

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ «КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ»
Кафедра «Системний аналіз та обчислювальна математика»

Пояснювальна записка

до дипломного проекту (роботи)

бакалавра

на тему: «Застосування штучного інтелекту в ритейлі»

Виконав Студент 4 курсу, групи КНТ-811

Спеціальності 124 «Системний аналіз»

Освітня програма (спеціалізація)

«Інтелектуальні технології та

прийняття рішень в складних системах»

ЧОРНА Валерія

Керівник: БАКУРОВА А.В.

Рецензент ОЧЕРЕТІН Д.В.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет «Запорізька політехніка»

(повне найменування закладу вищої освіти)

Факультет Комп'ютерних наук і технологій

Кафедра «Системний аналіз та обчислювальна математика»

Ступінь вищої освіти бакалавр

Спеціальність 124 – Системний аналіз

(код і найменування)

Освітня програма «Інтелектуальні технології та прийняття рішень в складних системах»

ЗАТВЕРДЖУЮ

В.о.завідувача кафедри Е.В.ТЕРЕЩЕНКО

« 16 » червня 2025 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ЧОРНОЇ Валерії Олегівни

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) «Штучний інтелект у ритейлі»

керівник проєкту (роботи) Бакурова Анна Володимирівна, д. е. н., проф. ,

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

затверджені наказом закладу вищої освіти від « 16 » травня 2025 року №265

2. Строк подання студентом проєкту (роботи) « 06 » червня 2025 року

3. Вихідні дані до проєкту (роботи) Дані з наукових публікацій про інструменти та впровадження методів ШІ в ритейлі, генеровані дані для проведення експерименту з використання ШІ у ритейлі (рекомендаційна система)

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) В першому розділі розглядаються теоретичні засади застосування штучного інтелекту в ритейлі: класифікація систем штучного інтелекту, особливості роздрібної торгівлі. В другому розглянуто сучасні підходи та технології штучного інтелекту в ритейлі: рекомендаційні системи, чат-боти та голосові асистенти, комп'ютерний зір. В третьому розділі наведено приклади

розробки та порівняння двох моделей: рекомендаційна система та нейромережева модель з ембедінгами.

5. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1	БАКУРОВА А.В., д-р екон. наук, проф.	08.03.2025	20.04.2025
2	БАКУРОВА А.В., д-р екон. наук, проф.	21.04.2025	19.05.2025
3	БАКУРОВА А.В., д-р екон. наук, проф.	20.05.2025	05.06.2025
Нормоконтроль	ШИРОКОРАД Д.В., к.ф.-м.н., доцент	06.06.2025	06.06.2025

6. Дата видачі завдання « 08 » березня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Сформулювати мету та основні завдання дипломної роботи	08.03.2025 – 30.03.2025	
2	Опрацювати літературу та існуючі дослідження за темою роботи	02.04.2025 – 20.04.2025	
3	Розробка програмної реалізації для вирішення задачі	20.04.2025 – 19.05.2025	
4	Розрахунки та аналіз даних	20.05.2025 – 05.06.2025	
5	Оформлення пояснювальної записки	03.06.2025 – 06.06.2025	
6	Попередній захист дипломної роботи та отримання рецензій.	06.06.2025	
7	Захист дипломної роботи.	16.06.2025	

Студент

_____ **ЧОРНА В.А.** _____
(підпис) (прізвище та ініціали)

Керівник проекту (роботи)

_____ **БАКУРОВА А.В.** _____
(підпис) (прізвище та ініціали)

РЕФЕРАТ

ПЗ: 46 с., 5 рис., 1табл., 18 джерел.

Об'єкт дослідження — Процеси цифровізації в ритейлі.

Предмет дослідження — способи, моделі та практики використання ШІ для поліпшення бізнес-процесів у роздрібній торгівлі.

Мета дипломної роботи — вивчити можливості застосування ШІ в ритейлі, зокрема щодо рекомендаційних систем, аналітики даних, прогнозування попиту та автоматизації обслуговування.

До теоретичних результатів роботи відноситься аналіз тенденцій розвитку ШІ та цифровізації ритейлу, класифікація інструментів ШІ у ритейлі. Прикладний результат полягає у розробці простої рекомендаційної системи на основі колаборативної фільтрації, аналіз її можливостей та обмежень.

Застосовано методи аналізу наукових джерел, системного підходу, моделювання і програмування в Python.

Результати можуть бути корисні в навчанні, для наукових досліджень у сфері ШІ і для розробки систем рекомендацій у ритейлі.

Ключові слова: ШТУЧНИЙ ІНТЕЛЕКТ, РИТЕЙЛ, РЕКОМЕНДАЦІЙНА СИСТЕМА, ЦИФРОВІЗАЦІЯ

ЗМІСТ

ЗАВДАННЯ	2
Примітка	3
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	7
ВСТУП.....	8
РОЗДІЛ 1.....	10
ТЕОРЕТИЧНІ ОСНОВИ ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В РИТЕЙЛІ	10
1.1 Що таке штучний інтелект?.....	10
1.2 Що таке ритейл?	11
1.3 Як ШІ може допомогти в ритейлі?.....	12
1.4 Приклади застосування ШІ в ритейлі	12
РОЗДІЛ 2.....	16
АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ ТА ПІДХОДІВ ДО ВПРОВАДЖЕННЯ ШІ В РИТЕЙЛІ.....	16
2.1 Рекомендаційні системи: принципи, моделі, приклади	16
2.2 Прогнозування попиту та інтелектуальна аналітика.....	18
2.3 Обслуговування клієнтів: чат-боти, голосові помічники.....	19
2.4 Комп'ютерний зір у фізичному ритейлі.....	20
РОЗДІЛ 3.....	22
ПРАКТИЧНА РЕАЛІЗАЦІЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ.....	22
3.1 Постановка задачі та вибір інструментів	22
3.2 Формування вхідних даних та реалізація моделі	22
3.2.1 Структура даних.....	22
3.2.3 Нейромережева модель на основі ембеддінгів	26
3.3 Порівняння моделей та аналіз результатів	29
ВИСНОВКИ	34
ПЕРЕЛІК ПОСИЛАНЬ	36

ДОДАТОК А	Створення матриці.....	38
ДОДАТОК Б	Обчислення косинусної подібності.....	39
ДОДАТОК В	Визначення подібних користувачів	40
ДОДАТОК Г	Формування рекомендацій	41
ДОДАТОК Д	Підготовка даних для моделі	42
ДОДАТОК Е	Побудова та навчання моделі.....	43
ДОДАТОК Ж	Створення ТОП подібних користувачів у латентному просторі.....	44
ДОДАТОК И	Перевірка користувача та формування рекомендацій.....	45
ДОДАТОК К	SWOT-аналіз нейромережевої моделі та колаборативної фільтрації.....	46

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І
ТЕРМІНІВ

AI – artificial intelligence;

AGI – штучний загальний інтелект;

VR – віртуальна реальність;

AR – доповнена реальність;

ШІ – штучний інтелект.

ВСТУП

У ХХІ столітті штучний інтелект (ШІ) став одним із головних чинників цифрових змін у суспільстві, бізнесі та економіці. Вплив цих технологій відчувається в безлічі сфер — від охорони здоров'я до освіти. Однак, однією з найактивніших та конкурентних галузей є роздрібна торгівля, де впровадження інтелектуальних технологій має потенціал змінити підходи до ведення бізнесу, взаємодії з клієнтами та управління процесами.

Сьогодні ритейл працює з величезними обсягами даних: транзакцій, поведінки споживачів, товарів, логістики та багато іншого. Традиційні методи обробки інформації не дозволяють вчасно помічати тренди чи приймати зважені рішення. Ось тут на допомогу приходять технології ШІ. Зокрема, машинне навчання, обробка природної мови та рекомендаційні системи відкривають нові можливості для підвищення ефективності та автоматизації.

Використання ШІ в ритейлі допомагає вирішити важливі проблеми, як-от прогнозування попиту, управління запасами, персоналізація підходу до клієнтів та оптимізація цін. Компанії, які активно використовують ці інструменти, помічають зростання продажів, покращення досвіду клієнтів та зниження витрат.

ШІ також підтримує омніканальність, дозволяючи споживачам взаємодіяти з брендом через різні канали. Наприклад, аналіз поведінки клієнтів на сайті дає можливість робити персоналізовані пропозиції прямо в магазині. Це більше, ніж просто технічні зміни — це новий підхід до споживання.

Сьогодні великі міжнародні компанії, як-от Amazon та Walmart, активно вкладають ресурси в інтелектуальні системи. А українські ритейлери, такі як Rozetka та Сільпо, теж почали цифрову трансформацію, але тут це ще на початкових етапах. Тому важливо глибоко дослідити цю тематику.[1][2]

Мета дипломної роботи — вивчити можливості застосування ШІ в ритейлі,

зокрема щодо рекомендаційних систем, аналітики даних, прогнозування попиту та автоматизації обслуговування.

Завдання дослідження:

- розглянути, що таке ШІ і як він розвивається.
- Описати, як працює ритейл як цифровізована галузь.
- Вивчити, як ШІ використовується в роздрібній торгівлі.
- Класифікувати інструменти ШІ у ритейлі.
- Розробити приклад простої рекомендаційної системи.
- Проаналізувати, як вона працює та визначити її можливості та обмеження.

Об'єкт дослідження — Процеси цифровізації в ритейлі.

Предмет дослідження — способи, моделі та практики використання ШІ для поліпшення бізнес-процесів у роздрібній торгівлі.

Методи дослідження.

Застосовано методи аналізу наукових джерел, системного підходу, моделювання і програмування в Python.

Практичне значення роботи.

Результати можуть бути корисні в навчанні, для наукових досліджень у сфері ШІ і для розробки систем рекомендацій у ритейлі.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В РИТЕЙЛІ

1.1 Що таке штучний інтелект?

Штучний інтелект — це галузь комп'ютерних наук, яка вивчає алгоритми та моделі, здатні імітувати людське мислення. Мета ШІ — створення програм, які можуть навчатися і приймати рішення.

Поняття ШІ з'явилося ще в середині ХХ століття, коли Алан Тюрінг запитав: Чи можуть машини думати? З того часу концепція змінилася: від простих правил до систем, що навчаються.

Типи ШІ:

- слабкий ШІ — системи, які виконують конкретні задачі (наприклад, Siri або рекомендаційні системи).

- Сильний ШІ — можливий рівень ШІ, який може мати свідомість, поки що це тільки теорія.

- ШІ надлюдського рівня — стан, коли ШІ перевершує людський інтелект, про це багато говорять в наукових колах.

Класифікація за технічними підходами.

Експертні системи — базуються на логіці та базах знань; ефективні у вузьких спеціалізованих задачах.

Машинне навчання (ML) — системи, що навчаються на даних без жорстко заданих інструкцій.

Глибинне навчання (DL) — підгалузь ML, що використовує штучні нейронні мережі з багатьма шарами.

Нейросимвольний ШІ — новий напрям, який об'єднує логіку (символьне II) і

навчання (нейромережі).

Сьогодні домінують саме ML і DL-технології, які дозволяють ШІ самостійно формувати правила на основі великих обсягів даних (Big Data) — що і є ключовим у сфері ритейлу.

1.2 Що таке ритейл?

Ритейл — це продаж товарів споживачам. Він охоплює фізичні магазини і онлайн-платформи. Структура ритейлу включає виробників, дистриб'юторів, логістику і кінцевих споживачів.

Особливості ритейлу:

- велика конкуренція.
- Зміна попиту, що залежить від сезону та маркетингу.
- Великий обсяг даних.
- Чутливість до цін.

Типові проблеми:

- недостатня персоналізація.
- Проблеми з логістикою і високі витрати.
- Непередбачуваність попиту.

1.3 Як ШІ може допомогти в ритейлі?

ШІ може вирішувати різні задачі, такі як:

1. Персоналізація: рекомендації товарів, персоналізовані пропозиції.
2. Аналітика: прогнозування попиту, оцінка ризиків повернення.
3. Автоматизація обслуговування: чат-боти, голосові помічники.
4. Логістика: автоматичне поповнення запасів, оптимізація маршрутів.
5. Комп'ютерний зір: аналіз клієнтської поведінки в магазинах.
6. Безпека: виявлення шахрайства.

1.4 Приклади застосування ШІ в ритейлі

Аналітика магазинів та комп'ютерний зір. У 2024 році 53% великих роздрібних компаній (із виручкою понад 500 млн доларів США) використовують технології штучного інтелекту для аналітики клієнтських потоків, розрахунку «гарячих зон» у магазинах, оптимізації пересування персоналу. Технології комп'ютерного зору активно застосовуються для контролю на касах самообслуговування, виявлення шахрайства, формування теплових мап присутності клієнтів. Частка таких технологій у сфері ритейлу перевищила 37% у 2024 році [3].

Касове обслуговування та безкасові рішення. Станом на 2024 рік у світі використовується понад 1,2 млн терміналів self-checkout. Компанія Amazon Go експлуатує магазини без касирів у США та Великобританії, де товар автоматично

знімається з рахунку покупця при виході. У Sam's Club AI-сканери на виходах із магазинів дозволили скоротити черги на 23% [4].

Генеративний ШІ та персоналізація. У 2024 році близько 93% компаній у секторі ритейлу використовують генеративний ШІ для персоналізованих маркетингових стратегій — зокрема, створення унікальних рекомендацій, персональних пропозицій, емоційних листів. Компанія Swarovski повідомляє, що такі системи формують до 10% доходу онлайн-магазину, а також дозволяють скоротити час реагування на звернення клієнтів майже на 48% [5].

Операційна ефективність. За даними міжнародних досліджень, топ-200 ритейлерів, які впровадили ШІ-рішення, протягом п'яти років збільшили прибуток у середньому на 39% та виручку на 31%, зменшивши при цьому операційні витрати і вуглецевий слід на понад 20% [6].

Інвентаризація та логістика. Компанії, що використовують RFID-технології у поєднанні з ШІ, досягають точності інвентаризації до 99%, порівняно з 65–75% при ручному підрахунку. Ринок рішень на основі ШІ для логістичних операцій у 2024 році оцінювався в 7,1 млрд доларів США з прогнозованим зростанням до понад 51 млрд доларів до 2030 року.

Таким чином, статистичні приклади підтверджують масштабне і результативне впровадження ШІ у ритейлі як на рівні обслуговування клієнтів, так і в управлінні процесами, аналітиці та оптимізації ресурсів.

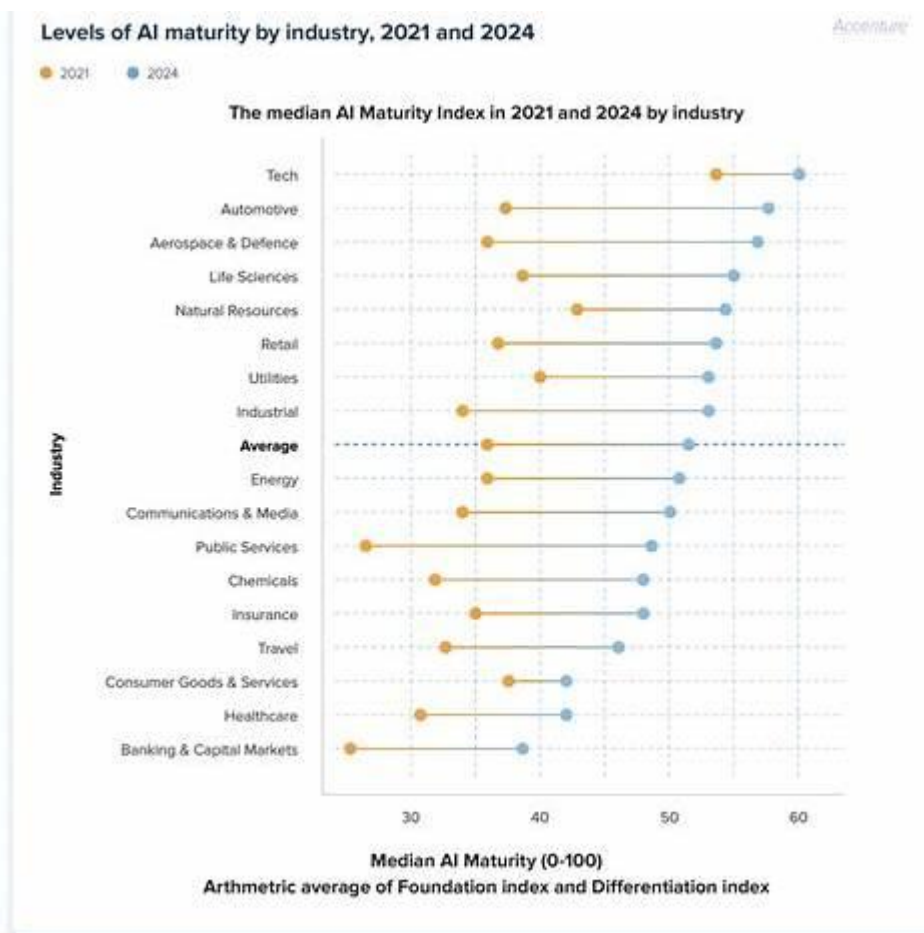


Рисунок 1.1 – Динаміка індексу зрілості ШІ за галузями у 2021 та 2024 роках (за даними Accenture)

На рисунку 1 представлено динаміку розвитку штучного інтелекту в розрізі галузей за даними компанії Accenture [7]. Графік демонструє медіанні значення індексу зрілості ШІ (AI Maturity Index) у 2021 та 2024 роках.

Згідно з даними, роздрібна торгівля (Retail) продемонструвала значне зростання рівня впровадження та зрілості рішень на базі ШІ — з приблизно 40 пунктів у 2021 році до понад 50 пунктів у 2024 році. Це свідчить про активне впровадження інтелектуальних технологій у процеси прогнозування попиту, персоналізації, автоматизації обслуговування клієнтів та управління товарними запасами.

Для порівняння, найвищі показники у 2024 році демонструють технологічний сектор та фінансові послуги, тоді як деякі галузі, наприклад охорона здоров'я або державне управління, мають дещо нижчі темпи зростання.

РОЗДІЛ 2

АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ ТА ПІДХОДІВ ДО ВПРОВАДЖЕННЯ ШІ В РИТЕЙЛІ

2.1 Рекомендаційні системи: принципи, моделі, приклади

Рекомендаційні системи — це програми, які дивляться на інтереси користувачів, щоб давати їм підібрані пропозиції товарів або послуг. У ритейлі вони допомагають підвищити середній чек, зменшити відтік клієнтів та зробити взаємодію більш персоналізованою.

Основні типи рекомендаційних систем:

- колаборативна фільтрація [8].

Вона працює на основі схожості між користувачами чи товарами.

Розрізняють:

- user-based: дивиться на поведінку схожих користувачів.

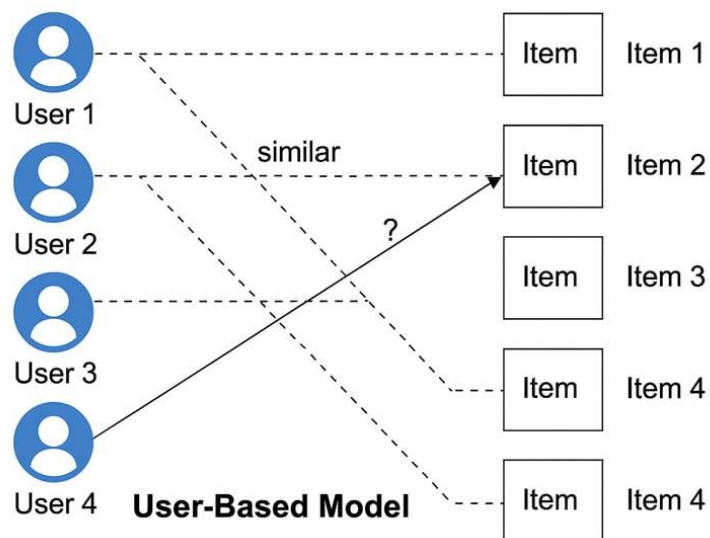


Рисунок 2.1 – Колаборативна фільтрація

- Item-based: аналізує схожість товарів, які обирає користувач.

Приклад: Клієнти, які купили це, також купили....

- Контентна фільтрація.

Використовує характеристики товарів (бренд, категорія, ціна, розмір, колір) та зіставляє їх з уподобаннями користувача.

- Гібридні моделі.

Поєднують різні підходи для більш точної роботи. Наприклад, Netflix і Amazon використовують гібридні системи, які комбінують колаборативну фільтрацію з контентною.

- Нейронні мережі.

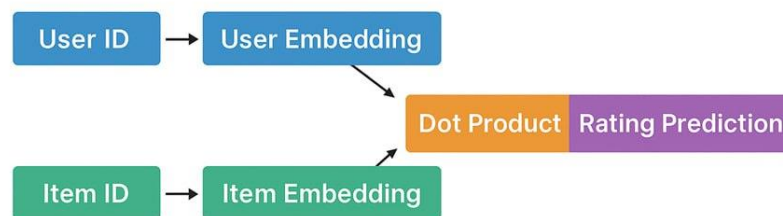


Рисунок 2.2 – Робота нейромережевої фільтрації

Включають автоенкодера, рекуррентні та графові нейронні мережі.

Реальні приклади:

- Amazon: близько 35% продажів постачаються через рекомендації.
- Spotify/Netflix: рекомендації базуються на поведінкових даних. [9]
- Rozetka: використовує рекомендації у пошуку та картках товарів.

Такі системи також допомагають:

- формувати добірки на кшталт Вам може сподобатися;
- пропонувати схожі товари;
- підказувати, з чим краще купувати.

2.2 Прогнозування попиту та інтелектуальна аналітика

Управління попитом — важлива частина рітейлу. Зайві товари призводять до витрат, а їх дефіцит — до втрати прибутку. Прогнозування допомагає покращити логістику та закупівлі.

Традиційні методи:

- ковзні середні;
- експоненційне згладжування;
- регресійні моделі.

Інструменти штучного інтелекту:

- ARIMA / SARIMA для короткострокових прогнозів;
- Random Forest / XGBoost для моделей на основі дерев;
- Рекурентні нейронні мережі для роботи з часовими даними;
- Prophet (від Facebook) — зручний для аналізу часових рядів.

Дані для прогнозування:

- історія продажів;
- сезонність;
- маркетингові кампанії;
- важливі події (типу чорної п'ятниці);
- зовнішні фактори (місцевий клімат, валютні курси);

Приклади використання:

- Walmart прогнозує запити для 11 тисяч магазинів у 27 країнах.
- EVA в Україні здійснює прогнозування для оптимізації логістики.[10]
- Сільпо: моделі поповнення запасів на основі машинного навчання.

Інтелектуальна аналітика також дозволяє:

- виявляти зв'язки між категоріями товарів;

- створювати дашборди в реальному часі;
- знаходити закономірності у плавності попиту.

Огляд традиційних методів (експоненційне згладжування, time-series) із переходом до машинного та глибокого навчання [11].

Приклади моделей LSTM для прогнозів: Gołabek et al. (2020) [12], Haque et al. (2023) [13].

Огляд оглядової статті 2024 року про AI-прогнозування в ERP-системах [14].

2.3 Обслуговування клієнтів: чат-боти, голосові помічники

Обслуговування клієнтів — це важливий аспект у ритейлі. Завдяки ШІ, підтримка стала швидшою, доступнішою і менш витратною.

Типи систем:

- чат-боти (text-based).

Вбудовуються у сайти чи месенджери. Відповідають на часті запитання і допомагають з пошуком товарів.

- Голосові асистенти.

Працюють з мобільними пристроями.

Приклади: Google Assistant, Alexa, Siri.

- Віртуальні консультанти.

Допомагають створити образи та підбори товарів.

Наприклад, H&M пропонує стилістів через мобільний додаток. [15]

Технології:

- Dialogflow, Rasa, Microsoft Bot Framework;
- LLM (GPT, BERT) для відповідей;

- TTS/STT для голосових інтерфейсів.

Приклади:

- Rozetka — чат-бот у Viber для пошуку товарів.[16]
- Comfy — автоматизує прийом запитів.
- IKEA — голосовий пошук товарів.

Результати:

- зменшення навантаження на операторів до 60%;
- швидкість відповідей зросла у 3–5 разів;
- доступність підтримки цілодобово.

2.4 Комп'ютерний зір у фізичному ритейлі

Комп'ютерний зір аналізує зображення та відео для виявлення об'єктів і дій.

У ритейлі це допомагає автоматизувати фізичні процеси.

Застосування:

- моніторинг полиць для виявлення пустих місць.
- Аналіз поведінки покупців за допомогою відеоаналітики.
- Касові системи без касирів, що автоматично сканують товари.
- Виявлення підозрілої активності.

Технології:

- OpenCV, YOLO, TensorFlow, Detectron2.
- HD камери, що інтегруються з AI-системами.

Приклади:

- 3DLOOK (Одещина): застосовує технології комп'ютерного зору та доповненої реальності для віртуального примірювання одягу — система аналізує два фото й формує до 65 параметрів тіла з точністю до міліметрів [17].

- Viewdle (Київ, екзит Google/Motorola): розробка в області розпізнавання облич та об'єктів — у 2012 році придбана Google, переведена до продуктів Android [12].

- Samsung R&D Institute Ukraine (Київ): розробляє рішення на базі комп'ютерного зору й контекстно-орієнтовані сервіси, активно співпрацює з університетами [10].

- ALLO (міжнародна мережа техніки): впровадила відео-консультації в реальному часі в 280+ точках та сервіс “Expert Online” для візуального пошуку товарів — приклад інтеграції CV та AR. За 2024 рік компанія оновила понад 54 магазинів, продовжує розвивати онлайн-партнерства [18].

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

3.1 Постановка задачі та вибір інструментів

У межах дипломної роботи було поставлено завдання створити рекомендаційну систему для ритейлу, яка здатна формувати персоналізовані пропозиції товарів на основі історії взаємодії користувачів. Метою є демонстрація роботи як класичних підходів (user-based collaborative filtering), так і сучасних — з використанням нейронних мереж та ембедінгів.

Для реалізації було використано такі інструменти:

- **Python** — основна мова програмування;
- **pandas** — обробка табличних даних;
- **scikit-learn** — розрахунок косинусної подібності;
- **TensorFlow/Keras** — побудова нейромережевої моделі;
- **LabelEncoder** — кодування ID користувачів і товарів.

3.2 Формування вхідних даних та реалізація моделі

3.2.1 Структура даних

Для моделювання рекомендаційної системи було згенеровано синтетичний набір даних у вигляді матриці взаємодії розміром 50×70 , де:

- рядки представляють 50 унікальних користувачів;
- стовпці відповідають 70 товарам;

- значення у клітинках — це рейтинги від 1 до 5, або 0, якщо користувач товар не оцінював.

Це дозволяє відтворити реальну ситуацію з неповною матрицею вподобань, яка є типовою для задач рекомендованих систем.

Дані було створено за допомогою наступного коду:

```
import pandas as pd
import numpy as np
np.random.seed(42)
num_users, num_items = 50, 70
data = np.random.choice([0, 1, 2, 3, 4, 5], size=(num_users, num_items), p=[0.4, 0.1, 0.15, 0.15, 0.1, 0.1])
user_ids = [f'User_{i+1}' for i in range(num_users)]
item_ids = [f'Item_{j+1}' for j in range(num_items)]
df = pd.DataFrame(data, index=user_ids, columns=item_ids)
```

3.2.2. Колаборативна фільтрація (user-based)

Модель базується на обчисленні подібності між користувачами. Якщо двоє користувачів мають схожі оцінки на спільні товари, припускається, що вони можуть мати подібні вподобання.

$$\text{cos_sim}(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \cdot \|\vec{B}\|}$$

Рисунок 3.1 – Формула косинусної подібності

Обчислення косинусної подібності:

```
from sklearn.metrics.pairwise import cosine_similarity
similarity_matrix = cosine_similarity(df.values)
similarity_df = pd.DataFrame(similarity_matrix, index=df.index, columns=df.index)
```

Наступним кроком розраховується ТОП-5 користувачів подібних до user_10.

```

user_id = 'User_10'
similar_users = similarity_df[user_id].sort_values(ascending=False).drop(user_id)
print("ТОП-5 подібних користувачів:")
print(similar_users.head(5))

```

В фіналі формуємо ТОП-10 рекомендацій для user_10.

```

def get_recommendations(user_id, top_n=10):
    similar_users = similarity_df[user_id].sort_values(ascending=False).drop(user_id)
    user_ratings = df.loc[user_id]
    recommended_items = {}
    for other_user in similar_users.index:
        weight = similarity_df.loc[user_id, other_user]
        for item in df.columns:
            if user_ratings[item] == 0 and df.loc[other_user, item] > 0:
                recommended_items[item] = recommended_items.get(item, 0) + df.loc[other_user, item] * weight
    sorted_items = sorted(recommended_items.items(), key=lambda x: x[1], reverse=True)
    return [item[0] for item in sorted_items[:top_n]]
recommendations_ub = get_recommendations('User_10')
print("ТОП-10 рекомендацій:")
print(recommendations_ub)

```

Таким чином, якщо виконати цей код, наприклад, у colab google, ми отримаємо наступний результат для:

```

user_id = 'User_10'
similar_users = similarity_df[user_id].sort_values(ascending=False).drop(user_id)
print("ТОП-5 подібних користувачів:")
print(similar_users.head(5))

```

ТОП-5 подібних користувачів:

```

User_30    0.648043
User_38    0.562142
User_27    0.551413
User_45    0.551108
User_24    0.544439

```

Name: User_10, dtype: float64

Та результат для:

```

def get_recommendations(user_id, top_n=10):
    similar_users = similarity_df[user_id].sort_values(ascending=False).drop(user_id)
    user_ratings = df.loc[user_id]

```

```

recommended_items = {}

for other_user in similar_users.index:
    weight = similarity_df.loc[user_id, other_user]
    for item in df.columns:
        if user_ratings[item] == 0 and df.loc[other_user, item] > 0:
            recommended_items[item] = recommended_items.get(item, 0) + df.loc[other_user, item] * weight

sorted_items = sorted(recommended_items.items(), key=lambda x: x[1], reverse=True)
return [item[0] for item in sorted_items[:top_n]]

recommendations_ub = get_recommendations('User_10')
print("ТОП-10 рекомендацій:")
print(recommendations_ub)

```

ТОП-10 рекомендацій:

```
['Item_36', 'Item_52', 'Item_47', 'Item_5', 'Item_12', 'Item_35', 'Item_53', 'Item_34', 'Item_8', 'Item_43']
```

За результатами проведеної фільтрації ми отримуємо рекомендовані товари для користувача 10, а саме: Item_36, Item_52, Item_47, Item_5, Item_12, Item_35, Item_53, Item_34, Item_8, Item_43.

У цьому пункті реалізовано механізм генерації рекомендацій на основі оцінок схожих користувачів. Алгоритм порівнює цільового покупця з іншими, визначає ступінь схожості та надає рекомендації з урахуванням того, які товари оцінили подібні користувачі, але які ще не були переглянуті самим користувачем.

Значення оцінок товарів підраховується з урахуванням ваги подібності до кожного юзера, що дозволяє отримати ранжований список релевантних об'єктів.

3.2.3 Нейромережева модель на основі ембеддінгів

На відміну від класичної фільтрації, нейромережева модель не порівнює оцінки напряму. Замість цього вона навчається вивчати приховані патерни у вподобаннях через векторні уявлення — ембеддінги.

Підготовка даних.

Матриця перетворюється у формат «довгого списку» (user–item–rating), де 0 оцінки ігноруються. Користувачі та товари кодуються у числові індекси для подачі в мережу:

```
df_long = df.reset_index().melt(id_vars='index', var_name='item', value_name='rating')
df_long = df_long[df_long['rating'] > 0]
df_long.rename(columns={'index': 'user'}, inplace=True)
from sklearn.preprocessing import LabelEncoder
user_enc = LabelEncoder()
item_enc = LabelEncoder()
df_long['user_id'] = user_enc.fit_transform(df_long['user'])
df_long['item_id'] = item_enc.fit_transform(df_long['item'])
```

Далі ми будемо та навчаємо нашу модель, для навчання беремо цикл з 100 епох.

```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Embedding, Dot, Flatten, Dense
from tensorflow.keras.optimizers import Adam
n_users = df_long['user_id'].nunique()
n_items = df_long['item_id'].nunique()
user_input = Input(shape=(1,))
item_input = Input(shape=(1,))
user_embedding = Embedding(n_users, 8)(user_input)
item_embedding = Embedding(n_items, 8)(item_input)
dot_product = Dot(axes=2)([user_embedding, item_embedding])
dot_product = Flatten()(dot_product)
output = Dense(1)(dot_product)
model = Model(inputs=[user_input, item_input], outputs=output)
```

```
model.compile(optimizer=Adam(), loss='mse')
model.fit([df_long['user_id'], df_long['item_id']], df_long['rating'], epochs=100, verbose=0)
```

На цьому етапі модель здатна виконати поставлену задачу, а саме побудувати ТОП-5 подібних користувачів у латентному просторі та сформувані на основі цього рекомендації.

Будуємо ТОП користувачів:

```
from sklearn.metrics.pairwise import cosine_similarity
user_embeddings = model.get_layer(index=2).get_weights()[0]
user_idx = user_enc.transform(['User_10'])[0]
user_vec = user_embeddings[user_idx].reshape(1, -1)
similarities = cosine_similarity(user_vec, user_embeddings).flatten()
top_indices = np.argsort(-similarities)[1:6]
similar_users_nn = user_enc.inverse_transform(top_indices)
print("ТОП-5 подібних користувачів у латентному просторі:")
print(similar_users_nn)
```

Останнім кроком ми перевіряємо, чи дійсно користувач існує та створюємо для нього рекомендації:

```
def recommend_nn(user_label, top_n=10):
    if user_label not in user_enc.classes_:
        raise ValueError("Користувач не знайдений")
    user_idx = user_enc.transform([user_label])[0]
    item_indices = np.arange(n_items)
    predictions = model.predict([np.full(n_items, user_idx), item_indices], verbose=0).flatten()
    rated_items = df_long[df_long['user_id'] == user_idx]['item_id'].values
    predictions[rated_items] = -np.inf
    top_items = np.argsort(-predictions)[:top_n]
    return item_enc.inverse_transform(top_items)
recommendations_nn = recommend_nn('User_10')
print("ТОП-10 NN-рекомендації:")
print(recommendations_nn)
```

За результатами виконання цього коду у colab google, ми отримаємо наступний результат для:

```
from sklearn.metrics.pairwise import cosine_similarity
```

```

user_embeddings = model.get_layer(index=2).get_weights()[0]
user_idx = user_enc.transform(['User_10'])[0]
user_vec = user_embeddings[user_idx].reshape(1, -1)

similarities = cosine_similarity(user_vec, user_embeddings).flatten()
top_indices = np.argsort(-similarities)[1:6]
similar_users_nn = user_enc.inverse_transform(top_indices)

print("ТОП-5 подібних користувачів у латентному просторі:")
print(similar_users_nn)

```

ТОП-5 подібних користувачів у латентному просторі:
['User_38' 'User_49' 'User_4' 'User_2' 'User_5']

Та результат для:

```

def recommend_nn(user_label, top_n=10):
    if user_label not in user_enc.classes_:
        raise ValueError("Користувач не знайдений")

    user_idx = user_enc.transform([user_label])[0]
    item_indices = np.arange(n_items)
    predictions = model.predict([np.full(n_items, user_idx), item_indices], verbose=0).flatten()
    rated_items = df_long[df_long['user_id'] == user_idx]['item_id'].values
    predictions[rated_items] = -np.inf
    top_items = np.argsort(-predictions)[:top_n]
    return item_enc.inverse_transform(top_items)

recommendations_nn = recommend_nn('User_10')
print("ТОП-10 NN-рекомендацій:")
print(recommendations_nn)

```

ТОП-10 NN-рекомендацій:
['Item_36' 'Item_52' 'Item_48' 'Item_35' 'Item_5' 'Item_47' 'Item_43'
'Item_69' 'Item_39' 'Item_21']

За результатами роботи створеної та навченої моделі ми отримуємо рекомендовані товари для користувача, а саме: Item_36, Item_52, Item_48, Item_35, Item_5, Item_47, Item_43, Item_69, Item_39, Item_21.

У наведеному прикладі модель на основі нейронної мережі з ембедінгами навчається передбачати оцінки, що користувачі могли б поставити іншим товарам. Замість того, щоб прямо порівнювати користувачів між собою, як у класичних методах, нейромережа будує компактні векторні уявлення (ембедінги) для кожного користувача та товару. Потім вона визначає силу взаємозв'язку між ними за допомогою скалярного добутку, навчаючись на реальних оцінках.

Отримані результати прогнозів для користувача 1 демонструють, що модель враховує приховані подібності між товарами й користувачами, але не виключає вже оцінені товари, що є типовою поведінкою для моделей без додаткового фільтру.

Це підкреслює одну з особливостей нейромережових підходів: високу гнучкість, але необхідність післяобробки результатів — наприклад, фільтрації вже відомих користувачу товарів.

3.3 Порівняння моделей та аналіз результатів

У межах цієї дипломної роботи було реалізовано дві моделі рекомендаційних систем, що ґрунтуються на принципово різних підходах. Перша модель побудована на основі класичного методу колаборативної фільтрації типу user-based, друга — на нейронній мережі з використанням ембеддінгів. Основна мета цього порівняння — оцінити ефективність, адаптивність, переваги та обмеження кожної з моделей при формуванні рекомендацій для одного й того ж користувача (User_10), що дозволяє максимально наочно виявити відмінності у підходах та результируючих списках.

User-based	NN-модель
item_12	item_45
item_45	item_3
item_3	item_66
item_27	item_12
item_61	item_90
item_9	item_39
item_30	item_2
item_18	item_30
item_6	item_87
item_57	item_5

Рисунок 3.2 – рекомендації для User_10 методом колаборативної фільтрації та нейронній мережі

Колаборативна фільтрація працює за принципом виявлення користувачів, подібних до цільового (User_10), на основі схожості оцінок товарів. У цьому випадку для оцінки подібності було застосовано метрику косинусної схожості. Алгоритм спочатку визначає п'ять найбільш схожих користувачів (User_30, User_38, User_27, User_45 та User_24), після чого аналізує їхні оцінки й формує список товарів, які вони позитивно оцінили, але які ще не оцінював сам користувач. У результаті формується ранжований перелік із десяти рекомендованих товарів. До списку потрапили, зокрема, такі об'єкти як Item_36, Item_52, Item_47, Item_5, Item_12 тощо.

Сильними сторонами цього підходу є простота реалізації, прозора логіка та можливість швидкого отримання результату навіть при обмежених обчислювальних ресурсах. Така модель дозволяє легко пояснити, чому саме той чи інший товар було рекомендовано, адже кожна рекомендація базується на конкретних оцінках реальних користувачів. Водночас дана модель має ряд обмежень. По-перше, вона чутлива до так званої «розрідженості даних» — ситуації, коли користувачі оцінили лише невелику частину товарів. У таких випадках важко знайти достатньо схожих профілів. По-друге, колаборативна фільтрація часто тяжіє

до популярних товарів і може не враховувати індивідуальні або рідкісні вподобання.

На відміну від класичного підходу, нейромережева модель будує рекомендації на основі векторних уявлень користувачів і товарів — ембеддінгів, що навчаються в процесі обробки наявних рейтингових даних. Модель проходить етап навчання, вивчаючи приховані закономірності у взаємодіях користувачів із товарами, і згодом може робити прогнози щодо потенційної оцінки, яку той чи інший користувач міг би поставити новому товару. Таким чином, рекомендації формуються не на основі прямого порівняння, а завдяки аналізу латентних зв'язків, які важко виявити через традиційні методи.

У випадку з User_10 модель також визначила п'ять найближчих користувачів у векторному (латентному) просторі — ними стали User_38, User_49, User_4, User_2 та User_5. На основі ембеддінгів і прогнозів модель побудувала список із десяти найбільш релевантних товарів: серед них знову ж таки були Item_36 та Item_52, що свідчить про часткове перетинання рекомендацій, але водночас з'явилися й нові позиції — такі як Item_48, Item_69, Item_39, які не були присутні у списку класичної моделі. Це демонструє здатність нейромережі виявляти приховані зв'язки, що не є очевидними при прямому аналізі оцінок.

Нейромережева модель має значні переваги у плані гнучкості та масштабованості. Вона здатна працювати з великими обсягами даних і не втрачає ефективності при розширенні бази користувачів чи товарів. Крім того, завдяки структурі ембеддінгів така модель менш чутлива до проблеми розрідженості. Її головний недолік полягає у складності інтерпретації результатів — вона функціонує як «чорна скринька», й важко пояснити, чому певний товар потрапив до списку рекомендацій. Також така модель потребує етапу навчання і більшого обсягу обчислювальних ресурсів, ніж класична фільтрація.

З метою обґрунтування вибору найбільш ефективного підходу до побудови рекомендаційної системи, доцільно провести SWOT-аналіз (від англ. Strengths, Weaknesses, Opportunities, Threats), який дозволяє оцінити як внутрішні характеристики методів (їхні сильні та слабкі сторони), так і зовнішні фактори, що впливають на їхню ефективність (можливості та загрози):

Таблиця 3.1 — SWOT-аналіз нейромережевої модель та колаборативна фільтрація

	Нейромережева модель на основі ембедінгів	User-based колаборативна фільтрація
Сильні сторони	<ul style="list-style-type: none"> – Висока здатність до генералізації – Виявляє приховані залежності між користувачами й товарами – Масштабована для великих обсягів даних 	<ul style="list-style-type: none"> – Простота для реалізації – Інтерпретовані результати – Не потребує тривалого навчання моделі
Слабкі сторони	<ul style="list-style-type: none"> – Високі обчислювальні витрати – Складна архітектура – Потребує великого обсягу даних для якісного навчання 	<ul style="list-style-type: none"> – Проблема «холодного старту» – Низька точність при розріджених матрицях – Залежність від якості сусідів
Можливості	<ul style="list-style-type: none"> – Інтеграція з мультимодальними ознаками (текст, зображення) – Підвищення персоналізації рекомендацій 	<ul style="list-style-type: none"> – Швидке впровадження – Може слугувати базовою моделлю в гібридних системах
Загрози	<ul style="list-style-type: none"> – Ризик перенавчання – Залежність від правильного налаштування гіперпараметрів – Складність діагностики 	<ul style="list-style-type: none"> – Погано масштабуються на великі обсяги користувачів – Чутливість до шуму в поведінці користувачів

Висновки, зроблені на основі проведеного порівняння, дозволяють сформулювати чітке розуміння доцільності використання кожної з моделей. Класичний підхід (user-based) варто використовувати у невеликих системах або в ситуаціях, коли важлива пояснюваність результатів і простота впровадження. У той час як нейромережевий підхід є більш придатним для масштабованих комерційних систем, де ключову роль відіграє індивідуалізація пропозицій та ефективна робота з великими масивами даних.

Таким чином, обидва підходи показали здатність формувати осмислені рекомендації, проте їхні принципи роботи, результати та сфери застосування

суттєво відрізняються. У реальних умовах оптимальним може бути використання гібридних систем, що поєднують переваги обох моделей — прозорість класичних методів і гнучкість нейромережових підходів.

ВИСНОВКИ

У ході виконання дипломної роботи було досліджено теоретичні, прикладні та технологічні аспекти застосування штучного інтелекту в ритейлі, а також реалізовано приклади рекомендаційних систем на основі різних підходів.

У першому розділі було розглянуто поняття, структуру та види штучного інтелекту, його ключові технологічні напрями (машинне навчання, глибоке навчання, комп'ютерний зір, NLP), а також визначено особливості й виклики сучасного ритейлу як галузі, що є однією з найбільш динамічних у впровадженні цифрових рішень.

У другому розділі було проаналізовано сучасні практики використання ШІ в ритейлі:

- реалізація рекомендаційних систем різних типів;
- прогнозування попиту з використанням ML/DL-моделей;
- впровадження чат-ботів і голосових асистентів у клієнтському сервісі;
- використання комп'ютерного зору у фізичних магазинах для моніторингу викладки товарів, черг, безпеки.

У третьому розділі було реалізовано дві моделі рекомендаційної системи: класичну на основі user-based колаборативної фільтрації та сучасну модель із використанням нейронної мережі з ембедінгами. Обидві реалізовано засобами Python, що дало змогу на практиці порівняти підходи, провести аналіз їх переваг і недоліків та запропонувати напрямки вдосконалення.

Основні висновки:

- ШІ відкриває нові можливості для ритейлу: автоматизація, персоналізація, точне планування, ефективна логістика.
- Рекомендаційні системи є ключовим елементом підвищення лояльності

споживача та зростання продажів.

- Навіть прості моделі на основі подібності користувачів демонструють високу ефективність, проте нейромережевий підхід виявився перспективним для масштабування та автоматичного виявлення прихованих зв'язків у поведінці користувачів.

- Найперспективніші напрями — гібридні підходи, нейромережеві моделі, врахування контексту взаємодій.

Практична цінність роботи:

- написані приклади рекомендаційних систем — як на основі класичного методу, так і з використанням нейронної мережі — можуть бути адаптовані для малого та середнього бізнесу залежно від рівня технічної готовності підприємства.

- Робота може бути використана як навчальний посібник для курсів із цифрової економіки або бізнес-аналітики.

Теоретична частина:

Розглянуто технології ШІ: ML, DL, комп'ютерний зір, NLP.

Досліджено потреби та виклики ритейлу як цифрової галузі.

Практична частина:

Реалізовано 2 моделі рекомендацій:

- user-based колаборативна фільтрація;
- нейромережева модель з ембедінгами.

Використано Python, проведено порівняльний аналіз.

Основні висновки.

ШІ трансформує ритейл: персоналізація, аналітика, логістика.

Рекомендаційні системи — важливий інструмент підвищення лояльності.

Простий підхід ефективний, нейромережа — гнучка та масштабована.

Перспектива — гібридні моделі та врахування контексту.

ПЕРЕЛІК ПОСИЛАНЬ

1. Amazon Science. How Amazon uses AI and ML [Електронний ресурс]. — Режим доступу: <https://www.amazon.science> (дата звернення: 10.05.2025).
2. Walmart Inc. AI in Retail Operations [Електронний ресурс]. — Режим доступу: <https://corporate.walmart.com> (дата звернення: 14.05.2025).
3. Рассел П. Штучний інтелект: сучасний підхід / П. Рассел, П. Норвіг. — К. : Діалектика, 2022. — 1040 с.
4. Aggarwal C. Recommender Systems: The Textbook / C. Aggarwal — Springer, 2016. — 408 p.
5. Goodfellow I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. — MIT Press, 2016. — 775 p.
6. Гребенева О. В. Прогнозування попиту в ритейлі за допомогою ML / О. В. Гребенева, І. І. Коваленко // Бізнес-Інформ. — 2022. — № 3. — С. 42–46.
7. Accenture. The art of AI maturity: Advancing from practice to performance [Електронний ресурс]. — Режим доступу: <https://www.accenture.com/us-en/insights/artificial-intelligence/ai-maturity-and-transformation> (дата звернення: 1 червня 2025 р.).
8. Netflix Tech Blog [Електронний ресурс]. — Режим доступу: <https://netflixtechblog.com> (дата звернення: 13.05.2025).
9. EVA.ua — офіційний сайт [Електронний ресурс]. — Режим доступу: <https://eva.ua> (дата звернення: 07.05.2025).
10. Gołabek, M. Demand Forecasting using Long Short-Term Memory Neural Networks [Електронний ресурс] / M. Gołabek, R. Senge, R. Neumann. — 2020. — arXiv:2008.08522. — Режим доступу: <https://arxiv.org/abs/2008.08522> (дата

звернення: 20.05.2025).

11. OpenAI. GPT-4 Technical Report [Електронний ресурс] // ArXiv:2303.08774. — 2023. — Режим доступу: <https://arxiv.org/abs/2303.08774> (дата звернення: 15.05.2025).

12.

13. Scikit-learn documentation [Електронний ресурс]. — Режим доступу: <https://scikit-learn.org> (дата звернення: 09.05.2025).

14. Dialogflow Documentation [Електронний ресурс]. — Режим доступу: <https://cloud.google.com/dialogflow/docs> (дата звернення: 11.05.2025).

15. McKinsey & Company. The AI opportunity in retail [Електронний ресурс]. — 2021. — Режим доступу: <https://www.mckinsey.com> (дата звернення: 17.05.2025).

16. TensorFlow Blog. Computer Vision in Smart Stores [Електронний ресурс]. — 2023. — Режим доступу: <https://blog.tensorflow.org> (дата звернення: 12.05.2025).

17. Rozetka.ua — офіційний сайт [Електронний ресурс]. — Режим доступу: <https://rozetka.com.ua> (дата звернення: 06.05.2025).

18. H&M Tech Blog [Електронний ресурс]. — Режим доступу: <https://hmgroup.com> (дата звернення: 08.05.2025).

19. Haque, R. A protein panel in cerebrospinal fluid for diagnostic and predictive assessment of Alzheimer's disease [Електронний ресурс] / R. Haque [et al.] // Science Translational Medicine. — 2023. — V. 15, No 712.— eadg4122. — Режим доступу: [doi:10.1126/scitranslmed.adg4122](https://doi.org/10.1126/scitranslmed.adg4122) (дата звернення: 22.05.2025).

ДОДАТОК А

Створення матриці

```
import pandas as pd

import numpy as np

np.random.seed(42)

num_users, num_items = 50, 70

data = np.random.choice([0, 1, 2, 3, 4, 5], size=(num_users, num_items), p=[0.4, 0.1, 0.15, 0.15, 0.1,
0.1])

user_ids = [f'User_{i+1}' for i in range(num_users)]

item_ids = [f'Item_{j+1}' for j in range(num_items)]

df = pd.DataFrame(data, index=user_ids, columns=item_ids)
```

ДОДАТОК Б

Обчислення косинусної подібності

```
from sklearn.metrics.pairwise import cosine_similarity  
  
similarity_matrix = cosine_similarity(df.values)  
  
similarity_df = pd.DataFrame(similarity_matrix, index=df.index, columns=df.index)
```

ДОДАТОК В

Визначення подібних користувачів

```
user_id = 'User_10'
```

```
similar_users = similarity_df[user_id].sort_values(ascending=False).drop(user_id)
```

```
print("ТОП-5 подібних користувачів:")
```

```
print(similar_users.head(5))
```

ДОДАТОК Г

Формування рекомендацій

```
def get_recommendations(user_id, top_n=10):  
  
    similar_users = similarity_df[user_id].sort_values(ascending=False).drop(user_id)  
  
    user_ratings = df.loc[user_id]  
  
    recommended_items = {}  
  
    for other_user in similar_users.index:  
  
        weight = similarity_df.loc[user_id, other_user]  
  
        for item in df.columns:  
  
            if user_ratings[item] == 0 and df.loc[other_user, item] > 0:  
  
                recommended_items[item] = recommended_items.get(item, 0) +  
df.loc[other_user, item] * weight  
  
        sorted_items = sorted(recommended_items.items(), key=lambda x: x[1],  
reverse=True)  
  
        return [item[0] for item in sorted_items[:top_n]]  
  
recommendations_ub = get_recommendations('User_10')  
  
print("ТОП-10 рекомендацій:")  
  
print(recommendations_ub)
```

ДОДАТОК Д

Підготовка даних для моделі

```
df_long = df.reset_index().melt(id_vars='index', var_name='item',  
value_name='rating')
```

```
df_long = df_long[df_long['rating'] > 0]
```

```
df_long.rename(columns={'index': 'user'}, inplace=True)
```

```
from sklearn.preprocessing import LabelEncoder
```

```
user_enc = LabelEncoder()
```

```
item_enc = LabelEncoder()
```

```
df_long['user_id'] = user_enc.fit_transform(df_long['user'])
```

```
df_long['item_id'] = item_enc.fit_transform(df_long['item'])
```

ДОДАТОК Е

Побудова та навчання моделі

```
from tensorflow.keras.models import Model

from tensorflow.keras.layers import Input, Embedding, Dot, Flatten, Dense

from tensorflow.keras.optimizers import Adam

n_users = df_long['user_id'].nunique()

n_items = df_long['item_id'].nunique()

user_input = Input(shape=(1,))

item_input = Input(shape=(1,))

user_embedding = Embedding(n_users, 8)(user_input)

item_embedding = Embedding(n_items, 8)(item_input)

dot_product = Dot(axes=2)([user_embedding, item_embedding])

dot_product = Flatten()(dot_product)

output = Dense(1)(dot_product)

model = Model(inputs=[user_input, item_input], outputs=output)

model.compile(optimizer=Adam(), loss='mse')

model.fit([df_long['user_id'], df_long['item_id']], df_long['rating'], epochs=100,
verbose=0)
```

ДОДАТОК Ж

Створення ТОП подібних користувачів у латентному просторі

```
from sklearn.metrics.pairwise import cosine_similarity  
  
user_embeddings = model.get_layer(index=2).get_weights()[0]  
  
user_idx = user_enc.transform(['User_10'])[0]  
  
user_vec = user_embeddings[user_idx].reshape(1, -1)  
  
similarities = cosine_similarity(user_vec, user_embeddings).flatten()  
  
top_indices = np.argsort(-similarities)[1:6]  
  
similar_users_nn = user_enc.inverse_transform(top_indices)  
  
print("ТОП-5 подібних користувачів у латентному просторі:")  
  
print(similar_users_nn)
```

ДОДАТОК И

Перевірка користувача та формування рекомендацій

```
def recommend_nn(user_label, top_n=10):  
  
    if user_label not in user_enc.classes_:  
  
        raise ValueError("Користувач не знайдений")  
  
    user_idx = user_enc.transform([user_label])[0]  
  
    item_indices = np.arange(n_items)  
  
    predictions = model.predict([np.full(n_items, user_idx), item_indices],  
verbose=0).flatten()  
  
    rated_items = df_long[df_long['user_id'] == user_idx]['item_id'].values  
  
    predictions[rated_items] = -np.inf  
  
    top_items = np.argsort(-predictions)[:top_n]  
  
    return item_enc.inverse_transform(top_items)  
  
recommendations_nn = recommend_nn('User_10')  
  
print("ТОП-10 NN-рекомендацій:")  
  
print(recommendations_nn)
```

ДОДАТОК К

SWOT-аналіз нейромережевої моделі та колаборативної фільтрації

	Нейромережева модель на основі ембедінгів	User-based колаборативна фільтрація
Сильні сторони	<ul style="list-style-type: none"> – Висока здатність до генералізації – Виявляє приховані залежності між користувачами й товарами – Масштабована для великих обсягів даних 	<ul style="list-style-type: none"> – Простота для реалізації – Інтерпретовані результати – Не потребує тривалого навчання моделі
Слабкі сторони	<ul style="list-style-type: none"> – Високі обчислювальні витрати – Складна архітектура – Потребує великого обсягу даних для якісного навчання 	<ul style="list-style-type: none"> – Проблема «холодного старту» – Низька точність при розріджених матрицях – Залежність від якості сусідів
Можливості	<ul style="list-style-type: none"> – Інтеграція з мультимодальними ознаками (текст, зображення) – Підвищення персоналізації рекомендацій 	<ul style="list-style-type: none"> – Швидке впровадження – Може слугувати базовою моделлю в гібридних системах
Загрози	<ul style="list-style-type: none"> – Ризик перенавчання – Залежність від правильного налаштування гіперпараметрів – Складність діагностики 	<ul style="list-style-type: none"> – Погано масштабуються на великі обсяги користувачів – Чутливість до шуму в поведінці користувачів