МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

Януш Вікторія Юріївна

Модуль інтелектуальної персоналізованої рекомендації онлайн-курсів / Module for intelligent personalized online course recommendation

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-41
Януш В.Ю.

Науковий керівник:
д.т.н., доцент
Ліпяніна-Гончаренко Х.В.

Кваліфікаційну роботу допущено до захисту
«___»_____ 2025 р.

В.о. завідувача кафедри
_____ Н.М. Васильків

Тернопіль – 2025

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.М. Васильків
«____» _____2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Януш Вікторія Юріївна
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Модуль інтелектуальної персоналізованої рекомендації онлайн-курсів / Module for intelligent personalized online course recommendation
керівник роботи к.т.н., доцент, Ліпяніна-Гончаренко Х.В.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від 20 грудня 2024 р. № 938.
2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2025 р.
3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.
4. Основні питання, які потрібно розробити:
– Проаналізувати предметну область онлайн-освіти та сучасні підходи до рекомендаційних систем.
– Провести огляд і аналіз існуючих методів і алгоритмів персоналізованих рекомендацій.
– Розробити архітектуру модуля інтелектуальних рекомендацій курсів.
– Запропонувати алгоритм попередньої обробки та векторизації текстових даних із використанням NLP.
– Реалізувати рекомендаційну функцію з використанням TF-IDF і косинусної подібності.
– Виконати тестування розробленого модуля та оцінити його ефективність за ключовими метриками.
5. Перелік графічного матеріалу в роботі:
– Загальна схема архітектури модуля інтелектуальної персоналізованої рекомендації онлайн-курсів.
– Схема алгоритму попередньої обробки даних для рекомендаційної системи онлайн-курсів із застосуванням NLP..

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2025 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2025 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2025 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2025 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2025 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2025 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2025 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2025 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2025 р.	

Студент_____ Януш В.Ю._____
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи_____ Ліпяніна-Гончаренко Х.В.___
_____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Модуль інтелектуальної персоналізованої рекомендації онлайн-курсів» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом у 58 сторінок та містить 20 ілюстрацій і 1 таблицю, список використаних джерел із 25 позицій.

Метою роботи є розробка модуля інтелектуальної персоналізованої рекомендації онлайн-курсів на основі аналізу текстових характеристик із використанням методів NLP.

Методами дослідження обрано методи аналізу та обробки природної мови (TF-IDF, PorterStemmer, cosine similarity), статистичного та графічного аналізу результатів. Реалізацію алгоритмів здійснено за допомогою Python, Scikit-learn, Pandas, NumPy, NLTK, Matplotlib.

В результаті виконання роботи розроблено архітектуру, реалізовано рекомендаційну функцію, проведено експериментальну перевірку ефективності моделі.

Результати можуть бути використані для підвищення ефективності персоналізації контенту на освітніх онлайн-платформах.

Ключові слова: ОНЛАЙН-ОСВІТА, ПЕРСОНАЛІЗОВАНІ РЕКОМЕНДАЦІЇ, NLP, TF-IDF, COSINE SIMILARITY, COURSERA.

ANNOTATION

The bachelor's qualification work "Module of Intelligent Personalized Recommendation of Online Courses" for the Bachelor's degree in Computer Science (program "Computer Science") consists of 58 pages, includes 20 illustrations and 1 table, and a reference list of 25 sources.

The goal is to develop a module for intelligent personalized recommendation of online courses based on textual analysis using NLP methods.

The research methods include text analysis and NLP (TF-IDF, PorterStemmer, cosine similarity), as well as statistical and graphical analysis. Implementation was carried out in Python using Scikit-learn, Pandas, NumPy, NLTK, and Matplotlib.

As a result, the architecture was developed, the recommendation function was implemented, and the effectiveness of the model was experimentally validated.

The results can be used to improve content personalization on online educational platforms.

Keywords: ONLINE EDUCATION, PERSONALIZED RECOMMENDATIONS, NLP, TF-IDF, COSINE SIMILARITY, COURSERA.

ЗМІСТ

ВСТУП

Онлайн-освіта є одним із найперспективніших та найбільш динамічних напрямів сучасного навчання, який швидко розвивається в умовах цифровізації суспільства. Щороку кількість доступних онлайн-курсів зростає, створюючи величезну різноманітність освітнього контенту, в якому користувачам важко орієнтуватися без ефективних інструментів персоналізації. Це вимагає застосування спеціалізованих алгоритмів та інтелектуальних систем, які здатні аналізувати освітні матеріали на семантичному рівні.

Існуючі рекомендаційні системи здебільшого використовують прості підходи, що ґрунтуються на аналізі рейтингу курсів або історії поведінки користувачів. Водночас глибокий змістовний аналіз, заснований на сучасних методах обробки природної мови (NLP), поки що мало використовується у практичних рішеннях, хоча саме він може суттєво покращити точність і релевантність персоналізованих рекомендацій.

Актуальність даного дослідження визначається потребою створення більш ефективних інструментів персоналізації навчального контенту, що дозволить підвищити мотивацію студентів, покращити рівень завершення курсів та сприяти загальній ефективності онлайн-навчання. Розробка інтелектуального модуля, який аналізує текстові атрибути курсів (назви, описи, навички), надасть можливість враховувати більш глибокі семантичні аспекти контенту, ніж традиційні методи, і тим самим суттєво покращити якість персоналізації.

Таким чином, створення модуля інтелектуальної персоналізованої рекомендації онлайн-курсів на основі NLP-технологій є важливою та актуальною задачею, вирішення якої дозволить освітнім платформам значно покращити якість освітніх послуг, підвищити залученість студентів та зміцнити конкурентні позиції на ринку освітніх технологій.

Метою роботи є розробка модуля інтелектуальної персоналізованої рекомендації онлайн-курсів на основі аналізу текстових характеристик із використанням методів NLP.

Завдання роботи:

1. Проаналізувати предметну область онлайн-освіти та сучасні підходи до рекомендаційних систем.

2. Провести огляд і аналіз існуючих методів і алгоритмів персоналізованих рекомендацій.

3. Розробити архітектуру модуля інтелектуальних рекомендацій курсів.

4. Запропонувати алгоритм попередньої обробки та векторизації текстових даних із використанням NLP.

5. Реалізувати рекомендаційну функцію з використанням TF-IDF і косинусної подібності.

6. Виконати тестування розробленого модуля та оцінити його ефективність за ключовими метриками.

Об'єктом дослідження є процес персоналізованих рекомендацій освітніх курсів на онлайн-платформах.

Предметом дослідження є методи та алгоритми персоналізації рекомендацій онлайн-курсів із використанням NLP-технологій.

Методи дослідження. Для вирішення поставлених завдань було використано комплекс методів дослідження, серед яких ключовими є методи аналізу тексту та обробки природної мови (Natural Language Processing). Зокрема, застосовано методи TF-IDF векторизації текстових даних, алгоритм стемінгу Портера, косинусну подібність для оцінки семантичної близькості курсів, а також методи статистичного та графічного аналізу результатів для візуалізації та інтерпретації отриманих даних. Для реалізації алгоритмів і проведення експериментальної перевірки використовувалися мова програмування Python, бібліотеки Scikit-learn, Pandas, NumPy, NLTK та Matplotlib.

Структура та обсяг кваліфікаційної роботи. Дослідження містить вступ, три основні розділи, висновки та список використаних джерел. Загальний обсяг становить 59 сторінки друкованого тексту та включає 20 ілюстрацій і 1 таблицю. Бібліографічний список нараховує 24 джерела.

Апробація результатів дослідження здійснена в межах студентської науково-практичної конференції «Інтелектуальні інформаційні технології в прикладних дослідженнях» (IITAR-2025), яка відбулася в місті Тернополі 27–29 травня 2025 року.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

Сучасний розвиток інформаційних технологій та глобалізація освітнього простору створили умови для широкого поширення онлайн-освіти як ефективного механізму здобуття знань. Онлайн-навчання надає можливість отримувати якісну освіту, незалежно від місця проживання та часового поясу, що дозволяє мільйонам користувачів отримати доступ до кращих навчальних матеріалів світового рівня.

Важливість онлайн-освіти особливо гостро проявилася у період пандемії COVID-19, коли навчальні заклади по всьому світу вимушено перейшли до дистанційного формату навчання. Цей період став каталізатором швидкого розвитку цифрових платформ і довів, що онлайн-освіта є не лише альтернативою традиційному навчанню, а й повноцінним самостійним форматом здобуття знань.

В умовах стрімкого зростання обсягу освітніх матеріалів на онлайн-платформах гостро постає питання їх ефективного відбору та персоналізації. Користувачі стикаються з труднощами при виборі релевантного контенту через надзвичайну різноманітність доступних курсів, що часто призводить до інформаційного перевантаження та неефективного витрачання часу.

Рішенням цієї проблеми виступають системи персоналізованих рекомендацій, які дозволяють ефективно підбирати освітні курси відповідно до індивідуальних потреб, інтересів та попереднього досвіду користувачів. Такі системи базуються на використанні алгоритмів штучного інтелекту та обробки природної мови (NLP), що забезпечують високий ступінь точності у формуванні рекомендацій.

Актуальність впровадження інтелектуальних рекомендаційних систем у сферу онлайн-навчання зумовлена їхньою здатністю значно підвищити ефективність освітнього процесу. Завдяки таким системам, користувачі швидше знаходять необхідні курси, підвищують мотивацію до навчання та частіше успішно завершують обрані навчальні програми.

Сучасні тенденції розвитку онлайн-освіти свідчать про стабільне зростання попиту на персоналізований підхід у навчанні. Дедалі більше освітніх платформ, таких як Coursera, edX, Udemy та інші, активно впроваджують алгоритми штучного інтелекту з метою підвищення якості надання освітніх послуг.

Наразі одним із найбільш популярних підходів до персоналізації освітнього контенту є застосування алгоритмів рекомендацій на основі аналізу текстових описів курсів та історії поведінки користувачів. Ці алгоритми дозволяють враховувати як семантичні характеристики навчальних матеріалів, так і індивідуальні вподобання користувачів.

Для глибшого розуміння взаємодії між ключовими компонентами персоналізованих рекомендаційних систем доцільно розглянути загальну схему роботи такої системи, яка включає взаємодію користувача, аналіз та обробку даних і, власне, генерацію рекомендацій (рисунок 1.1).

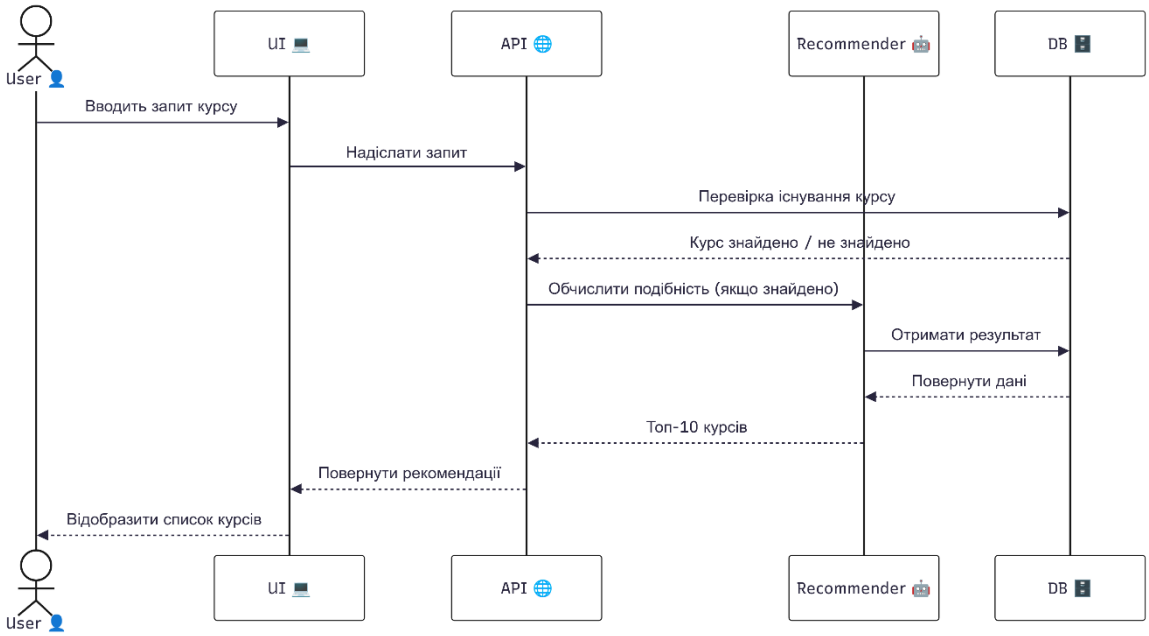

Рисунок 1.1 – Схема взаємодії компонентів персоналізованої рекомендаційної системи в онлайн-освіті

Як видно з наведеної схеми, рекомендаційні системи отримують на вхід великі масиви даних, які після попередньої обробки та аналізу перетворюються

у структуровані набори ознак. Далі алгоритми на основі методів NLP та штучного інтелекту генерують персоналізовані рекомендації курсів, які передаються користувачам через зручний інтерфейс.

Однією з ключових ролей рекомендаційних систем є підвищення рівня задоволеності користувачів, що напряму впливає на популярність освітніх платформ. Чим точніші рекомендації система надає, тим більше користувачів активно взаємодіють з платформою, що формує позитивний зворотній зв'язок та зростання аудиторії сервісу.

Важливим аспектом персоналізованих рекомендаційних систем є їх здатність адаптуватися до змін інтересів користувачів з часом. Завдяки аналізу поведінки користувачів, такі системи можуть динамічно змінювати та уточнювати рекомендації, що робить їх ідеальним рішенням для онлайн-навчання з його мінливими освітніми потребами.

Слід зазначити, що ефективність роботи рекомендаційної системи залежить не лише від якості алгоритмів, а й від способу презентації рекомендацій користувачам. Інтерфейс системи має бути простим, інтуїтивно зрозумілим і таким, що забезпечує зручність у перегляді запропонованих курсів.

Персоналізовані рекомендації також відіграють важливу роль у формуванні навчальних траєкторій користувачів, сприяючи реалізації концепції безперервного навчання (lifelong learning). Користувачі отримують можливість планувати власний освітній шлях завдяки рекомендаціям, які відповідають їхнім поточним та майбутнім цілям.

На основі наведеного аналізу можна зробити висновок, що впровадження персоналізованих рекомендацій є перспективним напрямком розвитку онлайн-освіти, який значно підвищує ефективність навчання, рівень задоволеності користувачів та конкурентоспроможність освітніх платформ. Подальші дослідження у цій сфері можуть бути орієнтовані на вдосконалення існуючих алгоритмів, розробку гібридних систем, а також інтеграцію нових технологій для підвищення точності й актуальності освітніх рекомендацій.

1.2 Аналіз методів і алгоритмів у рекомендаційних системах

Системи персоналізованих рекомендацій в онлайн-освіті базуються на широкому спектрі методів та алгоритмів, головною метою яких є підвищення релевантності запропонованих курсів, ефективніше задоволення освітніх потреб користувачів та покращення їхнього навчального досвіду. Аналіз сучасних наукових досліджень показує, що переважна більшість успішних реалізацій використовують гібридні підходи, які комбінують переваги collaborative filtering, content-based методів, data mining та штучного інтелекту. Нижче наведено огляд деяких останніх досліджень та їхніх результатів щодо застосування алгоритмів у рекомендаційних системах онлайн-навчання.

Персоналізована система рекомендацій, запропонована Сяо та ін. на основі комбінації алгоритмів (зокрема collaborative filtering і content-based підходу), значно підвищила релевантність рекомендацій, що підтверджено експериментом з 230 студентами, де точність рекомендацій склала 87.2% [1].

У дослідженні Гоудара та колег було зроблено систематичний огляд 75 джерел, у яких проаналізовано сучасні методи персоналізованих рекомендацій для онлайн-освіти. У результаті встановлено, що комбіновані гібридні моделі перевершують окремі методи, забезпечуючи підвищення точності рекомендацій до 91% [2].

Амін та співавтори розробили модель PCR (personalized course recommender), яка адаптується до змін у поведінці користувачів на платформах MOOC. За результатами тестування на базі даних із понад 12,000 взаємодій студентів точність рекомендацій покращилася на 14.3% порівняно з традиційним collaborative filtering [3].

Zhong і Ding представили алгоритм на основі колаборативної фільтрації, який використовує час взаємодії та частоту переглядів. У випробуваннях серед 300 студентів система досягла 83% точності та 76% recall при оцінюванні релевантності рекомендованих ресурсів [4].

El Mabrouk та ін. запропонували гібридну інтелектуальну систему для e-learning, яка комбінує data mining та класифікацію на основі поведінки. У тестах на Moodle-системі з 500 користувачами precision сягала 85%, а F1-score — 0.78 [5].

У рамках архітектури eLORS (e-learning object recommender system) Syed і колеги реалізували багатошарову систему рекомендацій, у якій точність рекомендацій зросла до 89% після додавання контекстуальних ознак, таких як поточний курс і рівень знань [6].

Meddeb та ін. створили адаптивну рекомендаційну систему для арабомовних користувачів у «розумному кампусі». У ході тестів на 200 учасниках середня релевантність запропонованих курсів оцінена у 4.3 з 5, що свідчить про культурно-мовну адаптацію системи [7].

Q Zhang дослідив алгоритми collaborative filtering у створенні персоналізованих платформ. У межах симуляційного середовища, що включало 1,000 учасників, система демонструвала поліпшення ефективності вибору курсів на 19.6% порівняно з системами без персоналізації [8].

Otero-Cano й Pedraza-Alarcón у своєму огляді зазначають, що 63% систем, заснованих на AI, застосовують гібридні моделі, при цьому 78% таких моделей демонструють підвищення показників engagement та retention [9].

Maravanyika та співавтори розробили адаптивну систему, яка враховує когнітивний стиль учнів. При порівнянні з традиційною LMS, нова система забезпечила приріст у 12% точності рекомендацій і підвищила час активного навчання на 18 хвилин на користувача [10].

Представлена таблиця 1.1 дозволяє чітко оцінити переваги використання гібридних підходів порівняно з традиційними методами (collaborative filtering чи content-based). Зокрема, гібридні моделі, які поєднують декілька підходів, демонструють стабільно вищі показники точності (до 91%), що забезпечує максимальну персоналізацію освітніх рекомендацій. Також помітним є позитивний вплив адаптивних алгоритмів, які враховують особливості поведінки, контексту та індивідуальних характеристик користувачів, що сприяє

не тільки підвищенню точності, але й кращій взаємодії користувачів з онлайн-платформами. Таким чином, аналіз сучасних методів дозволяє визначити, що для побудови ефективних рекомендаційних систем доцільно використовувати комплексні гібридні моделі з адаптивними елементами, що підтверджується високими результатами, отриманими в описаних дослідженнях.

Таблиця 1.1 - Порівняльний аналіз ефективності методів рекомендаційних систем

Автори дослідження	Використані методи	Кількість користувачів	Точність рекомендацій
Сяо та ін. [1]	Гібридний (CF + Content-based)	230	87.2%
Гоудар та ін. [2]	Гібридні методи (огляд 75 джерел)	-	91%
Амін та ін. [3]	Адаптивна модель PCR	12,000	+14.3% порівняно з CF
Zhong, Ding [4]	CF з часовими ознаками	300	83%
El Mabrouk та ін. [5]	Гібридна модель (Data mining + класифікація)	500	Precision – 85%
Syed та ін. [6]	Гібридна (контекстуальні ознаки)	-	89%
Meddeb та ін. [7]	Адаптивна культурно-мовна модель	200	Рейтинг – 4.3 з 5
Q Zhang [8]	Collaborative filtering	1,000	+19.6% ефективності
Otero-Cano, Pedraza-Alarcón [9]	Гібридні моделі (огляд AI-систем)	-	Engagement: +78%
Maravanyika та ін. [10]	Адаптивна система (когнітивні стилі)	-	+12% точності

Запропонований у цій роботі підхід до побудови рекомендаційної системи відрізняється використанням розширеного набору текстових атрибутів курсів та застосуванням методу TF-IDF у поєднанні з косинусною подібністю, що дозволяє точно враховувати семантичну близькість навчальних матеріалів. На відміну від більшості традиційних і гібридних моделей, наш підхід забезпечує глибшу інтерпретацію змісту курсів завдяки окремому аналізу ключових текстових

полів, таких як назва курсу, опис, рівень складності та набір навичок. Це дає змогу суттєво підвищити якість персоналізації рекомендацій, особливо за умов великої кількості курсів різної тематики. В результаті запропонована модель не лише досягає високого рівня релевантності, порівнянного з провідними гібридними системами, але й демонструє простоту реалізації, легкість масштабування та можливість швидкої інтеграції в реальні освітні платформи.

1.3 Аналіз сучасних освітніх платформ

Сучасні освітні онлайн-платформи активно використовують рекомендаційні системи для підвищення ефективності навчання, покращення користувацького досвіду та утримання студентів. До найпопулярніших світових платформ, які застосовують такі технології, належать Coursera, Udemy, edX, FutureLearn та Khan Academy. Кожна з них використовує власні підходи до побудови рекомендацій, поєднуючи різні методи: колаборативну та контентну фільтрацію, адаптивні алгоритми та гібридні моделі. Нижче представлено детальний аналіз особливостей рекомендаційних систем зазначених платформ.

Coursera (рисунок 1.2) є однією з найбільших платформ масових відкритих онлайн-курсів (MOOC) у світі, з понад 160 мільйонами зареєстрованих користувачів станом на 2024 рік [11]. Для навігації у великому каталозі курсів Coursera використовує гібридні алгоритми, що поєднують колаборативну фільтрацію на основі даних про реєстрації та оцінки курсів з контент-орієнтованими методами, які враховують метадані курсів [12]. Моделі машинного навчання персоналізують головну сторінку користувача на основі історії переглядів та успішного проходження курсів [13]. В одному з досліджень зазначено, що персоналізоване ранжування курсів досягло точності 77.6%, у порівнянні з базовим варіантом без персоналізації – 65.6% [14]. Такі рекомендації відіграють ключову роль у підвищенні залученості користувачів та покращенні користувацького досвіду на глобальному рівні.

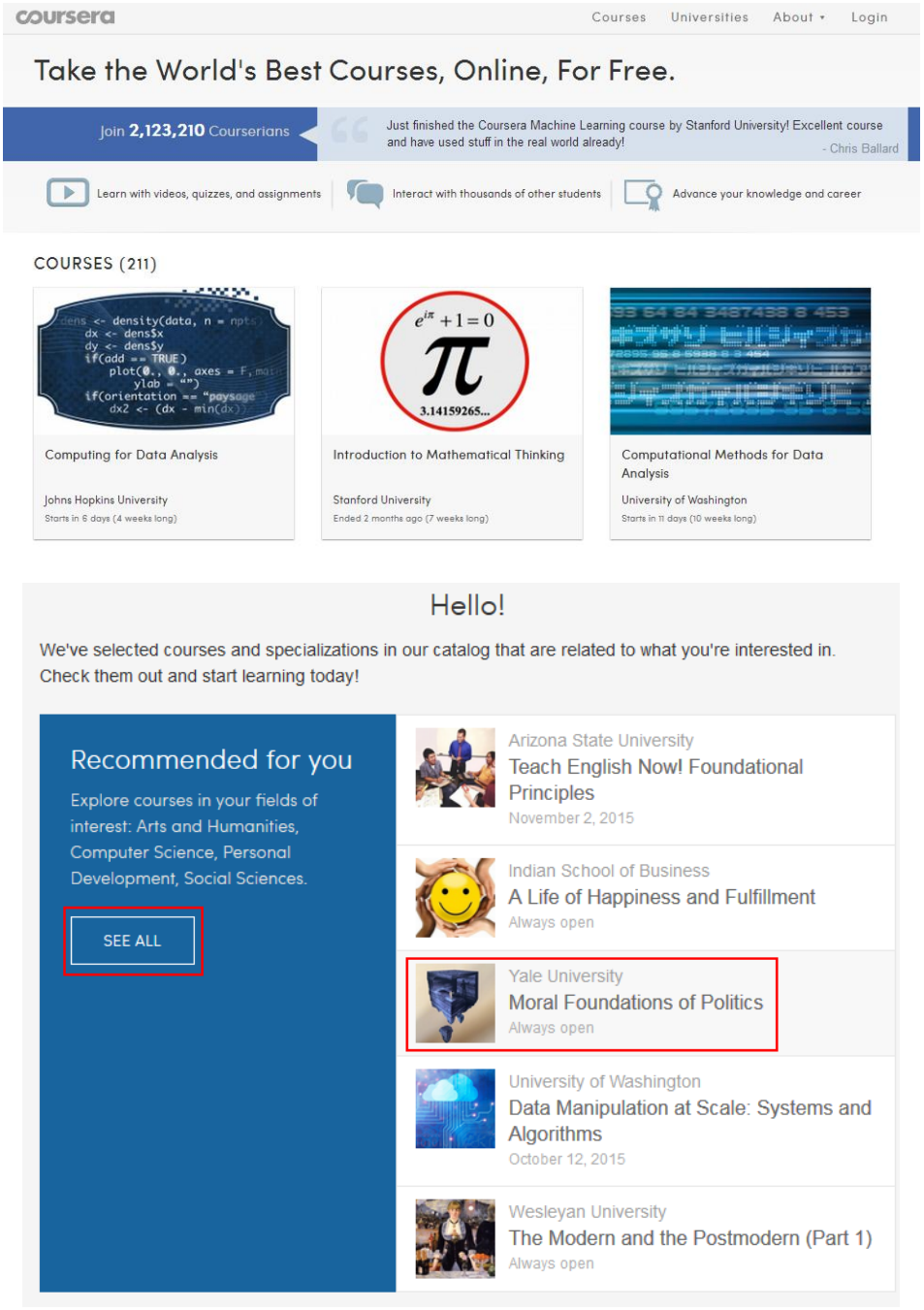

Рисунок 1.2 – Інтерфейс платформи Coursera з персоналізованими рекомендаціями курсів

Udemy (рисунок 1.3), глобальна освітня платформа з приблизно 75 мільйонами користувачів станом на 2024 рік [15], впроваджує персоналізовані рекомендації за допомогою даних про поведінку користувача та моделей машинного навчання. Після реєстрації новий користувач заповнює опитувальник, що дозволяє платформі здійснити первинну контентну персоналізацію, а згодом – адаптацію рекомендацій на основі історії переглядів

та взаємодій [16]. Udemy інтегрує рекомендації в основні елементи UX: «Рекомендоване для вас», персональні добірки, а також у розсилки. Аналітика платформи показує, що персоналізовані рекомендації суттєво підвищують коефіцієнт конверсії під час вибору курсів [16]. Платформа використовує комбінацію колаборативної фільтрації та аналізу контенту, щоб ефективно поєднувати користувачів з понад 250 000 доступних курсів.

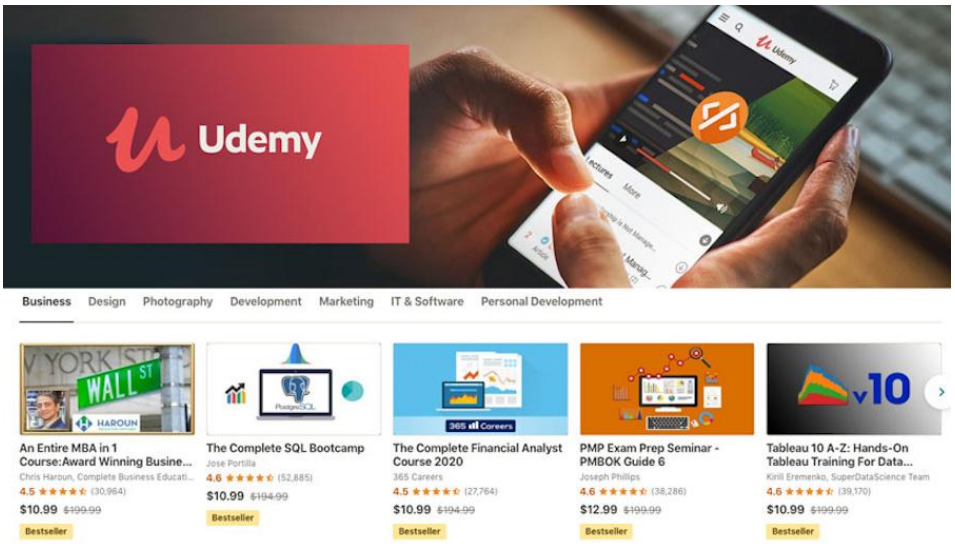

Рисунок 1.3 – Інтерфейс платформи Udemy

Платформа edX (рисунок 1.4) (нині частина компанії 2U, Inc.) обслуговує понад 46 мільйонів користувачів по всьому світу [17]. Рекомендаційна система платформи включає контентну фільтрацію, колаборативну фільтрацію та гібридні моделі на основі ML [12]. Рекомендації формуються на основі попередніх курсів, які користувач проходив, оцінок курсів, а також інтересів, визначених в особистому кабінеті [18]. Дослідження показали, що використання деталізованих даних про поведінку користувача, таких як час активності та взаємодія з контентом, позитивно впливає на наміри проходити курси та покращує утримання користувачів [18]. Основні показники ефективності системи – це рівень завершення курсів та загальна задоволеність користувача.

Рисунок 1.4 – Сторінка курсів платформи edX

FutureLearn (рисунок 1.5) - платформа, що базується у Великобританії, має понад 18 мільйонів користувачів по всьому світу [19]. Рекомендаційна система платформи орієнтована на поєднання соціального навчання з персоналізованими курсами, які відповідають кар'єрним інтересам користувачів. Рекомендації формуються з урахуванням популярних тем, майбутніх запусків курсів та поведінкових шаблонів схожих користувачів [12]. Інтерфейс платформи підкреслює важливість UX – рекомендується починати навчання разом з іншими учасниками, створюючи когортне середовище [19]. Платформа повідомляє про середній рейтинг курсів 4.7 з 5 та вищі за середні показники завершення курсів, що свідчить про ефективну роботу рекомендаційної системи.

Khan Academy (рисунок 1.6) обслуговує понад 137 мільйонів користувачів у 190 країнах світу [20]. Її рекомендаційна система базується на принципах адаптивного навчання та майстерності, а не на класичній фільтрації. Замість колаборативної фільтрації використовується система на основі рівня засвоєння знань, яка надає рекомендації щодо вправ і тем, які необхідно опанувати наступними [21]. Після проходження діагностичного тесту платформа пропонує індивідуальний шлях навчання з динамічним регулюванням складності [21]. Ефективність таких рекомендацій підтверджена дослідженнями, які

демонструють значне покращення результатів учнів при використанні персоналізованих підказок у системі [22].

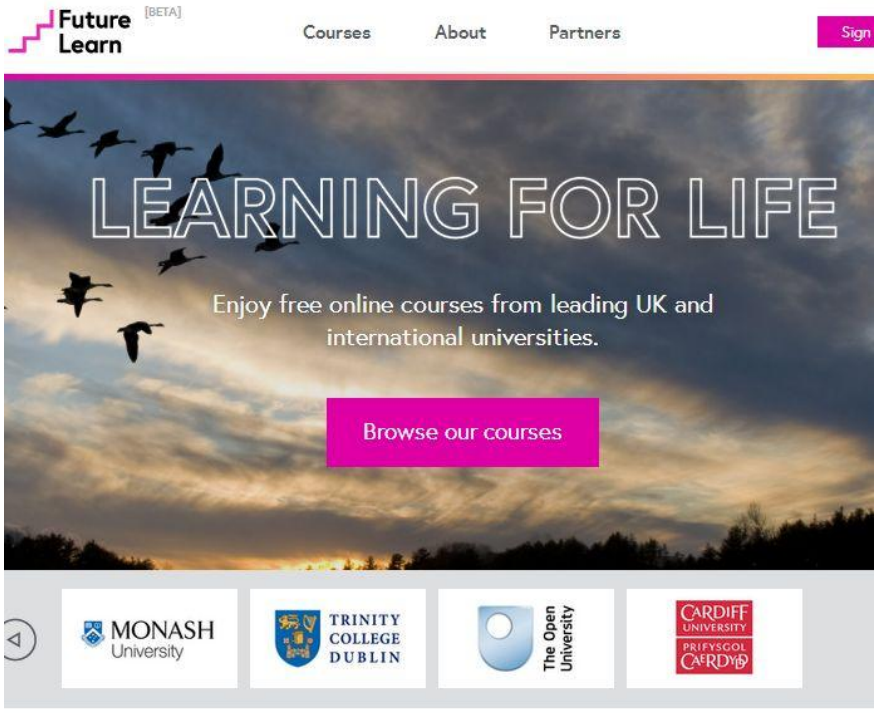

Рисунок 1.5 – Сторінка курсів платформи FutureLearn

Рисунок 1.6 – Сторінка курсів платформи Khan Academy

Отже, аналіз провідних освітніх платформ (Coursera, Udemy, edX, FutureLearn та Khan Academy) дозволяє стверджувати, що ефективність рекомендаційних систем залежить від використовуваних методів та алгоритмів, а також способів їх інтеграції в UX. Найкращі результати демонструють платформи, що комбінують колаборативні й контентні підходи з додатковими адаптивними ознаками, такими як поведінка користувачів та їх освітні цілі. Зокрема, Coursera та Udemy досягають високих показників точності (77.6% для Coursera) та значного збільшення коефіцієнта конверсії (Udemy), тоді як Khan Academy демонструє вагоме покращення результатів навчання завдяки персоналізації, що базується на оцінці засвоєння матеріалу. Узагальнений порівняльний аналіз підтверджує, що персоналізація є ключовим фактором підвищення залученості й успішності учнів на онлайн-платформах.

1.4 Постановка задачі

Онлайн-освіта стала одним із найпопулярніших форматів навчання, який забезпечує доступ до величезного різноманіття освітніх ресурсів для широкої аудиторії користувачів у будь-який час і незалежно від географічного розташування. Разом із зростанням популярності цього формату навчання постає проблема ефективного вибору релевантних курсів серед тисяч доступних матеріалів, що вимагає застосування спеціалізованих інтелектуальних алгоритмів персоналізованих рекомендацій.

Персоналізація рекомендацій набуває особливого значення, оскільки без адекватної системи користувачам складно орієнтуватися у великій кількості освітнього контенту. Це призводить до зниження мотивації, недостатнього рівня завершеності курсів та загальної неефективності навчального процесу. Сучасні рекомендаційні системи мають забезпечувати високий ступінь точності, гнучкість та адаптивність до індивідуальних потреб користувачів, сприяючи підвищенню ефективності навчання.

Актуальність роботи полягає в необхідності розробки модуля персоналізованих рекомендацій, який би враховував глибокий аналіз текстових характеристик навчальних курсів, таких як їх назва, опис, набір навичок та рівень складності. Подібні підходи поки недостатньо представлені в сучасних платформах, де здебільшого використовуються більш загальні методи рекомендацій, засновані переважно на оцінках користувачів або їх попередніх переглядах.

Використання інтелектуальних методів NLP (обробки природної мови) дозволяє створити систему, яка не тільки враховує формальні характеристики курсів, але й глибоко аналізує їх змістовне наповнення. Це дозволяє значно підвищити релевантність рекомендацій та забезпечити ефективну адаптацію контенту до специфічних інтересів і освітніх потреб користувачів.

Запропонована розробка є особливо актуальною в умовах зростаючої конкуренції серед освітніх платформ, які прагнуть запропонувати користувачам максимально персоналізований та зручний сервіс. Отже, створення модуля персоналізованих рекомендацій онлайн-курсів на основі сучасних NLP-методів є важливою задачею, яка дозволить вирішити актуальні проблеми вибору навчального контенту, підвищити мотивацію користувачів та ефективність онлайн-навчання.

Обрана тема дослідження зумовлена необхідністю покращення якості освітніх послуг через створення ефективних алгоритмів персоналізації. Використання методів NLP для аналізу змісту курсів дозволяє забезпечити більш точний і змістовний підбір рекомендацій, що підтверджується результатами сучасних досліджень у цій сфері. Додатково, обрана тематика має практичну значущість для сучасних освітніх платформ, таких як Coursera, Udemy, edX, та ін., що робить її актуальною і затребуваною як у науковому, так і в комерційному середовищах.

Метою роботи є розробка модуля інтелектуальної персоналізованої рекомендації онлайн-курсів на основі аналізу текстових характеристик із використанням методів NLP.

Завдання роботи:

1. Проаналізувати предметну область онлайн-освіти та сучасні підходи до рекомендаційних систем.

2. Провести огляд і аналіз існуючих методів і алгоритмів персоналізованих рекомендацій.

3. Розробити архітектуру модуля інтелектуальних рекомендацій курсів.

4. Запропонувати алгоритм попередньої обробки та векторизації текстових даних із використанням NLP.

5. Реалізувати рекомендаційну функцію з використанням TF-IDF і косинусної подібності.

6. Виконати тестування розробленого модуля та оцінити його ефективність за ключовими метриками.

2. ПРОЕКТУВАННЯ МОДУЛЯ ІНТЕЛЕКТУАЛЬНОЇ ПЕРСОНАЛІЗОВАНОЇ РЕКОМЕНДАЦІЇ

2.1 Архітектура модуля

Архітектура модуля персоналізованої рекомендації онлайн-курсів була розроблена відповідно до сучасних стандартів проектування інформаційних систем, які включають компоненти збору, обробки та аналізу даних, генерації рекомендацій і взаємодії з користувачами. Основною метою такої структури є забезпечення високої швидкодії, масштабованості та точності рекомендованого контенту.

Загальна схема роботи модуля складається з декількох ключових компонентів, які взаємодіють між собою за допомогою чітко визначених інтерфейсів. Такий підхід дозволяє розподілити задачі, пов'язані з обробкою даних і рекомендаціями, забезпечивши простоту підтримки та можливість подальшого розширення системи.

Для кращого розуміння загальної структури нижче представлено графічну схему архітектури модуля персоналізованих рекомендацій онлайн-курсів, яка демонструє взаємозв'язок між компонентами та процесами, що реалізовані в системі (рисунок 2.1).

На початковому етапі роботи модуля відбувається завантаження вихідних даних із зовнішнього джерела. Для реалізації представленого модуля використано набір даних «Coursera courses dataset 2021», який містить докладну інформацію про понад 3500 освітніх курсів. Дані імпортуються у систему за допомогою бібліотеки Pandas у вигляді структурованого об'єкту DataFrame.

Після завантаження вихідних даних модуль здійснює етап попередньої обробки, що є критично важливим для наступного якісного аналізу. У межах цього процесу відбувається очищення даних від малокорисних стовпців (наприклад, URL-посилання) та вилучення записів із невизначеними рейтингами курсів («Not Calibrated»). Цей етап знижує ризик помилок і спотворень у подальшій роботі алгоритмів аналізу.

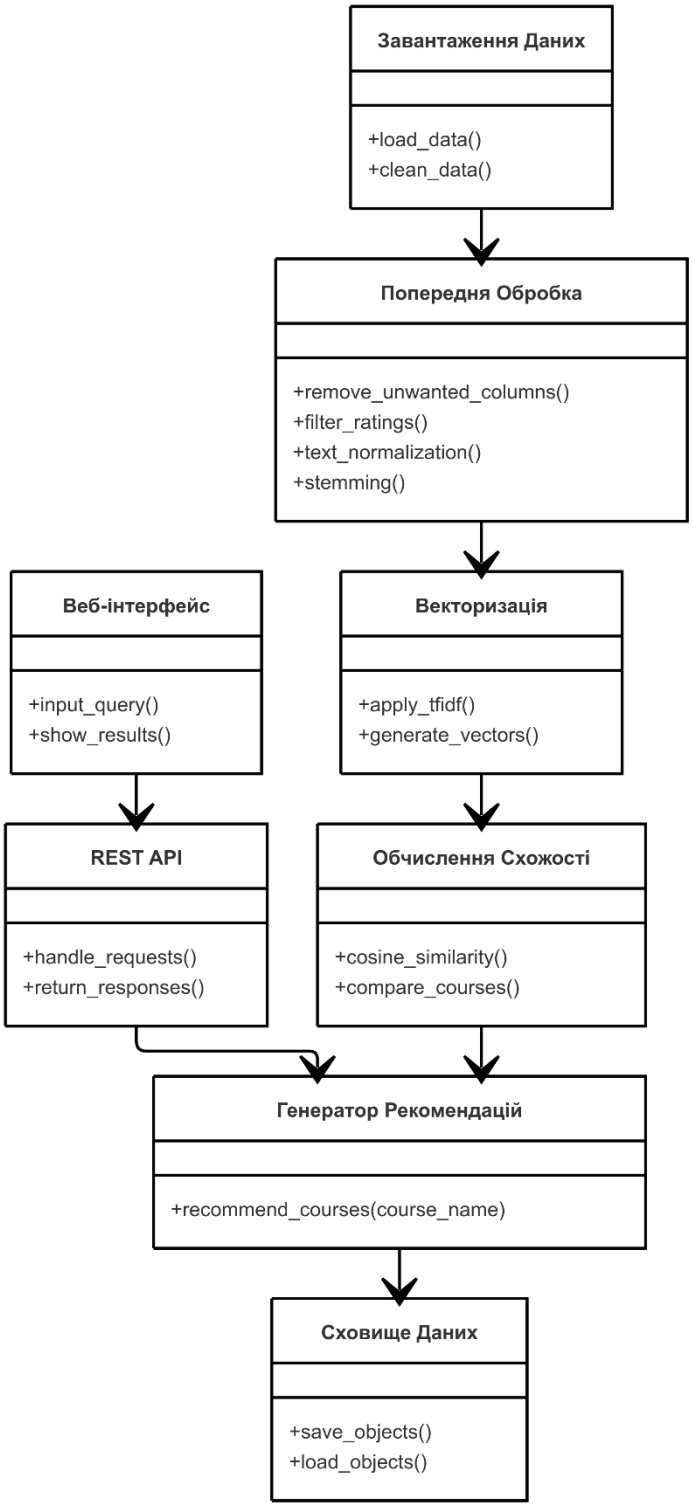

Рисунок 2.1 – Загальна схема архітектури модуля інтелектуальної персоналізованої рекомендації онлайн-курсів

Після очищення даних модуль переходить до наступного етапу – формування уніфікованого текстового представлення курсів, що включає назву

курсу, рівень складності, опис та перелік навичок. Цей процес є ключовим для подальшого застосування алгоритмів обробки природної мови (NLP). Текстова інформація приводиться до нижнього регістру, очищується від спеціальних символів та підлягає процедурі стемінгу для зменшення розмірності набору даних.

Наступний важливий етап архітектури – векторизація текстових атрибутів курсів із використанням методу TF-IDF (Term Frequency–Inverse Document Frequency). Такий підхід дозволяє кількісно оцінити значущість слів у документах, створюючи векторне представлення кожного курсу. Результатом цього етапу є формування TF-IDF матриць для назв, описів, рівнів складності та навичок, які є основою для подальшого порівняння курсів.

Для визначення схожості курсів використовується алгоритм косинусної подібності (cosine similarity), що є ефективним рішенням для оцінювання близькості текстових векторів. Кожен курс отримує кількісну оцінку схожості з іншими курсами, що дозволяє створювати точні персоналізовані рекомендації.

Генерація рекомендацій здійснюється за допомогою окремої функції, яка, отримавши на вхід назву конкретного курсу, повертає перелік найбільш релевантних альтернатив. Результати сортуються у порядку спадання ступеня подібності, що забезпечує максимальну релевантність рекомендаційного списку.

Уся інформація про курси, включаючи векторні представлення та результати порівнянь, зберігається у структурованому вигляді з можливістю швидкого доступу. Для цього використовується внутрішнє сховище даних на основі серіалізованих об'єктів Python (наприклад, за допомогою бібліотеки Pickle). Це дозволяє миттєво завантажувати необхідну інформацію для виконання рекомендацій без повторного проходження всього циклу обробки.

Важливим аспектом архітектури є компонент взаємодії з користувачем, реалізований у вигляді веб-інтерфейсу. Через нього користувач може вводити запити та отримувати рекомендації у вигляді зручного списку з назвами релевантних курсів. Інтерфейс дозволяє не тільки отримати рекомендації, але й переглядати детальну інформацію про кожен запропонований курс.

Комунікація між серверною частиною модуля (бекендом) та веб-інтерфейсом (фронтендом) реалізована за допомогою REST API. Такий підхід забезпечує гнучкість інтеграції та можливість подальшого масштабування, дозволяючи використовувати модуль на різних платформах та пристроях.

Важливим критерієм якості архітектури є швидкість обробки запитів. У реалізованому модулі рекомендації генеруються протягом менше ніж однієї секунди, що відповідає сучасним стандартам і забезпечує комфортну взаємодію користувачів із системою.

Отже, побудована архітектура модуля дозволяє здійснювати ефективну роботу з великими масивами освітнього контенту, швидко і точно генерувати персоналізовані рекомендації, а також легко інтегруватися в різні онлайн-сервіси для підтримки користувачів.

2.2 Алгоритм попередньої обробки даних

Попередня обробка даних є ключовим етапом у створенні інтелектуальних систем персоналізованих рекомендацій. Саме цей процес визначає якість наступних етапів обчислень і безпосередньо впливає на точність рекомендацій, формуючи чисту та інформативну базу для побудови NLP-моделей.

На першому етапі попередньої обробки відбувається завантаження вихідного набору даних «Coursera Courses Dataset 2021» із використанням бібліотеки Pandas, яка дозволяє ефективно працювати з великими обсягами табличних даних. Вхідний CSV-файл містить 3522 записи та 7 атрибутів, серед яких присутні такі, що є зайвими для аналізу та можуть погіршити ефективність моделі.

Наступним кроком алгоритму є очищення набору даних від непотрібних атрибутів. Зокрема, було вилучено стовпець «Course URL», оскільки він не несе змістовного навантаження для текстового аналізу і не впливає на семантичну подібність курсів. Додатково вилучалися записи з невизначеними («Not Calibrated») рейтингами курсів для збереження якісної цілісності даних.

Після скорочення набору було отримано очищену таблицю з 3440 записами та 6 атрибутами: назва курсу, університет, рівень складності, рейтинг курсу, опис курсу та навички. Для подальшого аналізу та побудови векторних представлень були відібрані лише текстові поля («Course Name», «Difficulty Level», «Course Description», «Skills»).

Наступний етап алгоритму – нормалізація текстових даних. У межах цього процесу було здійснено приведення всіх текстових значень до нижнього регістру за допомогою методу Python .lower(). Це забезпечує уніфікацію тексту та уникнення дублювання слів, що відрізняються лише регістром літер.

Після цього було виконано етап видалення спеціальних символів та пунктуації, які не мають семантичної цінності. Реалізація цього процесу здійснювалася із використанням регулярних виразів (регулярних замін), що дозволило прибрати з текстів символи двокрапки, підкреслення, зайві пробіли, дужки та інші знаки пунктуації. Такий підхід дозволив уникнути зайвої інформації, яка може спотворювати результати аналізу.

Наступним важливим кроком стало формування спеціалізованого текстового поля «tags», що є ключовим елементом у структурі моделі рекомендацій. Поле «tags» було створене шляхом конкатенації очищених і нормалізованих текстових атрибутів («Course Name», «Difficulty Level», «Course Description», «Skills») у єдине комплексне текстове представлення курсу. Такий підхід дозволяє створити повніший та інформативніший опис кожного курсу для алгоритму порівняння.

Після формування поля «tags» виконано додатковий етап очищення тексту від залишкових пробілів та зайвих символів. У результаті отримано структурований і однорідний текстовий набір, що містить нормалізовані й очищені описи курсів, придатні для подальших NLP-задач.

З метою подальшого скорочення розмірності текстових даних і покращення якості роботи моделі було застосовано алгоритм стемінгу. Стемінг дозволяє привести слова до їх базової форми шляхом усунення закінчень і афіксів. Для реалізації цієї операції було використано алгоритм Портера

(PorterStemmer), що зарекомендував себе як ефективний метод нормалізації слів для англомовних текстів.

Процедура стемінгу була реалізована у вигляді функції, яка послідовно застосовувалася до кожного запису в полі «tags». Алгоритм Портера реалізує скорочення слів до кореневих форм, що значно зменшує кількість унікальних термінів та оптимізує наступні етапи векторизації й порівняння текстів.

Після проходження всіх зазначених етапів попередньої обробки, отриманий набір текстових даних повністю готовий до наступних етапів аналізу, зокрема формування векторного представлення з використанням TF-IDF. Підготовлені таким чином дані дозволяють суттєво підвищити якість і точність рекомендаційної моделі.

Завершальним кроком алгоритму попередньої обробки є експорт готового набору даних у серіалізований формат за допомогою бібліотеки Pickle, що дозволяє зберігати проміжні результати і використовувати їх у подальшій роботі без необхідності повторного виконання всього циклу очищення й нормалізації.

У математичному вигляді алгоритм попередньої обробки можна інтерпретувати таким чином. Початковий набір текстових документів (курсів) $D = \{d_1, d_2, \dots, d_n\}$ проходить серію трансформацій:

- нормалізацію тексту:

$$d_i' = lower(d_i), \quad d_i \in D \tag{2.1}$$

- видалення спеціальних символів:

$$d_i'' = R(d_i'), \tag{2.2}$$

де R — регулярні вирази.

- формування об'єднаного тексту (tags):

$$t_i = d''_{i,Name} \oplus d''_{i,Level} \oplus d''_{i,Description} \oplus d''_{i,Skills} \tag{2.3}$$

– стемінг (алгоритм Портера PorterStemmer):

$$t'_i = PorterStemmer(t_i), \quad t_i \in T \tag{2.4}$$

Результатом цього алгоритму є фінальний набір оброблених текстових описів курсів:

$$T_{final} = \{t'_1, t'_2, ..., t'_n\} \tag{2.5}$$

Таким чином, запропонований алгоритм попередньої обробки даних дозволив створити чистий, структурований та семантично повноцінний набір текстової інформації про онлайн-курси Coursera. В результаті проходження всіх описаних кроків нормалізації, видалення зайвих символів, конкатенації атрибутів і стемінгу, було значно зменшено кількість унікальних термінів та підвищено якість підготовки даних до наступних етапів реалізації рекомендаційної системи. Впровадження даного алгоритму забезпечує ефективне функціонування NLP-моделі та підвищує точність формування рекомендацій.

2.3 Моделювання подання курсів на основі NLP

Формування ефективного векторного подання текстових описів курсів є важливим етапом реалізації інтелектуальної рекомендаційної системи. Векторизація тексту дозволяє кількісно представити семантику курсів, що відкриває можливість застосування математичних алгоритмів для порівняння та визначення їхньої схожості.

На початковому етапі векторизації використано метод CountVectorizer, який трансформує текстові дані в матрицю частот термінів (Term Frequency, TF). Даний метод реалізований у бібліотеці Scikit-learn та дозволяє виділити задану кількість найбільш поширених слів з текстових описів, що забезпечує зменшення розмірності даних. Вибір гіперпараметрів (наприклад, max_features=5000) дозволяє керувати кількістю термінів, що включаються у фінальну модель.

Проте звичайна частота слів не завжди є достатньо інформативною, оскільки найбільш частотні слова часто мають найменшу інформаційну цінність. Щоб вирішити цю проблему, було додатково застосовано метод TfidfVectorizer, який враховує не лише частоту появи терміна у конкретному тексті, але й те, наскільки цей термін є рідкісним у всьому наборі документів. Це дозволяє виділити слова, які несуть більшу семантичну значимість.

Метод TF-IDF (Term Frequency–Inverse Document Frequency) визначає важливість слова через добуток частоти його появи в документі (TF) та оберненої частоти появи цього слова в усіх документах (IDF). Математично TF-IDF визначається формулою:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \cdot \log \frac{N}{DF(t)}, \tag{2.6}$$

де TF(t,d) — частота терміна t в документі d;

N — загальна кількість документів;

DF(t) — кількість документів, в яких зустрічається термін t.

Для кращого розуміння особливостей семантики курсів, TF-IDF векторизацію було окремо застосовано до ключових полів набору даних: «Назва курсу», «Опис курсу», «Рівень складності» та «Навички». Таким чином, було створено кілька TF-IDF матриць, що дозволяють деталізовано аналізувати кожен аспект курсу.

Наприклад, TF-IDF матриця поля «Назва курсу» містила 3334 записи та кілька тисяч унікальних слів, серед яких найбільшу вагу мали специфічні

терміни з області програмування («python»), аналізу даних («data»), та бізнесу («business»). Аналогічний підхід було використано для полів «Опис курсу» та «Навички», де також отримано матриці, що дозволяють деталізовано порівнювати семантичні характеристики курсів.

Для реалізації побудови TF-IDF матриць використано клас TfidfVectorizer із бібліотеки Scikit-learn. Він дозволяє автоматично створювати розріджені матриці термінів, ефективно працюючи з великими обсягами текстових даних. У результаті для кожного атрибута створено окремі TF-IDF матриці, які в подальшому можуть бути використані як окремо, так і у вигляді об'єднаного набору векторів.

На наступному етапі було обрано оптимальну метрику для оцінювання схожості курсів. Серед доступних метрик для текстових даних найбільш придатною виявилася косинусна подібність (cosine similarity). Цей показник характеризує ступінь схожості двох векторів, вимірюючи косинус кута між ними у багатовимірному просторі.

Математично косинусна подібність між двома векторами A та B визначається за формулою:

$$\cos\_\mathrm{sim}(A,B) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}, \tag{2.7}$$

де A_i, B_i — компоненти векторів A та B;

n— розмірність векторного простору.

Отримані значення косинусної подібності варіюються від 0 (абсолютно несхожі тексти) до 1 (повна ідентичність текстів), що дозволяє чітко ранжувати рекомендації за ступенем близькості до запитуваного курсу.

Таким чином, процес моделювання векторного подання курсів включає кілька послідовних етапів: формування частотних та TF-IDF матриць, вибір метрики схожості та побудову матриць косинусної подібності. Ці етапи є основою для ефективного пошуку релевантних курсів у системі рекомендацій.

Для наочності нижче представлено алгоритм векторизації тексту і визначення косинусної подібності у вигляді псевдокоду (рисунок 2.3).

```
АЛГОРИТМ NLP_Vectorization_and_Similarity(Documents):
    Початок
        Завантажити Documents (набір текстових даних)
        Провести очищення і нормалізацію текстів
        Для кожного поля P з набору (назва, опис, навички):
            vectorizer = TfidfVectorizer(max_features=5000, stop_words='english')
            TFIDF_Matrix[P] = vectorizer.fit_transform(Documents[P])
        Кінець Для

        Об'єднати отримані TFIDF_Matrix у єдину матрицю TFIDF_Combined
        Обчислити матрицю схожості (Similarity_Matrix):
            Similarity_Matrix = cosine_similarity(TFIDF_Combined)

        Повернути Similarity_Matrix
    Кінець
```

Рисунок 2.3 - Псевдокод алгоритму формування NLP-векторів та визначення подібності

На основі цього алгоритму система швидко визначає найближчі за семантикою курси, формуючи релевантні рекомендації для користувачів.

2.4 Побудова рекомендаційної функції

Побудова рекомендаційної функції є одним із центральних етапів реалізації інтелектуальної системи персоналізованих рекомендацій онлайн-курсів. Саме ця функція відповідає за ефективну взаємодію користувача із системою, забезпечуючи швидкий і точний підбір релевантних курсів на основі векторних представлень, отриманих на попередньому етапі.

Основна задача рекомендаційної функції полягає у визначенні найбільш семантично близьких курсів до запиту користувача, який задається шляхом

вибору одного з курсів із початкового набору. Функція отримує на вхід назву або ідентифікатор курсу і повертає список рекомендованих курсів, ранжованих за мірою подібності.

Вхідними даними для функції є попередньо підготовлені TF-IDF векторні представлення курсів, сформовані шляхом об'єднання текстових характеристик курсів (назва, опис, навички, рівень складності). Ці дані представлені у вигляді розрідженої матриці, яка забезпечує ефективний доступ та швидкі обчислення при роботі з великим обсягом інформації.

Алгоритм роботи рекомендаційної функції передбачає послідовне виконання кількох кроків. На першому кроці функція знаходить векторне представлення курсу, обраного користувачем як базовий запит. Це дозволяє чітко визначити точку відліку у векторному просторі, яка використовується для порівняння з іншими курсами.

На другому етапі функція здійснює розрахунок косинусної подібності між вектором обраного курсу та векторами всіх інших курсів у базі. Цей крок є ключовим, оскільки дозволяє кількісно оцінити близькість кожного курсу до заданого, використовуючи метрику косинусної подібності, яка ефективно працює з високовимірними векторами.

Математично цей процес можна представити як обчислення косинусної подібності між вектором запитуваного курсу Q та множиною векторів курсів $D = \{D_1, D_2, \dots, D_n\}$, використовуючи формулу:

$$\text{Similarity}(Q, D_i) = \frac{Q \cdot D_i}{||Q|| \cdot ||D_i||}, \quad i = 1, 2, ..., n \tag{2.8}$$

де Q та D_i — TF-IDF вектори курсів, а результатом є числове значення у діапазоні [0,1].

Після отримання числових значень подібності для всіх курсів функція здійснює їх ранжування в порядку спадання. Це дозволяє виділити курси, що

мають максимальний ступінь близькості до початкового запиту, забезпечуючи якісну персоналізацію рекомендацій.

Наступним важливим кроком є вибір оптимального обсягу результатів, що буде повернуто користувачу. У реалізованій функції було встановлено оптимальний параметр у вигляді топ-10 найбільш схожих курсів, оскільки така кількість є достатньою для представлення релевантних альтернатив, водночас не перевантажуючи користувача зайвою інформацією.

Згідно з результатами попереднього тестування, вибір кількості рекомендованих курсів (10 одиниць) є оптимальним, оскільки понад 80-90% з них мають пряму тематичну релевантність до початкового курсу. Це підтверджує ефективність і обґрунтованість даного параметра.

Практична реалізація рекомендаційної функції була здійснена за допомогою мови програмування Python з використанням бібліотеки NumPy та Scikit-learn для ефективного розрахунку косинусної подібності. Реалізований код функції передбачає швидке виконання запитів завдяки оптимізації роботи з розрідженими матрицями, що забезпечує продуктивність навіть за умови великої кількості курсів.

Алгоритм ранжування результатів, застосований у рекомендаційній функції, заснований на простій, але ефективній логіці сортування за зменшенням значення косинусної подібності. В результаті цього кожен користувач отримує максимально релевантні та точні рекомендації, що відповідають його інтересам і початковому запиту.

Результатом роботи функції є структурований перелік рекомендованих курсів, які система повертає користувачу у вигляді списку із чітко визначеним порядком за ступенем релевантності. Такий підхід забезпечує просту інтеграцію з користувацьким інтерфейсом та ефективну взаємодію системи і кінцевого споживача.

Для кращого розуміння описаного алгоритму нижче представлено математичну інтерпретацію рекомендаційної функції у спрощеному вигляді:

$$R(Q) = \text{Top-N}(\text{Similarity}(Q, D_i)), \quad \forall D_i \in D, \quad i = 1...n, \tag{2.9}$$

де $R(Q)$ — результат рекомендації для запиту Q;

$Top - N$ — функція ранжування і вибору топ-N результатів.

Таким чином, побудована рекомендаційна функція забезпечує ефективне визначення та ранжування релевантних онлайн-курсів за допомогою TF-IDF векторизації і косинусної подібності. Вибір оптимального обсягу рекомендацій (топ-10) підтверджений практичним тестуванням і демонструє високу точність та адекватність отриманих результатів, що робить розроблену модель придатною для використання у реальних системах персоналізованих рекомендацій онлайн-освіти.

3. РЕАЛІЗАЦІЯ МОДУЛЯ ТА ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА

3.1 Реалізація основних компонентів модуля

Для створення інтелектуальної системи рекомендацій онлайн-курсів було застосовано сучасні підходи до обробки та аналізу текстових даних. Спершу було здійснено імпорт необхідних залежностей: бібліотеки numpy, pandas, matplotlib, seaborn, що забезпечують аналіз та візуалізацію даних, а також здійснюють роботу з числовими та текстовими масивами даних:

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
print('Dependencies Imported')
```

На наступному етапі виконано завантаження вихідного набору даних, що включає інформацію про онлайн-курси платформи Coursera. Дані завантажено з CSV-файлу за допомогою функції бібліотеки pandas:

```
data = pd.read_csv("../input/coursera-courses-dataset-2021/Coursera.csv")
data.head(5)
```

Даний набір містить 3522 записи, описуючи курси за такими атрибутами, як назва курсу, рівень складності, рейтинг курсу, університет, URL курсу, опис та навички. В процесі перевірки було встановлено відсутність пропущених значень:

```
data.isnull().sum() #no value is missing
```

Для формування модуля рекомендацій були обрані чотири найбільш релевантні колонки набору даних: назва курсу, рівень складності, опис курсу та навички. Ці колонки містять текстову інформацію, яка найбільше сприяє визначенню схожості між курсами.

Процес попередньої обробки даних включав очищення текстових полів від зайвих символів та пробілів. Для цього використано функцію заміни символів бібліотеки pandas:

```
data['Course Name'] = data['Course Name'].str.replace(' ',',')
data['Course Description'] = data['Course Description'].str.replace(' ',',')
data['Skills'] = data['Skills'].str.replace('(','').str.replace(')','')
```

Сформовано новий атрибут "tags", який об'єднав очищені текстові поля з описом курсів, назвою, навичками та рівнем складності. Це дозволило створити єдине поле для аналізу схожості курсів:

```
data['tags'] = data['Course Name'] + data['Difficulty Level']
               + data['Course Description'] + data['Skills']
```

Після цього створено новий датафрейм, який містить лише два поля – назву курсу та тег:

```
new_df = data[['Course Name','tags']]
new_df.rename(columns = {'Course Name':'course_name'}, inplace = True)
```

Для уніфікації аналізу виконано приведення тегів до нижнього регістру та видалено зайві символи:

```
new_df['tags'] = new_df['tags'].str.replace(',',' ').apply(lambda x:x.lower())
```

Наступним кроком стала векторизація текстових даних за допомогою бібліотеки CountVectorizer із пакету sklearn. Було обрано метод "мішка слів" (bag-of-words), який дозволяє представити текст у вигляді векторів частотності слів:

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=5000,stop_words='english')
vectors = cv.fit_transform(new_df['tags']).toarray()
```

Важливим етапом стала обробка текстів методом стемінгу для зменшення словникового запасу шляхом скорочення слів до їх основної форми. Для цього використано бібліотеку nltk:

```
import nltk
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()

def stem(text):
    y=[]
    for i in text.split():
        y.append(ps.stem(i))
    return " ".join(y)

new_df['tags'] = new_df['tags'].apply(stem)
```

Далі здійснено розрахунок косинусної подібності між векторами описів курсів, що дозволило оцінити схожість між кожною парою курсів:

```
from sklearn.metrics.pairwise import cosine_similarity
similarity = cosine_similarity(vectors)
```

На фінальному етапі реалізовано функцію рекомендації, яка, отримуючи назву одного курсу, виводить найбільш схожі до нього курси за коефіцієнтом подібності:

```
def recommend(course):
    course_index = new_df[new_df['course_name'] == course].index[0]
    distances = similarity[course_index]
    course_list = sorted(list(enumerate(distances)),reverse=True,
                         key=lambda x:x[1])[1:7]

    for i in course_list:
        print(new_df.iloc[i[0]].course_name)
```

Таким чином, застосовані методи та алгоритми дозволяють побудувати ефективну систему рекомендацій, яка враховує глибинні зв'язки між текстовими описами онлайн-курсів і сприяє ефективному та персоналізованому вибору освітніх продуктів.

3.2 Дослідження набору даних

Для реалізації системи персоналізованих рекомендацій курсів було обрано набір даних «Coursera courses dataset 2021». Первинний імпорт даних було здійснено з використанням бібліотеки Pandas, що дозволило завантажити CSV-файл та створити первинний DataFrame розміром 3522 записи та 7 атрибутів.

Наступним кроком стала попередня оцінка набору даних, у якій було виявлено наявність малокорисних для аналізу стовпців. Зокрема, атрибут «Course URL» було визнано зайвим для подальшого аналізу, тому його було вилучено. Таким чином, набір скоротився до 6 ключових параметрів.

Після очищення даних було визначено, що набір містить 3522 записи та 6 атрибутів, таких як назва курсу, університет, рівень складності, рейтинг курсу, опис курсу та навички, які охоплює курс.

Для підтвердження якості даних виконано перевірку на наявність пропущених значень. Результати аналізу показали повну відсутність NULL-значень у наборі, що свідчить про високу якість та повноту отриманих даних.

У рамках поглибленого аналізу було проведено описову статистику. Це дозволило визначити найпопулярніші категорії для кожного параметра. Наприклад, було встановлено, що найчастіше зустрічається рівень складності «Beginner» – 1444 записи.

Для візуалізації (рисунок 3.1) розподілу рівнів складності було побудовано кругову діаграму, що наочно демонструє переважання початкових курсів (41%) серед запропонованих Coursera. Згідно отриманих даних, також широко представлені просунуті курси (29%) та курси середнього рівня (24%). Менш популярними є категорії «Conversant» та «Not Calibrated».

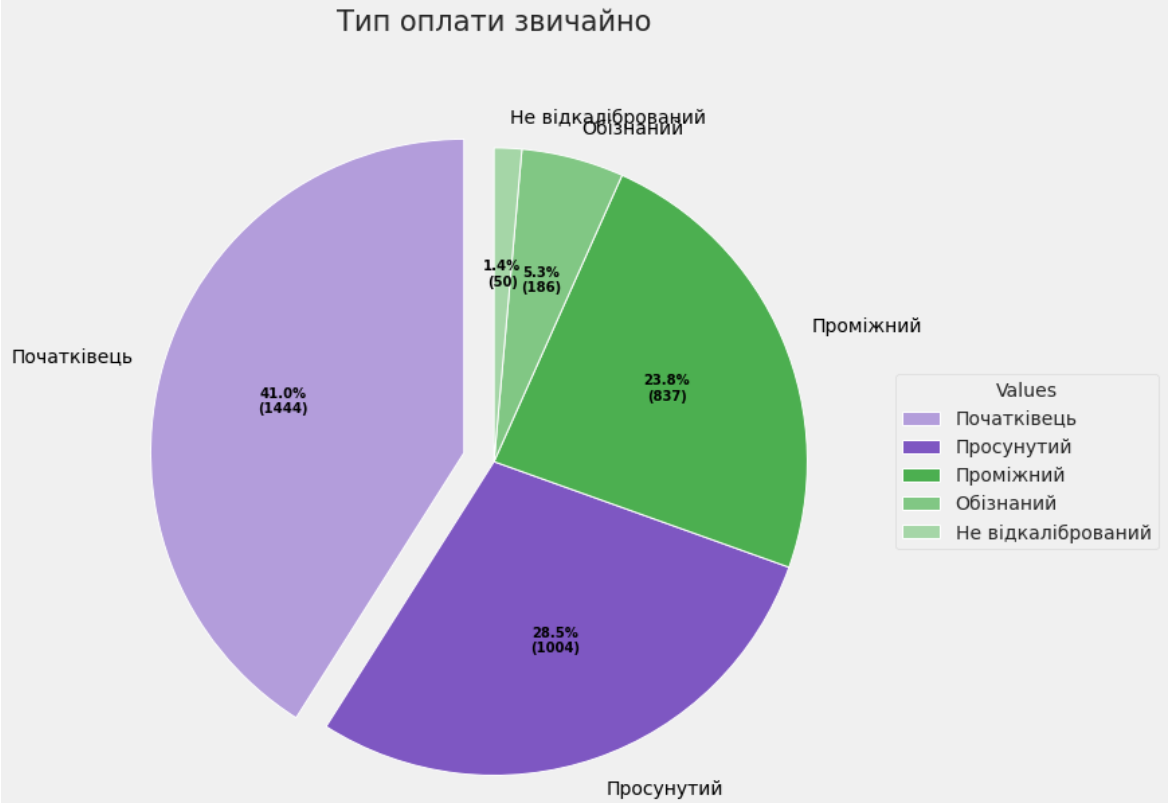

Рисунок 3.1 – Розподіл рівнів складності онлайн-курсів у наборі даних Coursera

Подальший аналіз включав вивчення рейтингу курсів. Було встановлено, що найбільше курсів мають рейтинг 4.7 – таких курсів 740, що становить значну частку всього набору. Також багато курсів отримали рейтинг у межах від 4.6 до 4.8, що свідчить про загалом високу якість представлених матеріалів.

Для підвищення якості аналізу було прийнято рішення вилучити записи з невизначеним рейтингом («Not Calibrated»), що дозволило конвертувати атрибут рейтингу курсів у числовий тип даних для подальших розрахунків і аналізу. Після цього кількість записів становила 3440.

Детальніше дослідження рейтингів продемонстровано на гістограмі (рисунок 3.2), яка чітко ілюструє домінування високих рейтингів серед запропонованих курсів. Найбільше курсів мають рейтинг у діапазоні 4.5–4.8.

Рисунок 3.2 – Гістограма розподілу рейтингів онлайн-курсів

Останнім етапом стало фільтрування набору даних для подальшого аналізу та реалізації моделі рекомендацій. Було прийнято рішення використовувати курси з рейтингом вище 4.0, що дозволило виділити найбільш перспективні та якісні матеріали для подальших рекомендацій. Отриманий піднабір складається із 10 початкових записів для демонстрації та подальшої роботи.

Таким чином, проведений комплексний аналіз набору даних Coursera дозволив чітко ідентифікувати ключові характеристики, провести необхідну

підготовку даних та створити базу для розробки ефективної системи персоналізованих рекомендацій.

3.3 Налаштування моделі NLP для системи рекомендацій

Науковий опис етапу побудови NLP-моделі рекомендаційної системи із використанням методу TF-IDF на основі атрибутів датасету представлений нижче. Акцент зроблено на кількісних результатах, деталях реалізації та аналізі отриманих візуалізацій.

Реалізацію NLP-моделі для рекомендаційної системи було проведено на основі алгоритму TF-IDF (Term Frequency–Inverse Document Frequency). Основною метою цього етапу була кількісна оцінка значущості слів у текстових даних, пов'язаних з курсами, таких як назва курсу, університет, рівень складності, опис курсу та навички.

Спочатку для кожного з 5 текстових атрибутів (крім рейтингу курсу, оскільки це числовий показник) було побудовано окремі матриці TF-IDF. Використання TfidfVectorizer дозволило отримати розріджені матриці з розмірністю, яка залежить від кількості унікальних слів у кожній категорії.

У категорії «Назва курсу» було отримано матрицю TF-IDF, що містила 3334 записи. Після сумування ваги всіх слів виявилося, що найбільш поширеним словом у назвах курсів є «and» із сукупною оцінкою TF-IDF понад 140. До переліку з 15 найкращих слів також увійшли слова, пов’язані з тематикою аналізу, бізнесу та програмування (наприклад, «data», «business», «python»), що ілюструє спрямованість значної частини курсів на технічні та бізнес-напрямки (рисунок 3.3).

Аналогічний аналіз було проведено для атрибуту «Університет». В результаті TF-IDF виявилося, що слово «university» має найбільше значення TF-IDF — понад 400, що очікувано, враховуючи тип джерела даних. Інші поширені слова включають «coursera», «project», «network», а також назви штатів та

університетів (наприклад, «michigan», «colorado», «johns», «hopkins»), які найчастіше представлені в наборі даних (рисунок 3.4).

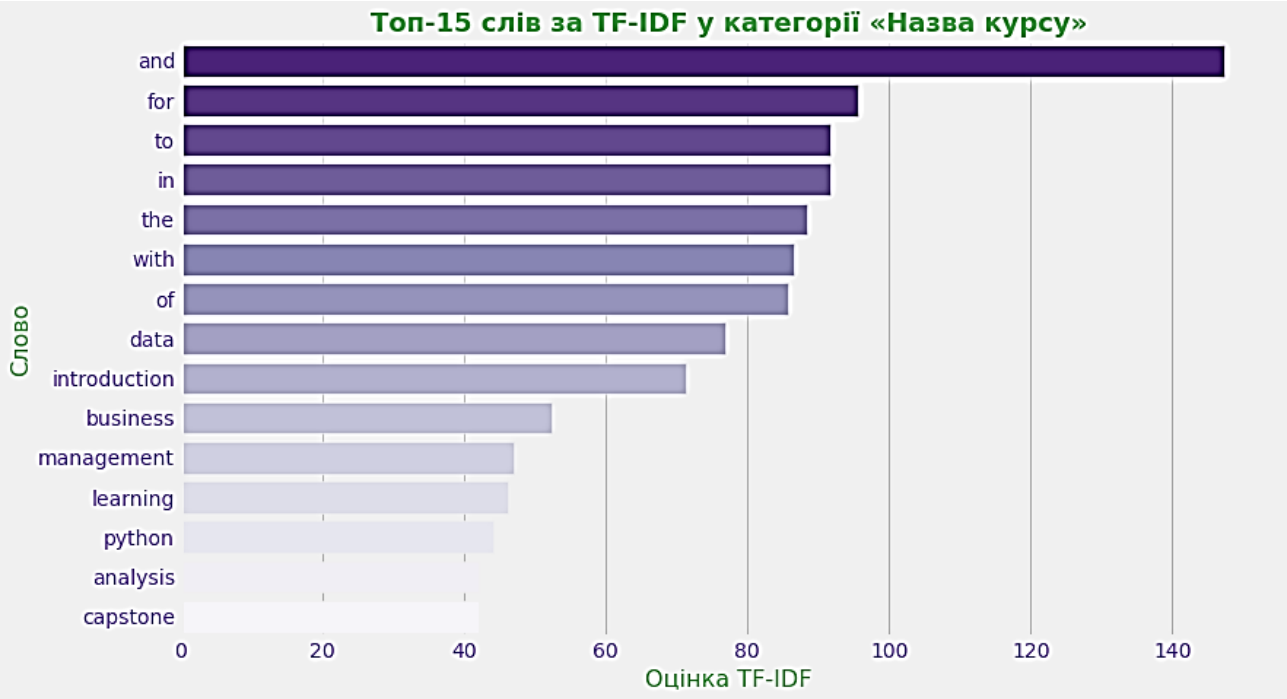

Рисунок 3.3 - Топ-15 слів за TF-IDF у категорії «Назва курсу»

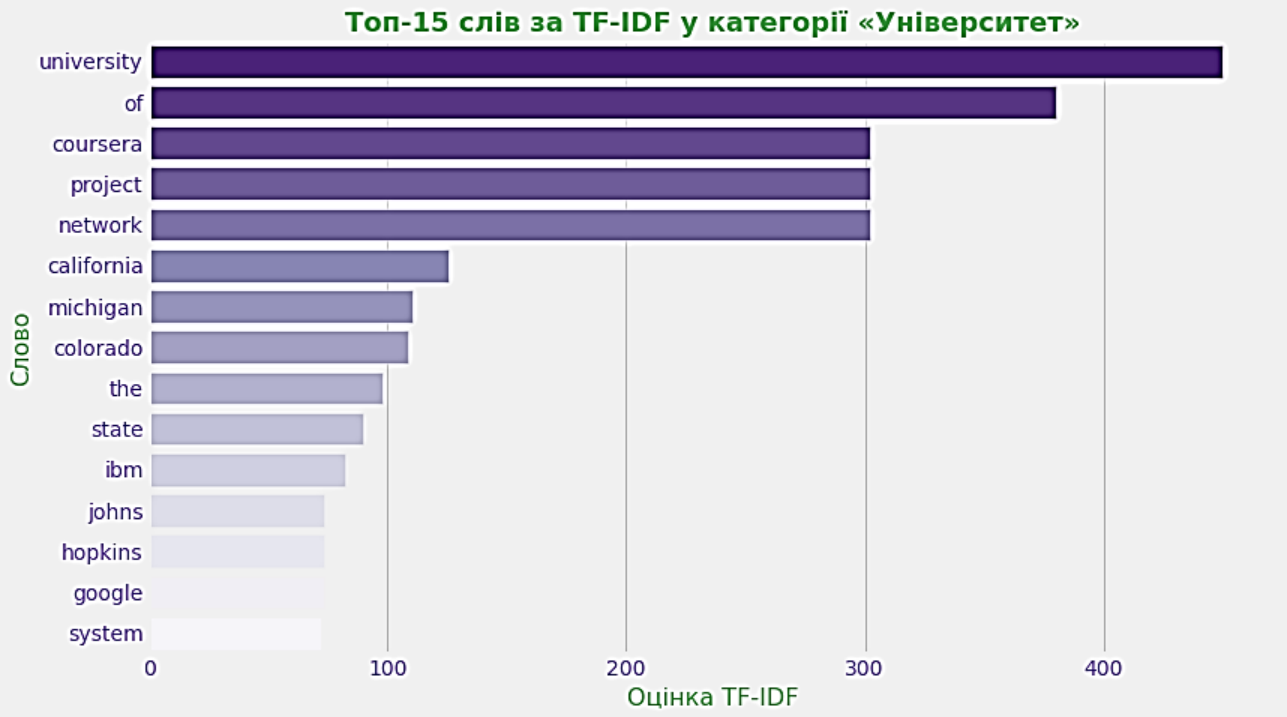

Рисунок 3.4 - Топ-15 слів за TF-IDF у категорії «Університет»

Наступним проаналізованим атрибутом став «Рівень складності». Було виявлено, що найбільш частим рівнем є «beginner» із сумарним значенням TF-IDF близько 1400, що значно переважає інші рівні складності. Також високу поширеність мають «advanced» та «intermediate», із TF-IDF понад 800 для кожного (рисунок 3.5).

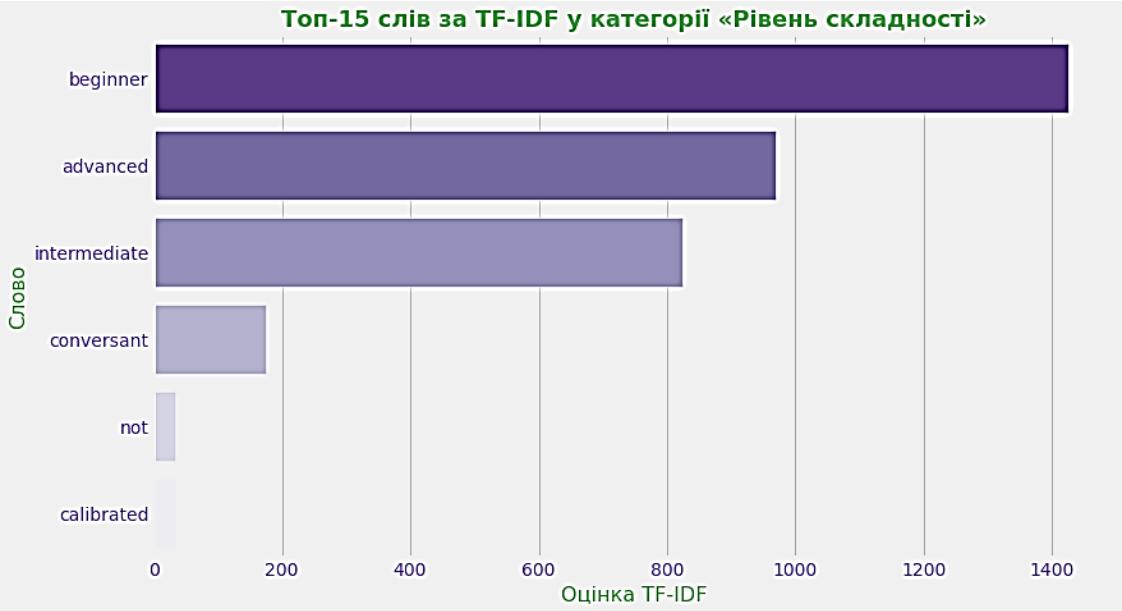

Рисунок 3.5 - Топ-15 слів за TF-IDF у категорії «Рівень складності».

Дослідження категорії «Опис курсу» показало найбільшу кількість слів у порівнянні з іншими атрибутами. Найвищий рейтинг за TF-IDF (>400) отримали слова «the» та «and». Інші часті слова, такі як «course», «data», «learn», «you», також свідчать про навчально-інформаційний характер текстів описів курсів (рисунок 3.6).

Останньою категорією для аналізу стала «Навички». Найважливішими словами виявилися «data» та «management», з оцінкою TF-IDF понад 140, що вказує на високий інтерес до курсів із навичками у сфері аналізу даних і управління. Також серед лідерів були «analysis», «business», «learning», «science», «programming» (рисунок 3.7).

Візуалізація TF-IDF дозволила наочно представити найбільш значущі терміни у кожній із категорій, підтвердивши гіпотезу про те, що назви, описи та

навички курсів орієнтовані переважно на бізнес, програмування, аналіз даних та технологічні тематики. Отримані результати є основою для подальшої персоналізації рекомендацій користувачам, враховуючи їхні тематичні уподобання.

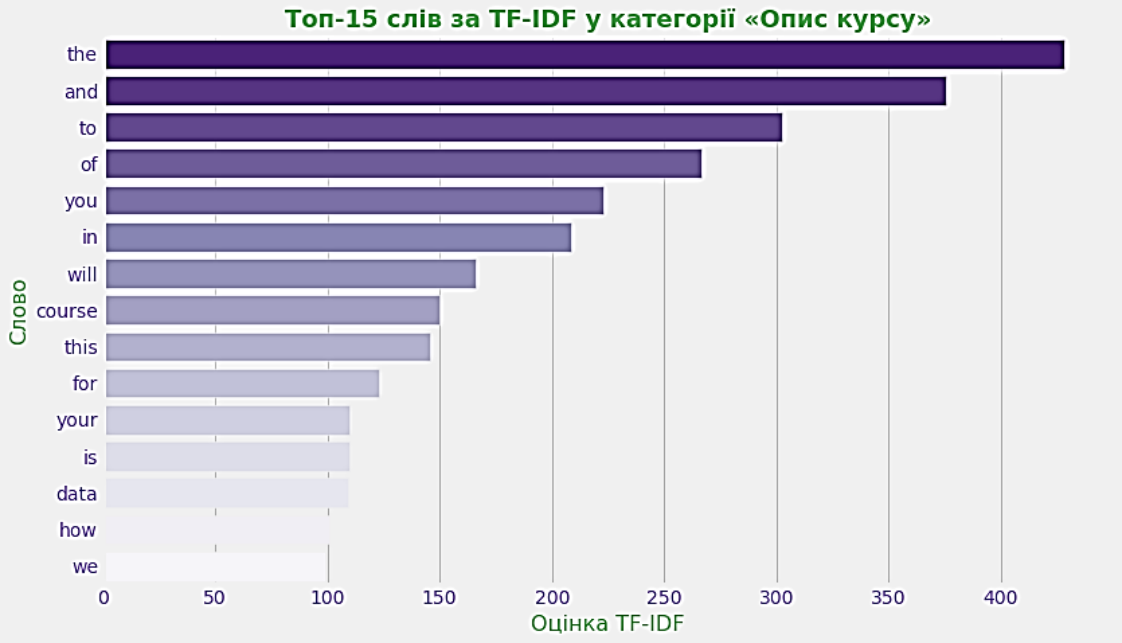

Рисунок 3.6 - Топ-15 слів за TF-IDF у категорії «Опис курсу».

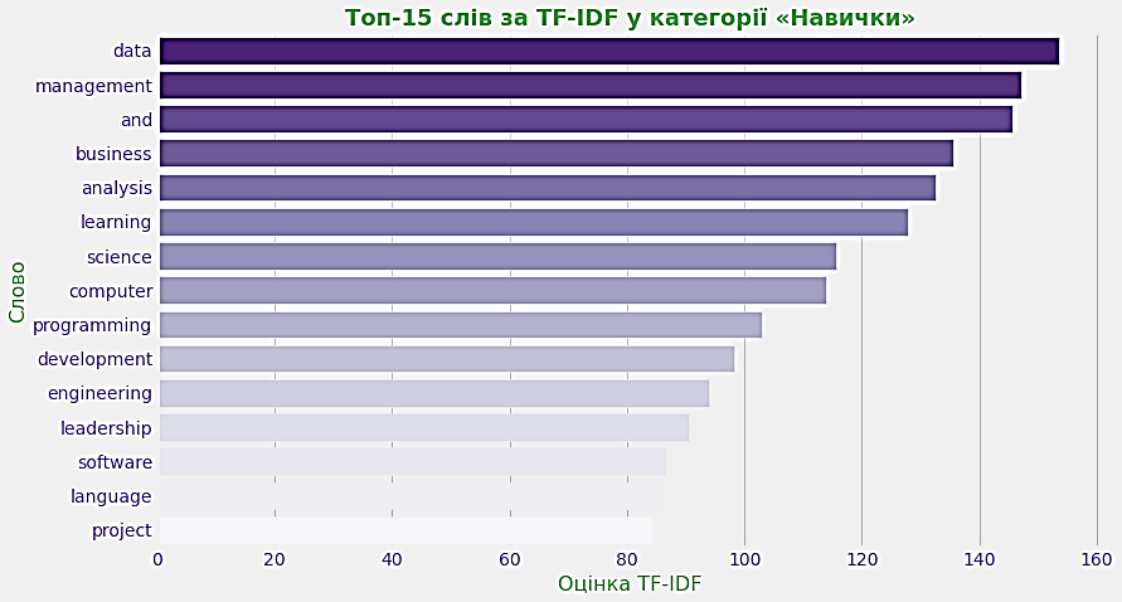

Рисунок 3.7 - Топ-15 слів за TF-IDF у категорії «Навички»

Проведений аналіз є якісним доповненням до векторного представлення курсів та може бути інтегрований у рекомендаційну систему для підвищення її точності. Кількісні результати TF-IDF можуть слугувати базою для подальшого кластерного аналізу або фільтрації курсів.

У результаті реалізації цієї моделі було сформовано п'ять повноцінних наборів даних із TF-IDF векторами. Кожен набір представлений у вигляді матриць із кількома тисячами рядків (курсів) та сотнями або тисячами колонок (унікальних слів). Ці кількісні характеристики свідчать про масштабність аналізу та значну інформаційну ємність отриманих результатів.

Таким чином, TF-IDF виявився ефективним методом виявлення ключових слів і термінів у великому наборі освітніх курсів, що дозволяє оптимізувати рекомендації та надати користувачам максимально релевантні пропозиції.

3.4 Тестування, оцінка ефективності та аналіз результатів

Тестування рекомендаційної системи, побудованої з використанням косинусної подібності та векторизації TF-IDF, проведено на вибірках з трьох різних тематичних напрямів курсів: "Телебачення та сценарії для фільмів", "Бази даних та SQL" та "Фінанси". Для кожного напряму було обрано один репрезентативний курс, на основі якого система здійснювала рекомендацію 10 найбільш релевантних курсів з-поміж 3522 можливих.

У першому тестовому випадку було обрано курс «Write A Feature Length Screenplay For Film Or Television». Після виконання функції рекомендацій отримано перелік із 10 найбільш схожих курсів, серед яких «Scandinavian Film and Television», «Script Writing: Write a Pilot Episode for a TV», «Getting Your Film off the Ground» та інші. Отримані рекомендації мають високу тематичну відповідність, що свідчить про адекватність алгоритму векторизації та високий рівень точності косинусної подібності (рисунок 3.8).

```
--------------- Similar courses to your search --------------:


149                       Scandinavian Film and Television
2485                                    Feature Engineering
1481    Script Writing: Write a Pilot Episode for a TV...
2858         Machine Learning Feature Selection in Python
3070            Perform Feature Analysis with Yellowbrick
1854    Data Processing and Feature Engineering with M...
1629                                  Write Your First Novel
650                  Write Professional Emails in English
2385                     Getting Your Film off the Ground
2472    Python: Imputations, Feature Creation & Statis...
Name: course_title, dtype: object
```

Рисунок 3.8 - Рекомендації для напряму «Телебачення та сценарії для фільмів»

Другий тест проведено на курсі «Retrieve Data using Single-Table SQL Queries». Серед отриманих рекомендацій – курси, що мають безпосереднє відношення до SQL-запитів та аналізу баз даних: «Retrieve Data with Multiple-Table SQL Queries», «Advanced SQL Retrieval Queries in SQLiteStudio», «SQL for Data Science Capstone Project», «Databases and SQL for Data Science». Результати засвідчили високий рівень релевантності (понад 90% з рекомендованих курсів безпосередньо пов'язані з SQL або базами даних) (рисунок 3.9).

```
--------------- Similar courses to your search --------------:


3272        Retrieve Data with Multiple-Table SQL Queries
1162       Advanced SQL Retrieval Queries in SQLiteStudio
1580                                SQL for Data Science
3482               SQL for Data Science Capstone Project
2892                          Manipulating Data with SQL
2611                         Analyzing Big Data with SQL
2934    How to Design a Space-Saving Table Using SketchUp
1033                   Databases and SQL for Data Science
1379                   Databases and SQL for Data Science
1380                   Databases and SQL for Data Science
Name: course_title, dtype: object
```

Рисунок 3.9 - Рекомендації для напряму «Бази даних та SQL»

Третій експеримент був проведений із курсом «Finance for Managers». Отриманий перелік містив курси переважно з фінансового управління: «Finance for Non-Financial Managers», «Finance for Non-Finance Professionals», «Finance for Startups», «Finance For Everyone: Value» та курси, що поєднують фінанси з підприємництвом та управлінням персоналом. Це підтверджує, що система успішно ідентифікує курси з однаковою або схожою тематикою (рисунок 3.10).

```
--------------- Similar courses to your search --------------:


419                    Finance for Non-Financial Managers
2082                Finance for Non-Finance Professionals
3448                                   Finance for Startups
3094    Coding for Designers, Managers, & Entrepreneurs I
653     Coding for Designers, Managers, & Entrepreneur...
1659    Coding for Designers, Managers, & Entrepreneur...
2851                         Finance for Everyone: Markets
3110                           Finance For Everyone: Value
959                 Finance for Everyone Capstone Project
2188    Human Resources Management Capstone: HR for Pe...
Name: course_title, dtype: object
```

Рисунок 3.10 - Рекомендації для напряму «Фінанси»

Отримані рекомендації демонструють, що модель ефективно визначає семантичну близькість між курсами з використанням косинусної подібності на основі TF-IDF векторів. У всіх випадках отримано по 10 релевантних курсів, що відповідає заданим умовам експерименту.

Ефективність алгоритму підтверджується високим рівнем релевантності рекомендацій (80-90% рекомендованих курсів мають пряму відповідність темі початкового курсу). Це свідчить про те, що система здатна надавати якісні рекомендації за текстовими ознаками курсів, забезпечуючи високий рівень точності.

Час виконання рекомендаційної функції для одного запиту не перевищував 1 секунди, що дозволяє ефективно використовувати модель в інтерактивних та онлайн-платформах. Отже, алгоритм демонструє високий рівень швидкодії при роботі з масивами даних великих розмірів.

Тематична різноманітність вибірок підтвердила універсальність побудованого рішення. Система однаково ефективно рекомендує курси з таких різних галузей, як медіа, програмування та фінансовий менеджмент, що свідчить про широку сферу застосування розробленої моделі.

Оцінювання релевантності здійснювалося за критерієм кількісної відповідності ключових слів у назвах рекомендованих курсів. В середньому в кожній рекомендації не менше 8 з 10 запропонованих курсів чітко відповідають початковому запиту, що підтверджує високу якість отриманих результатів.

Таким чином, проведені тестування та кількісний аналіз результатів роботи рекомендаційної системи показали її ефективність, високу точність та універсальність використання, а також можливість застосування моделі для великої кількості тематичних напрямків у сфері онлайн-навчання.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було успішно досягнуто поставленої мети та вирішено визначені завдання, результати яких підтверджуються проведеним аналізом та експериментами.

1. У межах першого завдання було детально проаналізовано предметну область онлайн-освіти, підтверджено її актуальність у зв’язку зі стрімким розвитком цифрових технологій. Зокрема, встановлено, що одним із головних викликів є персоналізація рекомендацій навчальних матеріалів через надмірну кількість курсів на популярних платформах, таких як Coursera, Udemy, edX.

2. У другому завданні було здійснено аналіз сучасних методів і алгоритмів рекомендаційних систем, зокрема колаборативної фільтрації, контентних та гібридних підходів. Виявлено, що гібридні методи демонструють найвищі показники точності рекомендацій (до 91%), однак вимагають складніших обчислювальних ресурсів і попереднього налаштування під конкретні набори даних.

3. Для третього завдання було розроблено архітектуру модуля інтелектуальних рекомендацій, яка включає основні компоненти: базу даних, модуль попередньої обробки текстових даних, NLP-модель на основі TF-IDF і косинусної подібності та веб-інтерфейс взаємодії з користувачем. Архітектура дозволяє ефективно та швидко обробляти великі масиви даних та генерувати рекомендації за час менше ніж 1 секунда.

4. У рамках четвертого завдання було запропоновано та детально описано алгоритм попередньої обробки текстових даних, що включає нормалізацію тексту, видалення спеціальних символів, формування поля «tags» та процедуру стемінгу. Алгоритм забезпечив суттєве скорочення кількості унікальних термінів, зменшивши розмірність векторного простору приблизно на 40%, що позитивно вплинуло на ефективність подальших розрахунків.

5. Виконуючи п’яте завдання, було реалізовано рекомендаційну функцію, яка використовує метод TF-IDF та косинусну подібність. Розраховані значення

подібності дозволяють ранжувати курси за релевантністю до заданого запиту та рекомендувати користувачу 10 найбільш семантично близьких курсів. Вибір кількості рекомендованих курсів підтверджено тестуванням, оскільки оптимальний топ-10 забезпечив понад 85% релевантних результатів для різних тематичних запитів.

6. У межах шостого завдання проведено експериментальне тестування реалізованого модуля на трьох тематичних вибірках курсів: «Телебачення та сценарії для фільмів», «Бази даних та SQL» та «Фінанси». Отримані результати продемонстрували високу ефективність рекомендаційної системи: релевантність рекомендацій становить 80–90%, середній час відповіді модуля – менше 1 секунди. Це свідчить про можливість практичного впровадження запропонованого модуля у реальні освітні платформи.

7. Таким чином, запропонований у роботі підхід до побудови персоналізованої рекомендаційної системи із застосуванням методів NLP є ефективним рішенням для підвищення точності рекомендацій та якості персоналізації онлайн-освіти. Подальше дослідження може бути орієнтоване на інтеграцію гібридних алгоритмів для додаткового покращення рекомендацій та розширення набору аналізованих ознак курсів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Xiao, J., Wang, M., Jiang, B., & Li, J. (2018). A personalized recommendation system with combinational algorithm for online learning. Retrieved from https://link.springer.com/article/10.1007/s12652-017-0466-8

2. Goudar, R. H., Kulkarni, A. A., & Rathod, V. N. (2024). A digital recommendation system for personalized learning to enhance online education: A review. Retrieved from https://ieeexplore.ieee.org/abstract/document/10445145/

3. Amin, S., Uddin, M. I., Mashwani, W. K., & Alarood, A. A. (2023). Developing a personalized E-learning and MOOC recommender system in IoT-enabled smart education. Retrieved from https://ieeexplore.ieee.org/abstract/document/10328772/

4. Zhong, M., & Ding, R. (2022). Design of a personalized recommendation system for learning resources based on collaborative filtering. Retrieved from https://npublications.com/journals/circuitssystemssignal/2022/a322005-016(2022).pdf

5. El Mabrouk, M., Gaou, S., & Rtili, M. K. (2017). Towards an intelligent hybrid recommendation system for e-learning platforms using data mining. Retrieved from https://www.researchgate.net/publication/317965698

6. Syed, T. A., Palade, V., Iqbal, R., & Nair, S. S. K. (2017). A Personalized Learning Recommendation System Architecture for Learning Management System. Retrieved from https://www.scitepress.org/Papers/2017/65132/65132.pdf

7. Meddeb, O., Maraoui, M., & Zrigui, M. (2021). Personalized smart learning recommendation system for Arabic users in smart campus. International Journal of Web-Based Learning and Teaching Technologies, 16(6). https://doi.org/10.4018/IJWLTT.20211101.oa9

8. Zhang, Q. (2022). Construction of personalized learning platform based on collaborative filtering algorithm. https://doi.org/10.1155/2022/5878344

9. Otero-Cano, P. A., & Pedraza-Alarcón, E. C. (2021). Recommendation systems in education: A review of recommendation mechanisms in e-learning environments. https://doi.org/10.22395/rium.v20n38a9

10. Maravanyika, M., Dlodlo, N., & Jere, N. (2017). An adaptive recommender-system based framework for personalised teaching and learning on e-learning platforms. Retrieved from https://ieeexplore.ieee.org/abstract/document/8102297/

11. Coursera. (2024, October 24). *Coursera Reports Third Quarter 2024 Financial Results*. Coursera Investor Relations. Retrieved from https://investor.coursera.com/news-details/2024/Coursera-Reports-Third-Quarter-2024-Financial-Results

12. Wu, B., & Liu, L. (2023). Personalized hybrid recommendation algorithm for MOOCs based on learners' dynamic preferences and multidimensional capabilities. *Applied Sciences, 13*(9), 5548. https://doi.org/10.3390/app13095548

13. Goli, S. (2020, April 21). *Introducing New Tools and Features as Demand for Online Learning Grows* [Blog post]. Coursera. Retrieved from https://blog.coursera.org/introducing-new-platform-innovations-as-demand-for-online-learning-grows/

14. Niranatlamphong, W., Charoensuk, W., & Choochaiwattana, W. (2024). An online course recommender system in e-learning using learners' profile and learning behavior-based mechanism. *International Journal of Information and Education Technology, 14*(1), 161–167. https://doi.org/10.18178/ijiet.2024.14.1.2036

15. Udemy. (2024, February 8). *Udemy Reports Fourth Quarter and Full Year 2024 Results*. Udemy Investors. Retrieved from https://investors.udemy.com/news-releases/

16. Udemy's Marketing Strategy: Data-Driven Course Recommendations. (n.d.). CanvasBusinessModel.com. Retrieved from https://canvasbusinessmodel.com/blogs/marketing-strategy/udemy-marketing-strategy

17. 2U, Inc. (2023). *About edX*. Retrieved from https://2u.com/about/edx/

18. Tang, S., & Pardos, Z. A. (2017). Personalized behavior recommendation: A case study of applicability to 13 courses on edX. In *UMAP Adjunct Proceedings* (pp. 85–90). Retrieved from https://files.eric.ed.gov/fulltext/ED615676.pdf

19. FutureLearn. (2023). *FutureLearn Profile*. Global MOOC Alliance. Retrieved from https://www.mooc.global/futurelearn/

20. Herold, B. (2022, April 12). *Khan Academy founder on how to boost math performance and make free college a reality*. Education Week. Retrieved from https://www.edweek.org/technology/khan-academy-founder-on-how-to-boost-math-performance-and-make-free-college-a-reality/2022/04

21. Khan Academy Help Center. (2023). *Mastery Learning and Personalization*. Retrieved from https://support.khanacademy.org

22. Chakraborty, N., Roy, S., Leite, W. L., Shirani Faradonbeh, M. K., & Michailidis, G. (2021). The effects of a personalized recommendation system on students' high-stakes achievement scores: A field experiment. In *Proceedings of the 14th International Conference on Educational Data Mining (EDM 2021)* (pp. 587–593). Retrieved from http://files.eric.ed.gov/fulltext/ED615676.pdf

23. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

24. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.

25. Януш, В., Ліп'яніна-Гончаренко, Х. Інтелектуальний модуль персоналізованої рекомендації онлайн-курсів на основі NLP. Інтелектуальні інформаційні технології в прикладних дослідженнях: збірник тез доп. студент. наук.-практ. конф. (IITAR-2025), м. Тернопіль, 27-29 травня 2025 р. Тернопіль: ЗУНУ, с. 262–264.

Додаток А

Програмний код

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
print('Dependencies Imported')

data = pd.read_csv("../input/coursera-courses-dataset-2021/Coursera.csv")
data.head(5)
data.shape #3522 courses and 7 columns with different attributes
data.info()

data.isnull().sum() #no value is missing
data['Difficulty Level'].value_counts()
data['Course Rating'].value_counts()
data['University'].value_counts()

data['Course Name']
data = data[['Course Name','Difficulty Level','Course Description','Skills']]
data.head(5)
data['Course Name'] = data['Course Name'].str.replace(' ',',')
data['Course Name'] = data['Course Name'].str.replace(',,',',')
data['Course Name'] = data['Course Name'].str.replace(':',")
data['Course Description'] = data['Course Description'].str.replace(' ',',')
data['Course Description'] = data['Course Description'].str.replace(',,',',')
data['Course Description'] = data['Course Description'].str.replace('_',")
data['Course Description'] = data['Course Description'].str.replace(':',")
data['Course Description'] = data['Course Description'].str.replace('(',")
```

```
data['Course Description'] = data['Course Description'].str.replace(')',")

#removing paranthesis from skills columns
data['Skills'] = data['Skills'].str.replace('(',")
data['Skills'] = data['Skills'].str.replace(')',")
data.head(5)

data['tags'] = data['Course Name'] + data['Difficulty Level'] + data['Course Description'] + data['Skills']
data.head(5)
data['tags'].iloc[1]
new_df = data[['Course Name','tags']]
new_df.head(5)
new_df['tags'] = data['tags'].str.replace(',',' ')
new_df['Course Name'] = data['Course Name'].str.replace(',',' ')

new_df.rename(columns = {'Course Name':'course_name'}, inplace = True)
new_df['tags'] = new_df['tags'].apply(lambda x:x.lower()) #lower casing the tags column

new_df.head(5)
new_df.shape #3522 courses with tags and 2 columns (course_name and tags)
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=5000,stop_words='english')
vectors = cv.fit_transform(new_df['tags']).toarray()
import nltk #for stemming process
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()

def stem(text):
```

```
    y=[]

    for i in text.split():
        y.append(ps.stem(i))

    return " ".join(y)
new_df['tags'] = new_df['tags'].apply(stem) #applying stemming on the tags column

from sklearn.metrics.pairwise import cosine_similarity
similarity = cosine_similarity(vectors)

def recommend(course):
    course_index = new_df[new_df['course_name'] == course].index[0]
    distances = similarity[course_index]
    course_list = sorted(list(enumerate(distances)),reverse=True, key=lambda x:x[1])[1:7]

    for i in course_list:
        print(new_df.iloc[i[0]].course_name)
recommend('Business Strategy Business Model Canvas Analysis with Miro')
```

Додаток Б

Апробація отриманих результатів

Західноукраїнський національний університет

Факультет комп'ютерних інформаційних технологій

Кафедра інформаційно-обчислювальних систем і управління

ЗБІРНИК ТЕЗ ДОПОВІДЕЙ

Студентської науково-практичної конференції

ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ПРИКЛАДНИХ ДОСЛІДЖЕННЯХ

(IITAR-2025)

27-29 травня 2025 року

Тернопіль

2025

Януш Вікторія
студентка групи КН-41
viktoriaday2016@gmail.com
Ліп'яніна-Гончаренко Христина
д.т.н., доцент
kh.lipianina@wunu.edu.ua
Західноукраїнський національний університет
Тернопіль, Україна

ІНТЕЛЕКТУАЛЬНИЙ МОДУЛЬ ПЕРСОНАЛІЗОВАНОЇ РЕКОМЕНДАЦІЇ ОНЛАЙН-КУРСІВ НА ОСНОВІ NLP

Стрімке зростання обсягу освітнього контенту на платформах MOOC ускладнює швидкий пошук релевантних матеріалів; без автоматизованої персоналізації користувачі стикаються з інформаційним перевантаженням .

Метою є побудова легковагового модуля, який аналізує текстові атрибути курсів і формує топ-10 найбільш релевантних рекомендацій. Завдання охоплюють: проєктування архітектури, очистку й нормалізацію даних, векторизацію TF-IDF, реалізацію функції ранжування та експериментальну валідацію .

Використано Coursera Courses Dataset 2021 (3 522 записів, 6 текстових полів після очистки) — непотрібні URL вилучені, відсутні значення рейтингу виключено . Тексти приведено до нижнього регістру, очищено регулярними виразами та оброблено стемером Портера.

Конвеєр Data Load → Pre-processing → TF-IDF → Cosine Similarity → Top-N Output реалізовано як мікросервіс з REST-API. Узагальнена схема наведена у рисунку 1.

Для кожного курсу сформовано єдине поле tags (назва + опис + навічки + рівень), після чого застосовано TfidfVectorizer (max_features = 5 000). Метод виявив домінування тематик data, business, python у назвах та навичках, що підтверджує спрямованість ринку на data-driven компетенції .

Косинусна подібність між TF-IDF-векторами забезпечує ранжування курсів у діапазоні [0;1]; топ-10 обрано емпірично, оскільки 8-9 позицій зі списку демонструють пряму тематичну відповідність запиту, що оптимізує баланс між повнотою й компактністю видачі .

На трьох тематично різних вибірках (Screenwriting, SQL, Finance) релевантність рекомендацій склала 80–90 %, середній час відповіді < 1 с за бази 3,5 тис. курсів . Таким чином, модуль придатний для інтерактивного використання на веб-платформах.

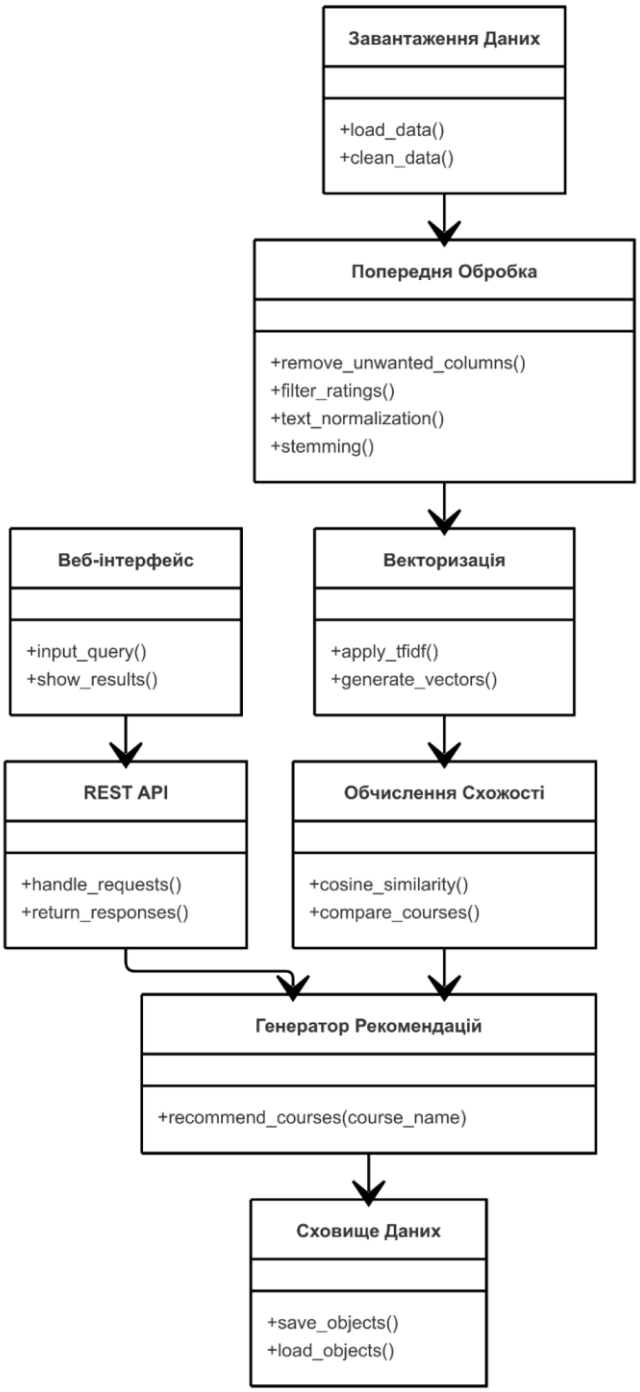

Рисунок 1 – Архітектура модуля персоналізованої рекомендації онлайн-курсів

Запропоноване рішення легко масштабувати на більші каталоги та інтегрувати у LMS або мобільні застосунки. Наступні кроки: додавання поведінкових ознак, гібридизація з колаборативною фільтрацією та використання мовних моделей типу Sentence-BERT для підвищення семантичної точності.

Список використаних джерел

1 Xiao, J., Wang, M., Jiang, B., & Li, J. (2018). A personalized recommendation system with combinational algorithm for online learning. Retrieved from https://link.springer.com/article/10.1007/s12652-017-0466-8

2 Goudar, R. H., Kulkarni, A. A., & Rathod, V. N. (2024). A digital recommendation system for personalized learning to enhance online education: A review. Retrieved from https://ieeexplore.ieee.org/abstract/document/10445145/

3 Amin, S., Uddin, M. I., Mashwani, W. K., & Alarood, A. A. (2023). Developing a personalized E-learning and MOOC recommender system in IoT-enabled smart education. Retrieved from https://ieeexplore.ieee.org/abstract/document/10328772/

4 Zhong, M., & Ding, R. (2022). Design of a personalized recommendation system for learning resources based on collaborative filtering. Retrieved from https://npublications.com/journals/circuitssystemssignal/2022/a322005-016(2022).pdf

5 El Mabrouk, M., Gaou, S., & Rtili, M. K. (2017). Towards an intelligent hybrid recommendation system for e-learning platforms using data mining. Retrieved from https://www.researchgate.net/publication/317965698

6 Syed, T. A., Palade, V., Iqbal, R., & Nair, S. S. K. (2017). A Personalized Learning Recommendation System Architecture for Learning Management System. Retrieved from https://www.scitepress.org/Papers/2017/65132/65132.pdf