Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____
(повна назва)

Кафедра _____Системотехніки_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

«Розробка та аналіз рекомендаційної системи формування індивідуальної освітньої траєкторії в закладах вищої освіти»_____
(тема)

Виконав:
студент _2__ курсу, групи _ІТПм-21-2_____
_____Павленко М.Ю._____
(прізвище, ініціали)

Спеціальність __122-Комп'ютерні науки_____
_____
(код і повна назва спеціальності)
Тип програми _освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)

Освітня програма __Інформаційні технології проектування_____
(повна назва освітньої програми)

Керівник ___доц. Білова Т.Г._____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ _____
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет_____Комп'ютерних наук_____

Кафедра_____Системотехніки_____

Рівень вищої освіти_____другий (магістерський)_____

Спеціальність_____122-Комп'ютерні науки_____
(код і повна назва)

Тип програми _____освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)

Освітня програма_____Інформаційні технології проектування_____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____»_____ 20 ___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові_____Павленко Марку Юрійовичу_____
(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка та аналіз рекомендаційної системи формування індивідуальної освітньої траєкторії в закладах вищої освіти»
затверджена наказом університету від 21 листопада 2022 р. № 1504 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 12 грудня 2022 р.
3. Вихідні дані до роботи Дані статей та публікацій, книг та електронних ресурсів за темою роботи.
4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ, 4.2 Аналіз предметної області, 4.2.1 Аналіз існуючої проблеми формування індивідуальної освітньої траєкторії студентами, 4.2.2 Соціальні мережі як середовище з вирішення наявних проблем, 4.2.3 Переваги на недоліки використання соціальних мереж для вдосконалення навчального процесу, 4.2.4 Опис предметної області, 4.2.5 Огляд існуючих систем-аналогів для формування освітньої траєкторії студента, 4.2.6 Актуальність та мета розробки, 4.2.7 Ключові методи з реалізації рекомендаційних систем, 4.2.8 Опис можливих проблем та способів їх вирішення, 4.2.9 Постановка задачі 4.3 Опис обраних технологій та методів проектування з реалізації системи 4.3.1 Технології та методи реалізації рекомендаційної системи 4.3.2 Метод колаборативної фільтрації 4.3.2.1 Метод колаборативної фільтрації, що базується на користувачах (User-based Collaborative filtering) 4.3.2.2 Метод колаборативної фільтрації, що базується на сутностях (Item-based Collaborative filtering) 4.3.3 Гібридні методи фільтрації 4.3.4 Опис ключових алгоритмів роботи рекомендаційних систем 4.3.4.1 Алгоритми кластеризації 4.3.4.2 Алгоритми сумісної фільтрації за користувачами 4.4 Програмна реалізація рекомендаційної системи 4.4.1 Опис архітектури програмного засобу 4.4.2 Обґрунтування вибору платформи СУБД 4.4.3 Обґрунтування вибору мови програмування 4.4.4 Реалізація рекомендаційної системи з вибору вибіркових освітніх компонент студентами ХНУРЕ 4.4.5 Проектування фізичної схеми баз даних додатку 4.4.6 Аналіз вимог до вхідного набору даних та результатів роботи, 4.4.7 Розробка гібридного рекомендаційного методу 4.4.8 Проектування графічного інтерфейсу системи, 4.4.9 Аналіз перспектив покращення роботи створеного рішення

4.4.10 Аналіз перспектив із покращення загального процесу дистанційного навчання 4.5 Висновки. 5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 5.1 Принцип роботи рекомендаційної системи мережі Facebook 5.2 Загальний принцип роботи рекомендаційної системи 5.3 Приклад матриці пріоритетних оцінок 5.4 Монолітна організація гібридної рекомендаційної системи 5.5 Паралельна організація гібридної рекомендаційної системи 5.6 Конвеєрна організація гібридної рекомендаційної системи 5.7 Приклад роботи сумісної фільтрації за користувачами 5.8 Схема принципу роботи шаблону MVC 5.9 Фізична схема бази даних 5.10 Програмна реалізація формули косинусові міри 5.11 Програмна реалізація функції формування матриці переваг 5.12 Програмна реалізація функції з обчислення транспонованої матриці 5.13 Програмна реалізація функції з перевірки матриці на розрідженість 5.14 Програмна реалізація функції зі знаходження схожих значень за середньою оцінкою 5.15 Програмна реалізація методу розрахунку середнього значення за обома результатами 5.16 Графічний інтерфейс системи на початку роботи програми 5.17. Графічний інтерфейс, що демонструє результати роботи програми 5.18 Результати поетапної роботи гібридного алгоритму.

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основний розділ	Доц. Білова Т.Г.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на дипломну роботу	1.09.2022	
2	Аналіз предметної області	02.09.2022 - 03.10.2022	
3	Огляд джерел інформації	04.10.2022 - 10.10.2022	
4	Огляд методів та технологій	11.10.2022 - 17.10.2022	
5	Постановка задачі дослідження	17.10.2022 - 24.10.2022	
6	Проектування інформаційної технології	25.10.2022 - 15.11.2022	
7	Аналіз результатів, отриманих за допомогою	16.11.2022 - 30.12.2022	
8	Оформлення пояснювальної записки	01.12.2022 - 15.12.2022	
9	Оформлення графічної частини та презентаційних	16.12.2022 - 17.12.2022	
10	Представлення на рецензування	20.12.2022	
11	Представлення дипломної роботи в деканат	20.12.2022	

Дата видачі завдання 01 вересня 2022 р.

Студент _____ Павленко М.Ю.
(підпис)

Керівник роботи _____ доц. Білова. Т.Г.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи містить: с. 67, рис. 20, формул 9, дод. 3, джерел 26.

ІНФОРМАЦІЙНА СИСТЕМА, ХНУРЕ, СОЦІАЛЬНА МЕРЕЖА, РЕКОМЕНДАЦІЙНА СИСТЕМА, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, ГІБРИДНА ФІЛЬТРАЦІЯ, ВЕКТОРИ ІДЕНТИЧНОСТІ, MVC, SQL, UUID, JS, REACT

Об’єктом дослідження є процес формування індивідуальної освітньої траєкторії здобувача вищої освіти.

Предметом дослідження є рекомендаційна система формування індивідуальної освітньої траєкторії в закладах вищої освіти

Мета дослідження: проектування та розробка рекомендаційної системи для вибору вибіркових освітніх компонент факультативних дисциплін здобувачами вищої освіти на базі комплексного удосконаленого гібридного методу.

Методи дослідження – системний аналіз, методи структурного аналізу, методи колаборативної фільтрації, засновані на користувачах (Item-based) та сутностях (User-Based), гібридні методи фільтрації.

Наукова новизна роботи полягає у дослідженні та розробці удосконаленого гібридного рекомендаційного методу, який поєднує методи колаборативної фільтрації та дозволяє інтерпретувати отримані результати згідно з розробленою системою перевірок.

Галузь застосування – соціальні мережі закладів вищої освіти як рекомендаційна система для вибору вибіркових освітніх компонент.

ABSTRACT

The explanatory note to the master's qualification thesis contains: 67 pages, 20 figures, 9 formulas, 3 appendices, 26 sources.

INFORMATION SYSTEM, KNURE, SOCIAL NETWORK, RECOMMENDATION SYSTEM, COLLABORATIVE FILTERING, ITEM-BASED, USER-BASED, HYBRID FILTRATION, IDENTITY VECTORS, MVC, SQL, UUID, JS, REACT

The object of the research is the process of formation of the individual educational trajectory of the student of higher education.

The subject of the research is a recommendation system for the formation of an individual educational trajectory in institutions of higher education.

The purpose of the research: to design and development of an recommendation system for the selection of selective educational components of optional disciplines by students of higher education based on the complex improved hybrid method.

Research methods – a system approach, methods of structural analysis, methods of collaborative filtering, based on users (Item-based Collaborative Filtering) and entities (User-Based Collaborative Filtering), hybrid filtering methods.

The scientific novelty of the work consists in the research and development of an improved hybrid recommendation method, which combines the methods of collaborative filtering and allows to interpret the obtained results according to the developed system of checks.

The field of application is the social networks of higher education institutions as a recommendation system for choosing selective educational components.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 8

ВСТУП 9

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 11

1.1 Аналіз інсуючої проблеми формування індивідуальної освітньої траєкторії студентами 13

1.2 Соціальні мережі як середовище з вирішення наявних проблем 15

1.3 Переваги на недоліки використання соціальних мереж для вдосконалення навчального процесу 16

1.4 Опис предметної області 17

1.5 Огляд існуючих систем-аналогів для формування освітньої траєкторії студента 18

1.6 Актуальність та мета розробки 24

1.7 Ключові методи з реалізації рекомендаційних систем 25

1.8 Опис можливих проблем та способів їх вирішення 26

1.9 Постановка задачі 28

2 ОПИС ОБРАНИХ ТЕХНОЛОГІЙ ТА МЕТОДІВ ПРОЕКТУВАННЯ З РЕАЛІЗАЦІЇ СИСТЕМИ 29

2.1 Технології та методи реалізації рекомендаційної системи 29

2.2 Метод колаборативної фільтрації 29

2.2.1 Метод колаборативної фільтрації, що базується на користувачах (User-based Collaborative filtering) 31

2.2.2 Метод колаборативної фільтрації, що базується на сутностях (Item-based Collaborative filtering) 35

2.3 Гібридні методи фільтрації 37
2.4 Опис ключових алгоритмів роботи рекомендаційних систем 41
2.4.1 Алгоритми кластерізації 41
2.4.2 Алгоритми сумісної фільтрації за користувачами 43
3 ПРОГРАМНА РЕАЛІЗАЦІЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ 45
3.1 Опис архітектури програмного засобу 45
3.2 Обґрунтування вибору платформи СУБД 47
3.3 Обґрунтування вибору мови програмування 48
3.4 Реалізація рекомендаційної системи з вибору вибіркових освітніх компонент студентами ХНУРЕ 49
3.5 Проектування фізичної схеми баз даних додатку 50
3.6 Аналіз вимог до вхідного набору даних та результатів роботи ... 52
3.7 Розробка гібридного рекомендаційного методу 54
3.8 Проектування графічного інтерфейсу системи 60
3.9 Аналіз перспектив покращення роботи створеного рішення 61
3.10 Аналіз перспектив із покращення загального процесу дистанційного навчання 62
ВИСНОВКИ 64
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 65
ДОДАТОК А Візуальний інтерфейс розробленої системи 68
ДОДАТОК Б Графічні матеріали кваліфікаційної роботи 72
ДОДАТОК В Текст програми 81
Відомість магістерської кваліфікаційної роботи 89

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення;

ПО – предметна область;

ІС – інформаційна система;

РС – рекомендаційна система;

ХНУРЕ – Харківський Національний Університет Радіоелектроніки;

СУБД – Система Управління Базою Даних;

MVC – Model-View-Controller – модель-уявлення-контролер;

SQL – Select Query Language – Мова Структурованих Запитів;

JS – JavaScript;

UUID – Universally Unique Identifier;

ВСТУП

З початком епідемії коронавірусу, у 2020 році, абсолютна більшість університетів перевели своїх студентів на дистанційне навчання. Ця подія поклала початок формування нової парадигми у навчанні, коли будь-яка навчальна діяльність повністю переведена у віртуальний формат. А сумнозвісні події кінця лютого 2022 року остаточно закріпили її на практиці, адже зараз організувати процес очного навчання в ХНУРЕ фізично неможливо, тому всі онлайн інструменти для комунікації продовжують активно використовуватись.

Серед них, на мою думку, значно виділяються соціальні мережі як найефективніші засоби з організації продуктивної, зручної та швидкої комунікації між її учасниками, якими в нашому випадку виступають студенти та викладачі та адміністративний персонал навчальних закладів.

Динамічний розвиток інформаційно-комунікаційних технологій вже досить давно породжує активну дискусію щодо можливостей використання соціальних мереж в освітньому середовищі. Віртуальна мережа загалом формує соціальні зв'язки, а під соціальною мережею розуміється онлайн-сервіс або веб-застосунок, що дозволяє будувати комунікації та поширювати різноманітну інформацію. Соціальні мережі складаються з множинуі агентів, як яких виступають окремі індивіди, групи людей чи спільноти. Зв'язки відбивають відносини між агентами мережі і можуть бути різних типів, наприклад, дружні, професійні чи споріднені.

Соціальні мережі нині вже перестали сприйматися як майданчик виключно для спілкування або розваг. У них можна знайти анонси заходів, освітні матеріали, спільноти однодумців за інтересами, можливо приймати участь у різних анкетуваннях та опитуваннях задля покращення та вдосконалення освітнього процесу, тощо. Для покращення навчального процесу також можна використовувати різні вбудовані інструменти що

можуть бути заточені саме під потреби коректування процесу дистанційного навчання.

Мета дослідження полягає у проектуванні та розробці системи для вибору вибіркових освітніх компонент факультативних дисциплін здобувачами вищої освіти на базі комплексного удосконаленого гібридного методу.

Об’єктом дослідження є процес формування індивідуальної освітньої траєкторії здобувача вищої освіти.

Предметом дослідження є рекомендаційна система формування індивідуальної освітньої траєкторії в закладах вищої освіти

Методами дослідження виступають: системний аналіз, методи структурного аналізу, методи колаборативної фільтрації, засновані на користувачах (Item-based Collaborative Filtering) та сутностях (User-Based Collaborative Filtering), гібридні методи фільтрації.

Наукова новизна роботи полягає у дослідженні та розробці удосконаленого гібридного рекомендаційного методу, який поєднує методи колаборативної фільтрації та дозволяє інтерпретувати отримані результати згідно з розробленою системою перевірок.

Галузь застосування – соціальні мережі закладів вищої освіти як рекомендаційна система для вибору вибіркових освітніх компонент.

За темою роботи були опубліковані тези на ІІІ Міжнародної студентської наукової конференції «ТЕОРЕТИЧНЕ ТА ПРАКТИЧНЕ ЗАСТОСУВАННЯ РЕЗУЛЬТАТІВ СУЧАСНОЇ НАУКИ». [19]

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Інформаці́йна систе́ма (Information system) – це сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів. Таке визначення може бути задовільним тільки при найбільш узагальненій і неформальній точці зору і підлягає подальшому уточненню.

В залежності від рівня автоматизації, інформаційні системи (надалі скорочено ІС) поділяються на ручні, напівавтоматизовані й повністю автоматизовані ІС[1]. У перших всі операції з переробки інформації виконуються виключно людьми, у других частина функціонуючих підсистем керуються та обробляються автоматично, а частина – за допомогою людських ресурсів, а в третьому випадку усі функції керування й опрацювання даних здійснюються технічними засобами без участі людини – наприклад, автоматичне керування високотехнологічними процесами.

За сферою практичного застосування, ІС поділяються на наступні класи:

- наукові дослідження (призначені для автоматизації діяльності науковців, аналізу статистичної інформації, керування експериментом);
- автоматизоване проектування – подібні системи призначені для автоматизації праці інженерів-проектувальників і розробників нової техніки або технологій;
- організаційне керування – дані системи призначені для автоматизації функції адміністративного персоналу. До цього класу відносяться ІС керування як промисловими (підприємства), так і непромисловими об'єктами (банки, біржа, страхові компанії, готелі), а також окремими офісами та системами;

- керування технологічними процесами – подібні системи призначені для автоматизації різноманітних технологічних процесів (гнучкі виробничі процеси, металургія, енергетика, тощо). [2]

Загалом, в контексті організації ефективного процесу дистанційного навчання, соціальні мережі можна віднести до напівавтоматизованих ІС, що мають в собі гібридну систему керування. Вони можуть бути в певній мірі задіяні в науковому процесі, хоча це і не є їх ключовим профілем, адже більшість соціальних мереж в першу чергу призначена задля спілкування та розваг, хоча в них й можуть існувати різні тематичні пабліки, з різними статями, публікації та дослідження та наукові та коло наукові тематики. Для їх керування в основному використовується модель організаційного керування – у пабліків будуть адміністратори, що будуть модерувати контент та слідкувати за його частотою оновлення.

Також, більшість соціальних мереж також мають в собі вбудовані модулі рекомендаційних систем – комплексу сервісів та програм, які намагаються передбачити, які саме сутності із переліка можливих можуть бути цікавими або корисними користувачу, основуючись на існуючій інформації про нього. І хоча ключова особливість більшості подібних існуючих рішень в тому, що в тій чи іншій мірі вони використовуються суто для комерційних цілей (щоб знайти відповідь на питання, який продукт користувач забажає придбати в певний проміжок часу), при бажанні, є можливість спроектувати та розробити власний метод надання рекомендацій, який буде здатний радити користувачам певні інформаційні сутності суто в учбово-наукових цілях.

1.1. Аналіз існуючої проблеми формуванні індивідуальної освітньої траєкторії студентами

Під час навчання в університеті та здобуття вищої освіти, кожному студенту потрібно сформувати свою індивідуальну освітню траєкторію навчання, в основу якої ляжуть вибіркові освітні компоненти. Ключова ідея її ефективної реалізації базується на тому, що кожен, хто навчається, може самостійно, відповідно до своїх побажань, скласти свій розклад і обрати ті дисципліни на вибір, які найбільш сильно цікавлять його [3]. Цей процес має бути максимально зручним, а перед тим, як обрати, студент мусить дізнатися максимально детальну інформацію про них, адже, зазвичай, дисципліни на вибір становлять до третини освітньої програми.

Індивідуальна освітня траєкторія дає можливість регулювати темп освоєння навчальних програм, завдяки особистим здібностям, отриманій раніше освіті та вже вивченому матеріалу. Наприклад, студент може пройти бакалаврську програму навчання у більш короткі терміни – не за чотири, а за три роки. Це зручно для студента, хоч і підвищує навантаження на навчально - методичне управління вузу: має бути складений навчальний розклад, що враховує індивідуальні побажання кожного учня.

Свобода вибору індивідуальної траєкторії освіти, яку надає дана модель кожному студенту, можливість вже на ранніх етапах навчання неодноразово здійснювати усвідомлений вибір курсів, модулів, напрями спеціалізації проводить, як не дивно, до більшої усвідомленості та відповідальності. Поєднання широти освіти з глибиною спеціалізації дає шанс студентам стати людьми готовими до вимог, що постійно змінюються, як ринку праці, так і суспільного життя. Також, дана освітня модель базується на сучасних освітніх технологіях, для яких характерна цифрова підтримка, відкритість, мобільність, орієнтація на дослідницьку та проектну складові, гарантія зворотного зв'язку по кожному виду роботи,

накопичувальна система оцінювання досягнень, ставка на дискусійний формат, розвиток нової грамотності

Така модель навчання – безумовний плюс, адже при цьому можливості учнів, їхні інтереси та здібності використовуються та задіються максимально. Серед мінусів можна виділити досить довгий та комплексний підхід до підготовки з її реалізації, поява величезної кількості індивідуальних планів, що спричиняє великі проблеми у реалізації навчального процесу. Якщо студент не справляється з індивідуальним планом, то, як правило, його переводять на загальну освітню програму, і він продовжує вчитися в плановому режимі, разом з основним потоком.

Але, на превеликий жаль, на сьогодні, дана модель формування навчального процесу зіштовхнулася із серйозними труднощами і викликами. Як вже було сказано вище, для студентів та викладачів більшості закладів вищої освіти всі навчальні процеси були переведені в дистанційний формат. Це відобразилось на ефективності комунікаційних процесів, адже студенти стали значно менше спілкуватися між собою. Швидкість зворотного відгуку від викладачів також зменшилась, адже зараз, через технічні комунікаційні проблеми, студентам стало складніше швидко отримувати відповіді на питання, що цікавлять їх.

Ці проблеми значно погіршили ефективність та швидкість процесів формування індивідуальних освітніх траєкторій студентами. Так, згідно розповіді мого знайомого, що навчається на бакалавраті, протягом поточного року, він значно менше спілкувався з одногрупниками та викладачами, а також рідше приймав участь у житті свого університету. Як результат, коли йому було потрібно обрати дисципліни для індивідуального вивчення, він опинився в ступорі, тому що банально не розумів, що саме йому варто вибрати і порадитися було особливо нема з ким. Така ж сама, якщо навіть і не гірша ситуація і зі студентами перших курсів як бакалаврських, так і

магістерських освітніх напрямків – повна неусвідомленість у навчальних процесах, освітніх напрямках та нерозуміння, як це можна виправити.

1.2. Соціальні мережі як середовище з вирішення наявних проблем

Звичайно, керівництво університету чудово освідомлено стосовно існуючих проблем та намагається вирішити їх усіма доступними засобами. Так, для налагодження нових та покращення існуючих шляхів комунікації, використовується такий засіб як соціальні мережі із усіма можливими інструментами, що включені до їх переліку. Раніше, вони також відігравали суттєву роль в інформаційній повістці університету. Але зараз вони відіграють одну з ключових ролей в освідомленні студентів та співробітників стосовно останніх подій з життя університету.

Уважно проаналізувавши поточну ситуацію та наявну інформацію, за моєю думкою, можна виділити наступні ключові напрямки використання соціальних мереж в університетському освітньому середовищі:

1. Організація індивідуальної та групової роботи студентів – є можливість створення учбових спільнот за певними направленнями або дисциплінами задля колективної роботи та оцінки її результатів. Крім того, можна збудувати індивідуальний освітній маршрут учня або групи в залежності від їх успішності.

2. Організація дистанційного навчання. Можна використовувати її для ознайомлення з методичними напрацюваннями, вивчення лекцій, презентацій, проведенням онлайн-трансляцій з демонстрацією наукових матеріалів, відео та аудіо файлів, що також особливо актуально для студентів усіх форм навчання на сьогодні.

3. Обмін ресурсами та інформацією. З’явиться можливість зберігати, публікувати та обмінюватися інформацією між всім учасникам

освітнього процесу, а самостійна пошукова діяльність буде сприяти активізації пізнавальної діяльності.

4. Можливість безперервної освіти та самоосвіти. Безперервність освітнього процесу розуміється як тривала взаємодія між учасниками у будь - який вдалий час. Крім того, є можливість консультування щодо практичних завдань чи здійснення інформаційної підтримки тим студентам, які пропустили заняття.

5. Організація швидкої та ефективної взаємодії між учасниками. Соціальні мережі виступають як інструмент комунікації учасників освітнього процесу, як усередині, так і за межами навчальних закладів. З їх допомогою є можливість постійно бути на зв'язку. В соціальній мережі спілкування стає більш відкритим і невимушеним. Студенти можуть без страху поставити викладачу різноманітні питання. Обговорення проектів може відбуватися або у спеціальних групах, або у чатах, або у режимі відеоконференції. Викладач також має можливість обмінюватися досвідом із колегами, демонструвати результати власних розробок та обговорювати їх [4].

6. Можливість проведення різноманітних анкетувань та опитувань. Це дозволить ефективно збирати інформацію для формування статистики щодо певних вподобань користувачів, можливих зауважень та особистих думок. На їх основі можливо удосконалювати учбовий процес, реалізуючи новий функціонал в якості нових модулів до вже існуючого переліку.

1.3. Переваги та недоліки використання соціальних мереж для вдосконалення навчального процесу

Серед ключових переваг використання соціальних мереж для вдосконалення навчального процесу можна виділити наступні:

- миттєва передача інформації, доступність звітів та завдань для вивчення;
- увесь навчальний матеріал доступний одночасно викладачеві та студенту, що вирішує проблему резервного копіювання та відсутності проблем із втратою електронної версії звітів;
- спрощення для викладача таких процедур, як інформування студентів, надання методичних посібників, доведення до відома завдань, прийом робіт в електронному вигляді для перевірки;
- можливість зворотного зв’язку та оцінки якості навчання.

Серед недоліків можна позначити наступні:

- проблеми з можливим етикетом для учасників – через це з’являється потрібність у постійній модерації чатів;
- занадто активна комунікація, великий потік інформації та різноманітність розважального контенту можуть сповільнювати або навіть відволікати від навчального процесу;
- достатньо високий рівень витрат для організації підтримки навчального процесу для викладачів.

1.4. Опис предметної області

Слід зауважити, що у контексті виконання даної магістерської роботи не буде розглядатися процес проектування та розробки повноцінної екосистеми для покращення процесів комунікації під час дистанційного навчання, адже під час виконання бакалаврської кваліфікаційної роботи вже був успішно розроблений функціонуючий прототип внутрішньо університетської соціальної мережі для комунікації в освітньому процесі.

В даній магістерській роботі буде сконцентровано увагу на рекомендаційних системах – допоміжних підсистемах, які дозволяють

користувачеві зробити свій вибір серед великого переліку найрізноманітнішого контенту, від музики і фільмів до пабліків та груп на несхожі тематики [5]. Дуже часто, в рамках екосистем соціальних мереж розробники проектують такі інструменти.

Але, зазвичай, вони вслуговують суто для комерційних цілей – задля того, щоб знайти відповідь на питання, яка сутність зацікавить користувача в певний проміжок часу. Так, з боку користувача соціальної мережі, розглядається питання, наскільки сильно він схвалює ти чи інші пропозиції, що потрапили до списку рекомендацій. Показники виміру даного питання це якість апроксимації оцінки та відсоток користувачів, які обрали певну сутність із списку рекомендованих. З боку системи, стоїть мета досягнення максимального прибутку з користувача, а також найбільший ступінь перегляду сутностей або послуг, що рекомендуються (чим далі від списку до списку сутностей, тим краще).

Метою виконання даної роботи є розробка рекомендаційної системи, що буде допомагати здобувачам вищої освіти формувати індивідуальну освітню траєкторію в умовах повністю дистанційного навчання.

1.5. Огляд існуючих систем-аналогів для формування освітньої траєкторії студента

Як було зазначено вище, для ефективного налагодження процесу дистанційного процесу навчальними закладами найчастіше використовуються, хоч і не в повну силу, вже готові інструменти, такі як рекомендаційні системи на базі існуючих соціальних мереж для внутрішніх потреб. І це зовсім не дивно, адже вже багато хто висказався за у користь використання цих інструментів у подібних цілях.

Так. в дослідженнях Р. Ших та Е. Вебб [6] продемонстровано, що коректне використання соціальних мереж та побудова у цьому середовищі ланцюжка взаємодії «викладач – той, хто навчається», може позитивно вплинути на рівень залучення студентів до навчального процесу. К. Джонс та Б. Чхао [7] стверджують, що якщо комунікація створена правильно, навчальні матеріали подаються у ненав'язливій формі, а викладач є грамотним ІТ-користувачем, то самі студенти визначають, що подібна взаємодія є вкрай дійсною, а академічні результати успішності підтверджують позитивний ефект. А в роботі [8] виконано огляд області рекомендаційних систем і описане поточне покоління методів рекомендацій, які зазвичай класифікуються на наступні три основні категорії: підходи до рекомендацій на основі контенту, підходи до співпраці та гібридні рекомендації. У цьому документі також описуються різні обмеження поточних методів рекомендацій і обговорюються можливі розширення, які можуть покращити можливості рекомендацій і зробити системи рекомендацій застосовними до ще ширшого кола застосувань

Розглянемо та проаналізуємо приклади з використання кожного з цих інструментів більш детально.

1.5.1. Соціальна мережа Facebook

Однією з найпопулярніших соціальних мереж, що активно використовується і в тому числі для навчального процесу виступає Facebook. [9] Він був створений аж 4 лютого 2004 року та спочатку зародився як сайт для спілкування студентів виключно Гарвардського університету. Засновником сервісу є Марк Цукерберг, який створив першу версію майбутнього додатку, знаходячись у стінах гуртожитку університету. Ключовою відмінністю Facebook від соціальних мереж, що

існували тоді, стала саме можливість контакту: Цукерберг запропонував людям простий і зручний спосіб обмінюватися інформацією один про одного. Саме цей фактор, на думку багатьох, зіграв вирішальну роль у стіль швидкій та масовій популяризації додатку серед Інтернет-користувачів.

Спочатку, сайт називався Thefacebook і аж до вересня 2005 року був доступний тільки для студентів Гарвардського університету, потім реєстрацію відкрили для інших університетів Бостона, а потім і для студентів будь-яких навчальних закладів США, які мають електронну адресу в домені .edu. Починаючи з вересня 2006 року, сайт доступний для всіх користувачів Інтернету віком від 13 років, які мають адресу електронної пошти.

Facebook входить до п'ятірки найбільш відвідуваних веб-сайтів світу. На червень 2017 року його активна аудиторія становила 2 мільярди користувачів. Добова активна аудиторія тоді складала 720 мільйонів осіб. Близько 35 відсотків користувачів складають молоді люди в проміжку між 18 та 40 роками, тобто студенти різних навчальних закладів також входять до їх числа. Станом на 2022 рік, в Україні за популярністю він посідає почесне третє місце [10]. Згідно дослідженням вчених, сучасний Інтернет-користувач знаходиться онлайн близько 7 годин на добу, з них близько половини –в соціальних мережах. Саме тому, будь-який університет має свою сторінку або групу (або обидві одночасно) в даній соціальні мережі.

Наш університет не став винятком – він має власні сторінки і групи (пабліки). Перші створюється з метою просування різних проектів, брендів та інших організацій у системі пошукового рейтингу (SEO) на якомога вищі позиції. Це робиться для завоювання більшого кола аудиторії. Функціонал цього облікового запису дозволяє залучати відвідувачів за допомогою численних інструментів, які часто поповнюються. Загальна її структура на прикладі інформаційної сутності “Сторінка Студент ХНУРЕ” зображена на рисунку 1.1.

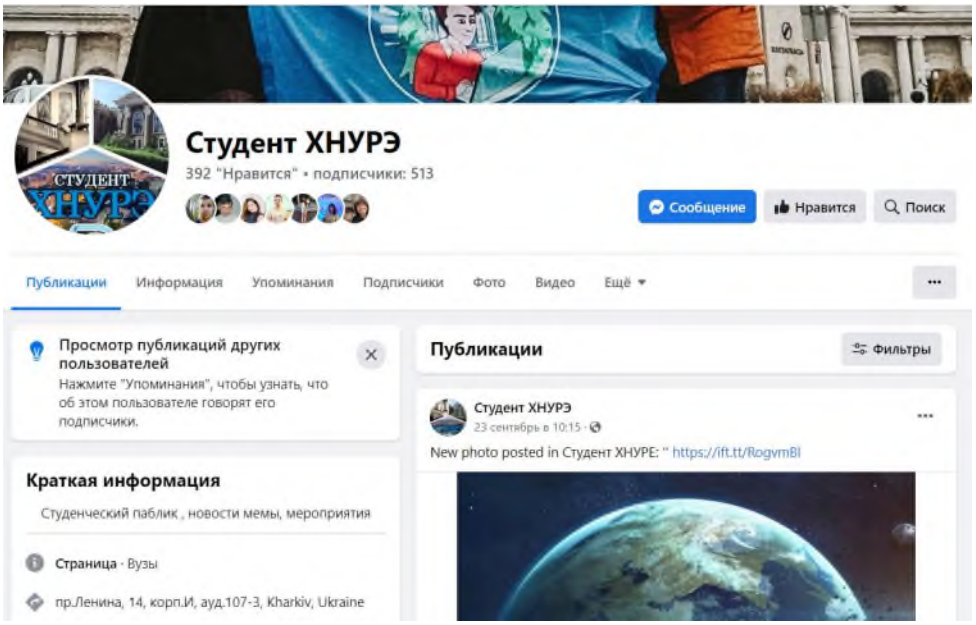

Рисунок 1.1 – Зовнішній вигляд ІС “Сторінка Студент ХНУРЕ”

Групи або пабліки, являють собою спільноти, за допомогою яких можна об'єднувати користувачів мережі за певними захопленнями та інтересами, вони створені для тематичних обговорень всередині групи. Загальна її структура на прикладі інформаційної сутності “Група ХНУРЕ-новини” зображена на рисунку 1.2.

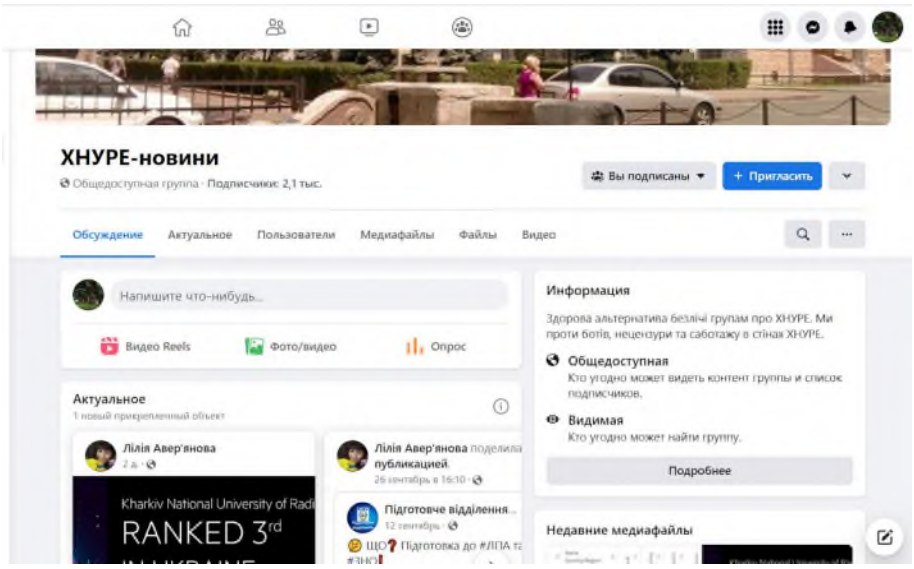

Рисунок 1.2 – Зовнішній вигляд ІС “Група ХНУРЕ - новини”

Обидві сутності роблять ключовий акцент на комунікаційних процесах – формують стрічку з різноманітними новинами та подіями, які трапились в житті закладу, дозволяють оцінювати новини, коментувати їх, вносити свої пропозиції, тощо.

1.5.2. Рекомендаційні системи

Рекомендаційні системи мають широку сферу застосування та призначення, тому їх можна поділяти на групи за різними параметрами: предметом, метою, джерелом рекомендації, користувальницьким контекстом і ступенем персоналізації. Наприклад, мета систем, які рекомендують фільми, книги, новини та інший контент – залучити користувача до споживання нової інформації на конкретній платформі. В цьому випадку пропозиція контенту базується не тільки на контексті дій самого користувача, але й на новизні інформації, оцінках експертів та популярності в інших користувачів. Інакше працюють системи, які радять придбати сутності повсякденного попиту. Їхнє завдання – не тільки запропонувати нові продукти, виходячи з контексту дій користувачів, а й нагадати про сутності, що часто купуються, за наявності історії покупок.

Facebook в цьому плані також не залишилась осторонь. Власна система платформи здатна визначити, які пости та в якому порядку показувати у стрічці новин. Ці прогнози ґрунтуються на тому, на кого підписано людину, чому ставив позначки реакції і з ким нещодавно спілкувався. Щоб розвінчати міфи навколо системи рекомендацій, Facebook опублікував докладну інформацію про те, як вона працює [11]. Принцип роботи даної системи порад зображений на рисунку 1.3.

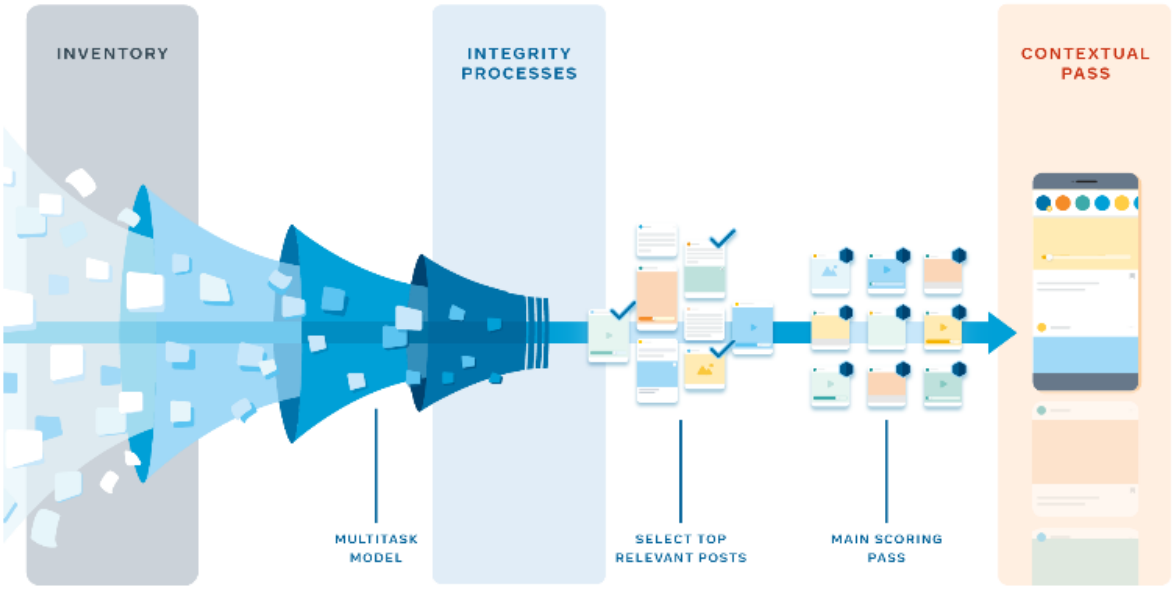

Рисунок 1.3 – Принцип роботи рекомендаційної системи мережі Facebook

Щоб вирішити, які з постів мають відображатися вище у стрічці новин користувача, потрібно передбачити, що для нього найважливіше. Для цього платформа може використовувати характеристики публікації, наприклад, хто відмічений на фотографії і коли вона була опублікована, щоб передбачити, чи вона сподобається користувачеві.

Можна привести наступний приклад: друг опублікував фото собаки, ще один користувач – відео з пробіжки, сторінка, на яку підписаний користувач – статтю про те, як краще побачити Чумацький Шлях, а кулінарна група – нові рецепти.

Якщо користувач часто переписується з певним користувачем або часто залишає коментарі до його публікацій, а його новий ролик з'явився нещодавно після спілкування в стрічці, висока ймовірність того, що пост від цього користувача йому сподобається. Якщо у минулому наш абстрактний користувач дивився більше відео, то його з меншою ймовірністю зацікавить фото собаки. У цьому випадку алгоритм ранжування Facebook надасть відео з пробіжки вище, ніж фото собаки.

Щоб звузити вибірку, система збирає всі пости-кандидати, які може ранжувати для користувача. Після цього система оцінює кожну одиницю контенту за його типом і схожістю з контентом, з яким зазвичай взаємодіє користувач. Щоб дати оцінку тисяч постів кожного з 2 млрд користувачів в реальному часі, ці моделі запускаються для всіх публікацій паралельно за допомогою нейронних мереж.

Далі система звужує коло кандидатів приблизно до 500 найбільш відповідних публікацій. Йде основний етап підрахунку очок, у якому відбувається переважна більшість персоналізації. Оцінка кожної публікації розраховується незалежно, і всі 500 постів вибудовуються за кількістю оцінок. Для деяких публікацій оцінка може бути вищою відмітки реакції, ніж за коментарі, оскільки комусь більше подобається висловлювати свою симпатію реакціями, ніж коментарями.

Після цього запускається контекстний фільтр де спрацьовують функції аналізу різноманітності типів контенту. Вони повинні перевірити, що у стрічці новин з'явиться різноманітний контент, і система не підсуне користувачеві кілька відеороликів поспіль.

1.6. Актуальність та мета розробки

З вищенаведеної інформації стає зрозуміло, що в рамках покращення дистанційного учбового процесу загальний метод реалізації не зовсім підходить. Саме тому, в рамках виконання даної дипломної роботи, за допомогою одного або декількох вищевказаних відповідних математичних методів або їх модифікацій, потрібно спроектувати власний математичний метод, що буде за конкретними параметрами формувати необхідний перелік рекомендацій з вибору певних освітніх напрямків.

В якості джерела отримання вхідного набора даних для роботи, практичного тестування та подальшої імплементації системи, було обрано

вже існуючий та коректно функціонуючий прототип веб-додатку університетської соціальної мережі для студентів та співробітників, який вже був розроблений в рамках виконання бакалаврської роботи [12].

Даний інструмент здатний надати повний базовий функціональний набір можливостей для повсякденного користування в учбових цілях, а саме: швидка, зручна та ефективна комунікація між учасниками, створення постів та записів, можливість проведення онлайн-лекцій та занять, вбудована система опитувань та анкетувань, оцінювання результатів робот студентів, тощо. Саме він зможе надати нам усе необхідне для простої, зручної та ефективної розробки рекомендаційного системи для вибору освітніх напрямків.

1.7. Ключові методи з реалізації рекомендаційних систем

Можна виділити наступні базові математичні методи, а також їх модифікації, що в тій чи іншій мірі використовуються при реалізації рекомендаційних систем:

- Summary-based Methods (неперсональні методи);
- Content-based Methods (моделі, засновані на описі сутності);
- Collaborative Filtering Methods (метод колаборативної фільтрації);
- Content Filtering Methods (метод контентної фільтрації);
- Matrix Factorization Methods (методи, засновані на матричному розкладі);
- Hybrid Filtration Methods (гібридні методи фільтрації).

Загальна мета рекомендаційних систем складається в пропозиції користувачеві певних сутностей, якими він раніше не цікавився, але які можуть виявитися корисними чи цікавими для нього. Узагальнений принцип роботи даної системи зображений на рисунку 1.4.

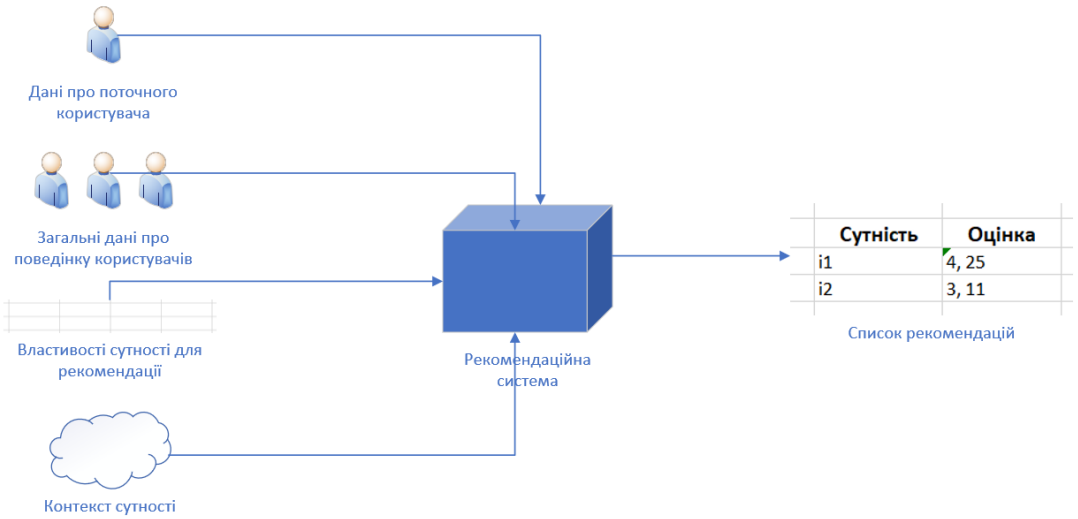

Рисунок 1.4 – Загальний принцип роботи рекомендаційної системи

Рекомендаційна система для надання порад в тій чи іншій сфері приймає на вході дані про поточного користувача, про поведінку всіх користувачів в цілому, про властивості об’єктів, що рекомендуються, і про контекст поточного інтересу користувача. А після їх обробки, на виході отримуємо сутність з певною оцінкою і чим остання більше, тим вище її пріоритет в рекомендаційній системі.

1.8. Опис можливих проблем та способів їх вирішення

Під час розробки рекомендаційних систем, досить часто виникає проблема «холодного старту» (Cold-start Problem, CSP), яка добре розписана у науковій роботі [13]. Вона виникає тоді, коли, наприклад, у системі, що пропонує «користувачам» деякі визначені «об’єкти» на підставі наявної історії переваг, з’являються нові елементи – ними можуть виступати нові користувачі, з порожньою історією переваг. У багатьох подібних ситуаціях CSP часто набуває характеру циклічних проблем для вже відомих користувачів або об'єктів – наприклад, у випадках, коли частина

користувачів з'являється в системі рідко, змінює свої інтереси (в системі переважає так звана волантильність користувачів) – так звана проблема постійного холодного старту (Continuous Cold-start Problem, CoCoS).

Вирішення задачі CSP визначається типом рекомендованих об'єктів – наприклад, широко відомі практики використання базових наборів рекомендацій та використання гібридних систем, що поєднують у собі комбінації колаборативної фільтрації та контентних рекомендацій (поява яких загалом очевидна у випадках, коли об'єктам можна призначити описові дескриптори) [14]. При доступі до сторонніх даних поширені практики отримання допоміжної інформації, наприклад, із соціальних мереж. Однак усі ці способи не підходять для повного вирішення проблеми CoCoS, оскільки припускають, що після того, як користувач став «відомим», він залишається таким необмежену кількість часу.

Найбільш популярний підхід при вирішенні завдань CoCoS – це комбінування спільної фільтрації та контентних методів за умови, що «похожим» користувачам пропонуються «схожі об’єкти». Цей спосіб чудово вирішує проблему Item Continuous Cold-Start. Однак для випадку User Continuous Cold-Start, цей спосіб не підходить, адже в цьому випадку «схожих» користувачів визначити неможливо, і він стає безсильним. В цьому випадку, краще застосувати інші методи, наприклад, метод «контекстного багаторукого бандита» [15] і метод для вирішення проблем користувацької волантильності [16]. Також можна відмітити метод на основі тем [17]. Даний метод моделює наміри користувача через ймовірне розподіл тем над об'єктами, а при взаємодії користувача з системою наслідки користувача передбачаються через марківський процес. Даний метод був апробований на реальних даних із системи електронної комерції, експерименти показали істотне підвищення ефективності системи рекомендації.

Також, існує ще одна проблема, властива великим рекомендаційним системам і пов'язана вона з їх масштабованістю. Базові методи добре працюють із порівняно невеликими обсягами даних, але зі збільшенням цих наборів отримання результатів на колишньому рівні якості за їх допомогою може стати проблемою. У разі обробки без постійного оновлення вхідних даних, це може не становити великої проблеми, але для сценаріїв реального часу необхідні спеціалізовані підходи.

Деякі проблеми породжують необхідність дотримання правил конфіденційності. Рекомендаційні алгоритми здатні розпізнавати різні закономірності, про існування яких користувачі можуть навіть не підозрювати.

1.9. Постановка задачі

Розробка рекомендаційної системи на базі гібридного методу мусить складатися з наступних задач:

- проведення аналізу вимог до обраної предметної області та її створення;
- виконання аналізу технологій, методів та алгоритмів, що будуть використовуватися при розробці;
- вибору та опису архітектуру програмного засобу;
- обґрунтування вибору СУБД та мови програмування;
- проектування та опису фізичної схеми баз даних додатку;
- аналізу вимоги до вхідних та вихідних даних системи;
- практичної реалізації системи;
- аналізу майбутніх перспектив із вдосконалення принципу роботи розробленого рішення та загального процесу дистанційного навчання за його допомоги.

2 ОПИС ОБРАНИХ ТЕХНОЛОГІЙ ТА МЕТОДІВ ПРОЕКТУВАННЯ Й РЕАЛІЗАЦІЇ СИСТЕМИ

2.1. Технології та методи реалізації рекомендаційної системи

Для теоретичного дослідження та практичної реалізації була обрана предметна область у вигляді розробки рекомендаційної системи з формування освітньої траєкторії на прикладі студентів кафедри Системотехніки ХНУРЕ. Дана система буде розроблена на гібридному методі, в основу якого ляжуть більш дрібні підсистеми, що будуть засновані на класичних допоміжних методах колаборативної фільтрації. Їх актуальність обумовлена простотою у проектуванні та високою ефективністю при правильному використанні. Розглянемо їх більш детально.

2.2. Метод колаборативної фільтрації

Колаборативна фільтрація (Collaborative filtering) – це метод рекомендації, при якому аналізується лише реакція користувачів на об'єкти: оцінки, що користувачі надають об'єктам. Оцінки можуть бути як явними (користувач явно вказує, на скільки «зірочок» він здатен оцінити об'єкт або якими параметрами він користується при голосуванні під час вибору об’єкту), так і неявними (наприклад, кількість переглядів одного ролика). Чим більше оцінок збирається, тим точніше вийдуть рекомендації. Як результат, користувачі формують один для одного перелік відфільтрованих об'єктів. Сам тому цей метод називається також спільною фільтрацією.

Під час розробки рекомендаційних систем за допомогою методів колаборативної фільтрації, дослідники виділяють два ключові підходи: колаборативна фільтрація, заснована на користувачах (User-based Collaborative filtering) та сумісна колаборативна фільтрація за елементами

(Item-based Collaborative filtering). Перший за основу бере набір чисельних параметрів з масиву даних про користувачів, що обрали ту чи іншу сутність та на їх основі обчислює, яку сутність рекомендувати. А другий підхід бере сам набір сутностей та на основі їх параметрів, наприклад, оцінки їх якості, проводить обчислення [18].

Ключовими перевагами під час розробки рекомендаційних систем на базі методу колаборативної фільтрації являються низькі вимоги до різноманітності початкового набору даних та простоту в реалізації програми за базовими версіями.

До недоліків розробки систем на базі цього методу можна віднести проблему «холодного старту», що виникає при додаванні нових користувачів і об'єктів для вибору [15], потребу у постійному отриманні актуальної інформації щодо взаємодії користувача з системою загалом та проведенні аналізу на основі множинуі показників активності (скільки часу було проведено користувачем за тим чи іншим видом діяльності, скільки людей переглянули певний контент, тощо), а також необхідність у об’ємній вибірці даних про різних користувачів.

Також, завелика кількість доступних даних може ускладнювати реалізацію системи рекомендації за цим методом. Наприклад, поведінка деяких користувачів добре піддається моделюванню, але інші користувачі не демонструють сталої поведінки. Велике кількість таких користувачів може призводити до зниження точності результатів рекомендаційної системи та до зменшення її ефективності. Крім цього, користувачі можуть задіяти рекомендаційну систему для підвищення значущості одного сутності щодо іншого сутності — до наприклад, за допомогою відправки позитивних відгуків про один сутність і негативних відгуків про його конкурентів. Добре спроектована рекомендаційна система має справлятися з цими проблемами.

2.2.1. Метод колаборативної фільтрації, що базується на користувачах (User-based Collaborative filtering)

Загалом, принцип роботи даного підходу можна описати наступним чином. У нас є вибірка даних, що містить в собі користувачів (ними виступають студенти) та об’єкти для рекомендації – в їх ролі знаходяться дисципліни, що були обрані конкретними студентами певну кількість раз.

Спочатку, для кожного користувача А потрібно обчислити, наскільки його інтереси збігаються з інтересами користувача В. Це можна зробити у декілька етапів. Спочатку, потрібно використати формулу векторів ідентичності для знаходження значень, що визначають схожість між одним та іншим користувачем на основі дисциплін, які були обрані обома ними [20].

Існує множина способів підрахунку близькості векторів, але найбільш розповсюдженою формулою є:

$$sim(x,y) = \cos(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{||\vec{x}||_2 \times ||\vec{y}||_2} = \frac{\sum\limits_{s\in S_{xy}} r_{x,s} r_{y,s}}{\sqrt{\sum\limits_{s\in S_{xy}} r_{x,s}^2}\sqrt{\sum\limits_{s\in S_{xy}} r_{y,s}^2}} \quad (2.1)$$

Далі, нам необхідно розрахувати косинус між отриманими векторами. Формула розрахунку виглядає наступним чином:

$$sim(u,a) = \frac{\sum_{i=1}^{m} r_{a,i}\cdot r_{u,i}}{\sqrt{\sum_{i=1}^{m} r_{a,i}}\cdot\sqrt{\sum_{i=1}^{m} r_{u,i}}} \quad (2.2)$$

У даній формулі sim(u,a) – міра близькості (схожості) користувачів a и u. ru,i – значення матриці R: u строка, i стовпець. sim(u,a) приймає значення з

відрізка [0,1]. Якщо користувач не вказав оцінку для якогось об'єкта, відповідне значення матриці дорівнює 0.

Існує ще один спосіб розрахувати міру схожості між користувачем і всіма іншими параметрами, якими можуть виступати, наприклад куплені сутності або прочитані статті. Ним виступає коефіцієнт кореляції Пірсона. Він розраховує лінійну залежність між двома об'єктами (користувачами) як їх параметрів. Але цей алгоритм не обчислює міру близькості по всьому набору об'єктів. Цей набір потрібний заздалегідь обробити, відфільтрував за зменшенням показника близькості (високорівневий показник схожості). Формула його знаходження виглядає наступним чином:

$$sim(u,a) = \frac{\sum_{i \in I}(r_{a,i} - \overline{r_a})(r_{u,i} - \overline{r_u})}{\sqrt{\sum_{i \in I}(r_{a,i} - \overline{r_a})^2 \cdot \sum_{i \in I}(r_{u,i} - \overline{r_u})^2}} \quad (2.3)$$

Тут I – множину об'єктів, які оцінив як користувач a, так і користувач (добавить формулу буквами) r_a та r_u – це середні оцінки користувачів a та u, відповідно.

У чисельнику підраховується множення відхилень оцінок двох користувачів від середніх значень одного об'єкта. Знаменник необхідний для того, щоб дана величина набувала значень з відрізка [-1,1]. Чим сильніше збігаються інтереси, тим ближче значення близькості до 1. Якщо коефіцієнт кореляції Пірсона негативний, то інтереси користувачів до певної досліджуваної сутності прямо протилежні.

Розрахунок косинусу та коефіцієнту кореляції Пірсона між векторами-рядками користувачів є найрозповсюдженими та найбільш ефективними мірами близькості, що використовуються у цьому методі. Коефіцієнт кореляції Пірсона буде залишатися високим, якщо всі оцінки користувачів, що порівнюються, відрізняються на постійну величину – один користувач дає

одній сутності більш високий пріоритет у виборі, інший же надає їй більш низький. У цьому випадку косинус зможе надати менше значення, але якщо пріоритетні оцінки двох користувачів відрізняються пропорцією, то косинус буде більше, а кореляція – менше. В обох випадках пріоритети користувачів будуть приблизно схожі. Тому, яку з цих метрик використати залежить від конкретного завдання. Після цього виберемо множину користувачів, найбільш близьких до a та передбачимо оцінку на основі оцінок об'єкта і "сусідами" попереднього кроку.

Далі нам потрібно обрати множину користувачів, що є найбільш близькими до a. Для цього потрібно обрати множину K найбільш схожих на a користувачів. Можна обрати всіх користувачів. Але оскільки користувачів з несхожими або інтересами, що незначно перетинаються, досить багато, то вони будуть негативно впливати на точність прогнозування оцінки для об'єкта I. Крім того, кількість користувачів впливає на обсяг обчислень на третьому етапі.

Одне з можливих рішень – встановити поріг міри близькості, обчисленої на першому етап за допомогою формули розрахунку косинусу та коефіцієнту кореляції Пірсона. До множини K увійдуть лише ті користувачі, у яких міра близькості перевищує поріг. Найчастіше усього, обирається ціла константа k. Потім усі користувачі сортуються за зменшенням міри близькості. І у множині K входять k користувачів, найближчих до a.

І нарешті, передбачимо оцінку на основі оцінок об'єкта і "сусідами" на попереднього кроці. Маючи множину K близьких користувачів, потрібно обчислити оцінку, яку поставив би користувач a об'єкту I. Нагадаємо, що розглядаються лише користувачі, які оцінили об'єкт I. [21]

Передбачувана оцінка користувача A для об’єкту I обчислюється за наступною формулою:

$$p_{a,i} = \overline{r_a} + \frac{\sum_{u \in K} (r_{u,i} - \overline{r_u}) \times sim(a,u)}{\sum_{u \in K} | sim(a,u) |} \quad (2.4)$$

За основу береться його середня оцінка r_a, а потім додається середнє відхилення оцінки інших користувачів із множини K для об'єкту I від їх середньої оцінки. Чим ближче користувач u до користувача a (за мірою близькості sim(a,u), яка була обчислена на першому кроці), тим більше його внесок у передбачення оцінки.

На основі розрахованих значень, ми формуємо матрицю пріоритетних оцінок, в якій в якості строк виступають користувачі, що поставили оцінки, а зверху – об’єкти, що оцінюються. Для цього нам необхідно виконати транспонування матриці – це дозволить алгоритму отримати певну залежність сутностей від людей задля перевірки її на розрідженість. Приклад матриці зображений на рисунку 2.1.

	1	*2*	...	*i*	...	*m*
1	5	3		1	2	
2		2				4
:			5			
u	3	4		2	1	
:					4	
n			3	2		
a	3	5		?	1	

Рисунок 2.1 – Приклад матриці пріоритетних оцінок

Таким чином, описаний алгоритм передбачає оцінки для об'єктів, які поточний користувач ще оцінив. Для того, щоб зробити рекомендацію для

даного користувача, достатньо передбачити оцінки для всіх неоціненних об'єктів та вибрати об'єкти з найбільшою передбаченою оцінкою. [22]

2.2.2. Метод колаборативної фільтрації, що базується на рекомендуємих сутностях (Item-based Collaborative filtering)

На відміну від вищенаведеного алгоритму, який ґрунтується на порівнянні між собою користувачів, даний алгоритм замість користувачів виконує порівняння рекомендуємих об’єктів. В даному випадку його принцип роботи майже такий самий та включає в себе наступні кроки:

1) для кожного об'єкта j обчислимо, наскільки він схожий на об'єкт i, для якого передбачається оцінка;
2) виберемо множину об'єктів, найближчих до i;
3) передбачимо оцінку на основі оцінок обраних на другому крок об'єктів користувачем a;

Об'єкти на першому кроці порівнюються так само, як і в першому методі - за допомогою формул розрахунку косинусу та коефіцієнту кореляції Пірсона. Формула порівняння об’єктів для у випадку роботи методу сумісної колаборативної фільтрації за елементами зображена нижче:

$$sim(i,j) = \frac{\sum_{u \in U} (r_{u,i} - \overline{r_i}) \cdot (r_{u,j} - \overline{r_j})}{\sqrt{\sum_{u \in U} (r_{u,i} - \overline{r_i})^2 \cdot \sum_{u \in U} (r_{u,j} - \overline{r_j})^2}} \quad (2.5)$$

Тут підсумовування ведеться по множинуі U користувачів, які оцінили об'єкти i та j, r_i – середня оцінка об’єкту i.

Другий крок аналогічний до другого кроку попереднього алгоритму. Ключова відмінність у тому, що в ролі K виступає множина близьких до i

об'єктів. На третьому етапі розраховується виважена середня оцінка схожих об'єктів, отримана користувачем a – формула розрахунку наведена нижче:

$$p_{a,i} = \frac{\sum_{j \in K} r_{u,j} \times sim(i,j)}{\sum_{j \in K} |sim(i,j)|} \tag{2.7}$$

У випадку третього кроку, в алгоритмі, заснованому на порівнянні об'єктів, використовувалася більш складна формула. Це пов’язано з тим, що середня оцінка об'єкта демонструє, наскільки добрим є об'єкт, а середня оцінка користувач показує, наскільки суворо користувач оцінює об'єкти. Формула (3) обчислює оцінку з урахуванням того, що строгість оцінки різних користувачів відрізняється.

Алгоритм, заснований на порівнянні об'єктів, на відміну від свого попередника, має кілька вагомих переваг. Оскільки інтереси користувачів часто змінюються, а середня оцінка об'єкта залишається постійною, то при порівнянні об'єктів достатньо порахувати міру близькості кожного з кожним та зберегти в пам'яті. У частому оновленні цих даних немає потреби.

У сучасних системах рекомендацій постійно зростає кількість користувачів та об'єктів, тому настає момент, коли необхідно використовувати розподілену архітектуру системи При цьому виникає проблема зберігання даних: було б непогано схожих користувачів чи схожі об'єкти зберігати поряд. Для її вирішення можна застосувати алгоритми кластеризації для користувачів або об'єктів, використовуючи міру близькості між ними: (2.1), (2.2), (2.4). Розбиття користувачів або об'єктів на групи дає ще одну перевагу: так як у кластері знаходяться схожі об'єкти, то потрібний об'єкт достатньо на першому кроці порівнювати з об'єктами зі свого кластера, що значно скорочує кількість обчислень. [23]

Описані вище методи колаборативної фільтрації називаються методами, заснованими на роботі з пам'яттю, тому що необхідно зберігати матрицю R. Існує інша група алгоритмів – методи, що базуються на моделях. Ідея полягає в тому, що маючи матрицю R можна побудувати модель, яка буде менше за обсягом пам'яті, що займається. Використовуючи отриману модель, можна обчислювати рекомендації. Прикладами таких моделей є Байєсовський алгоритм, у якому вихідне завдання зводиться до завдання класифікації; Марківські моделі, які дозволяють враховувати не тільки схожість об'єктів, але та їх цінність; а також інші моделі. [24]

2.3. Гібридні методи фільтрації

Як було зазначено вище, будь-яка рекомендаційна система, що розроблена на одному з базових методів, має свої суттєві недоліки, що звужує область його використання та знижує якість остаточних результатів. Цю проблему здатні вирішити великі системи рекомендацій, які використовують різні дані про об'єкти, користувачів, зв'язки між ними та являють собою складну гібридну систему. Загалом, виділяють три типи гібридних систем, кожна з яких буде розглянута більш детально:

- з монолітною організацією;
- з паралельною організацією;
- з конвеєрною організацією.

Кожна з цих систем може являти собою поєднання абсолютно різних підходів і алгоритмів, таких як: метод колаборативної фільтрації, аналізу вмісту, аналізу соціальних зв'язків та зв'язків між об'єктами, і так далі та має свій підхід до розробки [25]. Розглянемо кожний з цих підходів більш детально.

2.3.1. Монолітна організація

На відміну від паралельної та конвеєрних організацій гібридних систем, системи з монолітною організацією являють собою один модуль, що включає різні методи обробки наявних даних. Принцип її роботи зображений на рисунку 2.8.

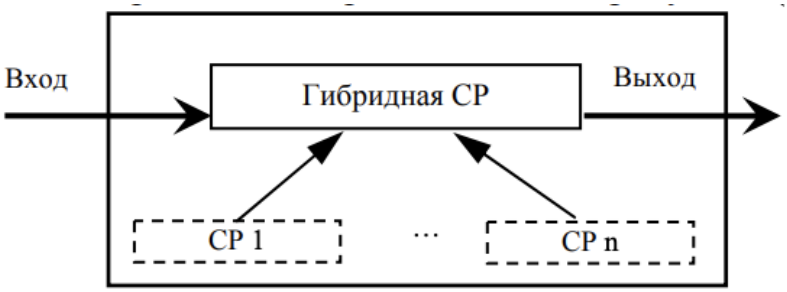

Рисунок 2.2 – Монолітна організація гібридної рекомендаційної системи

Більшість гібридних систем такого роду реалізують колаборативну фільтрацію, посилену особливостями вмісту. Наприклад, якщо алгоритм колаборативної фільтрації рекомендує деякому користувачеві кілька книг, то серед них можна вибрати найбільше придатну за допомогою порівняння жанрів цих книг із популярними жанрами цього користувача. Тобто. якщо користувач віддає перевагу фантастиці (що легко дізнатися, переглянувши жанри всіх оцінених/прочитаних користувачем книг або його профіль), то серед книг двох жанрів: історія та фантастика, система пропонує другий варіант.

Багато систем використовують систему псевдо оцінки об'єктів користувачами на основі прогнозу оцінки на основі аналізу вмісту. Іноді користувач запитує рекомендації об'єктів, що задовольняють деяким критеріям. Тут уже можуть використовуватися методи на знаннях. Але,

поєднання систем, які використовують бази знань з колаборативними методами, доки не вивчено та практично майже не використовується.

2.3.2. Паралельна організація

Принцип роботи гібридної системи з паралельною організацією виглядає наступним чином.

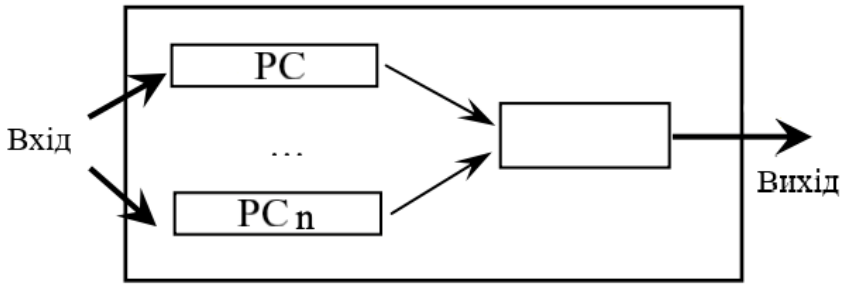

Рисунок 2.3 – Паралельна організація гібридної рекомендаційної системи

Вихідні дані надходять кожний модуль системи. Після того, як усі підсистеми зможуть надати відповідь, починає працювати модуль, що агрегує. Є три основні підходи у виборі множини найбільш релевантних рекомендацій.

Перший підхід – змішати всі рекомендації та надати їх користувачу. Формула обчислення наведена нижче.

$$rec(u) = \bigcup_{k=1}^{n} rec_k(u) \tag{2.8}$$

У ній rec(u) – результат роботи «агрегатора» для користувача u, $rec_k(u)$ – результат роботи k-ого модуля системи користувача u.

Другий підхід – для кожного об'єкта підрахувати суму рейтингів, отриманих паралельними модулями та вивести об'єкти з найбільшим рейтингом:

$$r(u,i) = \sum_{k=1}^{n} \beta_k \cdot r_k(u,i) \tag{2.10}$$

Тут r(u,i) – потрібна прогнозована оцінка об'єкта і користувачем u, r_k(u,i) – оцінка, отримана k-м модулем, β_k - вагові коефіцієнти.

І, нарешті, третій підхід – вибрати одну відповідну систему рекомендації, реалізовану в одному з модулів для кожної конкретної ситуації. Наприклад, якщо модуль колаборативної фільтрації не дає відповіді через брак даних (проблема «холодного старту»), система вибирає метод аналізу вмісту.

2.3.3. Конвеєрна організація

Схема гібридної системи з конвеєрною організацією представлена на рисунку 2.4.

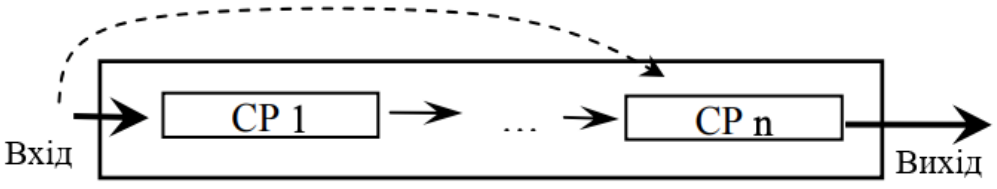

Рисунок 2.4 – Конвеєрна організація гібридної рекомендаційної системи

Під час даної реалізації, завдання з формування рекомендацій поділяється на етапи, що виконуються послідовно. Результат роботи чергового етапу використовується в наступних. І так до тих пір, поки рекомендація не буде сформована.

2.4. Опис ключових алгоритмів роботи рекомендаційних систем

Нижче наведений стислий опис найбільш розповсюджених алгоритмів, що використовуються по роздільності або в сумісності під час реалізації рекомендаційних систем. В нашому гібридному алгоритмі також будуть в тому чи іншому виді використовуватися ці алгоритми або їх ключові складові.

2.4.1. Алгоритми кластеризації

Алгоритми кластеризації виявляють структуру в рядах випадкових даних – тих, що без розпізнавальних міток. Вони засновані на виявлення подібності між об'єктами (наприклад, між покупцями інтернет магазину) за допомогою обчислення їх відстані від інших елементів у просторі ознак – кількості та якості сутностей, що купуються в одній транзакції. Кількість незалежних ознак визначає розмірність простору ознак. Якщо елементи подібні один до одного, то вони поєднуються в кластер.

Найпростішим із існуючих алгоритмів кластеризації є алгоритм k-середніх (k-means), який поділяє елементи на кластерів k. Спочатку елементи розподіляються за цими кластерами у довільному порядку. Потім кожного кластера обчислюється центр мас (чи навіть центр) як функція його членів. Потім порівнюється відстань кожного елемента кластера від цього кластера. Якщо елемент 14 виявляється ближче до іншого кластера, він переходить у цей кластер. Після перевірки всіх відстаней всім членів центри кластерів

обчислюються заново. Як тільки досягається стійкий стан (за результатом перевірки елементи не були переміщені) набір вважається належним чином кластеризованим, а потім алгоритм закінчує свою роботу.

Обчислення відстані між двома об'єктами може бути важким для візуалізації. Для вирішення цієї проблеми слід розглядати кожен елемент кластера як багатовимірний масив і обчислювати йому так звану евклідову відстань.

Також існують інші різновиди кластеризації, такі як: теорія адаптивного резонансу (Adaptive Resonance Theory), нечітка кластеризація методом С-середніх (Fuzzy C-means), ймовірнісна кластеризація за допомогою ЕМ-алгоритму (Expectation-Maximization) і т.д.

Загалом, в рекомендаційних механізмах може застосовуватися множину базових алгоритмів та їх модифікацій. Найбільш розповсюджені з них:

- байєсовські мережі довіри (Bayesian Belief Nets) - візуально можуть бути представлені як орієнтований ациклічний граф, ребра якого є пов'язані ймовірності змінних;
- ланцюги Маркова (Markov chains) – засновані на такому ж підході, як у байєсовських мереж довіри, але вирішують проблему вироблення рекомендації як послідовну оптимізацію, а не як просте прогнозування;
- класифікація за методом Роккіо (Rocchio classification) (заснована на векторній моделі) – використовує відгуки про релевантність елементів підвищення точності рекомендацій.

2.4.2. Алгоритм сумісної фільтрації за принципом користувачів

Сумісна фільтрація вмісту має обмеження за якістю опису, що надається контент-провайдером. Технічно існує обмеження на те, які ознаки можуть бути вилучені з обмеженого обсягу контентної інформації.

Вилучення «контентної» інформації в користувача теж непросте завдання. Насправді легше судити людей по тому, що вони роблять, ніж у тому, що вони говорять. Поведінкову інформацію, наприклад, те, що вони дивляться, як вони оцінюють предмети, набагато легше зібрати, а також це менш нав'язливо і точніше.

При сумісній фільтрації нам надається набір міток (labels), але не контентні ознаки. Проходить збір інформації про те, як проходить взаємодія з предметами – що ви оцінюєте, як ви оцінюєте, що ви переглядаєте, тощо.

Один із типових підходів полягає у створенні відповідної таблиці з рейтингами користувача, що містить оцінки, які користувач надав різним сутностям, що оцінюються. Завдяки їй ми знаходимо подібності і даємо рекомендації. Приклади отримання потрібних результатів за певними фільтрами наведені на рисунку 2.12.

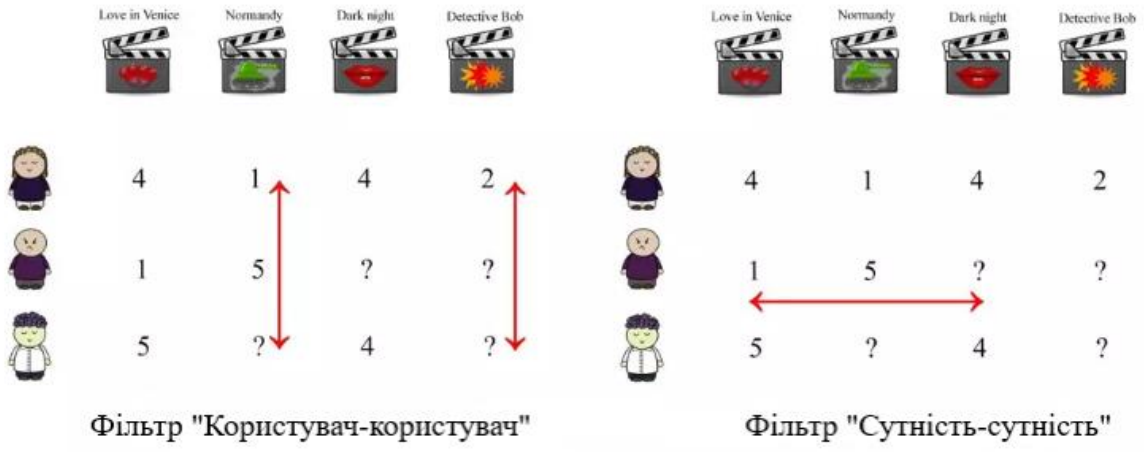

Рисунок 2.5 – Приклад роботи сумісної фільтрації за користувачами

Наприклад, отримано оцінки декількох фільмів. Система знаходить людей, які дали приблизно такі ж оцінки тим же фільмам, і пропонуємо ті фільми, які користувачі не дивились, але які отримали високі оцінки від цих користувачів (користувач-користувач).

Вищенаведений підхід представляє собою матрицю прийняття рішень. Це інструмент для оцінки та вибору найкращого варіанта з кількох можливих, що був винайдений британським інженером - проектувальником Стюартом Пью (Stuart Pugh). Даний метод може бути особливо корисним, якщо у вас більше одного варіанта вирішення проблеми та кілька факторів, які потрібно взяти до уваги для ухвалення остаточного рішення. Якщо критерії оцінки різних варіантів не однакові, то матриця прийняття рішень, швидше за все, буде кращим інструментом.

З іншого боку, замість знаходити подібності в людях, ми можемо знаходити подібності у фільмах (предмет-предмет) і робити пропозиції на основі того, що ви вже дивитеся. Щодо проблеми холодного старту, пропозиції виду предмет-предмет можуть бути простими, коли інформація про нового користувача обмежена.

Одне з можливих застосувань спільної фільтрації – це знаходження k-найближчих сусідів та використання їх середнього значення як прогноз. Його суть полягає у знаходженні найближчих транзакцій (у кількості К) до цієї транзакції. Потім використання середнього значення їх оцінок для передбачення невідомих оцінок.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

3.1. Опис архітектури програмного засобу

В рамках виконання дипломної роботи, було спроектовано програмний засіб, що складається з клієнтської, серверної частини, а також серверу з реляційною базою даних.

Клієнтська частина містить в собі графічний інтерфейс, який дозволяє представити наявний масив вихідних даних (існуючих користувачів та сутностей, що будуть обиратися) та підсумкові обчислені значення, що являють собою результати роботи додатку у вигляді, зрозумілому для простого користувача.

Серверна частина додатку містить в собі програмний код, що отримує необхідні вхідні дані з бази даних та використовує їх при роботі гібридного рекомендаційного алгоритму. База даних містить в собі всі необхідні параметри, можливі сутності для вибору, їх оцінки та сумарну цифру того, скільки раз вони були обрані.

Клієнтська та серверна частина з підключеної базою даних були реалізовані в рамках шаблону MVC (Model-View-Controller). Він описує простий спосіб побудови структури додатка, головною метою якого є відділення бізнес – логіки від призначеного для користувача інтерфейсу. В результаті чого, додаток легше масштабується, тестується, підтримується та практично реалізується. Принцип роботи шаблону MVC у вигляді концептуальної схеми зображений на рисунку 3.1.

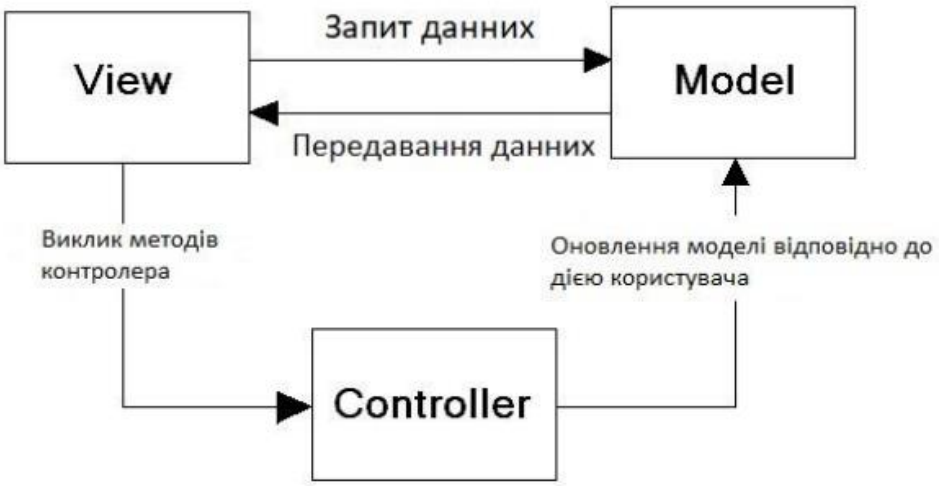

Рисунок 3.1 – Схема принципу роботи шаблону MVC

Шаблон моделі містить бізнес-логіку додатка і включає методи вибірки, обробки (наприклад, правила валідації) і надання конкретних даних. При цьому, він не повинен взаємодіяти безпосередньо з користувачем. Всі дані, що відносяться до запиту користувача повинні оброблятися в контролері. Він не здатний генерувати будь-яке відображення даних, адже цим займаються шаблони виду.

Шаблон виду використовується для завдання зовнішнього відображення даних, що були отримані з контролера та моделі. Зазвичай, містять в собі будь – яку розмітку та незначні вставки програмного коду, що використовуються для обробки, форматування і відображення отриманих програмним засобом даних. За своїм функціоналом поділяються на:

- загальні – містять в собі шаблони з розмітної для всіх сторінок, наприклад, header (шапку) і footer (підвал);
- часткові – використовуються для відображення даних з моделі або відображення форм введення даних [26].

Шаблон контролеру виступає ланкою, яка з`єднує між собою шаблони моделей, виду та інші компоненти у єдиний функціональний додаток, а також відповідає за обробку запитів користувача. При цьому, треба

пам’ятати, що контролер не повинен містити SQL/ NoSQL запитів – вони містяться в моделях. Також, контролер не повинен містити HTML і інший розмітки – вона виноситься в шаблони виду.

3.2. Обґрунтування вибору платформи СУБД

СУБД (система управління базами даних) – сукупність програмних та мовних засобів призначених для створення, збереження та модифікації даних деякою кількістю користувачів. Вимогами, які пред'являються до СУБД: атомарність, узгодженість, ізольованість, надійність (правила ACID - Atomicity, Consistency, Isolation, Durability).

В якості мови для створення СУБД була обрана MySQL – непроцедурна мова баз даних, що орієнтована на великий обсяг оброблюваної інформації. Вона була розроблена в 1993 році шведською компанією MySQL AB, ця СУБД є програмним забезпеченням для управління інформацією з відкритим вихідним кодом та призначається для підтримки реляційних БД. У цьому типі БД інформація зберігається у відносинах (таблицях), які взаємно пов'язуються. Оптимізація даних у таблиці здійснюється за допомогою скорочення рядків та/або стовпців. Запис будь-якої інформації є рядком і містить усі поля даних у стовпцях. Цей тип БД є найбільш застосованим у бізнес-рішеннях. Завдяки широкому та частому використанню MySQL система постійно розвивається та адаптується до потреб користувачів.

Серед ключових переваг використання MySQL в якості мови для проектування БД можна виділити наступні:

- можливість використання на різних платформах;
- швидкість обробки запитів;
- можливість використання для роботи з великими масивами;
- перебуває під ліцензією GPL (OpenSource);

- має підтримку від великої спільноти користувачів та розробників.

В якості середовища для роботи з СУБД, була обрана PostgreSQL – система з відкритим вихідним кодом, що розповсюджується за вільною ліцензією, має постійну підтримку актуальними версіями та багату документацію. Вона безкоштовна, якщо використовується для навчальних цілей, має зручний і доступний інтерфейс, а також інструментарій для роботи з реляційними БД.

Отже, спираючись на всю вищенаведені факти, в якості мови для реалізації бази даних було обрано SQL з використанням реляційного підходу, а в якості програмного середовища для роботи було обрано PostgreSQL.

3.3. Обґрунтування вибору мови програмування

Для розробки клієнтської частини був використаний React (іноді React.js або ReactJS) — JavaScript-бібліотека з відкритим вихідним кодом для розробки інтерфейсів користувача. React розробляється та підтримується Facebook, Instagram, Google та величезною спільнотою окремих розробників та корпорацій. Загалом, його використовують для розробки одно сторінкових та мобільних додатків. Ключова мета даного інструменту складається в здатності надавати розробникам високу швидкість, простоту та масштабованість під час створення програмних додатків. Головна особливість даного JS - фреймворку в тому, що створений на його основі програмний додаток складається з множинуі невеликих, але багатофункціональних файлів – компонентів. При цьому, розробнику не потрібно розбивати верстку, стилі та програмний код на окремі файли, адже можна розділити функціонал прямо у компоненти. Також, на базі цього фреймворку було написано множину міні-бібліотек, що значно

пришвидшують час розробки. Під час роботи додатку, компонент динамічно отримує список необхідних на даний момент часу даних, наприклад, перелік повідомлень на певній сторінці, автоматично форматує стилізовану розмітку (HTML, CSS) та програмний код (JS) і відображає кінцевий результат у браузері. У цьому йому допомагає JavaScript XML (JSX) - це розширення синтаксису JavaScript, яке дозволяє використовувати HTML-подібний синтаксис для опису структури інтерфейсу.

Для розробки серверної частини програмного додатку була використана програмна платформа Node.js, яка заснована на движку V8 та перетворює JavaScript з вузькоспеціалізованої мови в мову загального призначення. Node.js додає можливість JavaScript запускатися окремо від браузеру, взаємодіяти з пристроями введення-виведення через свій API, написаний на бібліотеках C ++, підключати інші зовнішні бібліотеки, написані на різних мовах, забезпечуючи виклики до них з JavaScript-коду. Переважно, Node.js застосовується на сервері, виконуючи роль того самого веб-сервера. В основі Node.js лежить подієво-орієнтоване і асинхронне (або реактивне) програмування з неблокуючим введенням / висновком. Також, до його складу входить власний зручний багатофункціональний установник пакетів NPM, за допомогою якого можна з комфортом встановлювати та підключати різноманітні сторонні пакети та модулі.

3.4. Реалізація рекомендаційної системи з вибору вибіркових освітніх компонент студентами ХНУРЕ

Як вже було сказано вище, під час виконання даної роботи була успішно спроектована та розроблена система з надання порад щодо вибору тих чи інших освітніх компонентів. Її функціональні можливості були продемонстровані на принципі вибору освітніх дисциплін для студентів Цей процес складався з таких етапів як: фізичне проектування бази даних

системи, аналіз вимог до її вхідних даних, розробка гібридного рекомендаційного алгоритму, що базується на декількох більш простих рішеннях – аналізує їх та на їх основі надає остаточну відповідь, проектування графічного інтерфейсу для відображення наявних даних та результатів роботи алгоритму. Також, був проведений аналіз можливостей покращення розробленого рішення у майбутньому, вдосконалення принципу роботи самого алгоритму.

3.5. Проектування фізичної схеми баз даних додатку

Для уточнення структури та типів даних БД необхідно побудувати фізичну модель бази даних рекомендаційної системи. Для цього був використаний MySQL Workbench – інструмент для візуального проектування баз даних, що інтегрує проектування, моделювання, створення та експлуатацію БД у єдине безшовне оточення для системи баз даних MySQL. Є наступником DBDesigner 4 від FabForce. Здатен надати достатній набор інструментів для проектування фізичної моделі баз даних будь-якого рівня складності.

Фізична схема бази даних зображена на рисунку 3.2. Слід зауважити, що під час фізичного проектування необхідно враховувати особливості конкретної СУБД, щоб використовувати тільки відповідні типи даних та структури даних.

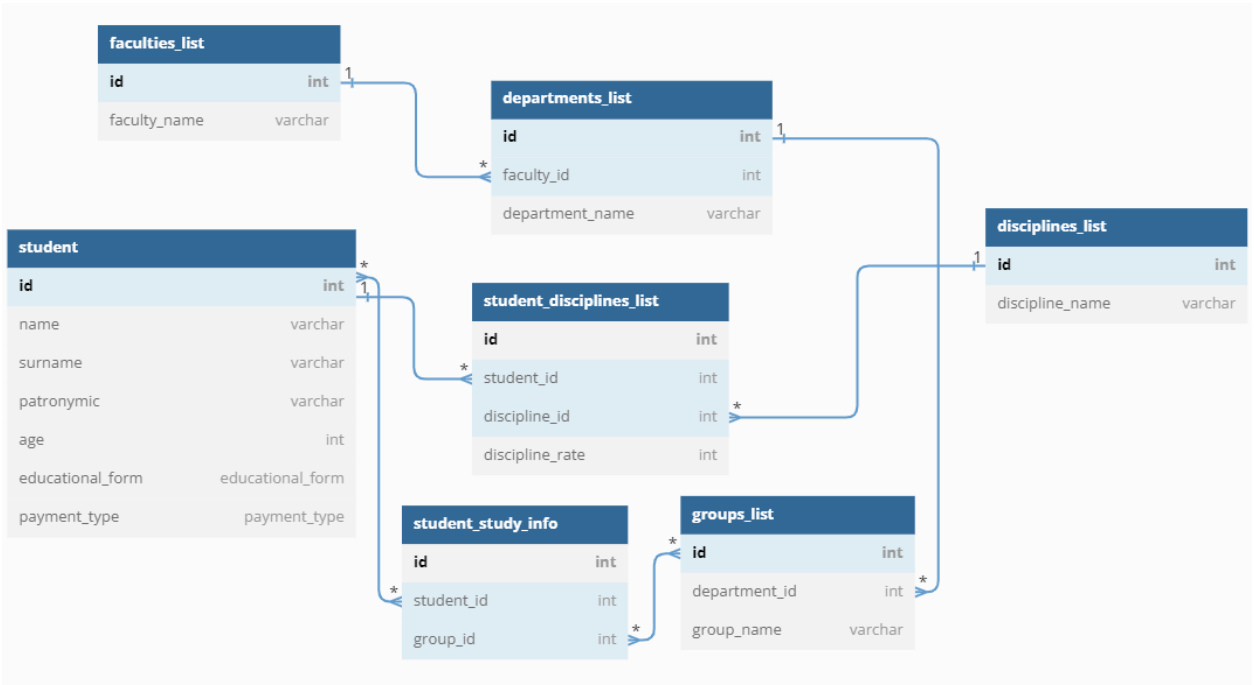

Рисунок 3.2 – Фізична схема бази даних

Кожна сутність містить свої атрибути з певними значеннями та зв’язки між потрібними таблицями. Слід зауважити, що кожна з цих сутностей має свій унікальний універсальний ідентифікатор у форматі первинного ключа (UUID). Розглянемо кожну з сутностей більш детально.

Сутність «faculties_list» відповідає за зберігання в собі інформації про всі доступні факультети. Вона містить в собі значення, що відповідають списку доступних факультетів університету. Ця сутність має зв’язок із сутністю «departments_list» типу «один до багатьох» – певна кафедра існує на певному факультеті.

Сутність «departments_list» відповідає за зберігання в собі інформації про всі доступні кафедри. Вона містить в собі ідентифікатори певних факультетів, до яких ці кафедри відносяться та значення, що відповідають списку доступних факультетів університету. Ця сутність має зв’язок із сутністю «groups» типу «один до багатьох» – на одній кафедрі може бути багато груп для навчання.

Сутність «groups_list» відповідає за зберігання в собі інформації про всі доступні групи. Вона містить в собі ідентифікатори певних кафедр, до яких ці групи відносяться та назву групи. Ця сутність має зв’язок із допоміжною сутністю «student_study_info». Дана допоміжна сутність необхідна для зберігання в ній ідентифікатору студенту та групи, на якій даний студент навчається. Вона має зв’язок типу «багато-до-багатьох» із сутністю з переліком груп та переліком студентів, адже 1 студент може навчатися в декількох групах та багато студентів можуть навчатися в одній групі одночасно.

Сутність «student» відповідає за зберігання в собі інформації про кожного зі студента. Вона містить в собі прізвище, ім’я та по-батькові студента, його вік, форму навчання (денна або заочна) та тип навчання – бюджет або контракт. Ця сутність має зв’язок типу «один до багатьох» із допоміжною сутністю «student_disciplines_list». Дана допоміжна сутність необхідна для зберігання в ній ідентифікатору навчальної дисципліни, ідентифікатору студенту, який цю дисципліну обрав та рейтингу, який був наданий студентом після її проходження.

Сутність «disciplines_list» відповідає за зберігання в собі інформації про всі доступні навчальні дисципліни. Вона містить в собі назву цієї дисципліни без підв’язки до конкретного факультету, кафедри чи групи. Ця сутність має зв’язок із сутністю «student_disciplines_list» типу «один до багатьох» – один студент вивчає багато дисциплін одночасно.

3.6. Аналіз вимог до вхідного набору даних та результатів роботи системи

На вхід до нашої системи будуть подаватися декілька наборів даних. Першим з них буде виступати повний список студентів, які входять до складу певної групи, навчаються на певній кафедрі та факультеті. Другим з

них буде загальним переліком дисциплін, на яких студенти навчались та які можуть бути порекомендованими. Третім набором виступатиме список, що буде містити в собі ідентифікатор дисципліни, студента, що її обрав та оцінка, яка була ним виставлена після її проходження.

Також, в якості можливого вхідного набору даних, у нас можуть виступати результатів опитування, що зберігаються у вигляді Microsoft Excel файлів, з яких легко можна експортувати дані у окремі файли формату CSV (Comma-Separated Values) – файлу текстового формату, що призначений для представлення табличних даних.. У ньому кожен рядок відповідає за окремий рядок таблиці, а стовпці відокремлені один від одного спеціальними символами – роздільниками (наприклад, комою). За допомогою програмних методів, з цих файлів можна отримати потрібні нам дані з аргументами того, скільки разів певна освітня компонента була обрана студентами та наскільки балів вони її оцінили. В рамках реалізованого алгоритму, для отримання потрібних асетів з даними, був використаний обидва ці підходи.

На виході, в рамках проміжних результатів роботи, система буде надавати нам чисельні значення, що будуть позначати, чи підходить нам бажана сутність. А в кінці, після обробки цих значень, буде надаватися остаточне чисельне значення, на основі якого і буде прийматися остаточне рішення.

В нашому випадку, для роботи системи була підготовлена вибірка даних з обраних дисциплін студентами групи ІТПм-21-2 кафедри системотехніки факультету Комп’ютерних наук у 2021 році у форматі CSV файлу. Вона містить в собі список студентів моєї групи, перелік можливих для вибору дисциплін, кількість раз, коли певна дисципліна була обрана та її сумарна рейтингова оцінка, що була отримана за формулою середнього арифметичного.

3.7. Розробка гібридного рекомендаційного методу

Розроблений гібридний рекомендаційний метод базується на двох допоміжних підсистемах – на базі методу колаборативної фільтрації, що заснована на користувачах (User-based Collaborative filtering) та сутностях (Item-based Collaborative filtering). Кожен з цих допоміжних методів аналізує вхідний набір даних за конкретними параметрами та на їх основі надає потрібну оцінку окремо один від одного, за своїм математичним шляхом.

В якості вхідного набору даних виступить певний перелік користувачів та дисциплін з конкретними критеріями, для яких буде проводитися процес рекомендації, що може бути отриманий як з бази даних, так і з CSV файлу.

На першому етапі, для певної кількості користувачів нам необхідно обчислити, наскільки сильно їх інтереси в контексті освітніх дисциплін збігаються із іншим користувачем. Виходячи з фундаментальних, загальних принципів роботи рекомендаційних алгоритмів, стає зрозуміло, що для того, щоб порадити користувачеві №1 будь-якої сутність, потрібно вибирати з тієї підмножини, що подобаються певним користувачам №2, №3 і т.д, які найбільш схожі за своїми оцінками на користувача №1.

Але як же нам отримати чисельний вираз цієї «схожості» користувачів? Допустимо, у нас є N сутностей. Оцінки, виставлені окремо взятим користувачем, є так званим вектором ідентичності M-мірному просторі усіх сутностей на основі певних критеріїв. Отже, для отримання чисельного значення схожості, нам потрібно порівняти між собою саме ці вектори. Існує багато способів з рішення цієї задачі, але було використано одне з найбільш популярних – порахувати косинусну міру між двома векторами, тобто, просто знайти косинус кута між ними. Це їх скалярний твір, поділений на довжину кожного з двох векторів. Формула розрахунку косинусної міри наведена нижче.

$$cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}||_2 \times ||\vec{y}||_2} \quad (3.1)$$

Програмна реалізація формули косинусної міри схожості між векторами ідентичності зображений на рисунку 3.3.

```
_compareVectors(firstVector: number[], secondVector: number[], excludeIndex: number): number {
    let numerator = 0;
    let denominator = 0;
    let firstOperand = 0;
    let secondOperand = 0;

    for (let i = 0; i < firstVector.length; i++) {
        if (i != excludeIndex) {
            numerator += firstVector[i] * secondVector[i];
            firstOperand += Math.pow(firstVector[i], 2.0);
            secondOperand += Math.pow(secondVector[i], 2.0);
        }
    }

    firstOperand = Math.sqrt(firstOperand);
    secondOperand = Math.sqrt(secondOperand);
    denominator = firstOperand * secondOperand;

    if (denominator === 0.0)
        return 0.0;

    return numerator / denominator;
}
```

Рисунок 3.3 – програмна реалізація формули косинусові міри

При реалізації було використано факт, що скалярне твір вектору себе дає квадрат довжини вектору – це найкраще рішення з погляду продуктивності.

Далі, нам потрібно сформувати матрицю переваг користувача на основі отриманих значень та за тими критеріями, за якими підсистема буде формувати рекомендацію. В якості строк в ній будуть виступати користувачі, що обрали освітній напрямок або поставили оцінки, а в якості стовбців –

сутності, що оцінюються та піддаються рекомендації. Функція розрахунку наведена на рисунку 3.4.

```
_buildItemsSimilarityMatrix(itemsMatrix: number[][]): number[][] {
    const results: number[][] = [];
    const invMatrix: number[][] = this._transposeMatrix(itemsMatrix);

    for (let i = 0; i < invMatrix.length; i++) {
        const row: number[] = [];

        for (let j = 0; j < invMatrix.length; j++) {
            if (i != j) {

                if (this._isSparse(invMatrix[i], invMatrix[j])) {
                    row.push(0.0);
                } else {
                    row.push(this._compareVectors(invMatrix[i], invMatrix[j], -1));
                }

            } else {
                row.push(-1);
            }
        }

        results.push(row);
    }

    return results;
}
```

Рисунок 3.4 – Функція формування матриці переваг

Під час її розрахунку в якості проміжного етапу, нам необхідно виконати транспонування матриці «користувач-сутність». Це робиться для декількох цілей. По – перше, для обчислення коефіцієнтів схожості, після чого для кожного з користувачів обчислюється набір найбільш ідентичних сутностей, використовуючи ту ж косинусну міру і запам'ятавши k найближчих. По друге, це робиться для отримання певної залежності сутностей від людей. Функція для обчислення транспонованої матриці наведена на рисунку 3.5.

```
_transposeMatrix(matrix: number[][]): number[][] {
    const res: number[][] = [];

    for (let i = 0; i < matrix[0].length; i++) {
        const row: number[] = [];

        for (let j = 0; j < matrix.length; j++) {
            row.push(matrix[j][i]);
        }

        res.push(row);
    }

    return res;
}
```

Рисунок 3.5 – Функція з обчислення транспонованої матриці

Також, під час формування матриці переваг, необхідно додатково перевірити її на розрідженість – наявність завеликої кількості пустих елементів. Функція з перевірки наведена на рисунку 3.6.

```
_isSparse(firstVector: number[], secondVector: number[]): boolean {
    let counter = 0;

    for (let i = 0; i < firstVector.length; i++) {
        if (firstVector[i] != 0 && secondVector[i] != 0)
            counter++;
    }

    return counter > 1 ? false : true;
}
```

Рисунок 3.6 – Формула з перевірки матриці на розрідженість

Далі, потрібно виконати обчислення попередньої рекомендації за кожним із підходів – заснованим на користувачах (User-based Collaborative filtering) та сутностях (Item-based Collaborative filtering). Як можна бачити, загалом, дані підходи дещо схожі, але, в іншу чергу, мають в собі ключові відмінності. В першому випадку, в якості набору даних береться перелік користувачів із освітніми напрямками, які вони вибрали – тут ключову роль буде відігравати той фактор, що скільки раз той чи інший освітній напрямок був обраний. У цьому випадку пошук буде виконаний в матриці ідентичності за потрібним індексом. У другому випадку за головний параметр буде братися оцінка пройденого освітнього напрямку користувачами, що була розрахована за середнім арифметичним усіх існуючих значень. Функція зі знаходження схожих значень за середньою оцінкою наведена на рисунку 3.7.

```
_predictMark(similarityMatrix: number[][], chosenMarks: number[], disciplineIndex: number): number {
    let numerator = 0;
    let denominator = 0;

    for (let i = 0; i < similarityMatrix.length; i++) {
        if (i != disciplineIndex) {
            numerator += similarityMatrix[disciplineIndex][i] * chosenMarks[i];
            denominator += similarityMatrix[disciplineIndex][i];
        }
    }

    if (denominator === 0.0)
        return 0.0;

    return numerator / denominator;
}
```

Рисунок 3.7 – Функція зі знаходження схожих значень за середньою оцінкою

Для формування заключної рекомендації, створений гібридний алгоритм бере значення, що були отримані під час роботи допоміжних методів та аналізує їх за певним набором правил, що були розроблені виключно для поточної задачі. Для цього обираються три певні дисципліни та на основі того, скільки раз вони були обрані та який середній бал мають,

гібридний метод проводить обчислення рекомендації. Цього вдалось досягти, задіявши певну системи перевірок та реагування на проміжні результати обчислення, отримані за допоміжними методів.

Так, наприклад, якщо освітня компонента була обрана достатньої кількістю користувачів, але її сумарна оцінка по 5-бальній шкалі менше ніж 3, то вона не може бути рекомендованою. У протилежному випадку, якщо дисципліна обиралась достатньо рідко, але має позитивну оцінку (більше, ніж 3), вона все одно може бути порекомендована користувачу, адже

Якщо для роботи методу, заснованого на сутностях або користувачах, недостатньо вхідних даних та порівнювати ні з чим, за остаточні результати беруться дані отримані під час роботи іншого методу, для роботи якого достатньо даних.

Також, у випадку, якщо результати роботи обох рекомендаційних допоміжних методів незадовільні (отримані оцінки мають менші значення, ніж 3), використовується метод розрахунку середнього значення, яке теж має бути більшим, ніж 3. Його програмна реалізація наведена на рисунку 3.8.

```
// Use all marks for discipline if not enough data
if (recommendData == 0 && predictedMark < this._markThreshold) {
    const marksSum: number = _.reduce([...marks.data, chosenMarks], (sum, curr) => sum += curr[chosenRecommendDiscipline], 0.0);
    const count: number = _.reduce([...marks.data, chosenMarks], (sum, curr) => sum += curr[chosenRecommendDiscipline] == 0.0 ? 0 : 1, 0);

    if (this._debugMode) {
        console.log(`Average mark: ${count == 0.0 ? 0.0 : marksSum / count}`);
    }

    if (count == 0) {
        return false;
    }

    return marksSum / count >= this._markThreshold;
}
```

Рисунок 3.8 – Метод розрахунку середнього значення за обома результатами

У випадку, якщо вхідна вибірка досить розріджена та для повноцінного аналізу даних не вистачає, ці результати комбінуються та виводяться їх середнє оцінка.

3.8. Проектування графічного інтерфейсу системи

Стартовий графічний інтерфейс розробленої системи на базі гібридного алгоритму. Для демонстрації принципу роботи системи, в якості вхідних даних був взятий список студентів групи ІТПм-21-2 кафедри Системотехніки факультету Комп'ютерних Наук та перелік можливих факультативних дисциплін для вибору другого курсу моєї магістерської програми.

У нас є таблиця з певним набором даних. В якості стовпців в ній виступає набір дисциплін, що доступний студентам для вибору. В якості строчок виступає список студентів за конкретною групою. Кожна клітинка в таблиці містить два значення. Перше з них – обрав студент конкретну дисципліну чи ні, чому відповідають чисельні значення 1 та 0 відповідно. Друге значення відповідає за середню оцінку дисципліни. За задумкою, кожен студент після проходження освітнього напрямку надає їй свою оцінку за 5-бальною шкалою – від 1 до 5 відповідно.

Задля того, щоб студент міг отримати рекомендацію стосовно того, чи підходить, студенту певна дисципліна, йому потрібно обрати не менше ніж 3 дисципліни, на основі яких буде обчислюватися рекомендація. Потім, студент мусить надати їм усім пріоритетну оцінку від 1 до 5 – наскільки сильно він хотів би обрати той чи інший освітній напрямок, де 5 – найвищий пріоритет, а 1 – найнижчий. Далі, потрібно обрати конкретну дисципліну для рекомендації та в кінці, натиснути на відповідну кнопку, що запустить поетапну роботу гібридного алгоритму.

Під час поетапної роботи гібридного алгоритму, проводиться обчислення усіх потрібних значень, на основі яких буде пораховано остаточний результат. Цей процес зображений на рисунку 3.9.

Рисунок 3.9 – Результати поетапної роботи гібридного алгоритму

І врешті-решт, згідно існуючого набору даних зі списку дисциплін, студентів, всіх вищеназваних параметрів та за описаним алгоритмом, система проводить обчислення та за отриманим чисельним значенням виводить повідомлення, підходить обрана дисципліна студенту чи ні та відображає відповідне повідомлення.

3.9. Аналіз перспектив з покращення принципу роботи розробленого рішення

На даний момент, з розробленим рішенням мож отримувати рекомендації з певних освітніх програм за такими параметрами як: кількість раз, коли певна дисципліна була обрана, її середня оцінка, пріоритетність цікавих, за думкою студента освітніх напрямків, на основі яких буде формуватися рекомендація.

У майбутньому, можливо додати нові параметри, за якими буде проводитися оцінка сутностей, що зацікавлять користувача. Їх можна буде отримувати із внутрішньо-університетської соціальної мережі, куди дане рішення планується повноцінно інтегруватися. Наприклад, якщо студент цікавиться новинами та подіями, які певним чином стосуються конкретного освітнього напрямку, а саме підписаний на відповідні групи та пабліки. Або якщо він слідкує за новинами на сторінці-профілі викладача, що викладає декілька дисциплін за однією загальною програмою. В такому випадку, засновуючись на наявних параметрах, система може порекомендувати конкретну дисципліну із даної предметної області до вибору.

3.10. Аналіз перспектив із покращення загального процесу дистанційного навчання

За допомогою розробленого рішення, у майбутньому, буде можливо покращити загальну ефективність процесу дистанційного навчання. Це можна буде робити різними способами.

Так, на основі рейтингової оцінки, що студенти надають певній дисципліні після її проходження, можна буде зробити певні висновки стосовно її якості. У випадку, якщо вона занадто низька, можна спробувати розібратися, чому це так – можливо, проблема в якості матеріалу, що викладається, підходу викладача до студентів або способу подачі матеріалу, тощо. І тому, після неупередженого аналізу ситуації, можливо, з’явиться потреба скоректувати освітню програму згідно адекватних побажань студентів.

Якщо реалізувати метод з розрахунку коефіцієнту популярності курсу, з’явиться можливість корегувати їх загальний список та перелік освітніх напрямків загалом. На основі отриманого значення будуть визначитися дисципліни, які є популярними серед студентів з певною формою навчання і які є незатребуваними. На основі цих даних, з’явиться можливість корегувати освітні напрямки загалом. Можна буде виділяти додаткових викладачів та більшу кількість людино-годин на викладання даних дисциплін, за рахунок скорочення непопулярних дисциплін, акцентуючи тим самим увагу на більш популярних напрямках. Ще, можна буде міняти перелік дисциплін освітнього напрямку – на основі наявних даних, виключати зовсім непопулярні предмети із загального переліку, а на їх місце додавати нові, що зможуть зацікавити студентів та ще більше занурити їх в учбовий процес.

У майбутньому, можна додати можливість реалізувати процес проведення ефективного оцінювання курсів, що є частинами певних освітніх

дисциплін та викладаються за конкретними навчальними програмами. Для цього можна, наприклад, запросити експертів, які працюють в цій сфері на комерційній основі вже достатню кількість часу. Вони мусять мати певний досвід у тому, за яким принципом відбуваються ті, чи інші робочі процеси, з якими труднощами можуть зіштовхнутися новачки у цій галузі, чому конкретно потрібно приділити більше уваги при вивченні матеріалів, а що можна скоротити або зовсім пропустити, тощо. Вони можуть проаналізувати вже існуючу або недавно створену освітню навчальну програму, надати свою незалежну оцінку стосовно її ефективності та підготовки майбутніх спеціалістів на її основі, а також задля допомоги у коректуванні вже існуючих освітніх програм.

Також, можна залучити спеціалістів не тільки для корегування вже існуючих програм, а й задля формування повністю нових навчальних напрямків зі своїми освітніми програмами. Для цього, вони можуть провести комплексний аналіз потреб ІТ-ринку на поточний момент часу. Це потрібно задля того, щоб зрозуміти, яких фахівців ринок потребує найбільше, а які напрями стають менш пріоритетними або зовсім безперспективними. На це зазвичай, впливає множина факторів – від глобального економічного стану певної галузі до особистих потреб конкретної країні. І на основі проведеного аналізу, буде можливість надати план з реформування освітніх прогам – з додання нових найбільш пріоритетних прогам, скороченням непопулярних напрямків та повним вилученням безперспективних учбових напрямків.

ВИСНОВКИ

У даній роботі була розглянута проблема формування індивідуальної освітньої траєкторії студентами вищих навчальних закладів під час дистанційного навчання, існуючі інструменти для вирішення проблеми, перспективи їх використання в контексті обраної теми. Була сформована потреба у розробці рекомендаційної системи для вирішення проблеми.

У ході виконання роботи, було досліджено найбільш розповсюджені існуючі математичні рекомендаційні методи, такі як методи колаборативної фільтрації, що засновані на користувачах (User-based) та сутностях (Item-based), гібридні методи фільтрації, алгоритми кластеризації та сумісної фільтрації за принципом користувачів. Для вирішення поставленої задачі були обрані методи колаборативної фільтрації.

Результатом дослідження стала рекомендаційна система, що базується на вже спроектованому прототипу внутрішньої університетської соціальної мережі. Був створений власний гібридний метод, в основу якого лягли результати обчислення, які згодом були оброблені з урахуванням власної системи перевірок – на основі отриманих значень, їх наявності, частоти вибору окремих сутностей та їх середніх оцінок.

Клієнт-серверна архітектура додатку була розроблена за шаблоном MVC. База даних була розроблена засобами СУБД MySQL. Програмна реалізація була виконання на базі фреймворків JS, таких як React та Node.js. У ході тестування було перевірено працездатність створеного рішення, перевірено та доведено коректність обчислених результатів.

Спроектоване рішення може бути вдосконалене в майбутньому за рахунок додавання нових параметрів для формування рекомендацій. А отримані результати можуть бути використані для ефективного та багатогранного покращення загального процесу дистанційного навчання – оцінки ефективності наявних напрямків навчання, створення нових, тощо.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Sytnikov D. Dimension reduction for network systems using structure modelaggregation // Sytnikov D., Gorbachov V., Ryabov O., Batiaa A.K., Ponomarenko O. //International Journal of Design and Nature and Ecodynamics Scopus.

2. Комплекс навчально-методичного забезпечення навчальної дисципліни "Проектування інформаційних систем" підготовки бакалавра спеціальності 122 – Комп'ютерні науки та інформаційні технології [Електронний ресурс] : освітньої програми "Комп'ютерні науки" / ХНУРЕ ; розроб. В. М. Левикін. – Харків, 2019. – 315 с.

3. Індивідуальна освітня траєкторія: запитання й відповіді. URL: https://sqe.gov.ua/individualna-osvitnya-traiektoriya/ (дата звернення: 05.12.2022).

4. Майнаєв Ф. Я. «ВИКОРИСТАННЯ СОЦІАЛЬНИХ МЕРЕЖ У ON-LINE-НАВЧАННІ. URL: https://core.ac.uk/download/pdf/145611689.pdf (дата звернення: 10.11.2022).

5. Рекомендаційна система. URL: https://clck.ru/32qxGL (дата звернення: 05.12.2022).

6. Shih R.C. Can Web 2.0 technology assist college students in learning English writing? Integrating Facebook and peer assessment with blended learning // Australasian Journal of Educational Technology. - 2011. №. 5. С. 829 – 845.

7. Tim O'Reilly. What Is Web 2.0// пер. з англ. Н. Казакова: Комп'ютерра, 2005.

8. G. Adomavicius, A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-artand possible extensions. In: IEEE Transactions of Knowledge and Data Engineering, 2005.

9. Народження Facebook та історія його успіху. URL: http://bussinesnews.org.ua/istorii-uspihu-istorija-uspihu/narodzhennja-facebook-istorija-uspihu.html (дата звернення: 05.12.2022).

10. Найпопулярніші соцмережі в Україні під час війни – дослідження Global Logic. URL: https://inlnk.ru/201jMd (дата звернення: 05.12.2022).

11. Дослідження методів побудови рекомендаційних систем в мережі Інтернет. URL: https://www.researchgate.net/publication/326107309 (дата звернення: 05.12.2022).

12. Павленко М.Ю. Університетська соціальна мережа для комунікації в освітньому процесі // Диплом бакалавра, спеціальність «Комп'ютерні науки». ХНУРЕ, 2020.

13. Lucas Bernardi, Jaap Kamps, Julia Kiseleva, Melanie J.I. Mueller. The Continuous Cold Start Problem in e-Commerce Recommender Systems, 2015.

14. Recommender systems survey / J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez // Knowledge-Based Systems. — 2013. — Vol. 46. — P. 109–132. — URL: https://www.sciencedirect.com/science/ article/pii/S0950705113001044

15. L. Tang, Y. Jiang, L. Li, and T. Li. Ensemble contextual bandits for personalized recommendation. In Proceedings of the 8th ACM Conference on Recommender Systems, pages 73–80, 2014.

16. J. Z. Sun, K. R. Varshney, and K. Subbian. Dynamic matrix factorization: A state space approach. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1897–1900, 2012.

17. M. Tavakol and U. Brefeld. Factored mdps for detecting topics of user sessions. In Proceedings of the 8th ACM Conference on Recommender Systems, pages 33 – 40, 2014.

18. Давидов А.О. Методи і засоби колаборативної фільтрації при побудові рекомендаційних сервісів у системах електронної комерції / А. О. Давидов – Тернопільський національний технічний університет імені Івана

Пулюя – Тернопіль, ТНТУ, 2018. – 8 с. URL: https://elartu.tntu.edu.ua/handle/lib/26406 (дата звернення: 05.12.2022).

19. Павленко М.Ю. Внутрішня університетська соціальна мережа як інструмент забезпечення ефективності дистанційного навчального процесу // Теоретичне та практичне застосування результатів сучасної науки : матеріали III Міжнародної студентської наукової конференції, м. Біла Церква, 7 жовтня, 2022 рік / ГО « Молодіжна наукова ліга». — Вінниця: ГО « Європейська наукова платформа», 2022. - С. 119-201.

20. W.P. Jones, G.W. Furnas; Pictures of Relevance: A Geometric Analysis of Similarity Measures; Journal of the American society for information science. 38(6): 420- 442, 1987; стр. 420-442. S. Choi, S. Cha, C.C. Tappert; A Survey of Binary Similarity and Distance Measures; Journal of Systemics, Cybernetics and Informatics, Vol 8 No 1 2010, с. 43-48.

21. X. Su, T.M. Khoshgoftaar; A Survey of Collaborative Filtering Techniques; Advances in Artificial Intelligence, 2009.

22. G. Shani, D. Heckerman, R. I. Brafman; An MDP-Based Recommender System; Journal of Machine Learning Research 6, 2005; с. 1265–1295.

23. Lucas Bernardi, Jaap Kamps, Julia Kiseleva, Melanie J.I. Mueller. The Continuous Cold Start Problem in e-Commerce Recommender Systems, 2015.

24. D. Jannach, M. Zanker, A. Felfernig, G. Friedrich; Recommender Systems. An Introduction; Cambridge University Press 32 Avenue of the Americas, New York, NY 10013-2473, USA, 2011; 352 с.

25. Види структур сайтів URL: http://alpha-spb.ru/helpfulinformation/43-types-of-sites (дата звернення: 05.12.2022).

26. Ethan B. Web Development with Node and Express: Leveraging the JavaScript Stack. New York: O`Reilly, 2019. 64 p