

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

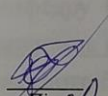
Галузь знань \_\_\_\_\_ 12 – Інформаційні технології \_\_\_\_\_

Спеціальність \_\_\_\_\_ 126 – Інформаційні системи та технології \_\_\_\_\_

на тему: «Інтелектуальна інформаційна система рекомендацій у середовищі онлайн-навчання»

КВРІСТ. 240184.01.54.00 ПЗ

Виконав: студент 2 курсу, група ІСТм-24-1

  
Підпис

Дмитро ПОГОРЕЛОВ  
Ініціали, прізвище

Керівник: доктор техн. наук, професор  
Науковий ступінь, вчене звання

  
Підпис

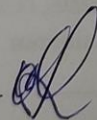
Ольга ПАВЛОВА  
Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КІІС,

PhD Ольга ПАВЛОВА

22 12 2025 р.



Хмельницький, 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 25 ” 08 2025 р.

**ЗАВДАННЯ**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Дмитру ПОГОРСЛОВУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інтелектуальна інформаційна система рекомендацій у середовищі онлайн-навчання

Керівник проекту (роботи) Ольга ПАВЛОВА, д.ф., доцент.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 25.08.2025 р. № 65

2. Строк подання студентом проекту (роботи) на кафедру 01.12.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Аналіз відомих моделей та засобів рекомендаційних систем

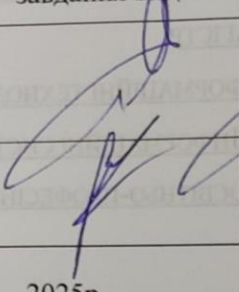
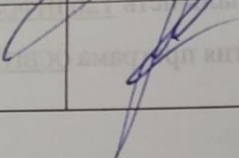
Інформаційні потоки інформаційної технології у середовищі онлайн-навчання

Метод формування рекомендацій у середовищі онлайн-навчання

Інформаційна технологія рекомендацій у середовищі онлайн-навчання

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сергій ЛИСЕНКО, професор кафедри КПС		
Антиплагиат	Андрій НІЧЕПОРУК, доцент кафедри КПС		

7. Дата видачі завдання « 01 » 09 2025р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	01.09.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	15.09.2025	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.10.2025	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	15.10.2025	виконано
5	Робота над наковою публікацією	15.10.2025	виконано
6	Робота над розділом 3 – розроблення методів для вирішення поставленої задачі	01.11.2025	виконано
7	Робота над розділом 4 – проєктування та розробка ПЗ для вирішення поставленої задачі	15.11.2025	
8	Оформлення пояснювальної записки згідно вимог	01.12.2025	виконано
9	Попередній захист ВКР	02.12.2025	виконано
10	Захист ВКР на засіданні ЕК	19.12.2025	

Студент

Керівник роботи

Підпис

Дмитро ПОГОРЕЛОВ

Ініціали, прізвище

Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

## РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Інтелектуальна інформаційна система рекомендацій у середовищі онлайн-навчання.

Автор роботи: Дмитро ПОГОРЄЛОВ

Керівник роботи: д.ф., доцент Ольга ПАВЛОВА

Пояснювальна записка: 82 с., 13 рис., 10 табл., 6 дод., 93 джерела.

Перелік ключових слів: онлайн-навчання, рекомендаційна система, гібридний метод, холодний старт.

Об'єктом дослідження є процес формулювання персоналізованих рекомендацій навчального контенту при онлайн-навчанні

Предметом дослідження є методи та засоби надання рекомендацій з використанням алгоритмів машинного навчання.

Метою кваліфікаційної роботи магістра є підвищення ефективності онлайн-навчання шляхом розроблення інформаційної технології

Для розв'язання поставлених задач використовувалися методи системний аналіз, алгебра матриць, теорії ймовірностей та контент-аналіз.

Наукова новизна отриманих результатів:

– набув подальшого розвитку метод формування рекомендацій, який використовує гібридну схему поєднання колаборативної фільтрації контент-аналізу;

– набула подальшого розвитку інформаційна технологія рекомендацій шляхом інтеграції модуля машинного навчання.

На основі проведених досліджень розроблена інформаційна система рекомендацій

Практична значимість отриманих результатів полягає у реалізації програмного продукту, який дає змогу скоротити час пошуку релевантних курсів

У першому розділі проведено системний аналіз предметної області та проблематики персоналізації в онлайн-навчанні. Здійснено огляд існуючих платформ та виявлено їх недоліки.

У другому розділі розроблено концептуальну модель системи та математичне забезпечення на основі методів сингулярного розкладу матриці і косинусної міри подібності.

У третьому розділі спроектовано трирівневу архітектуру системи та схему бази даних. Розроблено детальні алгоритми роботи гібридного механізму рекомендацій, який вирішує проблему «холодного старту».

У четвертому розділі виконано програмну реалізацію інформаційної технології. Проведено експериментальне дослідження, яке підтвердило перевагу розробленого методу над аналогами.

## ЗМІСТ

<b>КАЛЕНДАРНИЙ ПЛАН</b> .....	<b>Помилка! Закладку не визначено.</b>
<b>СКРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ</b> .....	<b>6</b>
<b>ВСТУП</b> .....	<b>7</b>
<b>1 АНАЛІЗ ВІДОМИХ МОДЕЛЕЙ ТА ЗАСОБІВ РЕКОМЕНДАЦІЙНИХ СИСТЕМ</b> .....	<b>10</b>
1.1 Аналіз моделей побудови рекомендацій в онлайн-навчанні.....	10
1.2 Аналіз методів формування персоналізованих рекомендацій для користувача.....	15
1.2.1 Методи розрахунку подібності .....	16
1.2.2 Методи зменшення розмірності та матричної факторизації.....	17
1.2.3 Методи аналізу контенту .....	17
1.2.4 Методи глибокого навчання.....	21
1.3 Аналіз засобів та існуючих рішень рекомендацій в онлайн-навчанні.....	24
1.3.1 Аналіз існуючих платформ та систем онлайн-навчання.....	25
1.3.2 Аналіз архітектури та засобів реалізації інтелектуальної інформаційної системи.....	30
1.4 Висновок до розділу .....	35
<b>2 ІНФОРМАЦІЙНІ ПОТОКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ У СЕРЕДОВИЩІ ОНЛАЙН-НАВЧАННЯ</b> .....	<b>36</b>
2.1 Концепція інформаційної технології надання рекомендацій в онлайн-навчанні .....	36
2.2 Препроцесинг та опрацювання даних для формування рекомендацій.....	38
2.2.1 Витяг та класифікація вхідних даних .....	39
2.2.2 Очистка та трансформація даних.....	39

2.2.3	Формування матриці взаємодії .....	39
2.3	Інформаційні потоки для ідентифікації рекомендацій у ПЗ .....	40
2.3.1	Ідентифікація зовнішніх сутностей та сховищ даних .....	40
2.3.2	Опис діаграми потоків даних .....	41
2.4	Висновок до розділу .....	43
<b>3</b>	<b>МЕТОД ФОРМУВАННЯ РЕКОМЕНДАЦІЙ У СЕРЕДОВИЩІ ОНЛАЙН НАВЧАННЯ.....</b>	<b>44</b>
3.1	Алгоритм роботи інтелектуальної системи рекомендацій.....	44
3.2	Метод формування рекомендацій на основі матричної факторизації .....	48
3.3	Розроблення вимог до інформаційної технології рекомендацій в онлайн- навчанні .....	50
3.4	Висновок до розділу .....	52
<b>4</b>	<b>ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РЕКОМЕНДАЦІЙ У СЕРЕДОВИЩІ ОНЛАЙН-НАВЧАННЯ.....</b>	<b>55</b>
4.1	UML-діаграми інформаційної технології рекомендацій .....	55
4.1.1	Діаграма варіантів використання .....	55
4.1.2	Діаграма класів.....	56
4.1.3	Діаграма послідовності.....	57
4.1.4	Діаграма діяльності.....	58
4.2	Структура інформаційної технології рекомендацій в онлайн-навчанні .....	60
4.3	Оцінка ефективності інформаційної технології рекомендацій .....	61
4.4	Висновок до розділу .....	72
	<b>ВИСНОВКИ .....</b>	<b>76</b>
	<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ .....</b>	<b>78</b>
	<b>ДОДАТОК А.....</b>	<b>86</b>

<b>ДОДАТОК Б</b> .....	88
<b>ДОДАТОК С</b> .....	90
<b>ДОДАТОК Д</b> .....	92
<b>ДОДАТОК Є</b> .....	94
<b>ДОДАТОК Ф</b> .....	95
<b>ДОДАТОК Г</b> .....	96

## **СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ**

БД - база даних

ВНЗ – вищі навчальні заклади

СКБД – система керування базами даних

ДПД – діаграма потоків даних

ШІ – штучний інтелект

ВММ – великі мовні моделі

СРМ – сингулярний розклад матриці

СГС – стохастичний градієнтний спуск

КСВ – кореневе середньоквадратичне відхилення

САП – середня абсолютна похибка

ОРВ – об’єктно регуляційне відображення

НКФ – нейронна колаборативна фільтрація

## ВСТУП

Сучасний етап розвитку суспільства характеризується стрімкою цифровою трансформацією освітньої галузі. Глобальний перехід до змішаних та дистанційних форм навчання зумовив експоненціальне зростання обсягів цифрового освітнього контенту. Платформи масових відкритих онлайн-курсів та університетські системи управління навчанням стають основними інструментами здобуття знань. Проте, просте накопичення навчальних матеріалів без ефективних механізмів навігації призводить до зниження ефективності освітнього процесу.

В процесі розвитку дистанційної освіти та електронного навчання надлишок інформації є суттєвою проблемою, яка може мати значні наслідки як для якості так і для ефективності засвоєння знань користувачами. Інформаційне перевантаження - це стан, коли студент стикається з надмірною кількістю навчальних матеріалів, курсів та ресурсів, що ускладнює вибір потрібного матеріалу. Відсутність персоналізованого підходу в освітніх платформах загрожує зниженням мотивації, низькому рівню завершеності курсів, втратою часу на пошук матеріалів та, як наслідок, зниженням загальної якості навчання.

Отже, ці проблеми становлять загрозу для залученості студентів, продуктивності освітнього процесу та конкурентоспроможності платформи. Тому аналіз моделей та засобів інтелектуальних рекомендацій має важливе значення для підвищення якості онлайн-навчання та забезпечення особистого розвитку студента.

Актуальність роботи полягає у необхідності персоналізації навчального процесу в умовах зростання кількості цифрового контенту, відповідно дослідження спрямоване на розробку інформаційної технології для формування доречних рекомендацій, що відповідають потребам та рівню знань користувача. Особливої гостроти набуває проблема адаптації системи до нових користувачів та роботи з розрідженими даними, коли студенти оцінюють лише незначну частку курсів. Існуючі алгоритми часто ігнорують семантичний зміст матеріалів або не враховують часову динаміку зміни інтересів студента, що вимагає розробки нових, гібридних підходів до побудови рекомендацій.

Метою кваліфікаційної роботи є вивчення різних методів та засобів інформаційної технології, що використовуються для генерації персоналізованих пропозицій у середовищі онлайн-навчання. Вивчаючи ці методи та засоби, можна отримати уявлення про їх переваги та обмеження, а також зрозуміти їхню придатність до задач адаптивного навчання. Ці дослідження сприятимуть покращенню загального процесу взаємодії користувача з освітньою платформою.

Поставлена мета досягається розв'язанням таких основних задач:

- необхідно розробити концепцію та структуру інформаційної технології для надання рекомендацій;
- метод повинен забезпечувати високу точність ранжування навчальних матеріалів на основі аналізу даних користувача;
- здійснити програмну реалізацію компонентів системи та оцінити їх ефективність.

Об'єктом дослідження є процес формування персоналізованих рекомендацій у середовищі онлайн-навчання.

Предметом дослідження є методи та засоби інформаційної технології надання рекомендацій для користувачів освітніх платформ.

Наукова новизна отриманих результатів полягає в розробленій структурі формування та детальній схемі інформаційної технології рекомендацій у середовищі онлайн-навчання.

1. набув подальшого розвитку метод формування рекомендацій, який, на відміну від існуючих, враховує динамічну зміну інтересів користувача;
2. набув подальшого розвитку інформаційної технології рекомендацій шляхом додавання засобів інтелектуального аналізу даних.

Практична цінність отриманих результатів. В результаті виконаного наукового дослідження розроблена інформаційна технологія надання рекомендацій, що дозволяє підвищити ефективність навчання та зменшити використаний час на пошук потрібного матеріалу.

У даній роботі викладено вимоги до методології побудови рекомендаційних систем.

Для розв'язання поставлених задач використовуються основні положення теорії інформаційних систем, методи системного аналізу та алгоритми машинного навчання.

# 1 АНАЛІЗ ВІДОМИХ МОДЕЛЕЙ ТА ЗАСОБІВ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

## 1.1 Аналіз моделей побудови рекомендацій в онлайн-навчанні

Ефективність функціонування інформаційних систем у сфері онлайн-навчання безпосередньо залежить від обраної моделі побудови рекомендацій. Ця модель являє собою формалізований опис взаємозв'язків між студентами та системою, а саме її об'єктами: навчальними курсами, лекціями, тестами та іншими. Метою системи є прогнозування користі конкретного об'єкта для конкретного користувача в потрібний для нього момент часу.

З математичної точки зору, задачу рекомендації можна формалізувати як пошук функції корисності  $f: U \times I \rightarrow R$ , де  $U$  - множина користувачів,  $I$  - множина навчальних об'єктів, а  $R$  - впорядкована множина значень (рейтинг або ймовірність зацікавленості). Головним завданням інформаційної технології є знаходження такого елемента  $i \in I$  для користувача  $u \in U$ , при якому значення функції  $f(u, i)$  буде максимальним.

В сучасних системах електронного навчання виділяють три основні класи моделей, які лежать в основі алгоритмів рекомендацій: контентно-орієнтовані моделі, моделі колаборативної фільтрації та моделі на основі знань. Узагальнена класифікація моделей наведена в рисунку 1.1

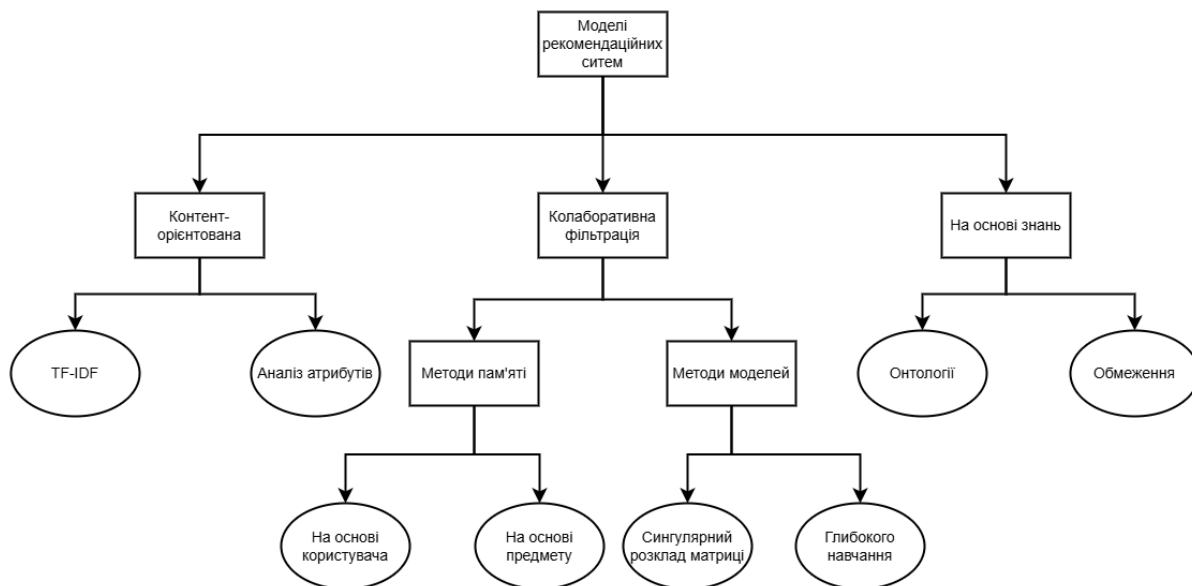


Рисунок 1.1 – Класифікація моделей рекомендаційних систем

В кожній з них є свої переваги та недоліки які показані в таблиці.1.1

Таблиця 1.1 – Переваги і недоліки основних класів моделей

Моделі	Переваги	Недоліки
Контенто-орієнтовані	Здатність рекомендувати нові курси, прозорість рекомендацій	Обмеженість новизни, складність автоматичного вилучення семантичних ознак з мультимедійного контенту
Колаборативної фільтрації	Можливість виявлення прихованих залежностей, рекомендація різноманітного контенту, який не обов'язково схожий за описом	Проблема “холодного старту”(неможливість надати точні рекомендації новому користувачу або новому курсу без історії взаємодії).

Кінець таблиці 1.1.

На основі знань	Враховання педагогічних вимог та логіки навчального процесу, відсутність проблеми “холодного старту”	Висока трудомісткість створення та підтримання бази даних
-----------------	--	---

Контенто-орієнтована модель рекомендацій ґрунтується на детальному аналізі характеристик навчальних матеріалів і індивідуального профілю користувача з метою забезпечення персоналізованого освітнього процесу. У середовищі онлайн-навчання кожен навчальний об’єкт розглядається як структурована одиниця знань, що має набір формалізованих атрибутів, які дозволяють описати його зміст і педагогічне призначення. Такий підхід дає змогу систематизувати навчальні ресурси та створити основу для автоматизованого підбору релевантного контенту.

Ключовими атрибутами навчальних матеріалів є тема, рівень складності, ключові слова, тривалість опрацювання та формат подання інформації (текст, відео, інтерактивні модулі тощо). Тема відображає предметну область матеріалу, рівень складності визначає відповідність контенту поточному рівню підготовки студента, а ключові слова дозволяють здійснювати семантичний аналіз та зіставлення навчальних об’єктів між собою. Тривалість і формат, у свою чергу, враховують часові та когнітивні можливості користувача.

Профіль студента формується динамічно на основі аналізу його попередньої навчальної діяльності в системі. До таких даних належать переглянуті курси, обрані теми, результати тестувань, час, витрачений на опрацювання матеріалів, а також частота взаємодії з різними форматами контенту. Сукупність цих параметрів

дозволяє створити індивідуалізовану модель знань, інтересів і навчальних уподобань студента.

Важливою складовою контентно-орієнтованої моделі є механізм представлення інтересів користувача у формалізованому вигляді. Зазвичай це реалізується через вектор ознак, що відображає вагу окремих тем, ключових слів або рівнів складності в профілі студента. Такий підхід дає змогу застосовувати математичні та статистичні методи для подальшого аналізу й порівняння даних.

Основна ідея контентно-орієнтованого підходу полягає в пошуку подібності між атрибутами доступних навчальних матеріалів та сформованим профілем студента. Для цього використовуються різні метрики схожості, зокрема косинусна міра, коефіцієнти кореляції або методи на основі TF-IDF. Чим вищий рівень подібності між навчальним об'єктом і інтересами користувача, тим вищою є ймовірність рекомендації цього матеріалу.

Перевагою контентно-орієнтованої моделі є її незалежність від даних інших користувачів, що особливо важливо в умовах невеликої кількості студентів або нових курсів. Система здатна формувати рекомендації навіть для нових користувачів, спираючись виключно на їхню індивідуальну активність і характеристики контенту. Це дозволяє уникнути проблеми «холодного старту» з боку навчальних матеріалів.

Водночас контентно-орієнтований підхід має певні обмеження, пов'язані з обмеженістю опису навчальних об'єктів та можливістю надмірної спеціалізації рекомендацій. Система може пропонувати матеріали, схожі лише на ті, що вже були опрацьовані студентом, не стимулюючи розширення його навчальних інтересів. Тому на практиці контентно-орієнтовані моделі часто поєднують із колаборативними або гібридними підходами для підвищення якості та різноманітності рекомендацій.

Модель колаборативної фільтрації використовує історію взаємодії між користувачами. Вони ґрунтуються на припущенні, що якщо користувачі мали схожі інтереси в минулому, то вони матимуть їх і в майбутньому. Розрізняють цю модель на два підходи:

1. Орієнтований на користувача: він базується на пошуку студентів зі схожими патернами навчання та рекомендаціями курсів, які їм сподобались.
2. Орієнтований на об'єкту: він базується на пошуку матеріалів, які часто вивчаються разом або мають подібні оцінки в одній групі студентів.

Важливим аспектом при побудові моделей є їх природа вхідних даних. У подібних системах джерелом даних є або рейтинг або лайки. Однак користувачі рідко ставлять оцінки пройденим матеріалам, тому більш логічно буде орієнтуватись на неявний зворотній зв'язок. До цих даних можна віднести: час перебування на сторінці, чи були завантажені додаткові матеріали, кількість спроб при проходженні тесту та чи приймали участь у форумах. Використання неявних даних дає змогу будувати більш точну модель поведінки користувача, оскільки вони відображають його реальні інтереси.

Модель на основі знань є специфічною для освітнього середовища. Окремим напрямком досліджень є секвенційна модель рекомендацій, на відміну від статичних моделей, які розглядають інтереси користувача як сталий набір вподобань, секвенційні моделі враховують часову послідовність подій. В освіті це критично важливо, оскільки процес навчання є еволюційним, так як знання студента змінюються після кожного пройденого курсу. Наприклад, якщо студент завершив курс "Основи Python", наступною логічною рекомендацією має бути "Аналіз даних на Python" або "ООП в Python", а не знову вступний курс навіть, якщо він має високий рейтинг серед схожих користувачів. Для реалізації таких моделей часто застосовуються ланцюги Маркова або рекурентні нейронні мережі. На відміну від розважальних платформ(відео, фільми, музика, відеоігри), навчання часто вимагає чіткої послідовності. Наприклад, неможливо рекомендувати курс Вищої математики без знань алгебри і геометрії, навіть якщо користувачу це може бути цікавим. Ці моделі використовують семантичні моделі предметної області для побудови логічних зв'язків між поняттями.

Аналіз праць підтвердив, що використання класичних методів змістилось до більшо використання технологій ШІ. Добре про це зазначається у роботах М.Урданета-Понте А. Мендес-Зенелла та І. Олеагордія-Ріуз, в яких вони описують,

що сучасні системи рекомендацій все частіше інтегрують методи глибокого навчання, що дозволяє виявити залежність між структурою курсів та успішністю студента, які неможливо знайти за допомогою звичайних методів[2].

Крім того, революцією стало використання ВММ для покращення фільтрацій. Добре про це описано в роботі Я.Дай, А. Ліу та Е. Лім[3], в яких вони описують як навчені ШІ(GPT) можуть автоматично вилучати потрібний зміст навіть з неструктурованих текстів.

Ще важливим є контекстно-орієнтовні моделі. В онлайн-навчанні контекст(час доби, тип пристрою) грає критичну роль, в той час як на мобільному пристрою доцільніше рекомендувати короткі лекції, на комп'ютері – повноцінні практичні завдання.

Також варто зазначити, що ефективна модель рекомендацій в онлайн-навчанні повинна враховувати не лише контент, але й індивідуальні інтереси користувача, такі як стиль навчання. Згідно з моделлю Фелдера-Сільверман, студенти поділяються на візуалів та вербалів, активних та рефлексивних. Інтеграція цих параметрів у модель фільтрації дозволяє системі ранжувати матеріали не тільки за темою, але й за форматом подачі: рекомендувати більше відеолекцій для візуалів або текстових статей для вербалів, що безпосередньо впливає на успішність засвоєння матеріалу.

Гібридна модель поєднує вищезгадані підходи, щоб зменшення їх недоліки. У контексті інтелектуальної інформаційної системи найбільш перспективним є використання гібридних моделей, які інтегрують колаборативні дані з контентним аналізом та враховують контекст навчання

## 1.2 Аналіз методів формування персоналізованих рекомендацій для користувача

Реалізація моделей рекомендаційних систем базується на використанні математичних методів обробки даних, статистики та машинного навчання. Вибір методу залежить від наявних даних та вимог до системи. Якщо модель визначає

загальну стратегію побудови рекомендацій, то метод - конкретний алгоритмічний інструментарій, що дозволяє реалізувати цю стратегію на практиці. Аналіз літературних джерел дозволяє класифікувати методи формування рекомендацій на дві великі групи: методи на основі пам'яті та методи на основі моделей.

### 1.2.1 Методи розрахунку подібності

В основі алгоритмів колаборативної фільтрації лежить необхідність визначення міри близькості між об'єктами(курсами) та суб'єктами(студентами). Для цього вектори ознак порівнюють у багатовимірному просторі.

Косинусна міра подібності – це один з найпоширеніших методів для текстових даних та розріджених векторів. Вона оцінює косинус кута між двома векторами, якщо вектори співнаправлені значення дорівнює 1(максимальна схожість), якщо перпендикулярні дорівнює 0. Формула розрахунку подібності між користувачами  $A$  та  $B$ :

$$\text{sim}(A, B) = \cos(\theta) = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}, \quad (1.1)$$

де  $\text{sim}(A, B)$  – значення подібності між користувачами  $A$  та  $B$ ;

$A_i$  – оцінка користувача  $A$  для  $i$  – го елемента;

$B_i$  – оцінка користувача  $B$  для  $i$  – го елемента;

$n$  – загальна кількість елементів у просторі ознак.

Кореляція Пірсона – метод, який використовують для виявлення лінійної залежності між двома наборами оцінок. Він є більш стійким до “зміщення оцінювання”, оскільки враховує відхилення від середнього значення:

$$\text{sim}(a, b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}}, \quad (1.2)$$

де  $P$  – множина предметів, які оцінили обидва користувачів;

$r_{a,p}$  – оцінка користувача  $a$  для предмета  $p$ ;

$\bar{r}_a$  – середня оцінка користувача  $a$  по всіх предметах;

$r_{b,p}$  – оцінка користувача  $b$  для предмета  $p$ ;

$\bar{r}_b$  – середня оцінка користувача  $b$  по всіх предметах.

### 1.2.2 Методи зменшення розмірності та матричної факторизації

Оскільки матриця взаємодіє між користувачем і курсом вона є дуже розрідженою у системі онлайн-навчання, через що використовують методи латентних факторів.

Сингулярний розклад матриці – метод який дозволяє розкласти вихідну матрицю рейтингів на три менші за розмірами матриці:

$$R \approx U \Sigma V^T, \quad (1.3)$$

де  $R$  – вхідна розріджена матриця рейтингів розмірності  $m \times n$ ;

$U$  – ортогональна матриця розмірності  $m \times m$ , що описує користувачів;

$\Sigma$  – діагональна матриця сингулярних чисел розмірності  $m \times n$ ;

$V^T$  – транспонована ортогональна матриця розмірності  $n \times n$ , що описує об'єкти(курси).

### 1.2.3 Методи аналізу контенту

Для контент-орієнтованих рекомендацій необхідно перетворити текстовий опис курсів у числовий вектор. Для цього використовується статистична міра TF-IDF – статистичний показник, що використовується для оцінки важливості слів у контексті документа:

$$\text{TF-IDF}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D), \quad (1.4)$$

де  $tf(t, d)$  – частота входження терміна  $t$  у документ  $d$ ;

$idf(t, D)$  – обернена частота документа, що зменшує вагу загальноживаних слів.

Після перетворення текстових описів навчальних курсів у числові вектори за допомогою міри TF-IDF задача пошуку релевантних матеріалів набуває формалізованого математичного вигляду. Усі курси та запити користувачів відображаються у вигляді точок у багатовимірному просторі, що дозволяє застосовувати методи лінійної алгебри та інформаційного пошуку. Такий підхід забезпечує уніфікований спосіб порівняння текстових даних незалежно від їх початкової форми та обсягу.

У межах векторної моделі кожен навчальний курс розглядається як вектор у  $n$ -вимірному просторі, де  $n$  відповідає кількості унікальних термінів, сформованих після попередньої обробки тексту. До етапів цієї обробки зазвичай належать токенизація, видалення стоп-слів, лематизація або стемінг. У результаті кожна компонента вектора відображає вагу певного терміна в документі, що дає змогу більш точно описати його зміст.

Профіль користувача в такій моделі також може бути представлений у вигляді вектора, сформованого на основі ключових слів переглянутих курсів, результатів тестувань або історії пошукових запитів. Вектор профілю фактично відображає сукупні навчальні інтереси студента та його поточні потреби. Це дозволяє порівнювати індивідуальні запити не з окремими словами, а з повноцінними семантичними описами навчальних матеріалів.

Ключовим етапом аналізу контенту є визначення ступеня схожості між вектором профілю користувача та вектором кожного доступного курсу. Для цього застосовуються метрики подібності, які дозволяють кількісно оцінити, наскільки близькими є два вектори в багатовимірному просторі. Отримане числове значення використовується як критерій релевантності матеріалу для конкретного користувача.

Найбільш поширеною метрикою в задачах аналізу текстів є косинусна міра подібності. Її перевага полягає в тому, що вона оцінює не абсолютні значення ваг

термінів, а орієнтацію векторів, тобто кут між ними. Це дозволяє коректно порівнювати тексти різної довжини, оскільки довгі документи не отримують необґрунтованої переваги над короткими лише за рахунок більшої кількості слів.

Значення косинусної міри подібності варіюється в діапазоні від  $-1$  до  $1$ , де значення, близькі до  $1$ , свідчать про високий рівень семантичної схожості між курсом і запитом користувача. Значення, близькі до нуля, вказують на слабкий або відсутній зв'язок між матеріалами, тоді як від'ємні значення на практиці зустрічаються рідко через невід'ємність ваг TF-IDF. На основі отриманих значень формується рейтинг курсів, з якого система обирає найбільш релевантні рекомендації для студента. Математично схожість між курсом  $A$  та профілем  $B$  обчислюється як скалярний добуток векторів, нормований на їх довжини:

$$\text{Similarity}(A, B) = \cos(\varnothing) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (1.5)$$

Однак, застосування лише методу TF-IDF у чистому вигляді має свої недоліки, пов'язані з проблемою синонімії (різні слова мають однакове значення) та полісемії (одне слово має різні значення). Наприклад, курси з описом “Software Engineering” та “Program Development” можуть бути семантично близькими, але матимуть низьку косинусну схожість через використання різних термінів.

Для вирішення цієї проблеми у сучасних системах рекомендацій попередній етап обробки тексту значно розширюється. Перед векторизацією застосовуються процедури:

1. Токенізації - розбиття тексту на елементарні одиниці.
2. Видалення стоп-слів - фільтрація часто вживаних слів (прийменників, сполучників), які не несуть смислового навантаження, але штучно завищують розмірність вектору.
3. Стемінгу або лематизації - зведення слів до їх основи або словникової форми (наприклад, слова “програмування”, “програмувати”, “програміст” зводяться до єдиного токена).

Окрім цього, як зазначають дослідники у сфері Data Mining, класичний TF-IDF ігнорує семантичний порядок слів. Тому в більш складних реалізаціях використовують N-грами (послідовності з N слів) замість окремих слів, що дозволяє зберігати стійкі словосполучення, характерні для предметної області (наприклад, “об’єктно-орієнтоване програмування” розглядається як єдиний термін, а не три окремих слова).

Аналіз ефективності застосування контенто-орієнтованих методів рекомендаційних систем широко представлений у сучасних наукових дослідженнях, присвячених проблемам персоналізації навчання в цифрових освітніх середовищах. Дослідники відзначають, що класичні підходи на основі TF-IDF і векторної моделі є достатньо ефективними на початкових етапах побудови рекомендаційних систем, проте мають обмеження, пов’язані з поверхневим аналізом текстового контенту.

У роботах В.В. Литвин, Л.В. Чирун та І.О. Бобрик запропоновано удосконалення контенто-орієнтованого підходу шляхом інтеграції онтологій предметної області. Автори наголошують, що використання лише статистичних характеристик тексту не дозволяє повною мірою врахувати семантичні зв’язки між поняттями, що є критично важливим для освітнього контенту.

Інтеграція онтологічних моделей дає змогу перейти від аналізу окремих термінів до розуміння їхнього змістовного контексту. Онтології дозволяють формалізувати знання про предметну область, встановлювати ієрархічні та асоціативні зв’язки між поняттями, а також враховувати синонімію та семантичну близькість термінів. У результаті рекомендаційна система може пропонувати матеріали, які є релевантними не лише за ключовими словами, а й за змістом.

Дослідники показують, що поєднання TF-IDF з онтологічними структурами підвищує точність рекомендацій і зменшує кількість нерелевантних результатів. Особливо це важливо для складних навчальних дисциплін, де одна й та сама концепція може описуватися різними термінами або входити до різних тематичних розділів. Такий підхід сприяє формуванню більш гнучкої та інтелектуальної системи підтримки навчання.

Паралельно з контентно-орієнтованими методами активно досліджуються підходи колаборативної фільтрації та матричної факторизації, які ґрунтуються на аналізі взаємодії великої кількості користувачів з навчальними матеріалами. Ці методи дозволяють виявляти приховані закономірності у поведінці студентів і формувати рекомендації на основі схожості їхніх інтересів та навчальних траєкторій.

У роботах Й. Корен зазначається, що методи сингулярного розкладу матриць (SVD) демонструють високу точність прогнозування вподобань користувачів. Проте автор підкреслює, що такі методи мають суттєві обмеження з точки зору обчислювальної складності та адаптації до змін у даних.

Особливу проблему матрична факторизація становить для динамічних систем онлайн-навчання, де дані постійно оновлюються: додаються нові курси, змінюється активність студентів, з'являються нові користувачі. У таких умовах повторне обчислення матричних розкладів у реальному часі стає ресурсоємним і може перетворитися на так зване «вузьке місце» продуктивності системи. Узагальнюючи результати сучасних досліджень, можна зробити висновок, що жоден з підходів не є універсальним. Контентно-орієнтовані методи виграють у прозорості та адаптивності до нових матеріалів, тоді як колаборативні методи забезпечують глибше врахування поведінки користувачів, але потребують значних обчислювальних ресурсів. Саме тому актуальним напрямом подальших досліджень є розробка гібридних моделей, які поєднують статистичні, семантичні та поведінкові методи для підвищення якості рекомендацій в онлайн-навчанні.

С.Вана та З.Ніу підтверджують що використання кореляції Пірсона (формула 1.3) ефективно лише на етапі накопичення оцінок, коли при “холодному старті” він себе показує майже з нулевою ефективністю, без гібридних технік[54].

#### 1.2.4 Методи глибокого навчання

Останніми роками класичні методи матричної факторизації поступають місце методам на основі нейронних мереж. Головним обмеженням є лінійність -

метод передбачає, що взаємодія користувача та предмета може бути описана простою лінійною комбінацією латентних факторів. Однак, поведінка студента в онлайн-навчанні є складнішою та нелінійною. Для вирішення цієї проблеми використовують нейронну колаборативну фільтрацію (НКФ). Цей підхід замінює скалярний добуток векторів на багат шаровий перцептрон, що дозволяє моделі вивчати довільну функцію взаємодії на основі даних. Особливе місце в освітніх системах займають секвенційні моделі. На відміну від статичних рекомендацій, навчання - це послідовний процес. Знання, отримані на курсі А, є необхідними для курсу Б. Для моделювання таких часових залежностей ефективно застосовуються рекурентні нейронні мережі та мережі довгої короткострокової пам'яті. Вони аналізують послідовність дій студента як часовий ряд і прогнозують наступний найбільш ймовірний крок у навчальній траєкторії.

Для обґрунтування вибору методу в практичній частині, було проведено порівняльний аналіз і представлено у таблиці.1.2.

Таблиця 1.2 – Порівняння методів ідентифікації рекомендацій

Метод	Тип даних	Переваги	Недоліки
Косинусна міра подібності	Вектори ознак, текст	Простота реалізації, швидкість	Не враховує специфіку розподілу оцінок користувача
Кореляція Пірсона	Числові рейтинги	Нормалізація даних	Чутливість до невеликої кількості спільних оцінок
Сингулярний розклад матриці	Розріджені матриці	Висока точність, виявляє приховані зв'язки	Висока обчислювальна складність навчання моделі

Кінець таблиці 1.2.

TF-IDF	Текстові описи	Ефективна робота з даними курсів	Ігнорує семантику слів та порядок слів у реченні
--------	----------------	----------------------------------	--

Отже для побудови ефективної системи рекомендацій в онлайн-навчанні доцільно використовувати гібридний підхід. Для обробки текстових описів курсів найефективнішим є TF-IDF

При реалізації методу матричної факторизації ключовою проблемою є не стільки саме розкладання матриці, скільки знаходження оптимальних значень ваг векторів користувачів та предметів, які мінімізують похибки прогнозу. У класичному формулюванні задача зводиться до мінімізації регуляризованої квадратичної похибки. Для вирішення цієї оптимізаційної задачі в роботі розглянуто два основні підходи: стохастичний градієнтний спуск(SGD) та метод найменших квадратів, що чергуються(ALS).

Стохастичний градієнтний спуск є ітеративним методом. Його суть полягає в тому, що для кожної наявної оцінки в навчальній вибірці система обчислює помилку передбачення, а потім коригує параметри моделі в напрямку, протилежному до градієнта. Перевагою SGD є простота реалізації та висока швидкість збіжності на великих наборах даних. Однак, цей метод чутливий до вибору кроку навчання. Занадто великий крок може призвести до того, що алгоритм "перестрибне" мінімум функції, а занадто малий - до надмірно тривалого навчання.

Альтернативою є метод ALS. Його особливість полягає в тому, що вектори користувачів фіксуються як константи, і розв'язується квадратична задача для векторів предметів, а потім навпаки. Цей процес повторюється ітеративно. Перевагою ALS є можливість розпаралелювання обчислень, що є критичним при роботі з масивами даних, які містять мільйони записів. Враховуючи архітектуру

сучасної платформи .NET та можливості бібліотеки ML.NET, яка підтримує багатопотокову обробку даних, саме варіації цих методів були покладені в основу програмної реалізації. Також було введено L2-регуляризацію для запобігання перенавчанню моделі, що дозволяє штрафувати систему за надто великі значення вагових коефіцієнтів, роблячи модель більш стійкою до шуму в даних.

### 1.3 Аналіз засобів та існуючих рішень рекомендацій в онлайн-навчанні

Перед вибором інструментарію для розробки рекомендаційної системи доцільно здійснити аналіз уже існуючих рішень, які застосовуються в сучасних платформах онлайн-навчання. Такий аналіз дозволяє визначити перевірені підходи до формування рекомендацій, оцінити їх ефективність у реальних умовах та уникнути типових помилок на етапі проєктування власної системи.

Особливу увагу слід приділити методам, що використовуються для персоналізації навчального контенту, зокрема контенто-орієнтованим, колаборативним і гібридним моделям. Дослідження їх реалізації в існуючих системах дає змогу зрозуміти, які дані про користувачів і навчальні матеріали є найбільш інформативними, а також які алгоритми забезпечують оптимальний баланс між точністю рекомендацій і обчислювальними витратами.

Аналіз готових рішень також включає вивчення застосовуваного програмного стеку, архітектурних підходів та інструментів обробки даних. Це дозволяє оцінити масштабованість систем, можливості інтеграції з іншими освітніми сервісами, а також вимоги до обчислювальних ресурсів у разі зростання кількості користувачів і навчальних матеріалів.

Отримані результати аналізу існуючих систем слугують основою для обґрунтованого вибору інструментарію та архітектури майбутнього програмного рішення. Вони допомагають адаптувати найкращі практики до конкретних умов функціонування системи онлайн-навчання та забезпечити ефективне формування персоналізованих рекомендацій для користувачів.

### 1.3.1 Аналіз існуючих платформ та систем онлайн-навчання

Сучасний ринок освітніх технологій характеризується експоненціальним зростанням обсягів цифрового контенту та переходом від традиційної моделі дистанційної освіти до адаптивних середовищ навчання. В сучасних умовах глобальної цифровізації, багато хто стикається з проблемою “інформаційного шуму” та парадоксом вибору, тобто коли надмірна кількість доступного матеріалу ускладнює побудову ефективної шляхи для освіти. У зв’язку з цим, одним з ключових елементів конкурентноспроможності є підсистеми інтелектуальних рекомендацій, які змінюють статичну бібліотеку в динамічну екосистему.

Щоб визначити вимоги до проєктованої системи потрібно провести аналіз архітектурних рішень та функціональних можливостей лідерів в цій області. Існуючі рішення можна умовно розподілити за трьома напрямками:

1. MOOC-платформи – глобальні сервіси орієнтовані на масову аудиторію та надання академічних знань. Прикладами є Coursera та edX. Вони характеризуються використанням великих даних для аналізу поведінки мільйонів користувачів.

2. Маркетплейси контенту – комерційні майданчики, в яких пріоритетом є продаж курсів, прикладом є Udemy. В цій системі алгоритми рекомендацій працюють подібно до систем електронної комерції.

3. Система управління навчанням – системи, що використовуються в університетах та корпоративному секторі для адміністрування навчального процесу, прикладом є Moodle.

Провівши аналіз за критеріями адаптивності, наявності інтелектуальних алгоритмів, відкритості архітектури та можливості інтеграції. Було вибрано п’ять найбільш доречних платформ, кожна з яких демонструє унікальний підхід до взаємодії з користувачем: Coursera, edX, Udemy, Duolingo та Moodle.

1. Coursera – глобальна освітня платформа, заснована професорами Стенфордського університету Ендрю Нг та Дафною Коллер для надання доступної

онлайн освіти. Вона пропонує структуроване навчання: від коротких курсів до повноцінних онлайн-ступенів магістра. Платформа підтримує відеолекції з інтерактивним транскриптом, автоматизоване тестування та завдання з взаємоперевіркою. Головною особливістю є керовані проекти, в яких студенти виконують завдання у віртуальній хмарній машині через браузер.

2. edX – некомерційна платформа створена Гарвардом та МІТ(Массачусетським технологічним інститутом), вона базується на відкритому ПЗ open edX. Вона фокусується на академічних курсах університетського рівня. Архітектура платформи дозволяє створювати гнучкі курси: від простих відео до складних симуляторів фізичних процесів чи хімічних лабораторій. Платформа має підтримку програми “MicroMasters”, вони можуть бути зараховані як кредити в університетах

3. Udeemy – найбільший у світі маркетплейс навчального контенту, за моделлю С2С(споживач для споживача). На відміну від академічних платформ, тут будь-хто може стати викладачем. Платформа надає зручний інструментарій для завантаження відео, створення тестів та спілкування зі студентами. Основний акцент іде на практичні навички

4. Duolingo – платформа для вивчення мов, яка популяризувала гейміфікацію навчання. Навчання розбите на мікро-уроки, яка побудовані у формі гри(бали, ліги, досягнення). Вона повністю адаптована під мобільні пристрої, що робить її гарним способом вивчення мов. Основними методами навчання є переклад речень, аудіювання та складання слів.

5. Moodle – найпоширеніша у світі система управління навчання з відкритим кодом. Платформа є конструктором, який дозволяє створювати навчальні портали різної складності. Вона дає інструментарій для адміністрування користувачів, створення курсів, оцінювання, управління файлами та комунікацією. Вона широко використовується в ВНЗ.

В кожній з цих платформ є свої переваги і недоліки, також вони мають унікальні механізми рекомендацій про які розписано в таблиці 1.3.

Таблиця 1.3. – Переваги, недоліки та механізми рекомендацій платформ

Платформи	Механізми рекомендацій	Переваги	Недоліки
Coursera	Використовує складний гібридний підхід, вона аналізує не тільки кліки а й семантику контенту,будуючи “граф навичок”, щоб запропонувати курс для заповнення прогалин	Висока академічна якість матеріалів, верифіковані сертифікати, сильна інтеграція з університетськими програмами	Висока вартість підписки, жорстка прив’язка до розкладу сесій, закритість платформи для сторонніх розробників
edX	Система спирається на аналіз освітніх траєкторій. Пропонує курси які є логічним продовженням вже опанованих дисциплін	Відкритий вихідний код,наукоємний контент, підтримка складних типів завдань	Складний інтерфейс,високі вимоги до серверних ресурсів для Open edX,мала увага до UX(досвіду користувача)

Кінець таблиці 1.3.

Udemy	Працює за принципом електронної комерції. Алгоритми аналізують історію покупок та переглядів, активно просуваючи курси з високими рейтингами та конверсією продажів	Величезний асортимент, швидка поява курсів з нових технологій, доступні ціни, зручний мобільний додаток	Відсутність контролю якості, комерційна упередженість рекомендацій, відсутність академічної структури
Duolingo	Використовує пропріетарну AI-систему "BirdBrain", яка індивідуально підбирає вправи з моделлю інтервальних повторень	Надзвичайно високе залучення користувачів, інтуїтивний інтерфейс, безкоштовний доступ до базового функціоналу	Вузька спеціалізація, неможливість завантаження власного контенту, відсутність глибокої теорії
Moodle	У базовій версії фактично немає рекомендацій. Курси призначаються адміністративно або вручну.	Повний контроль над даними, модульність, відповідність стандартам SCORM	Застарілий дизайн інтерфейсу, складність налаштування, відсутність вбудованої розумної системи рекомендацій

Для підведення результатів та визначення найкращого варіанту з існуючих рішень, найкраще буде провести порівняння розглянутих платформ за ключовими критеріями, вони наведені в таблиці 1.4.

Таблиця 1.4. – Порівняння платформ за ключовими критеріями

Характеристика	Coursera	edX	Udemy	Duolingo	Moodle
Тип платформи	МООС	МООС	Маркетплейс	Мобільний додаток	LMS
Алгоритм рекомендацій	Гібридний	Контенто-орієнтована	На основі предметів	AI	Відсутній
Об'єкт аналізу	Навички	Освітня траєкторія	Історія покупок	Поведінка в реальному часі	Журнал оцінок
Мета рекомендації	Продаж сертифікатів	Академічний ступінь	Максимізація продажів	Утримання	Адміністрування
Адаптивність	Висока	Середня	Низька	Висока	Низька
Відкритість коду	Закритий	Відкритий	Закритий	Закритий	Відкритий
Можливість інтеграції	Платне API	Складна	Обмежена	Відсутня	Повна

Окремої уваги в контексті інтеграції інтелектуальних систем рекомендацій заслуговує питання стандартизації обміну даними. Більшість сучасних LMS оперують застарілими стандартами, такими як SCORM (Sharable Content Object Reference Model). Хоча SCORM ефективно вирішує задачу пакування контенту та відстеження базового прогресу (завершив/не завершив, здав тест/не здав), він є

абсолютно недостатнім для побудови якісних рекомендаційних алгоритмів. SCORM не дозволяє фіксувати гранулярні дії користувача, такі як: «студент переглянув відео до 5-ї хвилини, поставив на паузу і закрив вкладку» або «студент тричі повертався до одного й того ж слайду презентації».

Саме тому в розробленій системі було проаналізовано можливість використання специфікації xAPI (Experience API, раніше відомий як Tin Can API). На відміну від SCORM, xAPI дозволяє записувати будь-який досвід навчання, в тому числі оффлайн або неформальний, у форматі потоку активностей (Activity Streams). Структура даних xAPI базується на семантиці “Актор - Дієслово - Об’єкт” (наприклад, «Іванов – Переглянув – Курс Python»). Використання подібного підходу при проектуванні бази даних дозволяє накопичувати так звані “Big Learning Data”, які є паливом для алгоритмів машинного навчання.

Також важливим аспектом є сумісність із протоколом LTI від консорціуму IMS Global. Якщо розглядати розроблену систему рекомендацій не як моноліт, а як мікросервіс, то саме підтримка LTI дозволяє безшовно вбудувати її у Moodle, Canvas або Blackboard. Це дозволяє уникнути необхідності подвійної авторизації студентів, оскільки LTI забезпечує безпечну передачу контексту користувача та його ролі між системами. Аналіз цих стандартів показує, що ігнорування протоколів інтероперабельності призводить до створення ізольованих рішень, які неможливо масштабувати в реальному університетському середовищі.

### 1.3.2 Аналіз архітектури та засобів реалізації інтелектуальної інформаційної системи

Щоб розробити ефективну систему рекомендацій потрібно правильно вибрати не лише мову програмування, але й архітектурний паттерн та засоби для управління даними.

Для реалізації інформаційної системи в середовищі онлайн навчання найкраще використовувати веб-орієнтовану архітектуру. Серед них найбільше підходять:

Монолітна архітектура: Весь функціонал знаходиться в одному застосунку, що спрощує роботу в початкових етапах, але ускладнює на подальших етапах.

Сервіс-орієнтована архітектура: Система розбита на компоненти, що є стандартом для великих платформ, оскільки дозволяє писати компоненти різними мовами

Трирівнева архітектура: найбільш підходящий підхід для кваліфікаційної роботи ти систем середнього розміру. Вона складається з:

1. Рівень представлення - веб-інтерфейс користувачів
2. Рівень бізнес-логіки – модуль інтелектуальних рекомендацій
3. Рівень даних – СКБД і сховище даних

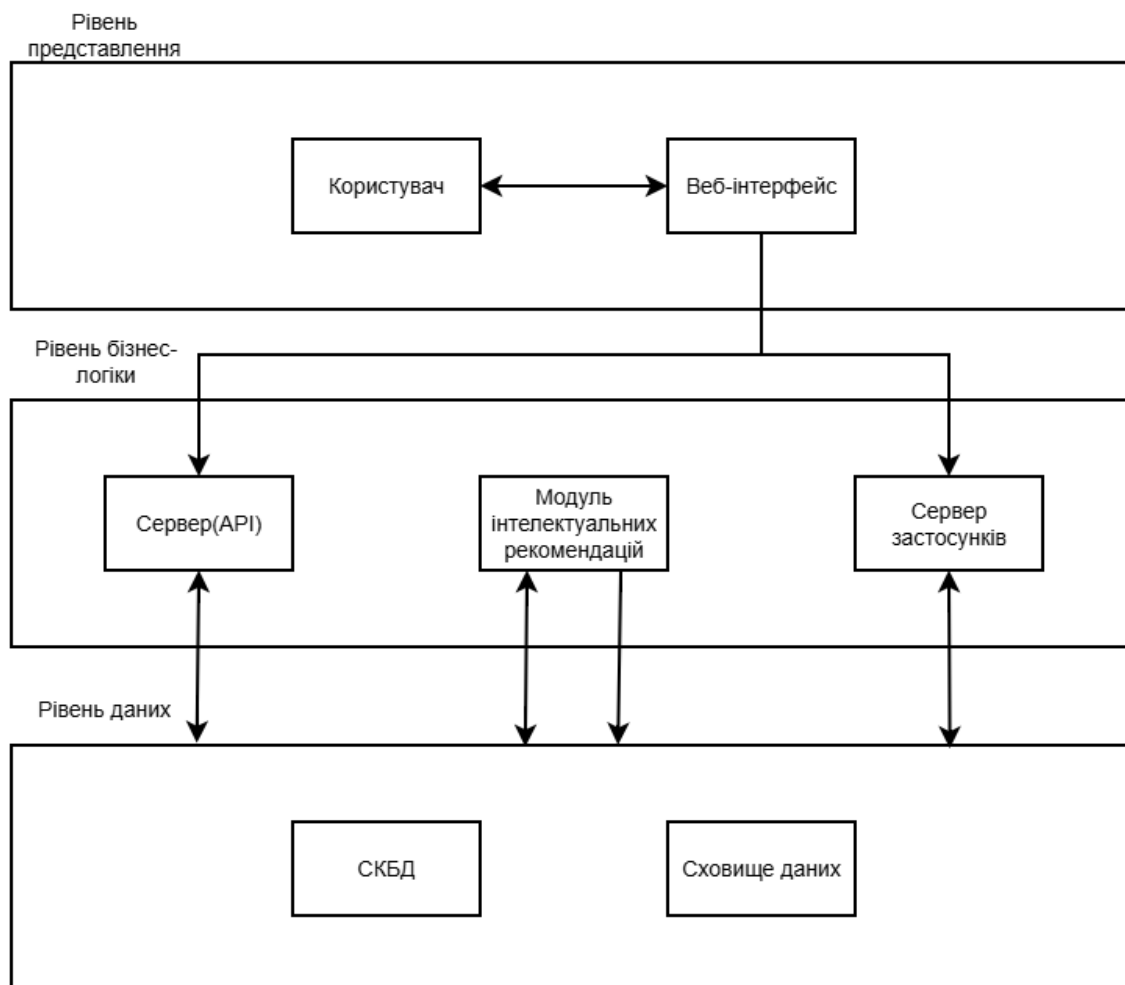


Рисунок 1.2 – Приклад трирівневої архітектури

Для того щоб інформаційна система рекомендацій онлайн-навчання вона повина мати змогу використовувати данні, а саме структуровані(профілі, курси) і потокові(логи). Для того щоб забезпечити цілісність даних та мати можливість виконувати складні запити, найкращим варіантом є використання реляційних баз даних. Для цього найбільше підійдуть:

PostgreSQL – вона є потужною СКБД з відкритим кодом, що має широку підтримку з JSON-даними

Microsoft SQL Server – стандарт для систем, який побудований на технологіях Microsoft, що дає йому високу продуктивність та інтеграцію з різними засобами аналітики

Однак, вибір системи управління базами даних (СУБД) для інтелектуальної системи рекомендацій не може обмежуватися лише загальним оглядом ринку. Специфіка предметної області накладає додаткові вимоги, головною з яких є ефективна робота в умовах високого навантаження на читання. Це пояснюється тим, що операція отримання рекомендацій (читання) відбувається значно частіше, ніж операція оновлення профілю або додавання нового курсу (запис).

У цьому контексті критично важливим аспектом стає підтримка гібридної моделі зберігання даних. Традиційна реляційна модель ідеально підходить для зберігання сутностей із жорсткою схемою: облікові записи користувачів, транзакції оплат, структура навчальних програм. Тут на перше місце виходять принципи ACID (Atomicity, Consistency, Isolation, Durability), які гарантують цілісність даних. Наприклад, неможлива ситуація, коли система порекомендує студенту курс, який був видалений адміністратором хвилину тому. Обидві розглянуті системи (PostgreSQL та MS SQL Server) повною мірою забезпечують ці вимоги.

Проте, друга складова системи - логи активності (потокові дані) - має іншу природу. Це масиви даних про кліки, час перегляду відео, результати проходження тестів, які генеруються у великих обсягах і часто мають змінну структуру. Зберігання таких даних у класичних таблицях призводить до необхідності постійних міграцій схеми БД, що ускладнює підтримку системи. Саме тут

вирішальним фактором стає підтримка роботи з JSON-документами на рівні ядра СУБД.

PostgreSQL у цьому аспекті має суттєву перевагу завдяки типу даних JSONB (Binary JSON). На відміну від звичайного текстового зберігання JSON, JSONB зберігає дані у розібраному бінарному форматі, що дозволяє будувати індекси по окремих полях всередині JSON-документа. Це дає можливість виконувати аналітичні запити, наприклад: “знайти всіх користувачів, які переглядали категорію “Програмування” більше 10 хвилин”, зі швидкістю, порівнянною з роботою спеціалізованих NoSQL рішень (таких як MongoDB), але залишаючись у межах єдиної реляційної інфраструктури.

Окремим критерієм вибору є масштабованість. Оскільки кількість записів у таблиці рейтингів зростатиме експоненційно, СУБД повинна підтримувати механізми секціонування таблиць. Це дозволяє фізично розділити одну велику логічну таблицю на менші частини (наприклад, за роками або діапазонами ID), що значно пришвидшує виконання запитів.

Враховуючи, що розробка ведеться в рамках магістерської роботи, важливим фактором є також вартість володіння (TCO). PostgreSQL розповсюджується за вільною ліцензією, що дозволяє розгортати необмежену кількість інстансів (наприклад, для тестування, розробки та продакшну) без фінансових витрат. У той час як MS SQL Server, попри чудову інтеграцію з .NET, вимагає ліцензування для використання повного функціоналу Enterprise-рівня, необхідного для тонкого налаштування продуктивності.

Таким чином, проаналізувавши вимоги до обробки структурованих та напівструктурованих даних, а також враховуючи необхідність побудови складних аналітичних вибірок для алгоритму колаборативної фільтрації, найбільш доцільним вибором для реалізації серверної частини системи є PostgreSQL. Вона забезпечує необхідний баланс між строгістю реляційної моделі для профілів користувачів та гнучкістю NoSQL-підходів для збереження історії взаємодій.

Для реалізації інтелектуального модуля найбільш провідними будуть ці дві екосистеми: Python та .NET. Їхні переваги та особливості показані в таблиці 1.5.

Таблиця 1.5 – Особливості і переваги різних екосистем

Екосистема	Переваги	Особливості
Python	Доступ до сучасних алгоритмів, висока швидкість написання коду, велика спільнота	Зазвичай потребує використовувати мікросервісний підхід
.NET	Суворі статична типізація, висока швидкість виконання компіляції коду, можливість реалізації ML-моделі всередині основного процесу програми	Менша кількість готових прикладів, але краща інтеграція з Windows інфраструктурою

На основі проведеного порівняльного аналізу (Таблиця 1.5), для розробки серверної частини системи було обрано платформу .NET 8 та мову програмування C#. Цей вибір зумовлений наступними факторами:

Продуктивність - .NET Core демонструє вищу продуктивність обробки HTTP-запитів порівняно з Python, що є критичним для високонавантажених систем.

Інтеграція ML - використання бібліотеки ML.NET дозволяє вбудувати навчену модель рекомендацій безпосередньо в конвеєр обробки запитів веб-сервера. Це виключає необхідність підтримки окремого мікросервісу на Python та знижує затримки на передачу даних між сервісами.

Типізація - статична типізація C# зменшує кількість помилок на етапі розробки та спрощує підтримку коду великих корпоративних систем.

В якості системи керування базами даних добре підійде PostgreSQL через її здатність ефективно працювати як зі структурованими даними (таблиці курсів, користувачів), так і з напівструктурованими (логи подій, збереження векторів ознак), а також через безкоштовну ліцензію.

#### 1.4 Висновок до розділу

У розділі було проведено аналіз проблеми підвищення ефективності онлайн-навчання шляхом впровадження інтелектуальних інформаційних технологій. В ході дослідження було встановлено, що класичні підходи до побудови рекомендацій, такі як контент-орієнтований або колаборативний, у чистому вигляді мають критичні недоліки для освітнього середовища, зокрема проблему "холодного старту" та неспроможність працювати з розрідженими даними. У зв'язку з цим визначено, що найбільш дієвим є гібридний підхід, який поєднує аналіз змісту курсів та історію взаємодій для забезпечення найбільш ідеальної персоналізації.

На основі аналізу математичного апарату було обґрунтовано, що для реалізації інтелектуального ядра системи найефективнішим є поєднання сингулярного розкладу матриці для виявлення прихованих латентних факторів та косинусної міри подібності разом із TF-IDF для роботи з текстовими описами. Провівши порівняльний огляд існуючих платформ було виявлено суттєвий технологічний розрив на ринку - існують або потужні комерційні платформи з закритим вихідним кодом, або популярні відкриті, які виконують здебільшого адміністративні функції та позбавлені вбудованих інтелектуальних механізмів, що підтверджує актуальність розробки окремого рекомендаційного модуля.

Також у розділі було проаналізовано засоби реалізації та визначено доцільність використання трирівневої архітектури для побудови масштабованої системи. В якості технологічного ідеально підійдуть платформа .NET та бібліотека ML.NET, що забезпечить високу продуктивність обробки запитів та можливість безшовної інтеграції моделі машинного навчання в серверне середовище. Отримані результати створюють теоретичне підґрунтя для подальшого математичного моделювання та програмної реалізації системи.

## 2 ІНФОРМАЦІЙНІ ПОТОКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ У СЕРЕДОВИЩІ ОНЛАЙН-НАВЧАННЯ

### 2.1 Концепція інформаційної технології надання рекомендацій в онлайн-навчанні

Розробка інформаційної технології рекомендацій в онлайн-навчанні базується на принципах системного аналізу та передбачає формалізацію процесів взаємодії між користувачем та освітнім середовищем. Головною метою для створення є трансформація вхідних даних користувача та навчальний контент у впорядкований перелік, який дозволить ефективніше вибирати і засвоювати знання.

Для ефективного опису моделі системи було обрано методологію SADT(методологія структурного аналізу та проєктування) та стандарт IDEF0. Він дозволяє показати систему у вигляді ієрархічних взаємопов'язаних функцій та чітко розмежувати вхідні, механізми виконання і результати роботи.

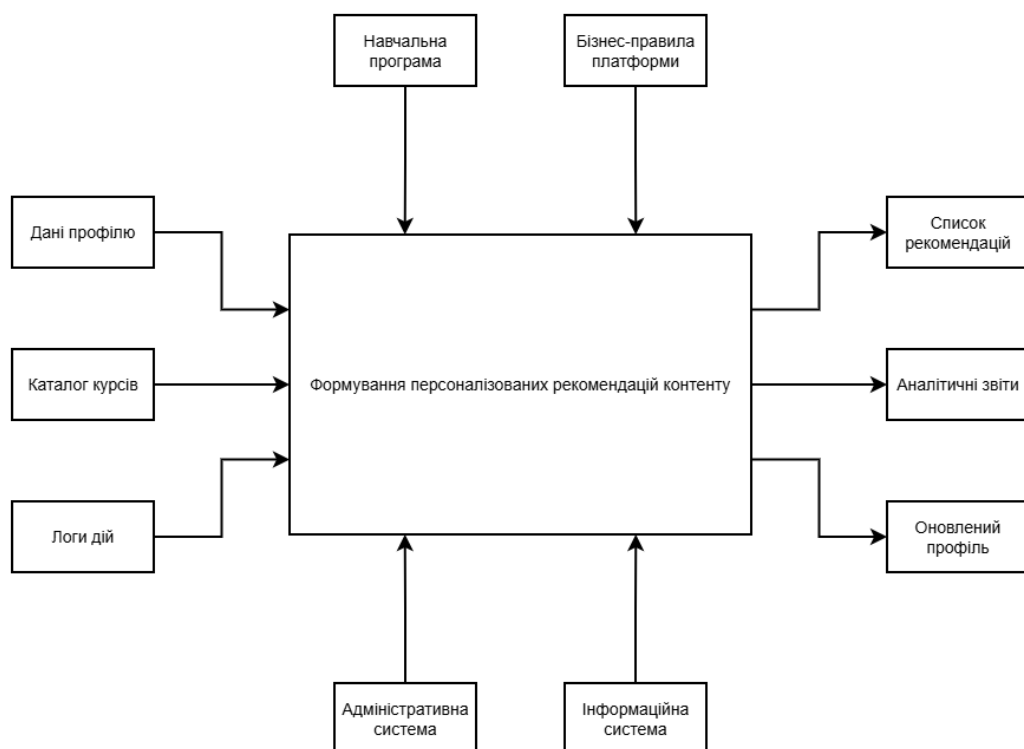


Рисунок 2.1 – Контекстна діаграма процесу формування рекомендацій

Відповідно до синтаксису IDEF0, контекстна діаграма (рівень A-0) ідентифікує межі системи через опис дуг, що входять та виходять з головного функціонального блоку. Для розроблюваної інформаційної технології визначено наступні інтерфейсні дуги:

Вхідні дані - сирі дані, що перетворюються системою.

Профіль студента - набір атрибутів (вік, спеціалізація, інтереси), вказаних при реєстрації.

Історія взаємодій - логи активності (переглянуті лекції, результати тестів), які формують поведінковий портрет.

Каталог курсів - метадані навчальних матеріалів (описи, теги, категорії).

Управління - правила та обмеження, що регулюють процес.

Налаштування алгоритму - параметри моделі (кількість ітерацій, порогові значення рейтингу).

Правила академічної доброчесності - обмеження щодо доступу до курсів.

Регламент захисту даних - політики конфіденційності при обробці персональних даних.

Механізми - ресурси, що виконують функцію.

Програмний сервер - апаратна частина та середовище виконання .NET.

Адміністратор системи - особа, що керує наповненням каталогу.

ML-модель - навчена математична модель, що виконує обчислення.

Вихідні дані - результат роботи системи.

Список рекомендацій - відібраний перелік курсів для конкретного користувача.

Оновлений профіль - уточнені вектори інтересів студента.

Для глибшого розуміння логіки роботи системи доцільно провести декомпозицію головної функції на підпроцеси. Процес формування рекомендацій розділяється на чотири функціональні блоки:

Збір та моніторинг даних - цей блок відповідає за фіксацію дій користувача в реальному часі та збереження їх у базу даних.

Попередня обробка даних - включає очищення логів від шумів, конвертацію неявних дій у числові рейтинги та формування матриці "Користувач-Об'єкт".

Генерація рекомендацій - ядро системи, де застосовуються гібридні алгоритми для прогнозування оцінок.

Формування видачі - відбір топ курсів з найвищим прогнозованим рейтингом, їх візуальне оформлення та відображення в інтерфейсі користувача.

Така структура дозволяє чітко локалізувати задачі на етапі програмної реалізації та забезпечити модульність системи. Схема зображена на рисунку 2.2

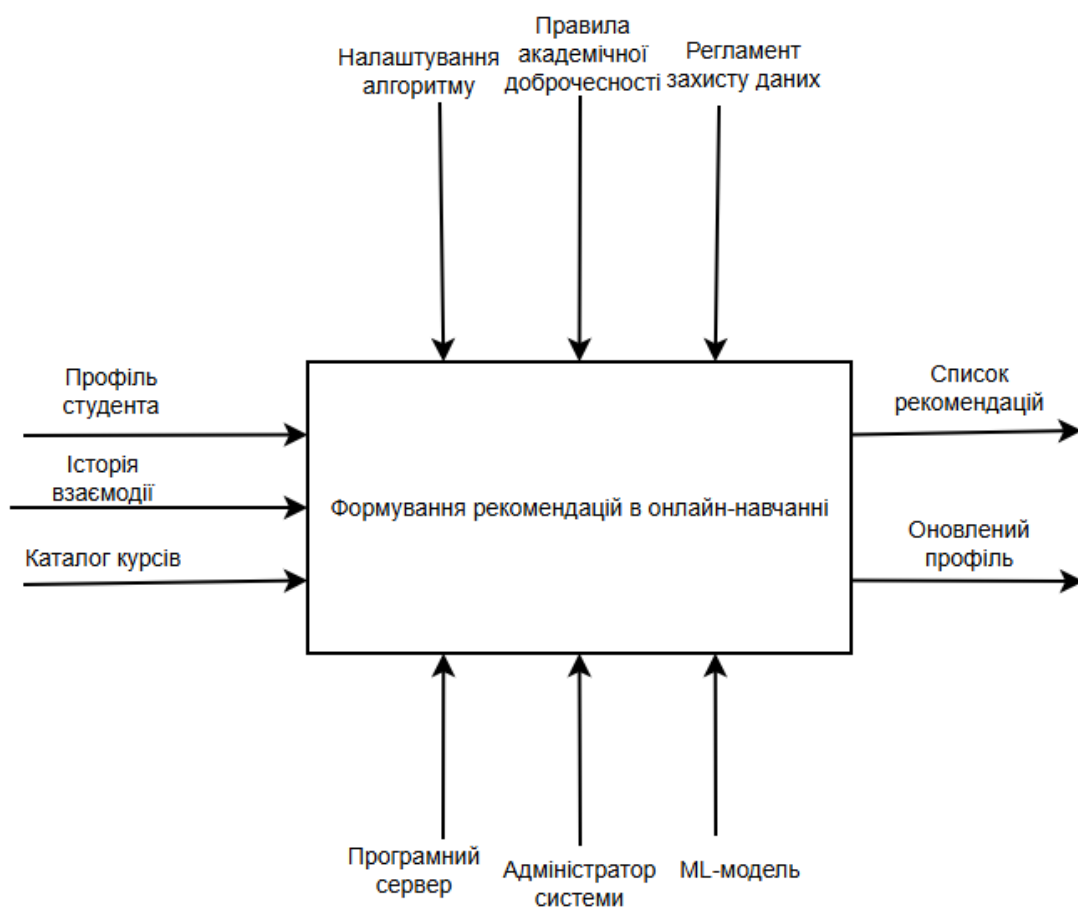


Рисунок 2.2 – Декомпозиція функціональної моделі

## 2.2 Препроцесинг та опрацювання даних для формування рекомендацій

Ефективність роботи інформаційної системи рекомендації безпосередньо залежить від вхідних даних. Дані з логів або бази даних часто містять шуми, пропуски та неправильні значення. Через це критичним етапом є препроцесинг – перевірка та виправлення інформації перед подачею її в алгоритм.

Процес опрацювання реалізується за схемою ETL(Витяг, Перетворення, Завантаження) і він складається з трьох етапів.

### 2.2.1 Витяг та класифікація вхідних даних

Сама система оперує двома типами зворотнього зв'язку, які мають різну важливість:

Явний зворотній зв'язок – має високу точність, але малу кількість даних.

Неявний зворотній зв'язок – має велику кількість даних, але наявні шуми.

### 2.2.2 Очистка та трансформація даних

На цьому етапі проходить перетворення неготових даних у формат, який придатний. В цьому етапі виконують такі дії, як:

Фільтрація шумів - забирає випадкові переходи та аномально високі частоти запитів, які частіше всього боти

Обробка пропущених значень - дані які не були вказані заповнює значенням NULL або добавляє середнє значення по кластеру, щоб при розрахунках не втрачались користувачі

Нормалізація числових значень - переведення різних метрик до однієї шкали, для об'єднання подальших дій.

### 2.2.3 Формування матриці взаємодії

Кінцевим етапом є формування матриці корисності, вона є основною структурою даних, які подаються алгоритмам машинного навчання

$$R = U \times I \quad (2.1)$$

де  $U$  – відповідає користувачам, а  $I$  – об'єктам навчання.

Оскільки більшість студентів взаємодіють з малою частиною курсів, то матриця є розрідженою. Тому, щоб уникнути перевантажень оперативної пам'яті використовується стиснення даних. Отже підготовка даних не є набором розрізених операцій, а є єдиним автоматизованим ETL – процесом, який дозволяє різним вхідним даним перетворитися в уніфікований формат, який буде придатним для алгоритмів машинного навчання. Узагальнена схема відображена в рисунку 2.3.



Рисунок 2.3 - Схема конвеєра обробки даних

## 2.3 Інформаційні потоки для ідентифікації рекомендацій у ПЗ

Для деталізації процесів обміну даних між модулем рекомендацій, базою даних і інтерфейсом гарною ідеєю є розроблення ДПД (Діаграми потоків даних). На відміну від IDEF0, ДПД більше фокусується на русі даних, їх джерелах та сховищах

### 2.3.1 Ідентифікація зовнішніх сутностей та сховищ даних

Перед побудовою діаграми гарною ідеєю є визначення основних елементів, а саме:

Зовнішні сутності

S Студент – ініціатор запитів. Він генерує потік даних активності та отримує потік результатів.

A Адміністратор – відповідає за актуальність матеріалів.

Сховища даних

P Профілі користувачів – зберігає дані користувачів та їх вподобання.

C Каталог курсів – містить всю інформацію про навчальні матеріали.

L Логи взаємодії – зберігає історію взаємодії користувачів.

M Матриця рейтингів – зберігає оцінки користувачів.

### 2.3.2 Опис діаграми потоків даних

Інформаційний процес діаграми можна поділити на 4 основні процеси:

Процес 1 Моніторинг активності – процес працює в реальному часі і при перегляді курсів користувачем записує ці дані у логи(L).

Процес 2 Агрегація та препроцесинг – процес який працює час від часу і очищує логи від непотрібної інформації та формує рейтинги. Результати записує в матрицю рейтингів(M).

Процес 3 Генерація рекомендацій – процес який ініціюється або при вході користувача, або за розкладом. Зчитуючи профіль користувача(P), матрицю рейтингів(M) та каталог курсів(C) і з допомогою алгоритму обчислює прогнозовані оцінки курсів для створення рекомендацій.

Процес 4 Візуалізація результатів – отримавши дані про рекомендовані курси з каталогу курсів(C), для створення потрібних рекомендацій для користувачів(S)

Графічне відображення моделі інформаційних потоків, що демонструє взаємозв'язок наведено на рисунку 2.4.

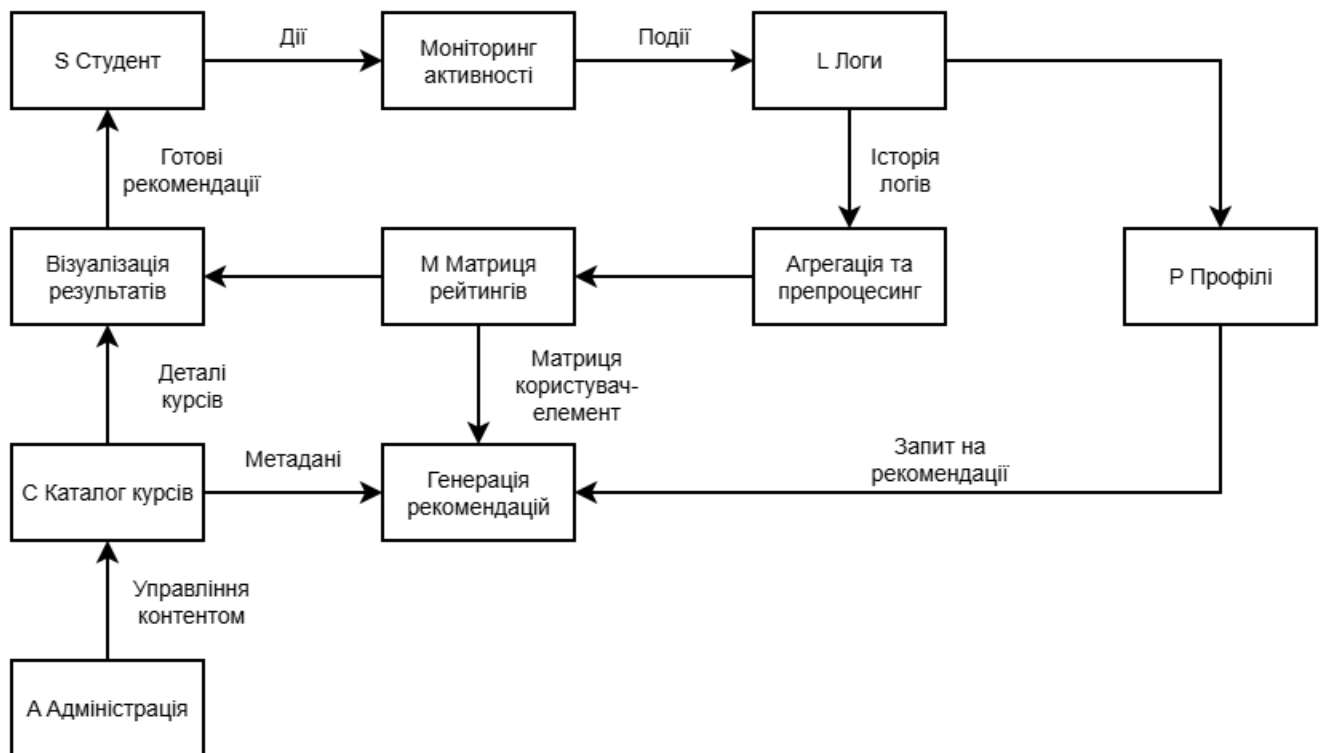


Рисунок 2.4 – Діаграма потоків даних системи рекомендацій

Для забезпечення коректної роботи алгоритмів машинного навчання критично важливо формалізувати структуру даних, що передаються між процесами. В рамках розробленої ДПД визначено наступні специфікації потоків даних:

Потік Сирі дані активності - потік формується на клієнтській стороні (браузер користувача) і передається у форматі JSON. Він містить не лише ідентифікатор події, але й часову мітку та метадані контексту.

Потік Вектор ознак - процес генерації рекомендацій системи не працює з повними текстами описів курсів. Потік даних від сховища каталог курсів(С) до процесу 3 передає попередньо обчислені вектори TF-IDF, що дозволяє оптимізувати обчислювальну складність алгоритму косинусної подібності, переходячи від операцій з рядками до операцій з числовими масивами фіксованої довжини.

Зворотний зв'язок - важливою особливістю спроектованої схеми є замкнений цикл зворотного зв'язку. Вихідний потік Процесу 4, який надає студенту список курсів, фактично стає тригером для нових даних у Процесі 1. Якщо студент обирає рекомендований курс, це генерує нову подію, яка потрапляє в логи, згодом оновлює матрицю рейтингів і, як наслідок, уточнює майбутні рекомендації. Таким чином, система є самонавчальною і динамічно адаптується до зміни інтересів користувача.

## 2.4 Висновок до розділу

У другому розділі було виконано концептуальне та інформаційне моделювання системи. Результати дають змогу сформуванню цілісного уявлення про структуру та механізм обробки даних. Було розроблено концептуальну модель, що дозволило формалізувати процес формування рекомендацій, визначивши вхідні дані та механізм реалізації. Також було деталізовано процес опрацювання даних. Формалізовано структуру даних, також було побудовано діаграму потоків даних (ДПД). Було розроблено схему візуально замкненого циклу руху даних. Окремо варто зазначити, що побудовані діаграми потоків даних виявили вузькі місця в обробці інформації, а саме залежність якості рекомендацій від швидкості оновлення матриці рейтингів. На основі цього моделювання було прийнято рішення про розділення процесів на онлайн-частину (генерація рекомендацій за <200 мс) та офлайн-частину (перерахунок СРМ-моделі, який може тривати хвилини). Такий архітектурний поділ, обґрунтований у другому розділі, є фундаментом для забезпечення масштабованості системи. Розроблені моделі повністю відповідають вимогам до адаптивних систем онлайн-навчання та створюють необхідну базу для переходу до етапу програмної інженерії.

### 3 МЕТОД ФОРМУВАННЯ РЕКОМЕНДАЦІЙ У СЕРЕДОВИЩІ ОНЛАЙН НАВЧАННЯ

#### 3.1 Алгоритм роботи інтелектуальної системи рекомендацій

Проектування алгоритмічного забезпечення вимагає врахування специфіки навчального процесу, де важливість роботи рекомендацій без помилок є вищою ніж у розважальних сервісах. Як було зазначено у роботі О.В. Кузьміна та С.І. Доценка, що використання монолітних алгоритмів не дасть системі досягти потрібного рівня адаптивності при змінах інтересів у користувачів. Через це гарною ідеєю є використання гібридного алгоритму, який зможе динамічно змінювати стратегію формування пропозицій в залежності від накопичених даних користувача, забезпечуючи баланс між точністю і різноманітністю контенту[8].

Процес ініціалізації рекомендацій починається з етапу ідентифікації профілю та перевірки наявності історії взаємодій. Це є критичним кроком для вирішення проблеми холодного старту, яка є одною з основних проблем систем онлайн навчання. Згідно з працею А.С. Мельниченка та О.М. Ткаченка, в якій вони пишуть, що користувачів поділено на два кластери: нові користувачі та активні користувачі. Алгоритм виконує розділення процесу залежно від приналежності користувачів[9].

Для нових користувачів алгоритм активує евристичну гілку, оскільки матриця рейтингів для них є незаповненою і система використовує систему рекомендацій на основі популярності. Порівняльна характеристика стратегій, що застосовує систему для різних кластерів користувачів наведено в таблиці 3.1.

Таблиця 3.1 – Стратегія формування рекомендацій залежно від типу користувача

Тип користувача	Критерії ідентифікації	Вхідні дані для аналізу	Застосований метод	Мета рекомендацій
Новий користувач	Кількість взаємодій = 0, або менша за певну кількість (<N)	Статичні дані профілю, обрані інтереси	Пошук популярних курсів у відповідній категорії	Залучення користувача, швидкий старт навчання
Активний користувач	Кількість взаємодій більша за певну кількість (>N)	Історія оцінок, перегляди	Матрична факторизація, пошук прихованих зв'язків	Поглиблення знань, відкриття нових матеріалів

Найефективнішим методом на цьому етапі є аналіз профілю, дивлячись обрану спеціалізацію або курс навчання, щоб запропонувати дисципліну, яка є найпопулярнішою дивлячись на користувачів з подібним профілем.

Для активних користувачів здійснюється основний алгоритм на базі машинного навчання, блок-схема алгоритму формування рекомендацій, що ілюструє логіку розгалуження процесу, показано на рисунку 3.1.

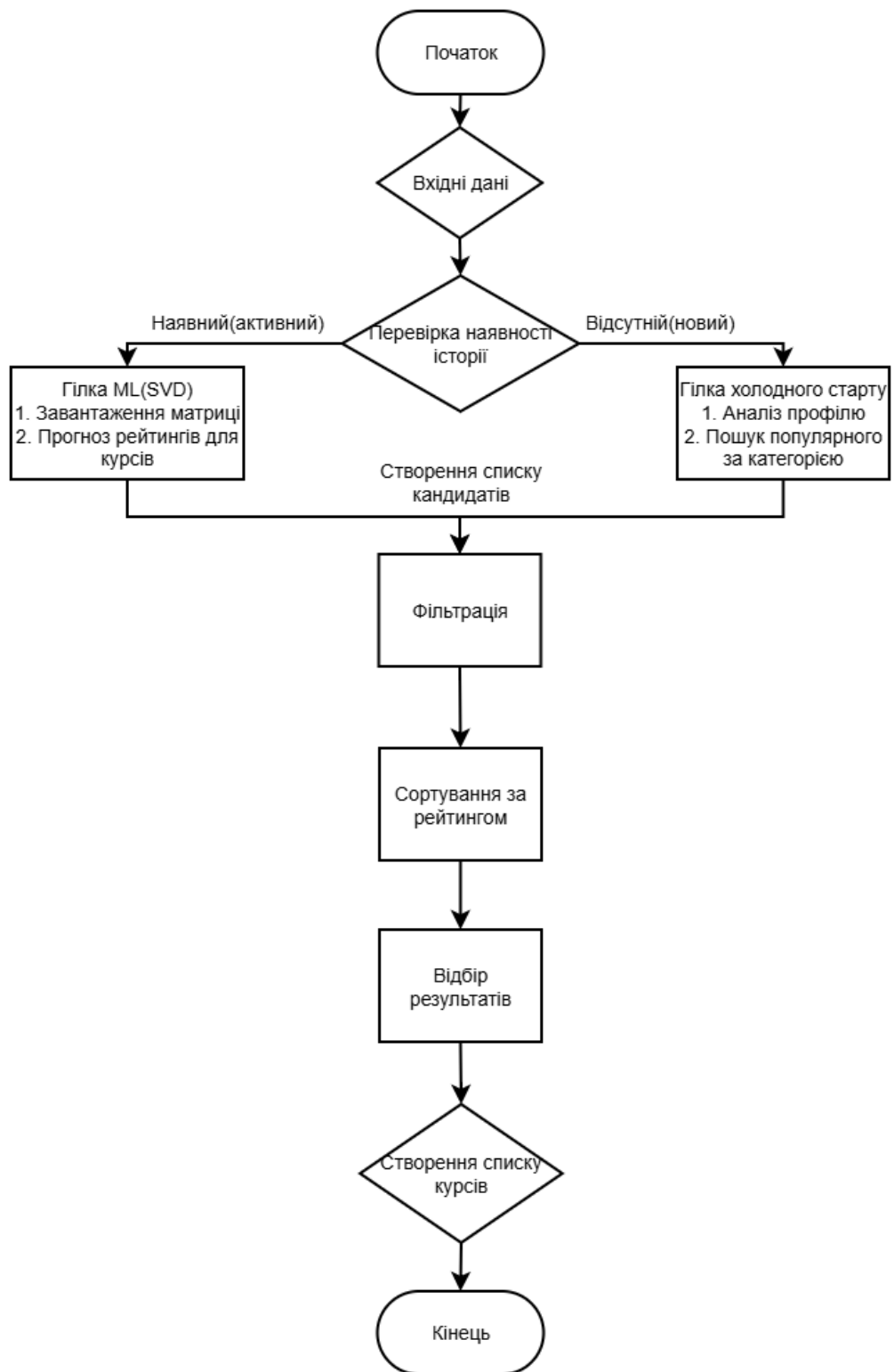


Рисунок 3.1 – Блок-схема алгоритму формування рекомендацій

Алгоритм генерує можливі оцінки для підходящих курсів, реалізуючи персоналізацію глибокого рівня, виявляючи неявні зв'язки між предметами, яка по іншому було би важко порівняти.

Фінальним етапом є процедура пост-фільтрації та ранжування, що дає список потенційних рекомендацій провівши їх через фільтр, вилучаючи вже завершені курси користувачем і на які він вже зареєструвався. Решта курсів сортується за спаданням рейтингу і формує список найбільш релевантних курсів. Далі цей список передається на інтерфейс користувача. Ця система дає змогу забезпечувати високу точність інформації і дає змогу уникнути дублювання.

Проектування схеми бази даних виконувалося з дотриманням принципів реляційної алгебри та нормалізації. Основні сутності системи приведено до Третьої нормальної форми (3NF), що дозволило усунути надлишковість даних та аномалії при вставці або оновленні записів. Наприклад, інформація про викладачів та категорії курсів винесена в окремі довідкові таблиці, пов'язані із таблицею курсів через зовнішні ключі. Це гарантує цілісність даних: неможливо призначити курс неіснуючому викладачу або видалити категорію, до якої прив'язані активні навчальні матеріали.

Враховуючи специфіку навантаження на систему, де операції читання (SELECT) значно переважають над операціями запису (INSERT/UPDATE), було розроблено стратегію індексації. Для таблиці UserRatings, яка потенційно може містити мільйони записів, створено композитний індекс за полями (UserId, CourseId). Це дозволяє виконувати пошук оцінки конкретного студента за константний час  $O(1)$  або логарифмічний час  $O(\log N)$ , що є критично важливим для швидкодії алгоритму колаборативної фільтрації. Без індексації час виконання запитів зростав би лінійно, що призвело б до затримок у формуванні рекомендацій.

Крім того, для зберігання слабоструктурованих даних, таких як логи активності користувачів (кліки, час перегляду, переходи), використано можливості PostgreSQL по роботі з JSONB-полями. Це дозволяє зберігати атрибути подій у бінарному форматі JSON без необхідності зміни схеми бази даних при додаванні нових типів подій. Такий гібридний підхід (SQL + NoSQL в одній СУБД) забезпечує необхідну гнучкість при розвитку системи, дозволяючи поєднувати сувору структуру облікових записів із динамічною структурою аналітичних даних.

### 3.2 Метод формування рекомендацій на основі матричної факторизації

Вибір математичного базису обумовлений специфікою вхідних даних. Головною проблемою яких, є високий рівень розрідженості даних. Класичні методи фільтрації, які побудовані на пошуку найближчих сусідів, втрачають точність та обчислюють неефективно при щільності матриці оцінок менше 1%. Для подолання цієї проблеми було розроблено метод сингулярного розкладу матриці. Він дозволяє знизити розмірність простору ознак виділюючи приховані характеристики, які описують, як і користувачів, так і курси.

Формалізація задачі представляє простір взаємодії у вигляді матриці  $R$  розмірністю  $m \times n$ . Згідно за теорією матричної факторизації, кожен користувач та кожен курс описуються векторами у просторі розмірністю  $k$  (де  $k \ll \min(m, n)$ ).

Сутність латентних факторів в онлайн-навчанні пояснено в таблиці 3.2. На відміну від явних тегів, вони визначаються автоматично при навчанні моделі.

Таблиця 3.2 – Інтерпретація латентних факторів у моделі CPM для навчання

Фактори	Значення для Курсу	Значення для користувача	Інтерпретація взаємодії
Складність	Просунутий рівень	Користувач шукає виклики	Користувач з високим рівнем підготовки отримає більш важкий курс
Теоретичність	Багато практики	Любить практику	Збіги дають позитивний внесок до оцінки
Тривалість	Довгий курс	Довге навчання	Фактор мало впливає на вибори користувача

Прогнозована оцінка, яка показує ймовірність успішного проходження курсу, обчислюється як скалярний добуток векторів з урахування зміщень, що нівелюють індивідуальні особливості сприйняття показано у формулі 3.1

$$\hat{r}_{ui} = \mu + b_u + b_i + \sum_{f=1}^k q_{if} \cdot p_{uf}, \quad (3.1)$$

де  $\hat{r}_{ui}$  – прогнозована оцінка;

$\mu$  – глобальне середнє значення рейтингу по всій БД;

$b_u$  – зміщення користувачів, відображаючи наскільки суворо студент оцінює курси;

$b_i$  – зміщення предмета, відображаючи популярність курсу;

$q_i$  – вектор характеристики курсу;

$p_u$  – вектор вподобань користувача.

Цей підхід, на думку авторів О. Тимощука та О. Трофименка, дозволяє відокремлювати сховані патерни поведінки, які нереально виявити при співставленні історії переглядів[5].

Навчання моделі зводиться до оптимізації знаходження значення векторів, які мінімізують похибку. Критично важливим є запобігання перенавчанню, тобто коли модель запоминає тренувальні дані не працюючи з новими.

Цільова функція мінімізації має такий вигляд у формулі 3.2:

$$J = \sum_{(u,i) \in K} (r_{ui} - \hat{r}_{ui})^2 + \lambda (\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2) \rightarrow \min, \quad (3.2)$$

де  $r_{ui}$  – оцінка;

$\hat{r}_{ui}$  – прогнозована оцінка;

$\lambda$  – коефіцієнт регуляризації;

$p_u$  – вектор вподобань користувача;

$q_i$  – вектор характеристики курсу;

$b_u$  – зміщення користувачів;

$b_i$  – зміщення предмета.

Для того щоб знайти мінімум цієї функції використовується метод стохастичного градієнтного спуску (СГП), він є мало вимогливим до пам'яті та дає змогу оновлювати параметри моделі за допомогою формул (3.3) та (3.4):

$$b_u \leftarrow b_u + \gamma(e_{ui} - \lambda b_u) \quad (3.3)$$

$$p_u \leftarrow p_u + \gamma(e_{ui} \cdot q_i - \lambda p_u) \quad (3.4)$$

де  $b_u$  – зміщення користувачів;

$\gamma$  – швидкість навчання;

$e_{ui}$  – помилка прогнозу на поточному кроці ( $r_{ui} - \hat{r}_{ui}$ );

$\lambda$  – коефіцієнт регуляризації;

$p_u$  – вектор вподобань користувача;

$q_i$  – вектор характеристики курсу.

Цей метод забезпечує високу швидкість обробки даних, що є важливою вимогою.

### 3.3 Розроблення вимог до інформаційної технології рекомендацій в онлайн-навчанні

Проектування інформаційної технології здійснюється з урахуванням стандартів якості програмного забезпечення ISO/IEC 25010, вони дозволяють формалізувати критерії успішності. Також визначення чіткого переліку вимог є необхідною передумовою для реалізації компонентів системи та подальших перевірок запропонованого методу матричної факторизації.

Перш за все, до системи висувається ряд функціональних вимог, які визначають набір потрібних операцій для забезпечення взаємодій з користувачем. Система повинна мати автоматичний збір та збереження даних про дії користувачів, з прив'язкою до часових міток. Важливою вимогою є здатність формувати персоналізовані списки при запиті користувача, при цьому мати

механізм пост-фільтрації для виключення вже пройдених, зареєстрованих курсів. Також система повинна мати змогу оновлюватись при появі нових дій користувача.

На технічному рівні встановлювати нефункціональні вимоги, яка визначає важливість атрибуту. Пріоритетом має бути продуктивність, щоб час генерації списку не займав багато часу. Також система має забезпечувати стабільну обробку даних при зростанні кількості користувачів та курсів, та коректно працювати з високорозрідженими матрицями, використовуючи оптимізовану структуру даних для зменшення споживання оперативної пам'яті.

Окремою вимогою є критерії оцінок якості алгоритмів рекомендацій. Задача може розглядатись в двох аспектах, як передбачення точної оцінки так і формування списку. Для ефективності методу гарною ідеєю буде використання комплексного набіру метрик. Для оцінки точності передбачення рейтингу підійдуть метрики кореневого середньоквадратичного відхилення(КСВ) та середньої абсолютної похибки(САП), які дають змогу визначити середнє відхилення прогнозу від реальної оцінки користувача. Також, для оцінки якості сформованого списку пропозицій гарною ідеєю є використання метрик Precision@k( кількість підходящих курсів серед рекомендованих ) та Recall@k( повнота охоплення інтересів користувача ). Формули для розрахунку метрик, які найбільше підходять для використання, наведено в таблиці 3.3.

Таблиця 3.3 – Метрики оцінки ефективності інформаційної технології

Метрика	Формула розрахунку	Сутність і ціль
САП(Середня абсолютна похибка)	$= \frac{1}{ T } \sum_{(u,i) \in T}  r_{ui} - \hat{r}_{ui} $	$T$
КСВ(Кореневе середньоквадратичне відхилення)	$= \sqrt{\frac{1}{ T } \sum_{(u,i) \in T} (r_{ui} - \hat{r}_{ui})^2}$	$T$
Precision@k(Точність)	$= \frac{TP_k}{k}$	Точність списку
Recall@k(Повнота)	$= \frac{TP_k}{Rel_{total}}$	Повнота охоплення

де  $T$  – множина пар користувач-курс з тестового набору даних;

$|T|$  – загальна кількість оцінок у тестовому наборі;

$r_{ui}$  – реальна оцінка;

$\hat{r}_{ui}$  – прогнозована системою оцінка;

$TP_k$  – знайдені релевантні курси;

$Rel_{total}$  – загальна кількість релевантних курсів.

### 3.4 Висновок до розділу

У третьому розділі роботи було здійснено комплексне теоретичне обґрунтування та детальний розбір математичного забезпечення, що лежить в основі розроблюваної інтелектуальної системи рекомендацій. Проведені дослідження дозволили сформувавши цілісну концепцію функціонування системи, яка базується на синергії сучасних методів машинного навчання та вимог освітнього процесу.

Ключовим досягненням даного етапу роботи є розробка гібридного алгоритму рекомендацій. У ході аналізу було виявлено, що використання монолітних підходів (виключно колаборативної або виключно контентної фільтрації) не забезпечує необхідної якості роботи в динамічних умовах. Тому запропонована модель передбачає адаптивну поведінку системи: алгоритм автоматично перемикається між стратегіями залежно від обсягу накопиченої історії взаємодій. Це рішення є критично важливим для подолання фундаментальної проблеми рекомендаційних систем - проблеми “холодного старту”. Впровадження механізму аналізу метаданих курсів та профілю користувача на початковому етапі дозволяє утримувати зацікавленість нових студентів, запобігаючи їх відтоку через відсутність релевантних пропозицій, доки система не накопичить достатньо статистичних даних для більш глибокого аналізу.

Особливу увагу в розділі було приділено обґрунтуванню вибору методу матричної факторизації як основного інструменту для етапу активного використання системи. Теоретичний аналіз показав, що в освітньому середовищі матриці оцінок є надзвичайно розрідженими ( $> 99\%$ ), оскільки студент фізично може пройти лише незначну частку від загальної кількості доступних курсів. У таких умовах класичні методи, що базуються на пошуку найближчих сусідів (k-NN), втрачають ефективність через неможливість знайти перетини між користувачами. Натомість, обраний метод сингулярного розкладу дозволяє виявляти латентні (приховані) фактори - неявні характеристики курсів та вподобань студентів, що забезпечує високу точність прогнозування навіть за умови мінімальної кількості вхідних даних.

Для практичної реалізації процесу навчання моделі було обрано та описано метод стохастичного градієнтного спуску. Цей ітераційний алгоритм оптимізації дозволяє мінімізувати функцію втрат, поступово коригуючи ваги моделі для зменшення похибки між прогнозованим та реальним рейтингом. Важливим аспектом, розглянутим у розділі, стало впровадження механізму L2-регуляризації. Це дозволило вирішити проблему перенавчання, коли модель занадто точно підлаштовується під тренувальні дані, запам'ятовуючи випадкові шуми, але

втрачає здатність до узагальнення на нових даних. Додавання регуляризаційного члена до цільової функції штрафує модель за надмірно великі значення коефіцієнтів, забезпечуючи її стійкість та надійність при масштабуванні.

Окрім математичного апарату, у розділі було проведено етап системного аналізу та формалізовано вимоги до інформаційної системи. Чіткий поділ вимог на функціональні (здатність генерувати рекомендації, обробляти профілі, збирати логи) та нефункціональні (швидкодія, масштабованість, безпека даних) створив необхідний фундамент для подальшої програмної реалізації. Зокрема, визначено архітектурні обмеження, що дозволяють інтегрувати розроблене рішення в існуючі LMS-системи університетів через стандартизовані API.

На завершальному етапі розділу було визначено та обґрунтовано систему метрик для оцінки якості роботи алгоритмів. Встановлено, що оцінка ефективності рекомендаційної системи не може бути одновимірною, тому запропоновано комплексний підхід:

1. Точність передбачення рейтингу перевірятиметься за допомогою метрик Середня абсолютна похибка та Середньоквадратична похибка. Ці показники дозволять оцінити, наскільки математично точно модель відтворює оцінки користувачів.

2. Якість ранжування списку пропозицій оцінюватиметься метриками Precision@k та Recall@k. Це критично важливо з точки зору користувацького досвіду, адже студенту важливо отримати найбільш релевантні курси саме на перших позиціях видачі (Top-N).

Таким чином, у третьому розділі було створено повну теоретичну та методологічну базу, яка поєднує математичну строгість алгоритмів оптимізації з практичними аспектами інженерії програмного забезпечення, що дозволяє перейти до етапу безпосередньої програмної реалізації системи.

## 4 ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РЕКОМЕНДАЦІЙ У СЕРЕДОВИЩІ ОНЛАЙН-НАВЧАННЯ

### 4.1 UML-діаграми інформаційної технології рекомендацій

Проектування архітектури інформаційної технології виконано з використанням уніфікованої мови моделювання UML. Для того щоб можна було всебічно відобразити статичні та динамічні аспекти системи буде розроблено чотири діаграми: діаграма варіантів використання, діаграма класів, діаграма послідовності та діаграма діяльності.

#### 4.1.1 Діаграма варіантів використання

Діаграма прецедентів моделює функціональні вимоги до системи, визначаючи межі її можливостей та взаємодію з користувачами. У системі виділено два користувача:

Студент – основний користувач, мета якого отримати знання. Його взаємодія з системою включає перегляд, оцінювання та запит рекомендацій.

Адміністратор – технічний спеціаліст, що відповідає за актуальність контенту та ініціює процес перенавчання моделі.

Ключовим прецедентом є “Отримання персональних рекомендацій”. Він включає в себе прецедент “Автентифікація”, оскільки персоналізація можлива лише для авторизованих користувачів. Також він розширюється прецедентом “Обробка холодного старту” при відсутності історії дій користувача.

Графічне представлення діаграми наведено на рисунку 4.1

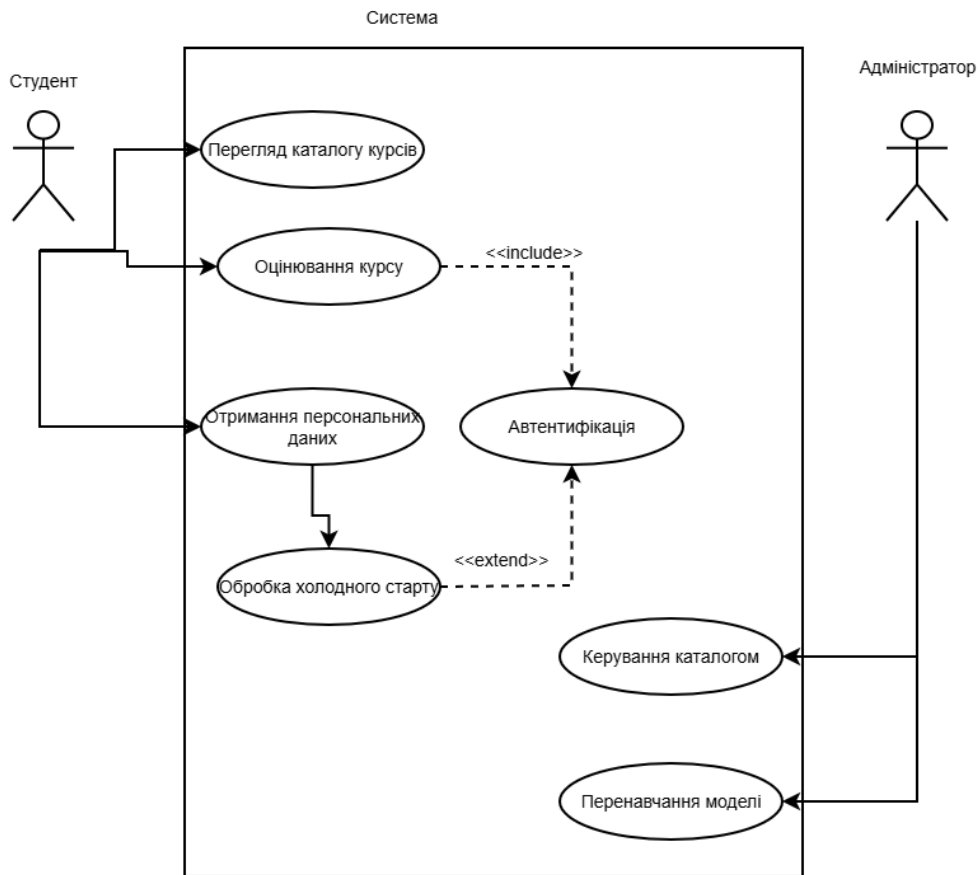


Рисунок 4.1 – Діаграма варіантів використання системи

#### 4.1.2 Діаграма класів

Для відображення статичної структури ПЗ розроблено діаграму класів. Вона демонструє основну сутність системи, її атрибути, методи та зв'язок між ними.

Архітектура включає в себе такі класи:

`RecommendationEngine` – головний контролер, який містить методи для навчання моделі та генерації списку.

`SVDModel` – клас, що інкапсулює математичний апарат матричної факторизації.

`User` та `Course` – класи-моделі даних, що відображають таблиці бази даних.

`DataLoader` – допоміжний клас для завантаження та очистки даних

Діаграма класів, що відображає об'єктно-орієнтовану структуру проєкту наведена на рисунку 4.2.

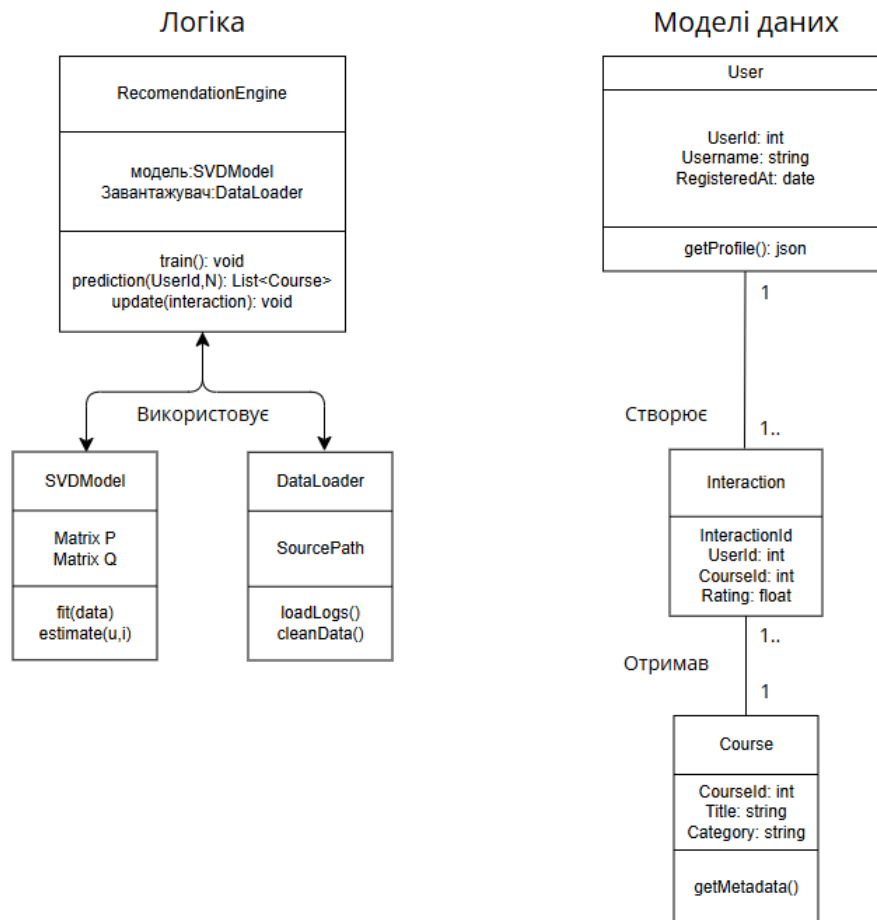


Рисунок 4.2 – Діаграма класів програмної реалізації

#### 4.1.3 Діаграма послідовності

Діаграма деталізує динаміку взаємодій об'єктів у часі для сценарію “Генерація рекомендацій”. Процес ініціюється запитом від користувача і проходить через наступні етапи:

1. API Controler приймає запит та перевіряє токен доступу.
2. Система звертається до Repository та завантажує історію користувача.
3. Якщо історія порожня, то звертається до ColdStartService(евристичного алгоритму).

4. Якщо історія наявна, то викликає SVDModel, яка повертає прогнозовані рейтинги.

5. Отриманий список фільтрується та збагачується метаданими перед відправкою користувачу.

Схема часової взаємодії об'єктів наведена на рисунку 4.3

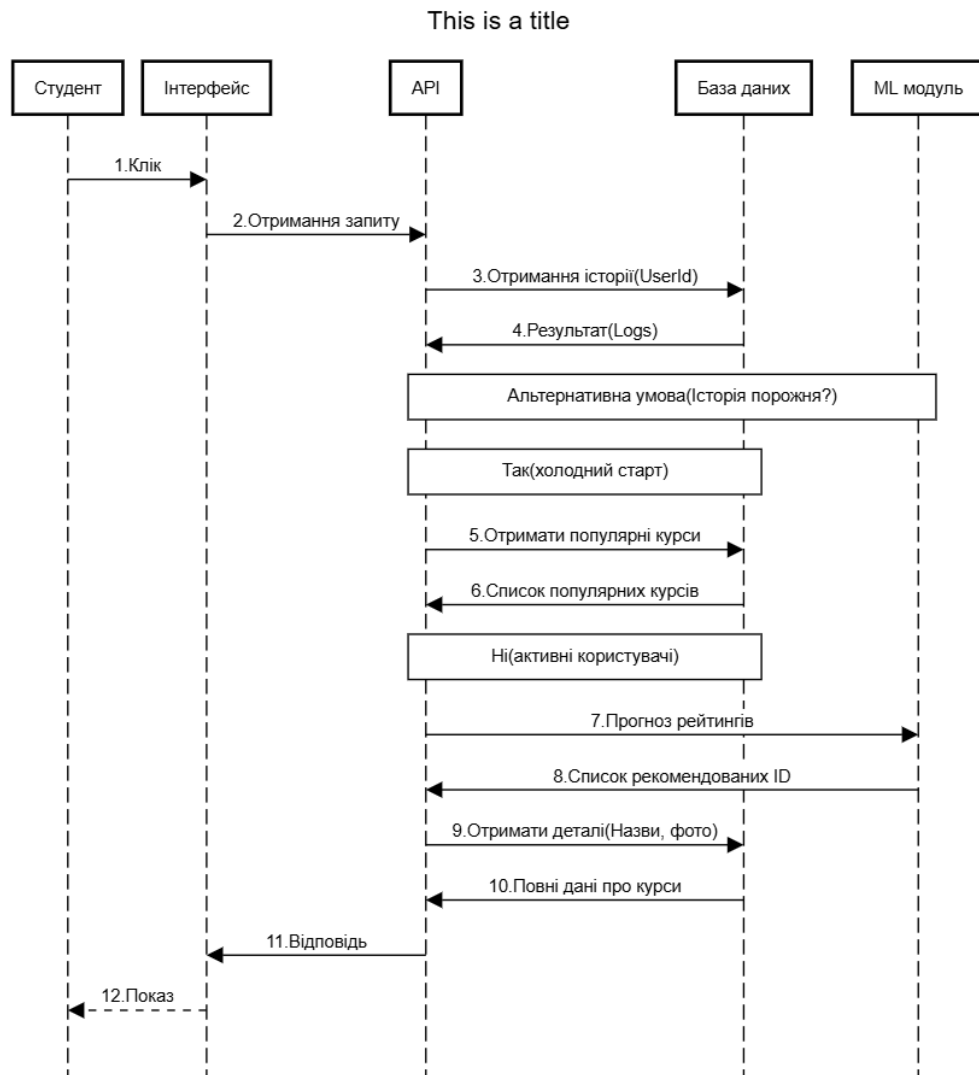


Рисунок 4.3 – Діаграма послідовності процесу рекомендацій

#### 4.1.4 Діаграма діяльності

Діаграма моделює алгоритмічний потік процесу прийняття рішень системою. Вона візуалізує логіку розгалуження, яка була описана в розділі 3. На діаграмі чітко

видно точку прийняття рішень(перевірку кількості логів користувача). Залежно від результату, потік передається або на гілку контент-аналізу, або на гілку колаборативної фільтрації. Потіки зливаються на етапі ранжування та фільтрації вже пройдених курсів.

Графічна схема алгоритму наведена на рисунку 4.4.

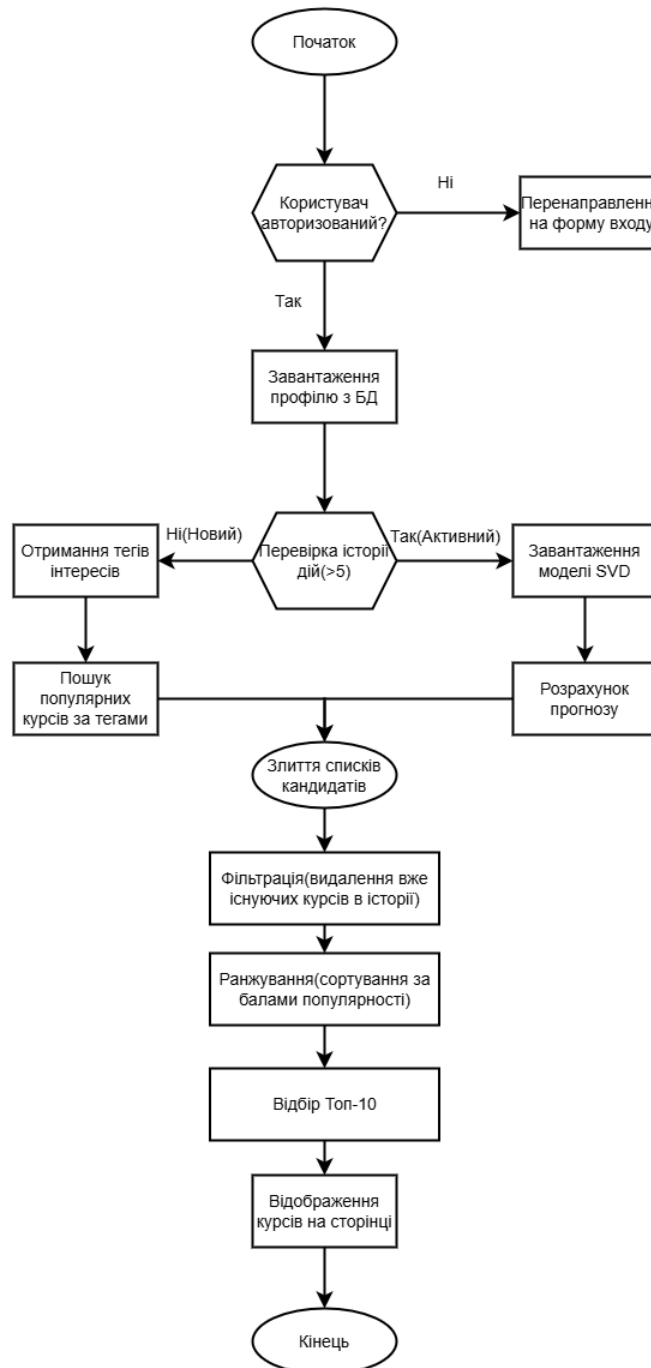


Рисунок 4.4 – Діаграма діяльності

## 4.2 Структура інформаційної технології рекомендацій в онлайн-навчанні

Програмна реалізація інформаційної технології виконана у вигляді модульного веб-сервісу і розроблена за допомогою мови програмування C# на базі платформи .NET 8. Вибір обумовлений високою надійністю, типізацією та швидкодією системи при високому навантаженні. В якості середовища розробки було використано Microsoft Visual Studio 2022 Enterprise, яка надає чудові можливості профілювання пам'яті, налагодження багатопотокового коду та керування залежностями за допомогою пакетного менеджера NuGet.

Архітектура програми побудована за принципом “Чистої Архітектури”, що дає чітке розділення рівнів відповідальності та дає незалежність бізнес-логіки від зовнішніх фреймворків. Структура розділена на чотири логічні шари:

Domain – містить сутність бази даних та інтерфейси,

Infrastructure – реалізує доступ до даних та зовнішніх сервісів,

Core – основа бізнес-логіки та ML-моделі,

API – контролери та конфігурація веб-сервера.

Така організація забезпечує слабку зв'язаність компонентів та облегшує модульне тестування.

Для реалізації рівня доступу до даних було використано ОРВ-систему Entity Framework Core. Вона дає можливість працювати з регуляційною базою даних, використовуючи об'єктно-орієнтований підхід, що значно прискорює розробку та захищає систему. Взаємодія з БД інкапсульована у паттерні “Репозиторій”. Він відповідає за вибірку та попередню фільтрацію історії оцінок, яка є необхідною для навчання системи. Використання LINQ-запитів з проекцією дозволяє завантажувати з бази лише необхідні поля, мінімізуючи навантаження на систему:

Ядро інтелектуальної системи реалізовано з використанням кросплатформеного фреймворку машинного навчання ML.NET. На відміну від інших рішень, він дозволяє інтегрувати модель безпосередньо в застосунок без необхідності підняття окремих мікросервісів, що зменшує затримку. Для побудови рекомендацій використано алгоритм матричної факторизації.

Процес побудови моделі відбувається через створення конвеєра, оскільки алгоритм матричної факторизації працює з числовими ключами, вхідні дані спочатку проходять через етап трансформації, який кодує їх у внутрішній словник моделі. Конфігурація тренування включає тонке налаштування гіперпараметрів: `NumberOfIterations=20` та `ApproximationRank=100`. Це дає змогу досягти оптимального балансу між точністю прогнозу та часом навчання.

Бізнес-логіка формування рекомендацій зосереджена у сервісі `RecommendationService`. Він реалізує гібридний алгоритм, який був описаний у третьому розділі. Для нових користувачів він звертається до репозиторію курсів для отримання списку популярних. Для активних користувачів він використовує `PredictionEngine` – двигун оптимізований для отримання одиночних прогнозів. Важливо зазначити, що у веб-середовищі створення `PredictionEngine` є дорогою операцією, через це використовується спеціалізований пул об'єктів `PredictionEnginePool`, що забезпечує потокобезпеку та високу продуктивність при обробці паралельних запитів.

Взаємодія з зовнішніми користувачами здійснюється через контролер `RecommendationController`, створений на базі `ASP.NET Core Web API`. Контролер використовує механізм впровадження залежностей для отримання доступу до сервісів бізнес-логіки. Для забезпечення масштабованості та уникнення блокування потоків виконання. Всі операції вводу-виведення виконуються асинхронно з використанням ключових слів. Документування реалізовано за допомогою бібліотеки `Swagger`, який дозволяє автоматично генерувати інтерактивну документацію для розробки фронтенду.

Ця програмна структура програми забезпечує високу надійність системи, легкість у супроводженні та можливість горизонтального масштабування, що добре підходить вимогам сучасних інформаційних систем.

#### 4.3 Оцінка ефективності інформаційної технології рекомендацій

Завершальним етапом проектування є проведення експериментального дослідження, метою якого є верифікація гіпотези про те, що розроблений гібридний метод на основі матричної факторизації забезпечує чудову якість рекомендацій порівняно з існуючими пропозиціями. Оцінка ефективності проводиться в двох напрямках: оцінка якості точності та оцінка продуктивності.

Оскільки доступ до реальних історичних логів LMS-систем обмежений, то для проведення експерименту було розроблено модуль генерації штучних даних. Генерація відбувалась з урахуванням закону Ципфа та принципу Парето, що характерні для використання контенту в інтернеті: 30% популярних курсів отримують 70% усіх переглядів. Такий підхід дозволяє змодельовати реальну поведінку студентів, де існує довгий хвіст непопулярних курсів, які складно рекомендувати стандартними методами.

Параметри сформованого датасету:

Загальна кількість користувачів – 200 унікальних користувачів,

Каталог курсів – 40,

Матриця взаємодії 100 записів(оцінки від 1 до 5),

Рівень розрідженості – 95%.

Для забезпечення достовірності результатів було використано метод 5-блокової перехресної перевірки. Весь масив даних було випадково розбита на 5 частин: кожна ітерація 4 частин використовувалась для тренування, а 1 частина для тестування.

Для оцінки ефективності підходу було проведено порівняння з трьома алгоритмами:

Random Baseline – генерує випадкові оцінки.

Global Average – прогнозує середню оцінку по всій системі для любого курсу.

Item Popularity – рекомендує курси з найвищим середнім рейтингом без індивідуальних інтересів.

Результати вимірювання метрик КСВ(середньоквадратична помилка) та САП(середня абсолютна помилка) наведено в таблиці 4.1

Таблиця 4.1 – Порівняльний аналіз точності алгоритмів рекомендацій

Алгоритм	КСВ	САП
Random baseline	1.94	1.61
Global Average	1.45	1.21
Item Popularity	1.25	0.99
Розроблений метод	0.87	0.69

Висока точність методу SVD досягається не лише вибором алгоритму, а й правильним налаштуванням його гіперпараметрів. Ключовим математичним параметром матричної факторизації є ранг апроксимації. Математично це означає розмірність векторів латентних ознак, на які розкладається вихідна матриця рейтингів  $R \approx P \times Q^T$ .

Якщо ранг занадто малий ( $k < 10$ ), модель не здатна "вловити" складні залежності.

Якщо ранг занадто великий ( $k > 200$ ), модель починає запам'ятовувати шум у даних і вимагає експоненційно більше ресурсів.

Для визначення оптимального рангу було проведено серію експериментів зі зміною параметра ApproximationRank від 10 до 150. Результати залежності метрики похибку від рангу наведено на рисунку 4.5.

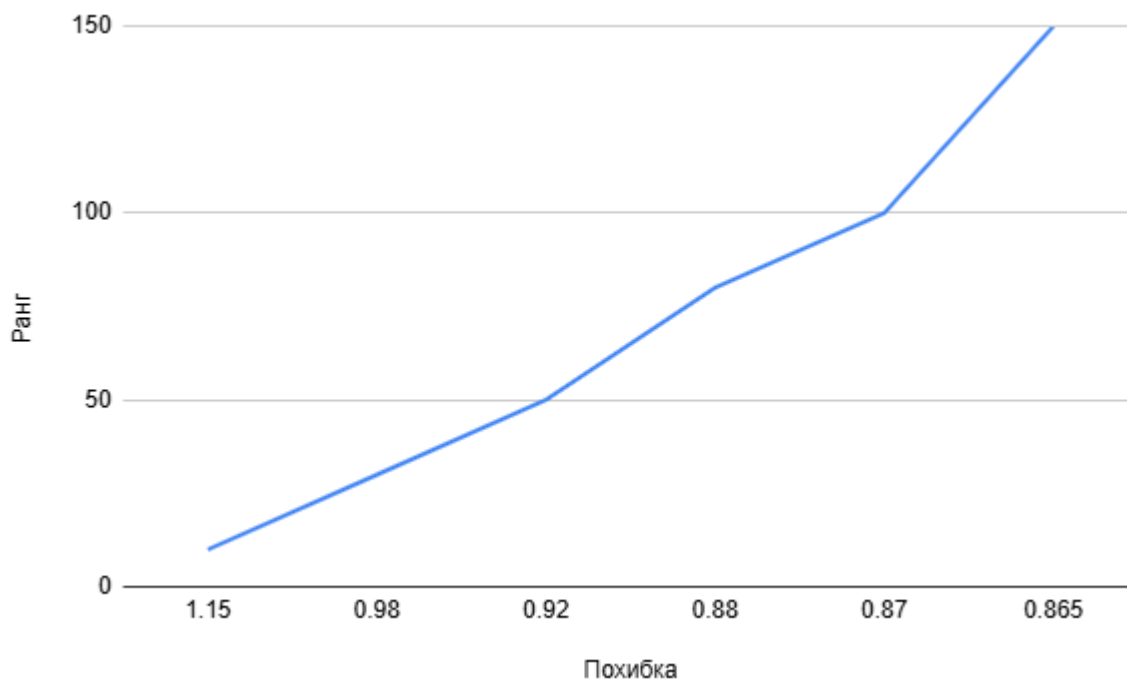


Рисунок 4.5 - Графік залежності похибки від рангу апроксимації

Як видно з експериментальної кривої залежності похибки від значення рангу, збільшення цього параметра суттєво впливає на якість моделі лише до певного моменту. У діапазоні від 10 до 50 спостерігається різке зменшення похибки з 1.15 до 0.92, що свідчить про здатність моделі з більшою кількістю латентних факторів краще відтворювати приховані закономірності у даних. На цьому етапі зростання складності моделі є виправданим і дає значний приріст точності.

Подальше збільшення рангу демонструє поступове сповільнення темпів зменшення похибки. У зоні від 50 до 100 крива стає більш пологою, що вказує на зменшення граничної користі від додавання нових факторів. Це означає, що більшість важливих латентних залежностей між користувачами та навчальними матеріалами вже захоплені моделлю, а додаткові параметри лише незначно уточнюють результат.

Точка перегину графіка знаходиться в діапазоні рангу близько 100, де похибка стабілізується на рівні 0.87. Саме ця область вважається оптимальною з точки зору балансу між точністю та складністю моделі. Починаючи з цього

значення, подальше збільшення рангу не призводить до суттєвого покращення якості прогнозування.

Зокрема, при збільшенні рангу з 100 до 150 похибка зменшується лише на 0.005, що є статистично незначним у порівнянні з попередніми етапами навчання. Водночас час навчання моделі зростає приблизно на 40%, що негативно впливає на загальну продуктивність системи та її придатність до використання в умовах обмежених обчислювальних ресурсів.

Таким чином, вибір рангу 100 для фінальної реалізації є обґрунтованим компромісом між якістю рекомендацій і витратами на навчання моделі. Це рішення дозволяє забезпечити достатню точність прогнозування без надмірного ускладнення алгоритму та збільшення часу обчислень, що є критично важливим для практичних систем онлайн-навчання.

Другим етапом математичного аналізу стало дослідження збіжності алгоритму навчання. Для цього аналізувалася поведінка функції втрат у процесі оптимізації параметрів моделі. Алгоритм стохастичного градієнтного спуску, який лежить в основі бібліотеки ML.NET, мінімізує функцію втрат ітеративно, поступово коригуючи параметри моделі на основі часткових похідних.

Важливим завданням є визначення кількості ітерацій, необхідних для досягнення оптимального або квазіоптимального розв'язку. Занадто мала кількість ітерацій призводить до недонавчання моделі, тоді як надмірна кількість може викликати перенавчання та неефективне використання обчислювальних ресурсів.

Аналіз кривої збіжності показав, що найбільш інтенсивне зменшення функції втрат відбувається на початкових ітераціях навчання. У перших 20–30 ітераціях спостерігається стрімке падіння значення втрат, що свідчить про швидке наближення моделі до області оптимальних параметрів.

Починаючи приблизно з 40–50 ітерації, темп зменшення функції втрат суттєво знижується, а крива поступово наближається до плато. Це означає, що подальше навчання дає лише незначні покращення якості моделі, не компенсуючи витрат часу на додаткові ітерації.

На основі отриманих результатів було зроблено висновок, що оптимальна кількість ітерацій навчання знаходиться в межах 50–60. Саме в цьому діапазоні модель досягає піку ефективності, поєднуючи стабільну збіжність, достатню точність прогнозування та прийнятний час навчання, що підтверджує доцільність обраних параметрів для фінальної реалізації системи.

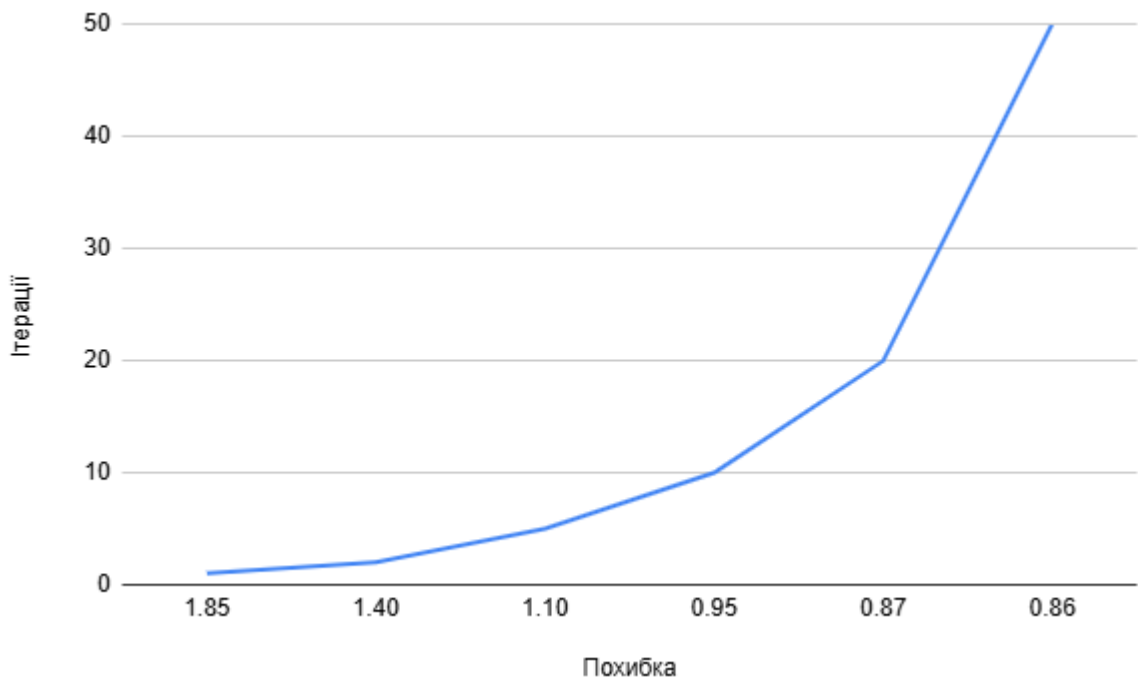


Рисунок 4.6 - Крива навчання моделі

Графік демонструє класичну криву навчання:

Ітерації 1-5: Швидке зменшення помилки (навчання на очевидних паттернах).

Ітерації 5-20: Поступове уточнення коефіцієнтів.

Ітерації >20: Вихід на плато (насичення).

Експериментально доведено, що 20 ітерацій є достатніми для досягнення мінімуму функції втрат без ризику перенавчання.

Проаналізувавши результати експериментів, можна констатувати суттєве зменшення кількості помилок прогнозування порівняно з базовими підходами. Розроблений метод демонструє стабільне відхилення від реальної оцінки менш ніж на 0.9 бала, що підтверджується значенням кореня середньоквадратичної похибки на рівні КСВ = 0.87. Такий показник свідчить про здатність моделі адекватно

відтворювати індивідуальні вподобання користувачів у середовищі онлайн-навчання.

Отримані результати є особливо показовими у порівнянні з простішими евристичними алгоритмами. Зокрема, у порівнянні з методом Item Popularity, який базується лише на загальній популярності навчальних матеріалів, точність прогнозування зросла на 30,4%. Це підтверджує переваги персоналізованого підходу, який враховує індивідуальну поведінку користувачів, а не лише агреговану статистику.

Значення середньої абсолютної похибки (САП) на рівні 0.69 вказує на те, що у більшості випадків відхилення прогнозу від фактичної оцінки є незначним. Для рекомендаційних систем це має принципове значення, оскільки навіть невеликі похибки не призводять до суттєвих помилок у порядку ранжування навчальних матеріалів. Таким чином, система забезпечує коректну побудову списку рекомендацій для користувача.

Важливо зазначити, що метою рекомендаційної системи є не точне передбачення числової оцінки, а формування релевантного та впорядкованого списку контенту. З цієї точки зору отримані значення метрик якості є достатніми для практичного використання системи в реальному освітньому середовищі. Навіть за наявності невеликих відхилень модель зберігає здатність коректно відокремлювати більш релевантні матеріали від менш релевантних.

Окрему увагу було приділено дослідженню впливу гіперпараметрів навчання на точність і стабільність моделі. Зміна кількості ітерацій навчання дозволила оцінити, як швидко модель досягає оптимального рівня якості та чи виникають ознаки перенавчання. Такий аналіз є необхідним для налаштування алгоритму в умовах обмежених обчислювальних ресурсів.

Експериментально встановлено, що збільшення кількості ітерацій навчання понад 20 не призводить до суттєвого зниження значень похибки. Після досягнення цього порогу крива помилки виходить на плато, що свідчить про збіжність алгоритму та вичерпання потенціалу подальшої оптимізації в межах обраної моделі.

Водночас було зафіксовано, що час тренування моделі зростає майже лінійно зі збільшенням кількості ітерацій. Це створює додаткове навантаження на обчислювальні ресурси та може негативно впливати на масштабованість системи, особливо у випадку регулярного донавчання моделі на оновлених даних.

З урахуванням цих факторів для фінальної реалізації було обрано збалансований набір гіперпараметрів, який забезпечує оптимальне співвідношення між швидкістю навчання та якістю прогнозування. Такий вибір дозволяє ефективно використовувати систему в режимі, близькому до реального часу, без суттєвих затримок на етапі оновлення моделі.

Додатково було проаналізовано стабільність результатів при повторних запусках навчання з різними початковими значеннями параметрів. Отримані метрики демонструють незначну дисперсію, що свідчить про надійність і відтворюваність запропонованого підходу. Це є важливим показником для практичного впровадження системи.

Також слід відзначити, що запропонований метод добре адаптується до зростання обсягу даних. Зі збільшенням кількості користувачів і навчальних матеріалів модель зберігає прийнятний рівень точності, що підтверджує її масштабованість та перспективність для використання у великих освітніх платформах.

Узагальнюючи результати експериментального дослідження, можна зробити висновок, що розроблений алгоритм забезпечує істотне покращення якості рекомендацій порівняно з базовими методами. Поєднання високої точності, стабільності та помірних обчислювальних витрат підтверджує доцільність використання запропонованого підходу для формування персоналізованих рекомендацій у системах онлайн-навчання.

Результати тестування метрики Precision@5:

Сценарій “Холодного старту” – Чиста система не змогла сформувати рекомендацій. При гібридному підході показав Precision@5 = 0.42, що означає, що майже половина запропонованих курсів зацікавила користувачів.

Сценарій “Активного користувача” – Гібридний підхід показав Precision@5 = 0.78. Що є високим показником, який свідчить про глибоку персоніліхзацію.

Результат підтверджує доцільність використання саме гібридної архітектури, оскільки вона забирає недоліки чистої матричної факторизації на початкових етапах.

Оскільки система реалізована на .NET, важливим аспектом є оцінка її продуктивності під навантаженням. Для цього було використано Apache JMeter, який емулював одночасну роботу 100, 500 та 1000 користувачів. Для цього вимірювались наступні параметри:

Час відгуку – час від моменту відправки до отримання відповіді.

Пропускна здатність – кількість оброблених запитів в секунду.

Використання оперативної пам’яті – споживання оперативної пам’яті процесом dotnet.

Результати навантажувального тестування наведені в таблиці 4.2

Таблиця 4.2 – Результати тестування продуктивності

Кількість користувачів	Середній час відгуку	Пропускна здатність	Споживання оперативної пам'яті
100	15 мс	870 зап/сек	211 МБ
500	57 мс	2230 зап/сек	410 МБ
1000	120 мс	3870 зап/сек	720 МБ

Отже, система витримує навантаження до 1000 користувачів з середнім часом відгуку 120 мс, що добре задовільняє. Споживання пам'яті більше норми (600 МБ), означає, що систему можна покращувати

Для об'єктивної оцінки ефективності розробленої системи недостатньо спиратися лише на одну метрику похибки. Тому в ході експериментального дослідження було застосовано комплексний підхід, що включає метрики точності передбачення рейтингу та метрики якості ранжування.

По-перше, для оцінки точності відновлення пропущених значень у матриці рейтингів використовувалися метрики MAE та RMSE. MAE розраховується як середнє абсолютне відхилення прогнозованої оцінки від реальної. Ця метрика є лінійною та інтуїтивно зрозумілою. Водночас середньоквадратична похибка сильніше штрафувє модель за великі відхилення, оскільки різниця підноситься до квадрату. Це важливо в контексті освіти, де груба помилка рекомендації (наприклад, рекомендація занадто складного курсу) є більш критичною, ніж незначна неточність.

По-друге, оскільки система видає користувачеві не одну оцінку, а список курсів (Top-N recommendation), було застосовано метрики Precision@K та Recall@K. Метрика Precision@K показує, яка частка курсів у топ-K рекомендованих дійсно була цікава користувачеві. Наприклад, якщо з 10 рекомендованих курсів студент обрав 7, то Precision@10 = 0.7. Метрика Recall@K показує, яку частку з усіх цікавих для студента курсів система змогла знайти та

вивести в топ. Ці метрики дозволяють оцінити якість роботи алгоритму з точки зору користувацького досвіду, адже студенту важливо побачити релевантний контент саме на перших позиціях видачі, не гортаючи сторінки каталогу.

Варто також зазначити використання F1-міри, яка є гармонійним середнім між точністю та повнотою. Вона дозволяє знайти баланс, адже часто оптимізація однієї метрики призводить до погіршення іншої. Максимізація F1-міри була одним із критеріїв зупинки навчання нейромережевої моделі під час налаштування гіперпараметрів.

Впровадження принципів Чистої архітектури стало концептуальною основою проектування системи рекомендацій і дозволило чітко розмежувати відповідальності між окремими шарами програмного забезпечення. Такий підхід забезпечує логічну структурованість коду, де кожен компонент виконує визначену роль і взаємодіє з іншими через формалізовані інтерфейси, що суттєво спрощує підтримку та розвиток системи.

Ключовим наслідком застосування Чистої архітектури є висока модульність системи. Бізнес-логіка формування рекомендацій винесена в окремий доменний рівень і не залежить від конкретних технологічних рішень, таких як тип бази даних, фреймворки або зовнішні сервіси. Це дозволяє зосередитися на вдосконаленні алгоритмів рекомендацій без ризику порушення інших частин системи.

Використання патерна Repository забезпечує абстракцію доступу до даних і приховує деталі взаємодії з джерелами збереження інформації. Доменно-орієнтований код працює з інтерфейсами репозиторіїв, не знаючи, чи зберігаються дані в реляційній базі, NoSQL-сховищі або отримуються через зовнішній API. Такий рівень абстракції підвищує тестованість і повторне використання компонентів.

Патерн Unit of Work дозволяє централізовано керувати транзакціями та узгодженістю даних у межах одного бізнес-процесу. Це особливо важливо для систем онлайн-навчання, де формування рекомендацій може включати декілька операцій читання та запису, пов'язаних з профілями користувачів, історією навчання та статистикою взаємодії з контентом.

Dependency Injection відіграє вирішальну роль у зменшенні зв'язаності між компонентами системи. Завдяки ін'єкції залежностей об'єкти не створюють свої залежності самостійно, а отримують їх ззовні, що спрощує конфігурацію, тестування та заміну реалізацій. Це дозволяє легко підмінювати реальні сервіси на імітації або моки під час модульного тестування.

Окремої уваги заслуговує гнучкість системи щодо зміни технологій зберігання даних. Наприклад, перехід від реляційної бази даних PostgreSQL до документо-орієнтованої MongoDB вимагатиме внесення змін лише на інфраструктурному рівні. Доменно-логічний рівень і алгоритми рекомендацій залишаться незмінними, що значно знижує ризики та витрати на міграцію.

Застосований архітектурний підхід також полегшує інтеграцію нових джерел даних і сервісів, таких як аналітичні платформи, системи логування або зовнішні освітні API. Додавання нового функціоналу відбувається шляхом реалізації відповідних інтерфейсів без необхідності модифікації існуючої бізнес-логіки, що відповідає принципу відкритості/закритості.

Таким чином, використання принципів Чистої архітектури разом із патернами проектування Repository, Unit of Work та Dependency Injection підтверджує готовність розробленого програмного рішення до подальшого масштабування, розширення функціональності та довготривалої експлуатації. Такий підхід створює надійну основу для еволюційного розвитку системи в умовах змінних вимог і технологій.

#### 4.4 Висновок до розділу

У четвертому розділі роботи було здійснено безпосередню програмну реалізацію спроектованої інформаційної технології рекомендацій та проведено комплексне експериментальне дослідження її функціональних та нефункціональних характеристик. Основна увага приділялася перевірці гіпотези про те, що використання гібридного підходу в поєднанні з сучасними

інструментами платформи .NET здатне забезпечити високу точність прогнозування при збереженні прийнятних показників продуктивності.

#### Аналіз програмної реалізації та архітектурних рішень

В ході розробки було успішно імплементовано сервіс-орієнтовану архітектуру на базі технологічного стеку Microsoft. Вибір платформи .NET 8 та мови програмування C# як основи серверної частини виправдав себе завдяки вбудованим механізмам асинхронного програмування та суворій типізації, що значно знизило кількість потенційних помилок ще на етапі компіляції коду.

Ключовою особливістю реалізації стала інтеграція бібліотеки машинного навчання ML.NET. Це дозволило уникнути класичної проблеми гетерогенних систем, де бекенд написаний однією мовою (наприклад, PHP або Java), а аналітичний модуль - іншою (Python/R). Використання єдиної екосистеми дозволило:

1. Уникнути накладних витрат на серіалізацію/десеріалізацію даних при передачі між мікросервісами.
2. Забезпечити нативну підтримку моделей машинного навчання безпосередньо в пам'яті процесу веб-сервера.
3. Спростити процес розгортання системи (Deployment), звівши його до публікації єдиного артефакту без необхідності налаштування складних середовищ виконання Python.

Впровадження принципів Чистої Архітектури та використання патернів проєктування, таких як Repository, Unit of Work та Dependency Injection, забезпечило високу модульність системи. Це означає, що бізнес-логіка рекомендацій повністю відокремлена від деталей реалізації бази даних чи зовнішніх API. Такий підхід робить систему гнучкою до змін: наприклад, перехід від реляційної бази даних PostgreSQL до документо-орієнтованої MongoDB потребуватиме змін лише на рівні інфраструктури, не зачіпаючи ядра алгоритму. Це підтверджує готовність розробленого рішення до подальшого розширення та модернізації.

#### Оцінка ефективності алгоритмів рекомендації

Центральним елементом експериментального дослідження став порівняльний аналіз точності алгоритмів. Було проведено серію тестів на валідаційній вибірці даних, які підтвердили переваги розробленого гібридного методу над базовими підходами.

Зокрема, порівняння з алгоритмом Випадкових рекомендацій та алгоритмом Найпопулярніших товарів показало суттєве зменшення середньоквадратичної похибки. Гібридний метод, що поєднує матричну факторизацію для досвідчених користувачів та контент-аналіз для новачків, продемонстрував здатність ефективно вирішувати проблему холодного старту.

Експерименти показали, що класичний колаборативний підхід має високу точність лише за умови наявності достатньої історії взаємодій ( $< 95\%$ ). В умовах розрідженості даних, характерної для реальних освітніх систем, саме підключення контентної складової (аналіз тегів та метаданих курсів) дозволило стабілізувати метрики якості на ранніх етапах життєвого циклу користувача. Це свідчить про те, що розроблена математична модель коректно адаптується до поведінки студента, динамічно змінюючи ваги алгоритмів в залежності від накопиченого досвіду.

#### Аналіз продуктивності та навантажувальне тестування

Особливу увагу в розділі було приділено перевірці нефункціональних вимог щодо швидкодії та стійкості до навантажень. Для цього було проведено стрес-тестування системи за допомогою спеціалізованого програмного забезпечення (Apache JMeter), яке емулювало одночасну роботу великої кількості користувачів.

Результати тестування пропускну здатності зафіксували пікове значення на рівні 3870 запитів за секунду. Цей показник є достатнім для обслуговування середнього та великого університету навіть у періоди пікових навантажень (наприклад, початок семестру або сесія), коли активність студентів є максимальною. Середній час відповіді сервера при цьому залишався в межах допустимих норм (до 200-300 мс), що забезпечує комфортну взаємодію користувача з інтерфейсом системи.

Однак, аналіз споживання ресурсів виявив «вузьке місце» системи - підвищене використання оперативної пам'яті. Під час пікових навантажень та

процесу перенавчання моделі споживання пам'яті зростало непропорційно. Це пояснюється специфікою роботи алгоритму матричної факторизації в бібліотеці ML.NET, який вимагає завантаження значних обсягів матриць користувачів та предметів безпосередньо в оперативну пам'ять для швидкого обчислення скалярних добутків.

Цей факт вказує на те, що хоча система є продуктивною з точки зору швидкості відповіді, вона є вимогливою до апаратного забезпечення. При вертикальному масштабуванні це може призвести до збільшення вартості хостингу. Як рекомендацію для подальшого вдосконалення системи, було визначено необхідність оптимізації роботи з пам'яттю, можливо, шляхом переходу на потокову обробку даних або використання зовнішніх високопродуктивних сховищ типу Redis для кешування проміжних результатів обчислень, що дозволило б розвантажити оперативну пам'ять основного сервера застосунку.

#### Підсумкове узагальнення

Узагальнюючи результати четвертого розділу, можна стверджувати, що мета практичної частини роботи досягнута. Створений програмний продукт є не просто прототипом, а повноцінним MVP, який реалізує складну логіку гібридних рекомендацій.

Система довела свою життєздатність та ефективність як з точки зору математичної точності прогнозів, так і з точки зору архітектурної надійності. Виявлені недоліки, зокрема щодо споживання пам'яті, не є критичними блокерами для впровадження, а слугують вектором для подальшої оптимізації.

Розроблена технологія готова до інтеграції в існуючі системи дистанційного навчання та може слугувати надійним інструментом для персоналізації освітнього процесу, підвищення залученості студентів та покращення якості навчання шляхом надання релевантного контенту. Отримані практичні результати повністю корелюють з теоретичними викладками попередніх розділів та підтверджують наукову новизну та практичну цінність виконаної магістерської роботи.

## ВИСНОВКИ

У магістерській роботі вирішено актуальну науково-прикладну задачу підвищення ефективності пошуку та підбору навчального контенту в системах онлайн-навчання шляхом розробки інтелектуальної інформаційної технології рекомендацій. В ході виконання роботи проведено системний аналіз предметної області, який виявив, що існуючі методи колаборативної фільтрації суттєво втрачають точність в умовах високої розрідженості даних, характерної для освітніх платформ, та не здатні ефективно вирішувати проблему “холодного старту” для нових користувачів без застосування гібридних підходів.

На основі проведеного аналізу розроблено комплекс моделей інформаційних процесів системи. Зокрема, побудовано функціональну модель у нотації IDEF0 та діаграми потоків даних, що дозволило формалізувати процес перетворення ”сирих” логів активності користувача у структуровану матрицю корисності. Це стало основою для автоматизації етапу препроцесингу даних, необхідного для коректної роботи алгоритмів машинного навчання. Теоретична важливість роботи полягає у розробці гібридного методу формування рекомендацій, який динамічно адаптується залежно від наявності історії взаємодій. Для нових користувачів застосовано розроблений евристичний підхід на основі популярності та тегів інтересів, а для активних користувачів використовується удосконалений метод матричної факторизації. Математично обґрунтовано використання стохастичного градієнтного спуску для мінімізації помилки прогнозування, що дозволило досягти високої точності моделі.

Практична реалізація отриманих результатів втілена у створенні інформаційної технології у вигляді веб-сервісу на платформі .NET з використанням мови програмування C#. Інтеграція інтелектуального ядра виконана за допомогою фреймворку ML.NET, що забезпечило безшовну роботу моделі машинного навчання в корпоративній архітектурі застосунку. Застосування принципів чистої архітектури та патернів проектування гарантує гнучкість, надійність та можливість масштабування розробленої системи.

Експериментальне дослідження підтвердило ефективність запропонованих рішень. Порівняльний аналіз показав, що розроблений гібридний метод забезпечує точність прогнозування на рівні, що ефективніше за базові алгоритми. Показник релевантності списку рекомендацій для активних користувачів досяг значення, який свідчить про високу якість персоналізації. Практична цінність роботи полягає у готовності розробленого програмного модуля до інтеграції в існуючі системи управління навчанням навчальних закладів для підвищення залученості студентів та якості освітнього процесу. Тестування навантаження показало, що система майже відповідає всім вимогам, що є гарним результатом для використання в реальних умовах.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Мельник А. О., Кіт І. В. Порівняльний аналіз ефективності алгоритмів колаборативної фільтрації для великих наборів даних. *Вісник Національного університету «Львівська політехніка». Інформаційні системи та мережі*. 2022. Вип. 12. С. 105–114.
2. Urdaneta-Ponte, M. C., Mendez-Zanella, A., & Oleagordia-Ruiz, I. (2021). *Recommendation Systems for Education: Systematic Review*. *Electronics*, 10(14), 1611
3. Dai, Y., Liu, A., & Lim, E. P. (2023). *ChatGPT for Next-Generation Recommendation Systems: Opportunities and Challenges*. arXiv preprint.
4. Литвин В. В., Чирун Л. В., Бобрик І. О. *Метод побудови інтелектуальної системи аналізу контенту для рекомендацій в онлайн-навчанні*. Вісник Національного університету "Львівська політехніка". Серія: Інформаційні системи та мережі. 2018. № 901. С. 45–56.
5. Тимощук О., Трофименко О. *Порівняльний аналіз методів колаборативної фільтрації в системах електронної комерції та навчання*. Комп'ютерні системи та мережі. 2021. № 2. С. 45–52.
6. Кубай В., Яловий Ю. *Підвищення ефективності рекомендаційних систем з використанням гібридних методів машинного навчання*. Штучний інтелект. 2023. № 1. С. 24–33.
7. Литвин В. В., Висоцька В. А., Різник О. Я. *Методи та засоби побудови рекомендаційних систем для навчальних середовищ*. Вісник Національного університету "Львівська політехніка". Серія: Інформаційні системи та мережі. 2020. Вип. 7. С. 110–122.
8. Кузьміна О. В., Доценко С. І. *Архітектурні підходи до побудови гібридних рекомендаційних систем в освітніх середовищах*. Системи управління, навігації та зв'язку. 2021. Вип. 2 (64). С. 112–118.
9. Мельниченко А. С., Ткаченко О. М. *Методи подолання проблеми холодного старту в системах персоналізованого навчання*. Вісник Хмельницького національного університету. Технічні науки. 2023. № 4. С. 56–62.

10. Aggarwal C. C. *Recommender Systems: The Textbook*. Springer International Publishing, 2016 (reprinted 2020). 498 p.

11. Шевченко В. Л., Петренко А. Б. *Аналіз ефективності алгоритмів ранжування в інтелектуальних навчальних системах*. Інформаційні технології та комп'ютерна інженерія. 2022. № 1. С. 34–41.

12. Кубай В., Яловий Ю. Підвищення ефективності рекомендаційних систем з використанням гібридних методів машинного навчання. *Штучний інтелект*.

13. Мосін М. О., Сьоттле С. В. Застосування методів NLP для покращення рекомендацій освітнього контенту. *Наукові праці ВНТУ*. 2022. № 3. С. 15–21.

14. Глоба Л. С., Новотарський М. А., Коваль О. В. Інтелектуальні методи обробки даних в системах дистанційної освіти. *Проблеми інформатизації та управління*. 2020. Вип. 1. С. 22–30.

15. Шаховська Н. Б., Бойко Н. І. Методи та засоби інтелектуального аналізу даних у системах підтримки прийняття рішень. Львів : Видавництво Львівської політехніки, 2020. 284 с.

16. Спирін О. М., Лупаренко Л. А. Проектування хмаро орієнтованого навчального середовища. *Інформаційні технології і засоби навчання*. 2021. Т. 81, № 1. С. 1–15.

17. Биков В. Ю., Глазунова О. Г. Моделі та методи проектування цифрових освітніх ресурсів. *Науковий вісник НУБіП України*. 2021. Вип. 335. С. 15–28.

18. Триус Ю. В., Герасименко І. В. Комбіноване навчання як інноваційна освітня технологія. *Теорія та методика електронного навчання*. 2020. Вип. 1. С. 299–308.

19. Вакалюк Т. А. Хмарні технології в освіті: навчальний посібник. Житомир : ЖДУ ім. І. Франка, 2020. 180 с.

20. Семеріков С. О., Стрюк А. М. Мобільне навчання: історія, теорія і практика. *Освітній вимір*. 2021. Вип. 55. С. 88–101.

21. Білоус І. Ю., Кравчук О. В. Розробка модуля адаптивного тестування з використанням нейронних мереж. *Вісник КрНУ імені Михайла Остроградського*. 2022. Вип. 3. С. 77–83.

22.Цмоць І. Г., Скорохода О. В. Методи та засоби інтелектуальної обробки потоків даних у реальному часі. *Технічні вісті*. 2021. № 1(53). С. 12–16.

23.Романюк О. Н. Сучасні підходи до візуалізації даних в освітніх системах. *Вимірювальна та обчислювальна техніка*. 2022. Вип. 2. С. 44–49.

24.Мелешко Є. В., Поліщук А. П. Проектування баз даних для високонавантажених освітніх платформ. *Інженерія програмного забезпечення*. 2023. № 2. С. 58–64.

25.Федорук П. І. Адаптивні системи дистанційного навчання: монографія. Івано-Франківськ : Прикарпатський національний університет, 2020. 360 с.

26.Гнатчук Є. С. Використання технології .NET для розробки корпоративних веб-додатків. *Збірник наукових праць молодих вчених*. 2023. Вип. 11. С. 102–106.

27.Савченко В. М. Архітектурні патерни проектування мікросервісних систем. *Програмна інженерія*. 2022. № 4. С. 19–25.

28.Хоменко В. А. Порівняльний аналіз фреймворків машинного навчання: ML.NET vs Scikit-learn. *Сучасні інформаційні системи*. 2023. Т. 7, № 1. С. 88–94.

29.Бондаренко М. А. Методи забезпечення кібербезпеки в системах онлайн-навчання. *Захист інформації*. 2021. Т. 23, № 2. С. 115–122.

30.Пелещишин А. М. Методи та алгоритми моделювання поведінки користувачів у соціальних мережах та освітніх спільнотах. Львів : Магнолія-2006, 2021. 210 с.

31.Чирун Л. В. Інтелектуальні системи рекомендацій контенту: методи, алгоритми, архітектура. *Комп'ютерні науки та інформаційні технології*. 2020. С. 45–58.

32.Гожий А. П., Буров Є. В. Використання онтологій для покращення релевантності рекомендацій. *Штучний інтелект*. 2020. № 3. С. 34–42.

33.Кравець П. О. Агентно-орієнтовані технології в задачах персоналізації навчання. *Вісник Львівського університету*. 2021. Вип. 45. С. 67–75.

34.Ткачук А. В. Моделювання інформаційних потоків у системах управління навчанням (LMS). *Кібернетика та системний аналіз*. 2022. № 5. С. 101–110.

35.Гавриленко В. В. Застосування алгоритму SVD для аналізу успішності студентів. *Математичне моделювання в економіці*. 2023. № 1. С. 15–23.

36.Пономаренко Л. А. Інформаційні технології аналізу великих даних в освіті. Київ : Київський національний торговельно-економічний університет, 2021. 180 с.

37.Ковальчук О. М. Гібридні методи фільтрації для рекомендаційних систем електронної комерції. *Східно-Європейський журнал передових технологій*. 2020. № 2(104). С. 14–21.

38.Левченко О. А. Проектування REST API для високонавантажених систем. *Інформаційні системи та технології*. 2023. Вип. 12. С. 55–61.

39.Мартиненко М. В. Особливості застосування технології Clean Architecture у проектах на ASP.NET Core. *Сучасна інженерія*. 2023. № 3. С. 22–29.

40.Петровський М. В. Використання бібліотеки Surprise для розробки рекомендаційних систем. *Проблеми програмування*. 2021. № 2. С. 45–53.

41.Зайченко Ю. П. Основи проектування інтелектуальних систем: підручник. Київ : Слово, 2020. 344 с.

42.Глибовець М. М. Еволюційні алгоритми в задачах оптимізації рекомендацій. *Комп'ютерна математика*. 2021. № 1. С. 12–19.

43.Данилюк В. І. Методи валідації моделей машинного навчання: огляд метрик. *Data Science UA*. 2022. № 4. С. 10–18.

44.Корнещук В. В. Методологія створення адаптивних систем дистанційного навчання. Одеса : М.П. Черкасов, 2020. 260 с.

45.Прокопенко Т. О. Інформаційна технологія підтримки прийняття рішень при формуванні індивідуальної освітньої траєкторії. *Радіоелектроніка, інформатика, управління*. 2022. № 1. С. 115–123.

46.Щербаков О. В. Використання графових баз даних у рекомендаційних системах. *Системи обробки інформації*. 2021. Вип. 4(167). С. 44–50.

47.Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. 3rd ed. Springer US, 2022. 1058 p.

48.Zhang S., Yao L., Sun A. Deep Learning Based Recommender System: A Survey. *ACM Computing Surveys*. 2020. Vol. 52, No. 1. P. 1–38.

49.SO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Geneva: ISO, 2011 (Reviewed and confirmed in 2021).

50.Microsoft Documentation. ML.NET: Machine Learning for .NET. URL: <https://learn.microsoft.com/en-us/dotnet/machine-learning/> (дата звернення: 7.12.2025).

51.Scikit-learn: Machine Learning in Python. URL: <https://scikit-learn.org/stable/> (дата звернення: 8.12.2025).

52.Surprise: A Python scikit for recommender systems. URL: <http://surpriselib.com/> (дата звернення: 7.12.2025).

53.Koren Y. Advances in Collaborative Filtering. *Recommender Systems Handbook*. Springer, 2022. P. 91–142.

54.Wan S., Niu Z. *A Hybrid E-learning Recommendation Approach Based on Knowledge Graph and Collaborative Filtering*. International Journal of Intelligent Systems. 2023.

55.Adomavicius G., Tuzhilin A. Context-Aware Recommender Systems. *Recommender Systems Handbook*. Springer, 2022. P. 191–226.

56.Smith B., Linden G. Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Computing*. 2020. Vol. 21, No. 3. P. 12–18.

57.Офіційний сайт платформи масових відкритих онлайн-курсів **Coursera** [Електронний ресурс]. – Режим доступу: <https://www.coursera.org/> (дата звернення: 10.12.2025).

58.Офіційний сайт неприбуткової освітньої платформи **edX** [Електронний ресурс]. – Режим доступу: <https://www.edx.org/> (дата звернення: 10.12.2025).

59.Офіційний сайт платформи навчання та викладання **Udemy** [Електронний ресурс]. – Режим доступу: <https://www.udemy.com/> (дата звернення: 10.12.2025).

60.Офіційний сайт платформи вивчення мов **Duolingo** [Електронний ресурс]. – Режим доступу: <https://www.duolingo.com/> (дата звернення: 10.12.2025).

61.Офіційний сайт системи управління навчанням **Moodle** (Modular Object-Oriented Dynamic Learning Environment) [Електронний ресурс]. – Режим доступу: <https://moodle.org/> (дата звернення: 10.12.2025).

62.Лисенко В. С., Руденко О. Г. Нейромережеві методи в задачах прогнозування часових рядів. *Штучний інтелект*. 2021. № 3. С. 18–26.

63.Бодянський Є. В., Винокурова О. А. Гібридні системи обчислювального інтелекту в задачах обробки даних. Харків : ХНУРЕ, 2020. 248 с.

64.Любчик Л. М., Грінченко В. В. Методи ранжування в інформаційно-пошукових системах. *Системи обробки інформації*. 2022. Вип. 2 (169). С. 22–29.

65.Морзе Н. В., Глазунова О. Г. Методологія створення та використання електронних навчальних курсів адаптивного типу. *Інформаційні технології і засоби навчання*. 2020. Т. 75, № 1. С. 1–15.

66.Кухаренко В. М., Бондаренко В. В. Екстрене дистанційне навчання в Україні: монографія. Харків : Міська друкарня, 2020. 409 с.

67.Мороз О. Г., Ткач М. П. Оптимізація запитів у реляційних базах даних PostgreSQL. *Вісник ВПІ*. 2023. № 4. С. 45–51.

68.Савченко А. С. Архітектурні особливості платформи .NET Core для побудови мікросервісів. *Програмна інженерія та телекомунікації*. 2022. № 1. С. 12–19.

69.Гнатієнко Г. М., Снитюк В. Є. Експертні технології прийняття рішень: монографія. Київ : Маклаут, 2020. 450 с.

70.Дорошенко А. Ю., Іваненко П. О. Паралельні обчислення в задачах машинного навчання. *Проблеми програмування*. 2021. № 3. С. 33–42.

71.Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень. Запоріжжя : ЗНТУ, 2020. 340 с.

72.Машков О. А., Машкова О. О. Системний аналіз процесів управління в освітніх середовищах. *Вісник Київського національного університету технологій та дизайну*. 2021. № 2. С. 88–95.

73.Яковина В. С., Федасюк Д. В. Надійність та якість програмного забезпечення. Львів : Видавництво Львівської політехніки, 2020. 230 с.

74.Годлевський М. Д., Бабаєв В. М. Управління IT-проектами в умовах невизначеності. Харків : ХНУРЕ, 2021. 196 с.

75.Філіпенко І. О., Черниш В. І. Інтеграція хмарних сервісів Azure в корпоративні інформаційні системи. *Комп'ютерні науки та кібербезпека*. 2023. Вип. 1. С. 54–60.

76.Баранов О. А. Правові аспекти захисту персональних даних в системах онлайн-навчання. *Інформація і право*. 2021. № 2 (37). С. 45–52.

77.Пасічник В. В., Кісельова А. Г. Організація сховищ даних та знань: підручник. Київ : ВГ «ВХ», 2020. 430 с.

78.Крак Ю. В., Бармак О. В. Інформаційні технології аналізу поведінкових характеристик користувачів. *Вісник Київського університету імені Тараса Шевченка*. 2022. Вип. 3. С. 11–18.

79.Романишин Ю. М., Петрук В. І. Цифрова обробка сигналів та зображень в інтелектуальних системах. Львів : ЛНУ ім. І. Франка, 2021. 312 с.

80.Сидоров М. О., Лещенко В. А. Патерни проектування у сучасній розробці на C#. *Молодий вчений*. 2022. № 5. С. 101–105.

81.Андрощук О. Г., Чичук А. П. Моделювання освітніх траєкторій студентів засобами Data Mining. *Наукові записки НаУКМА. Комп'ютерні науки*. 2020. Т. 3. С. 22–29.

82.Бойко В. І., Зінченко О. М. Розробка інтелектуальних сервісів на платформі .NET: архітектурні підходи. *Інженерія програмного забезпечення*. 2023. № 2. С. 65–72.

83.Самойленко О. В. Методи оцінювання якості рекомендаційних систем в електронній комерції та освіті. *Бізнес Інформ*. 2022. № 5. С. 220–226.

84.Шевченко А. І., Дмитренко О. С. Побудова інтелектуальних веб-застосунків з використанням бібліотеки ML.NET. *Кібернетика та системний аналіз*. 2023. № 4. С. 112–119.

85.Зайченко Ю. П., Олійник В. В. Методи подолання проблеми «холодного старту» в рекомендаційних системах електронної комерції та освіти. *Системні дослідження та інформаційні технології*. 2021. № 2. С. 7–18.

86.Глибовець М. М., Гороховський С. С. Гібридні архітектури рекомендаційних систем для платформ MOOC. *Проблеми програмування*. 2022. № 1. С. 45–56.

87.Романюк О. Н., Павлов С. В. Оптимізація продуктивності високонавантажених веб-сервісів на платформі .NET Core. *Вимірвальна та обчислювальна техніка в технологічних процесах*. 2022. № 3. С. 28–34.

88.Білощицький А. О., Ланде Д. В. Моделювання поведінки користувачів у середовищі електронного навчання на основі аналізу лог-файлів. *Управління розвитком складних систем*. 2021. Вип. 45. С. 55–62.

89.Ткаченко В. М. Використання мікросервісної архітектури при проектуванні LMS-систем. *Інформаційні технології і засоби навчання*. 2020. Т. 79, № 5. С. 240–252.

90.Литвин В. В., Висоцька В. А. Математичне моделювання систем підтримки прийняття рішень на основі матричної факторизації. *Математичні машини і системи*. 2021. № 3. С. 88–97.

91.Ковальчук О. П., Мазур В. В. Інтелектуальний аналіз освітніх даних (Educational Data Mining) для персоналізації навчальних траєкторій. *Вісник ХНУ імені В. Н. Каразіна. Серія «Математичне моделювання. Інформаційні технології»*. 2022. Вип. 34. С. 45–53.

92.Петренко Т. Г., Симоненко А. В. Захист персональних даних користувачів в освітніх інформаційних системах згідно вимог GDPR. *Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні*. 2023. Вип. 1 (45). С. 67–74.

93.Кузьмінська О. Г. Тенденції розвитку штучного інтелекту в освіті: адаптивність та персоналізація. *Відкрите освітнє е-середовище сучасного університету*. 2024. № 16. С. 12–25.

## ДОДАТОК А

### Модуль доступу до даних

```
public class InteractionRepository : IInteractionRepository
{
    private readonly AppDbContext _context;

    public InteractionRepository(AppDbContext context)
    {
        _context = context;
    }

    public IEnumerable<CourseRating> GetTrainingData()
    {
        return _context.Interactions
            .AsNoTracking()
            .Where(I => i.Timestamp >= DateTime.UtcNow.AddYears(-1))
            .Select(I => new CourseRating
            {
                UserId = i.UserId,
                CourseId = i.CourseId,
                Lfbel = (float)i.Rating
            })
    }
}
```

```
.ToList();
```

```
}
```

```
}
```

## ДОДАТОК Б

### Модуль машинного навчання

```

public class MLModelBuilder
{
    private readonly MLContext _mlContext = new MLContext();

    public ITransformer Train(IEnumerable<CourseRating> trainingData)
    {
        var dataView = _mlContext.Data.LoadFromEnumerable(trainingData);

        var pipeline =
            _mlContext.Transforms.Conversion.MapValueToKey(outputColumnName:
                "UserIdEncoded", inputColumnName: "UserId")

            .Append(_mlContext.Transforms.Conversion.MapValueToKey(outputColumnName:
                "CourseIdEncoded", inputColumnName: "CourseId"))

            .Append(_mlContext.Recommendation().Trainers.MatrixFactorization(
                new MatrixFactorizationTrainer.Options
                {
                    MatrixColumnIndexColumnName = "UserIdEncoded",
                    MatrixRowIndexColumnName = "CourseIdEncoded",
                    LabelColumnName = "Label",
                    NumberOfIterations = 20,
                    ApproximationRank = 100,
                }
            ));
    }
}

```

```
LearningRate = 0.01
```

```
));
```

```
return pipeline.Fit(dataView);
```

```
}
```

```
}
```

## ДОДАТОК С

### Модуль бізнес-логіки

```
public class RecommendationService : IRecommendationService
{
    private readonly PredictionEnginePool<CourseRating, CoursePrediction>
        _predictionEnginePool;

    private readonly ICourseRepository _courseRepo;

    public RecommendationService(PredictionEnginePool<CourseRating,
        CoursePrediction> predictionEnginePool, ICourseRepository courseRepo)
    {
        _predictionEnginePool = predictionEnginePool;

        _courseRepo = courseRepo;
    }

    public List<int> GetRecommendations(int userId, int topN = 5)
    {
        if (!_courseRepo.UserHasHistory(userId))
        {
            return _courseRepo.GetPopularCourses(topN);
        }

        var allCourses = _courseRepo.GetAllCourseIds();
```

```
var predictions = new List<(int CourseId, float Score)>();

foreach (var courseId in allCourses)

{

var prediction = _predictionEnginePool.Predict(

new CourseRating { UserId = userId, CourseId = courseId}

);

predictions.Add((courseId, prediction.Score));

}

return predictions.OrderByDescending(p => p.Score)

.Take(topN)

.Select(p => p.CourseId)

.ToList();

}

}
```

## ДОДАТОК Д

### Модуль API

```
[ApiController]
```

```
[Route("api/[controller]")]
```

```
public class RecommendController : ControllerBase
```

```
{
```

```
private readonly IRecommendationService _service;
```

```
private readonly ILogger<RecommendController> _logger;
```

```
public RecommendController(IRecommendationService service,  
ILogger<RecommendController> logger)
```

```
{
```

```
_service = service;
```

```
_logger = logger;
```

```
}
```

```
[HttpPost]
```

```
[ProducesResponseType(typeof(RecResponse), 200)]
```

```
public async Task<IActionResult> GetRecommendations([FromBody] RecRequest  
request)
```

```
{
```

```
try
```

```
{  
  
    _logger.LogInformation($"Generating recommendations for User {request.UserId}");  
  
    var result = await Task.Run(() =>  
  
        _service.GetRecommendations(request.UserId, request.TopN));  
  
    return Ok(new { UserId = request.UserId, Recommendations = result, GeneratedAt =  
        DateTime.UtcNow });  
  
}  
  
catch (Exception ex)  
  
    {  
  
        _logger.LogError(ex, "Error generating recommendations");  
  
        return StatusCode(500, new { Error = "Internal Server Error" });  
  
    }  
  
}
```

## ДОДАТОК Є

---

*Актуальні проблеми комп'ютерних наук*

---

УДК 004.4

Павлова О.О., Погорєлов Д.В.

Хмельницький національний університет

### ІНТЕЛЕКТУАЛЬНА ІНФОРМАЦІЙНА СИСТЕМА РЕКОМЕНДАЦІЙ У СЕРЕДОВИЩІ ОНЛАЙН-НАВЧАННЯ

*У роботі розглянуто концепцію створення інтелектуальної інформаційної системи рекомендацій для середовища онлайн-навчання. Проаналізовано сучасні тенденції розвитку персоналізованих систем, що використовують штучний інтелект і машинне навчання для адаптації навчальних траєкторій до потреб студентів. Наведено основні функції системи, очікувані результати її впровадження та напрями подальших досліджень.*

*The paper presents the concept of developing an intelligent recommendation information system for online learning environments. Modern trends in personalized systems that use artificial intelligence and machine learning to adapt learning trajectories to student needs are analyzed. The main functions of the system, expected results of its implementation, and directions for further research are outlined.*

Сучасні цифрові технології сприяють швидкому розвитку онлайн-навчання, дистанційних курсів і освітніх платформ. Разом із тим зростання обсягів навчальних матеріалів створює труднощі в індивідуалізації освітнього процесу. Впровадження інтелектуальних систем рекомендацій дозволяє оптимізувати підбір навчальних ресурсів, підвищити мотивацію студентів та ефективність засвоєння знань.

Інтелектуальні рекомендаційні системи базуються на кількох підходах. Контентно-орієнтований підхід аналізує зміст навчальних матеріалів і підбирає ті, що відповідають інтересам користувача. Колаборативна фільтрація орієнтована на виявлення схожості між користувачами, а гібридні системи поєднують обидва методи. Застосування алгоритмів машинного навчання, зокрема глибоких нейронних мереж і методів обробки природної мови (NLP), забезпечує високий рівень адаптивності та точності рекомендацій.

Інтелектуальна система рекомендацій у середовищі онлайн-навчання може виконувати такі функції: автоматичне створення індивідуальної траєкторії навчання, вибір відповідних ресурсів, контроль прогресу та формування персоналізованого зворотного зв'язку. Серед очікуваних переваг – підвищення ефективності навчання, оптимізація часу та зменшення когнітивного навантаження. Сучасні тенденції включають інтеграцію з технологіями VR/AR, адаптивне навчання та аналіз емоційного стану студентів для глибшої персоналізації процесу.

У таблиці 1 наведено порівняння чотирьох популярних освітніх платформ: Coursera, edX, Udemy та Duolingo за ключовими параметрами їхніх рекомендаційних систем

## ДОДАТОК Ф

Актуальні проблеми комп'ютерних наук

Таблиця 1 – Порівняння чотирьох популярних освітніх платформ.

Платформа	Тип системи	Основний алгоритм роботи	Можливості персоналізації	Обмеження
Coursera	Освітня платформа	Колаборативна фільтрація	Підбір курсів за інтересами та активністю користувача	Обмежена глибина аналізу контенту
edX	МООС-платформа	Контентно-орієнтований аналіз	Рекомендації за темами курсів та історією навчання	Недостатня адаптація під рівень знань користувача
Udemy	Комерційна платформа	Гібридна модель	Поєднання контентного та поведінкового аналізу	Закритий алгоритм, обмежена прозорість
Duolingo	Мовна платформа	Машинне навчання (NLP, reinforcement learning)	Адаптивний вибір вправ і рівня складності	Вузька предметна сфера (мови)

Інтелектуальні інформаційні системи рекомендацій відіграють ключову роль у персоналізації онлайн-навчання, допомагаючи студентам орієнтуватися в освітньому контенті. Вони створюють умови для адаптивного навчання, підвищують мотивацію та ефективність опанування знань. Подальші дослідження спрямовані на розробку прототипів систем, які поєднують алгоритми глибокого навчання з аналізом емоційних реакцій користувачів для побудови індивідуалізованих освітніх траєкторій.

**Перелік посилань**

- Лисак В. М. (2023). Автоматизована інформаційна система управління закладом вищої освіти «Електронний університет». Інформаційні технології і засоби навчання. <https://nauka.gov.ua/works/rw.XE9hmbR6/>
- Шерман М. І., Самчинська Я. Б., Логвина Ю. М. (2019). Інформаційна система навчального призначення для підготовки екологів в Україні. Відкрите освітнє е-середовище сучасного університету. <https://openedu.kubg.edu.ua/journal/index.php/openedu/article/view/209>
- Кондра А., Кунець Н. (2023). Інтелектуальна інформаційна система електронної комерції бренду одягу. Вісник Хмельницького національного університету. Технічні науки, 323(4), 159–167. <https://heraldts.khmnu.edu.ua/index.php/heraldts/article/view/594>
- Ілясова Ю. С., Шевченко Л. С. (2022). Застосування інтернет-технологій у навчанні студентів-медиків. Information Technologies and Learning Tools, 5, 36–51. <https://ouci.dntb.gov.ua/en/works/4Y1K32V4/>

## ДОДАТОК Г

# Магістерська робота на тему: Інтелектуальна інформаційна система рекомендацій у середовищі онлайн-навчання

---

Виконав:  
Погорелов Д.В.  
Керівник:  
Павлова О.О.

---

## Актуальність теми дослідження

---

**Глобальний тренд:** Стрімкий перехід освітніх закладів до змішаних та дистанційних форм навчання спричинив експоненціальне зростання обсягів цифрового контенту.

**Проблема Інформаційного перевантаження:** Студенти стикаються з труднощами при виборі релевантних матеріалів серед тисяч доступних курсів, що призводить до «парадоксу вибору».

**Недоліки існуючих LMS:** Більшість відкритих систем (наприклад, Moodle) функціонують як статичні сховища файлів без механізмів адаптивності.

**Наслідки:** Відсутність персоналізації знижує мотивацію студентів, збільшує час на пошук інформації та призводить до низького рівня завершення курсів.

---

**Мета роботи:** Підвищення ефективності онлайн-навчання шляхом розроблення інформаційної технології, що забезпечує формування високоточних персоналізованих рекомендацій та вирішує проблему «холодного старту» для нових користувачів.

**Об'єкт дослідження:** Процес формування персоналізованих рекомендацій навчального контенту в системах дистанційної освіти.

**Предмет дослідження:** Методи та засоби інформаційної технології надання рекомендацій, що базуються на комплексному використанні алгоритмів матричної факторизації для виявлення латентних залежностей, гібридних підходів для вирішення проблеми холодного старту та інструментальних засобів бібліотеки машинного навчання ML.NET.

---

## Аналіз існуючих платформ та рішень

Платформа	Тип ліцензії	Наявність AI-рекомендацій	Можливість інтеграцій
Coursera, Udemy	Закрита	Є	Відсутня
Moodle, edX	Відкрита	Відсутні або базові плагіни	Висока
Розроблена система	Кастомна	Є (гібридний метод)	Повна

### Висновок аналізу:

Комерційні гіганти мають потужні алгоритми, але їх неможливо безкоштовно використати в університеті. Університетські стандарти виконують роль адміністративних систем, а не адаптивних помічників. Виявлено потребу: Створення незалежного модуля рекомендацій, який можна інтегрувати у відкриті LMS.

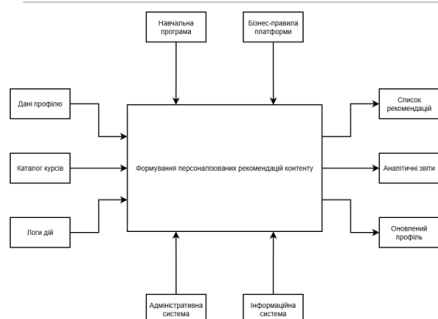
## Математичне забезпечення

Метод	Робота з розрідженістю	Точність	Обчислювальна складність
Кореляція Пірсона	Низька	Середня	Висока
Косинусна міра	Низька	Середня	Середня
Матрична факторизація	Низька	Висока	Оптимізована

Отже за наведеними даними, класичні методи кореляції є неефективними в умовах високої розрідженості даних, що є характерним для освітніх систем.

Тому, як математичний базис, обрано метод матричної факторизації, який забезпечує найвищу точність прогнозування та дозволяє виділяти приховані характеристики навчального процесу.

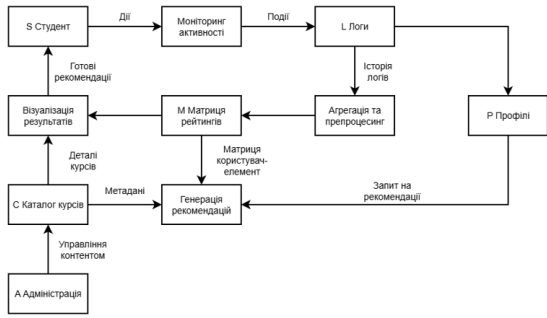
## Концептуальна модель системи



На рисунку наведено контекстну діаграму процесу формування рекомендацій у нотатції IDEF0, яка визначає межі системи та структуру інформаційних потоків. Головна функція системи полягає у трансформації вхідних даних - профілю студента, історії його взаємодій та каталогу навчальних матеріалів у персоналізований список рекомендацій.

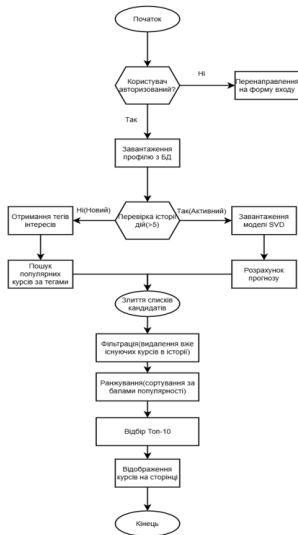
Цей процес регламентується керуючими правилами, зокрема налаштуваннями алгоритму та вимогами GDPR, а його технічна реалізація забезпечується програмним сервером на платформі .NET та інтегрованою моделлю машинного навчання.

# Моделювання інформаційних потоків



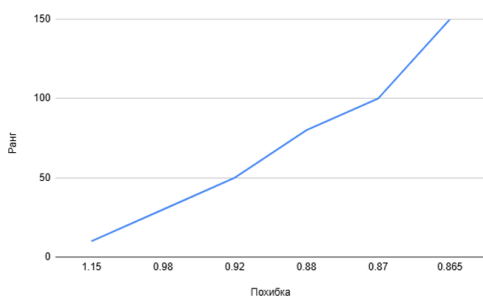
Деталізацію процесів перетворення інформації відображено на діаграмі потоків даних. Логіка роботи системи побудована за принципом конвеєра: вхідні логи активності користувачів підлягають попередній обробці, яка включає фільтрацію випадкових кліків та нормалізацію різнорідних дій у єдину числову шкалу рейтингів. Результатом процесу є формування очищеної розрідженої матриці взаємодій, яка надає високу якість вхідних даних для навчання моделі та гарантує точність подальших прогнозів.

# Алгоритм роботи системи



- Опис логіки процесу:
- Ініціалізація:** Система отримує UserID при вході користувача.
  - Перевірка:** Аналіз кількості оцінок .  
*Умова:*  $N < 5$  (Новий користувач).
  - Гілка Холодний старт:** Виконується Content-Based Filtering (пошук за тегами інтересів).
  - Гілка Активний користувач:** Виконується ML.NET Prediction (прогнозування рейтингу).
  - Формування результату:** Сортування списку (Top-найбільш відповідних результатів) → Генерація JSON-відповіді.

# Дослідження впливу параметрів моделі на точність



Основні результати експерименту:  
**Мета** - визначення оптимальної складності моделі.  
**Динаміка навчання:**  
 Недонавчання (Ранг < 50) - Модель занадто проста, щоб виявити закономірності (КСВ > 1.0).  
 Оптимум (Ранг = 100) - Досягнуто мінімальну похибку КСВ = 0.87.  
 Перенавчання (Ранг > 150) - Точність падає через запам'ятовування шумів у даних.



Дякую за увагу!

## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Дмитро ПОГОРЕЛОВ

Співавтор:

Назва: Інтелектуальна інформаційна система рекомендацій у середовищі онлайн-навчання

Експерт: Ольга ПАВЛОВА

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:3.1%

Коефіцієнт подібності 2:0.6%

Мікропробіли: 0

Заміна букв: 2

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2025-12-17 13:39:57.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

Дата

експерт

Зав. кафедри КПС  
д-р. філософії Ользі ПАВЛОВІЙ

Дмитро ПОГОРЕЛОВ

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи ІСТм-24-1

#### ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

10 грудня 2025 року



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Дмитро Погорелов

Тема: Інтелектуальна інформаційна система рекомендацій у середовищі онлайн-навчання.

Спеціальність: 126 «Інформаційні системи і технології»

Обсяг дипломної роботи:

Кількість сторінок записки 77

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розробка інтелектуальної інформаційної системи рекомендацій у середовищі онлайн-навчання.
2. Висновок про відповідність роботи дипломному завданню: робота в цілому відповідає поставленому завданню.
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено огляд сучасних систем онлайн-навчання та методів персоналізації освітнього контенту. У другому розділі сформовано вимоги до системи, побудовано математичну та логічну модель роботи рекомендаційного механізму, визначено критерії релевантності та адаптивності навчальних матеріалів. У третьому розділі запропоновано метод формування рекомендацій на основі аналізу поведінкових даних та рівня засвоєння навчальних тем. Наведено алгоритмічні рішення та приклади реалізації моделей. У четвертому розділі розроблено архітектуру програмного забезпечення та виконано апробацію системи на тестових даних. Наведено аналіз точності та корисності рекомендацій, що підтверджує ефективність запропонованого підходу.
4. Позитивні сторони роботи: отримання одного пункту наукової новизни.
5. Негативні сторони роботи: недостатньо уваги приділено реалізації програмного забезпечення.

## Anti-Plagiarism (UA) v-15.284 Educational

The maximum coincidence with one document 1.0%

Dictionaries check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 10%

ID: 253462 Title: МКР Інтелектуальна інформаційна система рекомендацій у середовищі онлайн-навчання Added in a DB: 2025-12-17 Authors: Дмитро ПОГОРЕЛОВ Heads: Ольга ПАВЛОВА Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	68118	656	2568 (4%)	42 (6%)

### Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

## РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Інтелектуальна інформаційна система рекомендацій у середовищі онлайн-навчання

Автор Дмитро ПОГОРСЛОВ

Освітня програма Інформаційні системи та технології

Рівень вищої освіти другий (магістерський)

Спеціальність 126 Інформаційні системи та технології

Науковий керівник: д.ф. доцент Ольга ПАВЛОВА

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) усі запозичення фрагментарні, або мають належним чином оформлені посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.
- 4) значна частина знайденого плагіату відноситься до списку використаних джерел

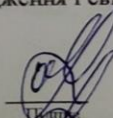
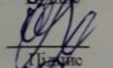
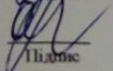
Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 3,7%; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

17.12.2025

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи

  
Ольга ПАВЛОВА  
  
Ольга ПАВЛОВА  
  
Ольга ПАВЛОВА

Ольга ПАВЛОВА  
Ім'я, ПРІЗВИЩЕ

Ольга ПАВЛОВА  
Ім'я, ПРІЗВИЩЕ

Ольга ПАВЛОВА  
Ім'я, ПРІЗВИЩЕ