2018)*. URL: https://link.springer.com/article/10.1007/s13735-018-0154-2.

12. Chang E. Building a song recommendation system with Spotify. *Medium. Towards Data Science*. URL: https://towardsdatascience.com/part-iii-building-a-song-recommendation-system-with-spotify-cf76b52705e7.

13. Wirfs-Brock J., Mennicken S., Thom J. Giving voice to silent data: designing with personal music listening history. *Spotify Research and Data*. URL: https://research.atspotify.com/2020/05/giving-voice-to-silent-data-designing-with-personal-music-listening-history/.

14. Персоналізація від NETFLIX та SPOTIFY: як бізнесу в Україні перейняти досвід. *Kyivstar business club. Тренди та аналітика*. URL: https://hub.kyivstar.ua/news/personalizacziya-vid-netflix-ta-spotify-yak-biznesu-v-ukrayini-perejnyaty-dosvid/.

15. 2018 Personalization pulse check survey. *Personalization Pulse Check. Accenture Interactive*. URL: https://www.accenture.com/_acnmedia/PDF-77/Accenture-Pulse-Survey.pdf.

16. Linden G., Smith B., York J. Amazon.com recommendations. Item to item collaborative filtering. *Industry report by IEEE Computer Society*.

17. How it works. *AWS. Amazon Personalize.* URL: https://aws.amazon.com/personalize/.

18. Krysik A. Amazon's product recommendation system in 2021: how does the algorithm of the ecommerce giant work?. *Recostream*. URL: https://recostream.com/blog/amazon-recommendation-system.

19. Biddle G. Search medium write sign up sign in top highlight A brief history of netflix personalization. *Medium*. URL: https://gibsonbiddle.medium.com/a-brief-history-of-netflix-personalization-1f2debf010a1.

20. Pereira D. Netflix business model. *The business model analyst*. URL: https://businessmodelanalyst.com/netflix-business-model/.

21. How netflix's recommendations system works. *Netflix help center*. URL: https://help.netflix.com/en/node/100639#:~:text=We%20estimate%20the%20likelihood%20that,preferences%20on%20our%20service,%20and.

22. Alvino C., Basilico J. Learning a personalized homepage. *Netflix Technology Blog*. URL: https://netflixtechblog.com/learning-a-personalized-homepage-aa8ec670359a.

23. Kasula C. Netflix Recommender System – A Big Data Case Study. The story behind Netflix's famous Recommendation System. *Towards Data Science*. URL: https://towardsdatascience.com/netflix-recommender-system-a-big-data-case-study-19cfa6d56ff5.

24. Rocca B. Introduction to recommender systems: overview of some major recommendation algorithms. *Towards data science*. URL: https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada.

25. Isinkaye F., Folajimi Y., Ojokoh B. Recommendation systems: principles, methods and evaluation. *Egyptian Informatics Journal (2015) 16, 261-273*. URL: https://www.sciencedirect.com/science/article/pii/S1110866515000341.

26. Payne M. Recommender systems for business - A gentle introduction. *Machine learning, Ai, data analytics, Width.ai*. URL: https://www.width.ai/post/recommender-systems-recommendation-systems.

27. Awartani M. Recommendation systems explained & simplified. *Medium.* URL: https://medium.com/@malak.awartani/recommendation-systems-explained-simplified-2fa4c3f08f45.

28. Model-based approach for collaborative filtering. *Research gate. Conference: Model-based Approach for Collaborative Filtering*. URL: https://www.researchgate.net/publication/321753015_Model-based_approach_for_Collaborative_Filtering.

29. Roy A. Introduction to recommender systems- 1: content-based filtering and collaborative filtering. *Medium. Towards Data Science*. URL: https://towardsdatascience.com/introduction-to-recommender-systems-1-971bd274f421.

30. Vas G. How recommendation systems tackle the cold start problem. URL: https://www.yusp.com/blog-posts/cold-start-problem/.

31. Chiang J. 7 types of hybrid recommendation system. *Analytics Vidhya*. URL: https://medium.com/analytics-vidhya/7-types-of-hybrid-recommendation-system-3e4f78266ad8.

32. Demystifying hybrid recommender systems and their use cases. *Bluepi*. URL: https://www.bluepiit.com/blog/demystifying-hybrid-recommender-systems-and-their-use-cases/.

33. Rakesh R. Neighborhood based collaborative filtering – part 4. *Fnplus Club*. URL: https://medium.com/fnplus/neighbourhood-based-collaborative-filtering-4b7caedd2d11.

34. Calkins K. G. Lesson 5 - correlation coefficients. *Applied Statistics*. URL:

https://www.andrews.edu/~calkins/math/edrm611/edrm05.htm#:~:text=Correlation%20coefficients%20whose%20magnitude%20are%20between%200.5%20and%200.7%20indicate,which%20have%20a%20low%20correlationhttps://uk.wikipedia.org/wiki/Кореляція#Коефіцієнт_кореляції_Пірсона.

35. Loukas S. Everything you need to know about Min-Max normalization: A Python tutorial. *Medium. Towards data science*. URL: https://towardsdatascience.com/everything-you-need-to-know-about-min-max-normalization-in-python-b79592732b79.

36. SPA (Single-page application). *MDN Web docs guides*. URL: https://developer.mozilla.org/en-US/docs/Glossary/SPA.

37. What is an isomorphic application?. URL: https://www.lullabot.com/articles/what-is-an-isomorphic-application.

38. A JavaScript library for building user interfaces. *React*. URL: https://legacy.reactjs.org.

39. Why spring?. *Spring by VMware Tanzu*. URL: https://spring.io/why-spring.

Додаток А.

Логіка алгоритмів колаборативних методів фільтрації

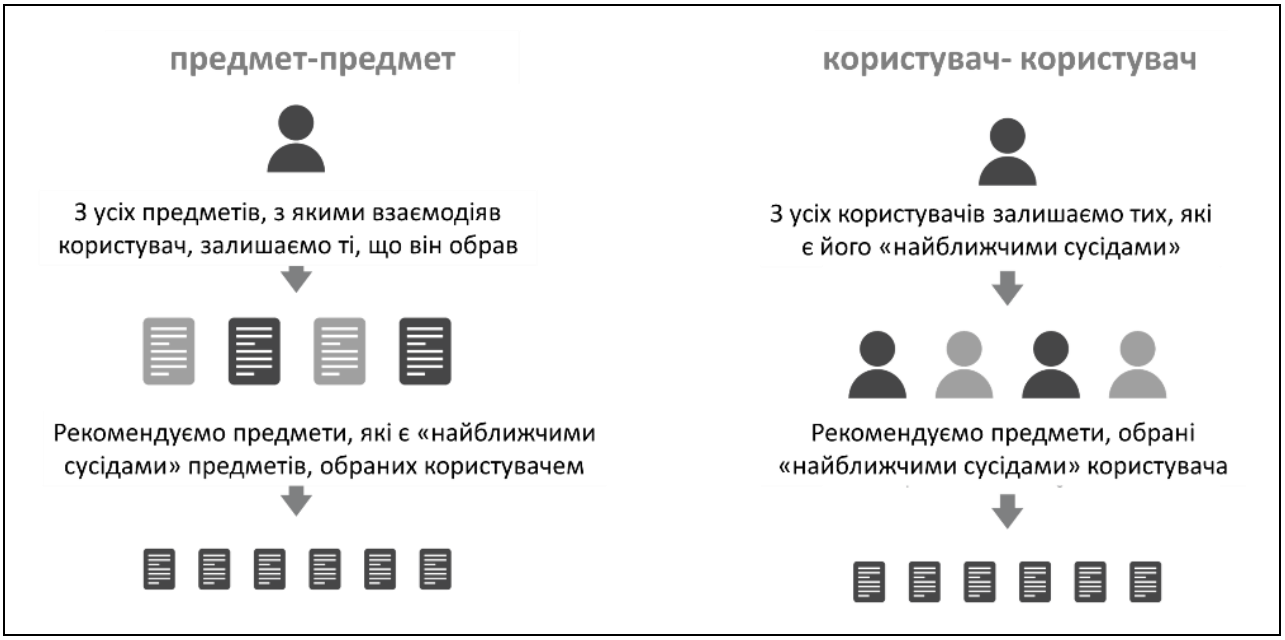

Рисунок.А.1. - Ілюстрація логіки алгоритмів «предмет-предмет» та «користувач-користувач» [24]

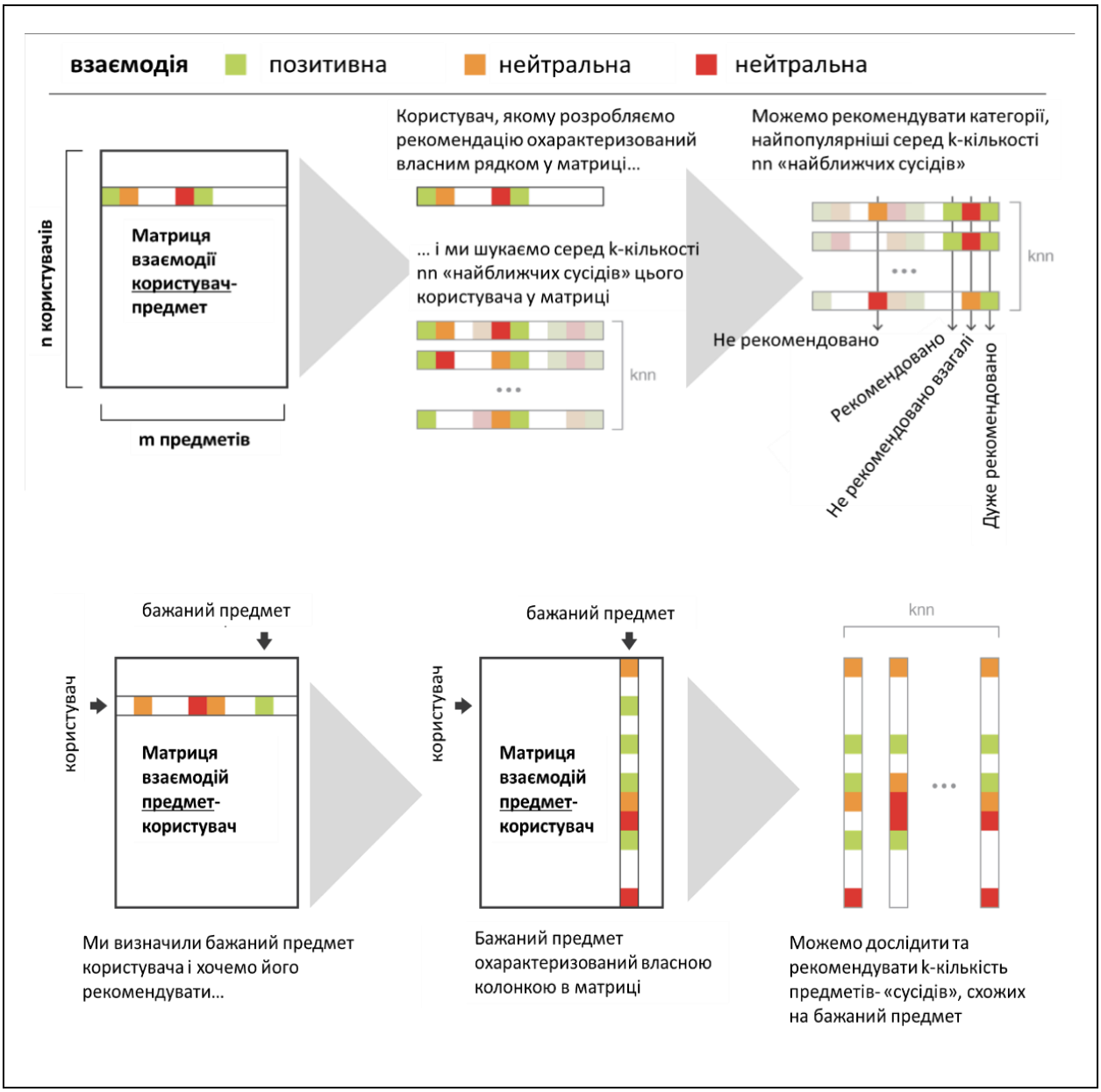

Рисунок.А.2. - Ілюстрація логіки взаємодії матриці «користувач-предмет» та «предмет-користувач» [24]

Додаток Б.

Структура серверної частини

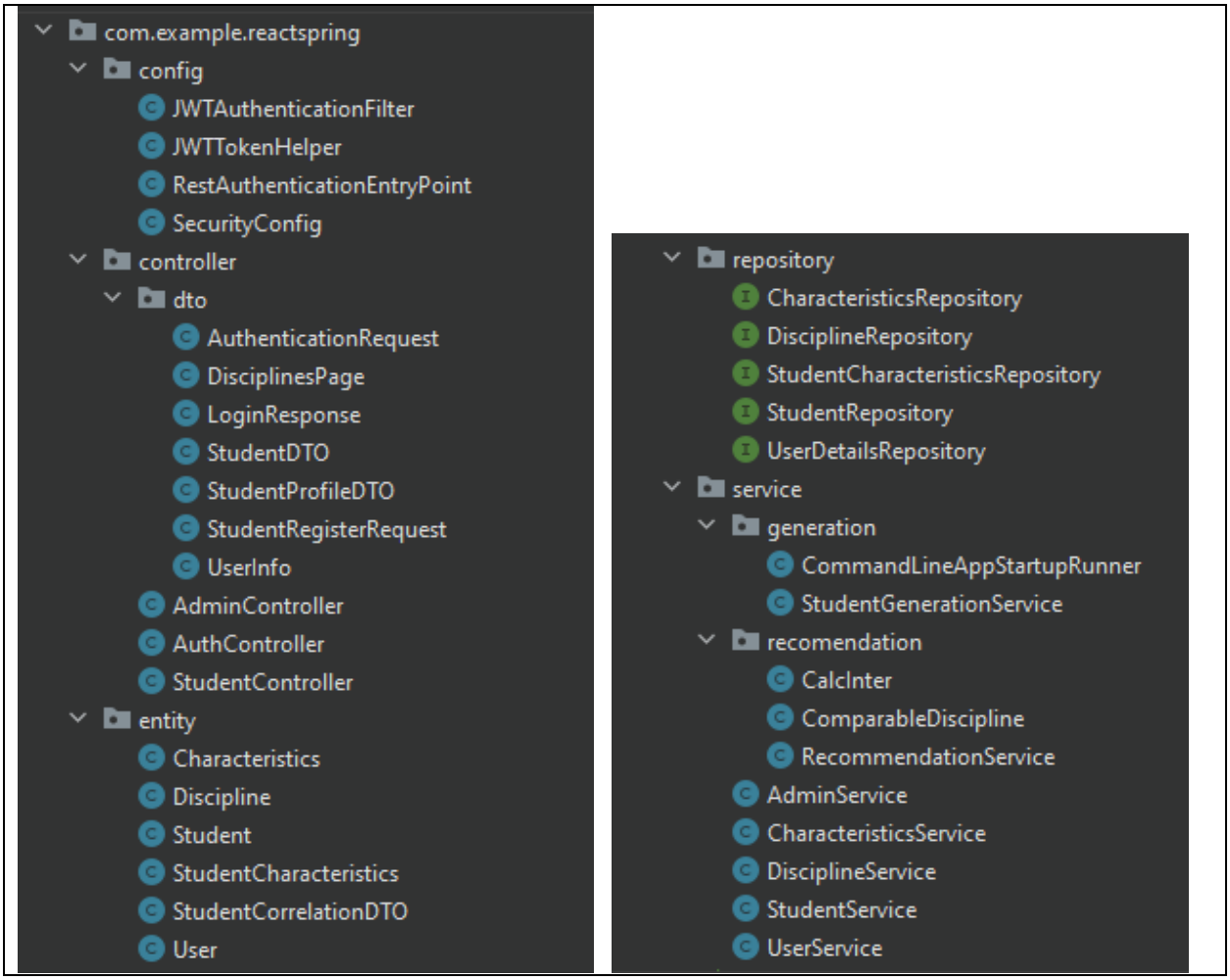

Рисунок.Б.1. - Структура модуля com.example.reactspring