Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: Розробка рекомендаційної системи для онлайн-магазину.

Виконав(ла): студент(ка) 4 курсу, групи СН-42
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

		Остапчук К.І.
	(підпис)	(прізвище та ініціали)
Керівник		Бонарчук І.О.
	(підпис)	(прізвище та ініціали)
Нормоконтроль		Липак Г.І.
	(підпис)	(прізвище та ініціали)
Завідувач кафедри		Боднарчук І.О.
	(підпис)	(прізвище та ініціали)
Рецензент		Коноваленко І.В.
	(підпис)	(прізвище та ініціали)

Тернопіль
2026

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
"8" червня 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

студенту Остапчук Костянтин Іванович
(прізвище, ім'я, по батькові)

1. Тема роботи **Розробка рекомендаційної системи для онлайн-магазину.**

Керівник роботи **к.т.н., доц. Боднарчук Ігор Орестович**
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від "*14*" *травня* 2026 року № *4/9-239*

2. Термін подання студентом завершеної роботи 19 червня 2026 р.

3. Вихідні дані до роботи Літературні джерела з тематики роботи

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ; 1 Принципи побудови рекомендаційних систем; 1.1 загальна характеристика рекомендаційних систем; 1.2 класифікація рекомендаційних систем; 1.3 рекомендації на основі популярності (popularity-based); 1.4 колаборативна фільтрація (collaborative filtering); 1.5 контентна фільтрація (content-based filtering); 1.6. матрична факторизація та сингулярне розкладання (svd); 1.7 кластеризація та векторизація тексту tf-idf; 1.8 метрики оцінки рекомендаційних систем; 2 Використана рекомендаційна система; 2.1 загальна архітектура системи; 2.2 сценарії застосування та логіка перемикання; 2.3 підсистема рекомендацій на основі популярності; 2.4 підсистема колаборативної фільтрації з svd; 2.5 інтеграція трьох підсистем; 3 Опис роботи програмного коду; 3.1 середовище розробки та використані бібліотеки; 3.2 завантаження та підготовка даних amazon; 3.3 реалізація popularity-based рекомендацій; 3.4 реалізація колаборативної фільтрації з svd; 3.5 реалізація кластеризації текстових описів; 3.6 результати та аналіз роботи системи; 4 Безпека життєдіяльності, основи охорони праці; Висновки; Список використаних джерел

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Титульний слайд; 2. Актуальність дослідження; 3. Мета роботи; 4. Класифікація рекомендаційних систем; 5. Колаборативна фільтрація; 6. Архітектура контентної фільтрації; 7. Сингулярне розкладання матриці; 8. TF-IDF, K-means; 9. Проблема холодного старту; 10. Архітектура гібридної системи; 11. Логіка перемикання; 12. Підготовка даних; 13. Реалізація колаборативної фільтрації; 14. Реалізація кластеризації; 15. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Гурик О.Я., к.т.н., доцент кафедри МТ		

7. Дата видачі завдання 26 січня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	26.01.2026 – 27.01.2026	*Виконано*
2.	Підбір джерел по темі роботи	28.01.2026 – 01.04.2026	*Виконано*
3.	Оформлення першого розділу	15.04.2026	*Виконано*
4.	Оформлення другого розділу	20.04.2026	
5.	Оформлення третього розділу	30.04.2026	*Виконано*
6.	Виконання завдання до підрозділу "Безпека		
7.	життєдіяльності, основи охорони праці"	15.05.2026	*Виконано*
8.	Оформлення кваліфікаційної роботи	07.06.2026	*Виконано*
9.	Перевірка на плагіат	07.06.2026	*Виконано*
10.	Нормоконтроль	09.06.2026	*Виконано*
11.	Попередній захист кваліфікаційної роботи	11.06.2026	*Виконано*
12.	Захист кваліфікаційної роботи	27.06.2026	
13.			
14.			
15.			

Студент _____ Остапчук К.І.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Боднарчук І.О.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

"Розробка рекомендаційної системи для онлайн-магазину" // Кваліфікаційна робота освітнього рівня "Бакалавр" // Остапчук Костянтин Іванович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-42 // Тернопіль, 2026 // с. – 61, рис. – 8, джерел – 31, додатків – 0.

Ключові слова: рекомендаційна система, електронна комерція, гібридна архітектура, машинне навчання, колаборативна фільтрація, сингулярне розкладання матриці (SVD), холодний старт, векторизація тексту, TF-IDF, кластеризація, K-means, python

Кваліфікаційну роботу присвячено дослідженню принципів побудови та практичній реалізації гібридної рекомендаційної системи для платформ електронної комерції, яка спрямована на подолання проблеми інформаційного перевантаження користувачів та вирішення проблеми «холодного старту».

У роботі здійснено комплексний теоретичний аналіз існуючих підходів до фільтрації інформації: популярнісних методів, колаборативної та контентної фільтрації, а також сучасних моделей на основі глибокого навчання. Обґрунтовано доцільність застосування гібридної архітектури для забезпечення безперервного циклу рекомендацій залежно від ступеня розрідженості та доступності даних.

Розроблено та програмно реалізовано мовою Python гібридну систему, яка об'єднує три автономні підсистеми:

1. Рекомендації на основі популярності (Popularity-based) – для нових користувачів без історії взаємодій.

2. Модельну колаборативну фільтрацію на основі сингулярного розкладання матриці (Truncated SVD) — для обчислення персоналізованих item-to-item рекомендацій постійним клієнтам у латентному просторі ознак.

3. Текстову кластеризацію описів товарів (TF-IDF + K-means) — для семантичного групування асортименту в умовах повного «холодного старту» нової системи або додавання нових позицій без транзакційної історії.

Тестування та аналіз ефективності алгоритмів проведено на реальних промислових наборах даних (Amazon Reviews Dataset та Home Depot Product Search Relevance). Емпірично доведено робастність запропонованого архітектурного підходу, його високу швидкість обробки розріджених даних та практичну цінність для інтеграції в реальні e-commerce платформи з метою підвищення конверсії та лояльності користувачів.

ANNOTATION

"Development of a Recommendation System for an Online Store" // Qualification work of the educational level "Bachelor" // Ostapchuk Kostiantyn // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, Group CH-42 // Ternopil, 2026 // p. – 61, fig. – 8, references – 31, annexes – 0.

Keywords: recommender system, e-commerce, hybrid architecture, machine learning, collaborative filtering, singular value decomposition (SVD), cold start, text vectorization, TF-IDF, clustering, K-means, python

The qualification thesis is devoted to the study of the design principles and practical implementation of a hybrid recommender system for e-commerce platforms, aimed at overcoming the problem of user information overload and addressing the "cold start" problem.

The paper provides a comprehensive theoretical analysis of existing information filtering approaches: popularity-based methods, collaborative and content-based filtering, as well as modern deep learning-based models. The expediency of using a hybrid architecture to ensure a continuous recommendation cycle depending on the degree of data sparsity and availability is substantiated.

A hybrid system has been developed and programmatically implemented in Python, combining three autonomous subsystems:

1. Popularity-based recommendations – for new users with no interaction history.

2. Model-based collaborative filtering based on singular value decomposition (Truncated SVD) – to calculate personalized item-to-item recommendations for regular customers in a latent feature space.

3. Text clustering of product descriptions (TF-IDF + K-means) – for semantic grouping of the assortment under conditions of a complete "cold start" of a new system or when adding new items without transactional history.

Testing and performance analysis of the algorithms were conducted on real-world industrial datasets (Amazon Reviews Dataset and Home Depot Product Search Relevance). The robustness of the proposed architectural approach, its high processing speed for sparse data, and its practical value for integration into real e-commerce platforms to increase conversion rates and user loyalty have been empirically proven.

ЗМІСТ

ВСТУП

Сучасний ринок електронної комерції характеризується експоненційним зростанням обсягів даних та широким асортиментом товарів. Великі онлайн-платформи, такі як Amazon, eBay, Walmart, AliExpress, пропонують мільйони продуктів, що створює для покупця проблему інформаційного перевантаження. У таких умовах рекомендаційні системи стають критично важливим інструментом, який допомагає користувачам знайти релевантні товари, а бізнесу – підвищити конверсію, середній чек та лояльність клієнтів.

За оцінками експертів, рекомендаційні системи Amazon генерують близько 35% доходу компанії, а персоналізовані рекомендації Netflix утримують близько 80% переглядів. Це свідчить про надзвичайно високу комерційну цінність ефективних алгоритмів рекомендацій. Розвиток методів машинного навчання, доступність обчислювальних ресурсів та накопичення великих обсягів даних про поведінку користувачів зробили рекомендаційні системи однією з найбільш активно досліджуваних галузей сучасного штучного інтелекту.

Розробка ефективної рекомендаційної системи є складною комплексною задачею, що вимагає поєднання різних підходів. Жоден окремий алгоритм не може ефективно працювати у всіх можливих сценаріях: для нових користувачів немає історії покупок, нові товари не мають оцінок, нові магазини взагалі не мають жодних транзакційних даних. Тому сучасні промислові рекомендаційні системи зазвичай є гібридними – вони поєднують декілька алгоритмів та автоматично обирають найкращий залежно від наявних даних.

Однією з ключових проблем у галузі є проблема холодного старту (cold-start problem) – ситуація, коли система не має достатньо даних для побудови якісних рекомендацій. Існує декілька варіантів цієї проблеми: холодний старт

для нового користувача, для нового товару та для нового бізнесу. Кожен сценарій вимагає окремого підходу до вирішення.

Об’єктом дослідження є процес персоналізованих рекомендацій товарів у системах електронної комерції з використанням методів машинного навчання.

Предметом дослідження є гібридна рекомендаційна система, що поєднує три алгоритми: рекомендації на основі популярності (popularity-based), модельну колаборативну фільтрацію зі сингулярним розкладанням матриці (SVD-based collaborative filtering) та кластеризацію текстових описів товарів за допомогою TF-IDF та K-means для вирішення проблеми холодного старту.

Метою кваліфікаційної роботи є дослідження принципів побудови рекомендаційних систем для електронної комерції та аналіз практичної реалізації гібридної системи, що поєднує три різні алгоритми для різних сценаріїв застосування, з метою забезпечення ефективних рекомендацій незалежно від наявності історичних даних.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

– проаналізувати теоретичні основи рекомендаційних систем, їхню класифікацію та основні алгоритми;

– дослідити принципи побудови популярнісних, колаборативних та контентних рекомендаційних систем;

– розглянути математичний апарат сингулярного розкладання матриці (SVD) та його застосування для рекомендацій;

– вивчити методи векторизації тексту (TF-IDF) та алгоритм K-means для кластеризації товарів;

– проаналізувати архітектуру гібридної рекомендаційної системи, що поєднує три алгоритми;

– описати практичну реалізацію системи у мові Python з використанням бібліотек pandas, scikit-learn та SciPy;

– проаналізувати результати роботи кожного з алгоритмів та інтерпретувати отримані рекомендації.

Методи дослідження. У роботі застосовано методи статистичного аналізу даних, методи лінійної алгебри (сингулярне розкладання матриці), методи машинного навчання (колаборативна фільтрація, кластеризація K-means), методи обробки природної мови (векторизація TF-IDF, токенізація, видалення стоп-слів), а також методи аналізу та проєктування програмного забезпечення.

Інформаційною базою дослідження є відкриті датасети електронної комерції: набір оцінок продуктів Amazon (Amazon Reviews Dataset), що містить оцінки користувачів для різноманітних товарів, та каталог продуктів Home Depot з детальними текстовими описами. Використання цих наборів даних дозволяє продемонструвати роботу всіх трьох алгоритмів на реальних промислових даних.

Наукова новизна роботи полягає у систематичному дослідженні гібридного підходу до побудови рекомендаційних систем, де вибір алгоритму здійснюється автоматично залежно від доступності даних. Запропонована система забезпечує безперервний цикл рекомендацій: від першого відвідування користувача без жодної історії до персоналізованих рекомендацій на основі складних математичних моделей.

Практичне значення отриманих результатів полягає у демонстрації архітектурного підходу, який може бути використаний при розробці реальних промислових рекомендаційних систем для e-commerce платформ. Принципи, описані у роботі, є застосовними до різних сфер: від онлайн-магазинів та маркетплейсів до стримінгових сервісів та платформ контенту.

1 ПРИНЦИПИ ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1 Загальна характеристика рекомендаційних систем

Рекомендаційна система (Recommender System, RS) – це програмний інструмент або алгоритм, який прогнозує оцінку, що користувач надав би певному об’єкту, або виявляє об’єкти, які з найбільшою ймовірністю зацікавлять користувача. Рекомендаційні системи є підкласом систем фільтрації інформації, спрямованих на боротьбу з інформаційним перевантаженням шляхом надання користувачеві персоналізованого вибору з великої кількості варіантів.

Історично перші рекомендаційні системи з’явилися наприкінці 1990-х років разом з розвитком електронної комерції та інтернет-сервісів. Однією з найбільш відомих ранніх систем була Tapestry (1992 рік), розроблена в Xerox PARC. Справжній прорив відбувся у середині 2000-х років, коли компанія Netflix у 2006 році оголосила конкурс «Netflix Prize» з призовим фондом у мільйон доларів за алгоритм, що покращить точність рекомендацій фільмів на 10%. Цей конкурс став каталізатором розвитку галузі і призвів до появи багатьох сучасних методів.

Сучасні рекомендаційні системи є невід’ємною частиною майже кожного великого онлайн-сервісу. Amazon рекомендує товари, Netflix – фільми та серіали, Spotify – музику, YouTube – відео, Facebook та Instagram – пости та рекламу. За оцінками експертів, близько 35% продажів Amazon генеруються через рекомендаційну систему, а 80% переглядів Netflix є результатом персоналізованих рекомендацій.

Формально задачу рекомендаційної системи можна описати наступним чином. Маючи множину користувачів $U = \{u_1, u_2, ..., u_m\}$ та множину товарів $I = \{i_1, i_2, ..., i_n\}$, а також функцію корисності $f: U \times I \rightarrow R$, яка зазвичай відома лише для частини пар (користувач, товар), необхідно для кожного користувача

u визначити товар i_0, який максимізує функцію корисності: $i_0 = \arg\max f(u, i)$ для $i \in I$.

Функція корисності може бути виражена різними способами: оцінкою товару користувачем (наприклад, від 1 до 5 зірок), бінарною ознакою (купив/не купив, переглянув/не переглянув), числовою метрикою взаємодії (кількість кліків, час перегляду, частота відвідувань). Вибір функції корисності суттєво впливає на архітектуру та якість роботи рекомендаційної системи.

Ключова складність задачі полягає у тому, що матриця оцінок R розміром $m \times n$ (де m – кількість користувачів, n – кількість товарів) є надзвичайно розрідженою: типовий користувач взаємодіє лише з невеликою часткою всіх доступних товарів. Наприклад, у датасеті Netflix оцінки складають лише близько 1,2% від усіх можливих пар «користувач-фільм». Задача рекомендаційної системи – заповнити пропущені значення матриці на основі наявних даних.

Бізнес-цінність рекомендаційних систем полягає у наступних аспектах: збільшення продажів за рахунок підвищення конверсії та середнього чека; покращення задоволеності користувачів та їхньої лояльності; зменшення відтоку клієнтів (churn); ефективна навігація по великому каталогу товарів; персоналізація користувацького досвіду; підтримка нових та маловідомих товарів через довгий хвіст (long tail) каталогу.

1.2 Класифікація рекомендаційних систем

Існує декілька класифікацій рекомендаційних систем за різними критеріями. Найпоширенішою є класифікація за типом даних та підходом до побудови рекомендацій, згідно з якою виділяють чотири основні типи систем (див. рис. 1.1).

Рисунок 1.1 – Класифікація рекомендаційних систем

Перший тип – рекомендаційні системи на основі популярності (Popularity-based). Найпростіший підхід, що рекомендує найпопулярніші товари незалежно від конкретного користувача. Така система не є персоналізованою, але є ефективним рішенням для нових користувачів, про яких немає інформації. Популярність може визначатися кількістю продажів, оцінок, переглядів або іншою метрикою.

Другий тип – системи колаборативної фільтрації (Collaborative Filtering, CF). Ці системи базуються на припущенні, що користувачі зі схожими смаками у минулому матимуть схожі смаки в майбутньому. Колаборативна фільтрація поділяється на два підтипи: на основі сусідства (memory-based) та модельна (model-based). Memory-based методи використовують безпосередньо матрицю оцінок для пошуку схожих користувачів або товарів. Model-based методи будують математичну модель (зазвичай через матричну факторизацію).

Третій тип – системи на основі вмісту (Content-based Filtering). Ці системи рекомендують товари, схожі на ті, що користувач уже оцінив високо. Схожість визначається на основі атрибутів товарів: для фільмів це може бути жанр, режисер, актори; для книг – автор, тематика; для товарів електронної комерції – категорія, бренд, опис, ціна. Контентні системи не страждають від

проблеми холодного старту нових товарів, але потребують структурованої інформації про товари.

Четвертий тип – гібридні системи (Hybrid Systems). Поєднують декілька підходів для отримання кращих результатів та усунення недоліків окремих методів. Існує декілька стратегій гібридизації: зважене (weighted) комбінування результатів декількох систем; перемикання (switching) між системами залежно від ситуації; змішування (mixed) рекомендацій від різних систем; послідовна (cascade) обробка, де одна система фільтрує результати іншої; пере-зважена (feature combination/augmentation) інтеграція ознак з різних джерел.

Окремою категорією є контекстно-залежні рекомендаційні системи (Context-aware Recommender Systems), які враховують контекст користувача: час доби, день тижня, геолокацію, тип пристрою, погоду, настрій. Наприклад, рекомендації музики можуть змінюватися залежно від того, чи слухає користувач її у спортзалі, на роботі чи перед сном. Контекстні системи є особливо актуальними для мобільних застосунків.

Також у останні роки активно розвиваються рекомендаційні системи на основі глибокого навчання (Deep Learning-based RS). Вони використовують нейронні мережі для виявлення складних нелінійних залежностей між користувачами та товарами. Прикладами є нейронна колаборативна фільтрація (Neural Collaborative Filtering, NCF), автоенкодери та рекурентні мережі для послідовних рекомендацій. Глибокі моделі забезпечують високу якість, але вимагають великих обсягів даних та обчислювальних ресурсів.

1.3 Рекомендації на основі популярності (Popularity-based)

Рекомендаційні системи на основі популярності є найпростішим підходом до побудови рекомендацій. Основна ідея полягає у тому, щоб рекомендувати кожному користувачеві ті товари, які є найбільш популярними

серед усіх інших користувачів. Така система не персоналізується для конкретного користувача, але є ефективною у деяких сценаріях.

Метрики популярності можуть бути різними залежно від наявних даних та бізнес-задач. Найпоширеніші: загальна кількість продажів товару за певний період; кількість оцінок або відгуків (популярність як зацікавленість); середня оцінка серед усіх оцінок (популярність як якість); кількість унікальних користувачів, що взаємодіяли з товаром; зважена комбінація кількості та оцінки (наприклад, формула IMDb Top 250).

Алгоритм роботи popularity-based системи зазвичай складається з трьох кроків: агрегація даних – для кожного товару обчислюється метрика популярності, наприклад, кількість оцінок або середня оцінка; сортування товарів за спаданням метрики популярності; повернення верхніх N товарів як рекомендацію.

Ключовою перевагою popularity-based систем є простота реалізації, висока швидкість роботи та відсутність потреби у складних математичних моделях. Така система може бути реалізована як простий SQL-запит або декілька рядків коду на pandas. Крім того, вона не страждає від проблеми холодного старту користувача – рекомендації можна надати навіть тим, про кого ніколи нічого не було відомо.

Основним недоліком є відсутність персоналізації. Усі користувачі отримують однакові рекомендації, що не враховує індивідуальні смаки. Це призводить до ефекту «популярного багатшає» (rich-get-richer): популярні товари отримують ще більше уваги, а маловідомі залишаються непоміченими. Також система не може рекомендувати товари з «довгого хвоста» (long tail), які можуть бути більш релевантними для конкретного користувача.

Незважаючи на обмеження, popularity-based підхід широко використовується у промислових системах як перший етап рекомендацій для нових користувачів та як базова лінія (baseline) для порівняння з більш складними алгоритмами. На багатьох сайтах розділ «Бестселери» або «Популярне зараз» є реалізацією саме цього підходу.

1.4 Колаборативна фільтрація (Collaborative Filtering)

Колаборативна фільтрація (Collaborative Filtering, CF) є одним із найбільш популярних та ефективних підходів у рекомендаційних системах. Назва походить від того, що цей метод використовує колективну поведінку багатьох користувачів (collaboration) для генерації рекомендацій конкретній особі. Основне припущення: якщо користувачі A та B мали схожі смаки в минулому, то A, ймовірно, сподобається те, що подобається B.

Колаборативна фільтрація поділяється на два основні підтипи: на основі сусідства (memory-based або neighborhood-based) та модельна (model-based). Підхід на основі сусідства, у свою чергу, поділяється на user-based (на основі схожості користувачів) та item-based (на основі схожості товарів).

User-based колаборативна фільтрація працює наступним чином. Для активного користувача u визначається множина схожих користувачів («сусідів») на основі схожості їхніх профілів оцінок. Для кожного товару i, який не оцінив u, прогнозована оцінка обчислюється як зважене середнє оцінок цього товару серед «сусідів». Товари з найвищими прогнозованими оцінками рекомендуються користувачеві.

Item-based колаборативна фільтрація, запропонована Amazon у 2003 році, працює навпаки. Для кожної пари товарів обчислюється схожість на основі того, які користувачі їх оцінювали і як. Для рекомендацій користувачеві u: для кожного товару i, що користувач не оцінив, обчислюється прогнозована оцінка як зважене середнє оцінок користувача u для товарів, найбільш схожих на i. Item-based підхід зазвичай є більш ефективним та стабільним, оскільки кількість товарів змінюється повільніше, ніж кількість користувачів.

Для обчислення схожості між користувачами або товарами використовуються різні метрики. Косинусна схожість (cosine similarity) є найпопулярнішою: вона вимірює кут між двома векторами і не залежить від їхньої довжини. Коефіцієнт кореляції Пірсона (Pearson correlation) враховує, що різні користувачі можуть мати різні «масштаби» оцінок (одні схильні

ставити завищені оцінки, інші – занижені). Скоригована косинусна схожість (adjusted cosine similarity) поєднує переваги обох підходів.

Модельна колаборативна фільтрація (Model-based CF) використовує методи машинного навчання для побудови математичної моделі, яка описує приховані залежності між користувачами та товарами. Найпопулярнішим підходом є матрична факторизація, де розріджена матриця оцінок розкладається на дві щільні матриці меншої розмірності, що відповідають латентним факторам. Схематично класифікація рекомендаційних систем показана на рисунку 1.2.

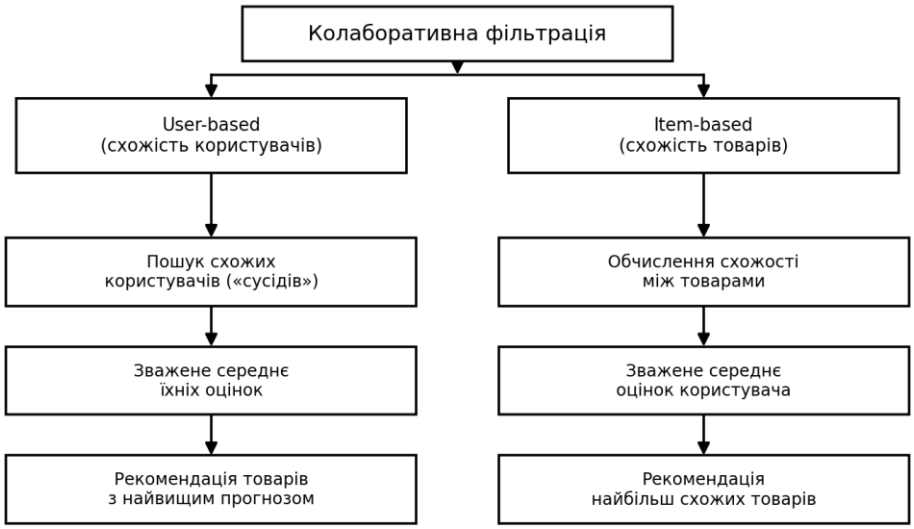

Рисунок 1.2 – Типи колаборативної фільтрації на основі сусідства

Перевагами колаборативної фільтрації є: відсутність потреби в інформації про вміст товарів (працює лише з оцінками); здатність виявляти несподівані рекомендації (serendipity); масштабованість на велику кількість користувачів та товарів. Недоліками є: проблема холодного старту (для нових користувачів та нових товарів немає даних); розрідженість матриці оцінок; обчислювальна складність при великій кількості користувачів.

1.5 Контентна фільтрація (Content-based Filtering)

Контентна фільтрація (Content-based Filtering, CBF) є альтернативним підходом до побудови рекомендацій, який ґрунтується на аналізі властивостей товарів. Замість використання інформації про інших користувачів, контентні системи аналізують, які атрибути товарів сподобались конкретному користувачеві, і рекомендують інші товари з подібними атрибутами.

Основна ідея: якщо користувач позитивно оцінив товари A, B, C, то система виявляє спільні характеристики цих товарів і знаходить інші товари з подібними характеристиками. Для фільмів це можуть бути жанр, режисер, актори, рік випуску; для книг – автор, жанр, тематика; для товарів електронної комерції – категорія, бренд, ціновий діапазон, ключові слова з опису.

Архітектура контентної рекомендаційної системи зазвичай включає три компоненти: модуль аналізу вмісту (Content Analyzer), що витягує ознаки з товарів і створює структуроване представлення; модуль профілю користувача (Profile Learner), що будує профіль смаків користувача на основі його оцінок; модуль фільтрації (Filtering Component), що порівнює профіль користувача з представленнями товарів і обирає найбільш схожі (див. рис. 1.3).

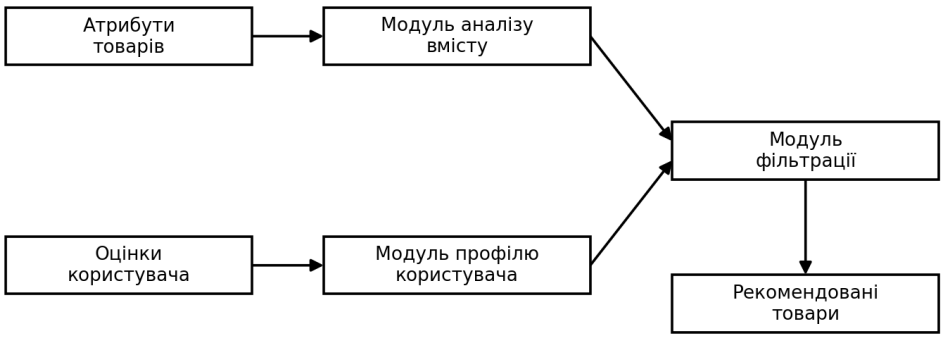

Рисунок 1.3 – Архітектура контентної рекомендаційної системи

Для текстових атрибутів (опис товару, рецензії) використовуються методи обробки природної мови. Найпоширенішою технікою є векторизація TF-IDF (Term Frequency-Inverse Document Frequency), що перетворює текст у

числові вектори. Після векторизації схожість між товарами обчислюється за допомогою косинусної схожості або інших метрик векторної відстані.

Перевагами контентної фільтрації є: відсутність проблеми холодного старту нових товарів (рекомендації можна давати одразу, як тільки товар з'явився у каталозі); незалежність від інших користувачів (працює навіть з одним користувачем); прозорість та інтерпретованість (можна пояснити, чому рекомендується саме цей товар – наприклад, «оскільки ви оцінили інші бойовики»).

Недоліками є: обмежена різноманітність рекомендацій (overspecialization) – система рекомендує лише схожі товари і не відкриває користувачеві нові категорії; проблема холодного старту нових користувачів – без жодної оцінки неможливо побудувати профіль; обмеженість якості рекомендацій якістю атрибутів товарів; потреба у структурованому каталозі товарів.

1.6. Матрична факторизація та сингулярне розкладання (SVD)

Матрична факторизація (Matrix Factorization, MF) є одним із найбільш ефективних підходів до колаборативної фільтрації. Метод став популярним після його успіху у конкурсі Netflix Prize, де варіанти матричної факторизації показали найкращі результати серед усіх досліджених алгоритмів.

Основна ідея матричної факторизації полягає у розкладанні розрідженої матриці оцінок R розміром $m \times n$ на дві матриці меншої розмірності: P розміром $m \times k$ (матриця користувачів) та Q розміром $n \times k$ (матриця товарів), де k – кількість латентних факторів (зазвичай k значно менше за $\min(m, n)$). Прогнозована оцінка користувача u для товару i обчислюється як скалярний добуток відповідних векторів: $\hat{r}_{ui} = p_u^T \cdot q_i$.

Латентні фактори можна інтерпретувати як приховані характеристики, що визначають смаки користувачів та властивості товарів. Наприклад, для фільмів латентні фактори можуть відповідати таким абстрактним

характеристикам, як «серйозність-комедійність», «художність-комерційність», «екшен-драматичність». При цьому модель самостійно виявляє ці фактори з даних, без явного їхнього задання.

Сингулярне розкладання матриці (Singular Value Decomposition, SVD) є фундаментальним методом лінійної алгебри, який дозволяє розкласти будь-яку дійсну матрицю M розміром $m \times n$ на три матриці: $M = U\Sigma V^T$, де U – $m \times m$ ортогональна матриця лівих сингулярних векторів; Σ – $m \times n$ діагональна матриця сингулярних чисел (всі невід'ємні); V^T – $n \times n$ ортогональна матриця правих сингулярних векторів.

Сингулярні числа в Σ зазвичай розташовуються у порядку спадання: $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq 0$, де r – ранг матриці. Найбільші сингулярні числа відповідають найважливішим напрямкам у даних. Залишивши лише k найбільших сингулярних чисел та відповідні їм вектори, отримуємо найкраще наближення матриці M рангу k у термінах норми Фробеніуса. Це і є основою для зменшення розмірності та виявлення латентних факторів.

Застосування SVD до рекомендаційних систем має певні особливості. Класичний SVD працює лише з повністю заповненими матрицями, тоді як матриця оцінок є розрідженою. Для роботи з розрідженими матрицями використовуються спеціальні модифікації: SVD з заповненням пропусків середніми оцінками, Funk SVD (запропонований Саймоном Фанком під час Netflix Prize), SVD++ Юехуди Корена (враховує неявний зворотний зв'язок), а також алгоритми мінімізації методом градієнтного спуску, що працюють лише з відомими оцінками.

У практичних застосуваннях для обчислення SVD на розріджених матрицях у Python використовується функція svds з модуля scipy.sparse.linalg, що базується на алгоритмі Lanczos. Для повноцінних промислових рекомендаційних систем використовується спеціалізована бібліотека Surprise, яка реалізує оптимізовані версії SVD, SVD++ та інших алгоритмів.

1.7 Кластеризація та векторизація тексту TF-IDF

Для побудови рекомендаційних систем у сценаріях, коли немає ні історії покупок, ні явних оцінок (повний холодний старт), використовуються методи аналізу текстового вмісту товарів. Опис товару містить багато інформації про його властивості, призначення та цільову аудиторію. Якщо вдається групувати товари у схожі категорії на основі описів, можна рекомендувати товари з тієї самої групи.

Векторизація тексту TF-IDF (Term Frequency-Inverse Document Frequency) є класичним методом перетворення текстових документів у числові вектори. Метод був запропонований Гердом Ремцем та К.А. Спарком Джонсом у 1972 році і досі залишається одним із найбільш ефективних базових методів обробки тексту.

TF (Term Frequency) – частота терміну в документі: TF(t, d) = (кількість входжень терміну t у документ d) / (загальна кількість термінів у d). IDF (Inverse Document Frequency) – обернена частота документів: IDF(t, D) = log(N / |{d ∈ D : t ∈ d}|), де N – загальна кількість документів. Спільне значення TF-IDF(t, d, D) = TF(t, d) × IDF(t, D).

Інтуїція TF-IDF: чим частіше термін зустрічається у конкретному документі, тим він важливіший для цього документа (TF); але якщо термін зустрічається у багатьох документах (наприклад, «і», «у», «або»), то він не несе специфічної інформації (IDF). Таким чином, TF-IDF підкреслює терміни, які є частими у конкретному документі, але рідкісними загалом. У результаті документи представлені як вектори у багатовимірному просторі ознак.

Алгоритм K-means – один із найпоширеніших методів кластеризації. Він поділяє n спостережень на k груп таким чином, щоб мінімізувати внутрішньокластерну дисперсію (відстань кожного спостереження до центроїда його кластера). Алгоритм є ітеративним: ініціалізує k центроїдів випадковим чином; для кожного спостереження визначає найближчий

центроїд; перераховує центроїди як середнє спостережень у кожному кластері; повторює два попередні кроки до збіжності.

Поєднання TF-IDF та K-means для рекомендацій працює наступним чином. По-перше, тексти описів товарів перетворюються у TF-IDF вектори. По-друге, K-means кластеризує товари у k кластерів за схожістю описів. По-третє, аналіз ключових термінів кожного кластера дозволяє інтерпретувати тематику товарів. По-четверте, при пошуковому запиті користувача система знаходить найближчий кластер до запиту та рекомендує товари з цього кластера.

Вибір кількості кластерів k є важливою задачею. Існують різні методи: метод ліктя (elbow method) аналізує залежність внутрішньокластерної дисперсії від k; силуетний коефіцієнт (silhouette score) оцінює якість кластеризації; статистика розриву (gap statistic) порівнює з випадковим розподілом. У практиці часто обирають k емпірично, виходячи з кількості очікуваних тематичних груп товарів.

1.8 Метрики оцінки рекомендаційних систем

Оцінка якості рекомендаційних систем є складною задачею, оскільки ціль системи (задоволення користувача) важко виміряти безпосередньо. Існує декілька підходів до оцінки: офлайн-метрики на основі історичних даних, онлайн-експерименти (A/B-тестування), користувацькі дослідження.

Офлайн-метрики поділяються на дві категорії: метрики прогнозування оцінок (rating prediction metrics) та метрики ранжування (ranking metrics). Метрики прогнозування оцінок використовуються, коли система прогнозує числову оцінку. Найпопулярніші: середньоквадратична помилка MSE = $(1/n)\Sigma(r_i - \hat{r}_i)^2$; корінь з MSE – RMSE = $\sqrt{MSE}$; середня абсолютна помилка MAE = $(1/n)\Sigma|r_i - \hat{r}_i|$. RMSE використовувалась у Netflix Prize як основна метрика.

Метрики ранжування використовуються, коли система видає впорядкований список top-N рекомендацій. Precision@K = (кількість

релевантних товарів у top-K) / K. Recall@K = (кількість релевантних товарів у top-K) / (загальна кількість релевантних товарів). F1@K = 2 · Precision · Recall / (Precision + Recall). MAP (Mean Average Precision) усереднює precision по різних значеннях відсічки.

Метрика NDCG (Normalized Discounted Cumulative Gain) є особливо корисною для ранжованих рекомендацій. Вона враховує не лише наявність релевантних товарів у списку, а й їхню позицію: товари на верхніх позиціях мають більшу вагу. NDCG нормалізована до діапазону [0, 1] і дозволяє порівнювати результати на різних запитах.

Окрім стандартних метрик точності, існують додаткові показники якості рекомендацій. Покриття (coverage) – частка унікальних товарів, що рекомендуються користувачам. Різноманітність (diversity) – наскільки різноманітні товари у списку рекомендацій. Новизна (novelty) – наскільки несподіваними є рекомендації. Несподіваність (serendipity) – чи рекомендує система товари, які користувач не очікував би побачити, але вони йому подобаються.

У промислових системах основним способом оцінки є A/B-тестування. Користувачі випадково розподіляються на контрольну та експериментальну групи, кожна з яких отримує рекомендації від різних алгоритмів. Порівнюються бізнес-метрики: конверсія, середній чек, час на сайті, частка кліків (CTR – Click-Through Rate). A/B-тестування є золотим стандартом оцінки, оскільки безпосередньо вимірює бізнес-цінність системи.

1.9 Проблема холодного старту

Проблема холодного старту (cold-start problem) є однією з найскладніших проблем у рекомендаційних системах. Вона виникає у ситуаціях, коли система не має достатньо даних для побудови якісних рекомендацій. Виділяють три варіанти цієї проблеми: новий користувач, новий товар та нова система.

Холодний старт нового користувача (new user cold-start) виникає, коли користувач вперше з’являється у системі і не має жодної історії взаємодій. У такому випадку колаборативна фільтрація не може видати персоналізовані рекомендації, оскільки немає даних для порівняння з іншими користувачами. Стандартні рішення: рекомендації на основі популярності; запит уподобань під час реєстрації; використання демографічної інформації (вік, стать, місце проживання) для перенесення знань (transfer learning).

Холодний старт нового товару (new item cold-start) виникає, коли у каталог додається новий товар, для якого ще немає оцінок або взаємодій. Колаборативна фільтрація не може рекомендувати такий товар, бо немає історії. Рішення: контентна фільтрація на основі атрибутів товару; гібридні моделі, що поєднують колаборативну та контентну фільтрацію; активне навчання, коли система свідомо показує новий товар обраним користувачам для збору даних.

Холодний старт нової системи (new system cold-start) є найскладнішою ситуацією: новий магазин або сервіс ще не має жодних даних про користувачів або товари. Це частина проблеми, відома як bootstrap problem. Можливі рішення: використання text mining та аналізу описів товарів для побудови рекомендацій на основі їхнього вмісту; перенесення знань з інших доменів; використання експертних правил; інтеграція з зовнішніми джерелами даних.

Гібридний підхід, описаний у даній роботі, систематично вирішує усі три варіанти холодного старту (див. рис. 1.4).

Для нового користувача без історії використовується popularity-based система. Після накопичення мінімальної історії взаємодій активується колаборативна фільтрація з SVD. Для нового бізнесу або нового товару без оцінок працює text clustering на основі описів.

Рисунок 1.4 – Три варіанти проблеми холодного старту та їх вирішення

Така архітектура забезпечує безперервний цикл рекомендацій незалежно від ситуації.

2 ВИКОРИСТАНА РЕКОМЕНДАЦІЙНА СИСТЕМА

2.1 Загальна архітектура системи

Розглянута у роботі рекомендаційна система є гібридною та реалізує три різних алгоритми, кожен з яких застосовується у відповідному сценарії. Архітектура системи побудована на принципі автоматичного вибору алгоритму залежно від доступності даних, що дозволяє забезпечити рекомендації навіть у найскладніших ситуаціях, включаючи усі варіанти проблеми холодного старту.

Розробка системи відштовхується від типового життєвого циклу клієнта на сайті електронної комерції. У момент першого відвідування користувача про нього ще нічого не відомо: немає історії покупок, переглядів, оцінок, демографічних даних. На цьому етапі система має запропонувати рекомендації, які, ймовірно, будуть цікавими максимально широкому колу користувачів. Після того як користувач здійснив декілька покупок або поставив оцінки, система може перейти до персоналізованих рекомендацій на основі колаборативної фільтрації.

Окремим складним сценарієм є запуск нового бізнесу або інтернет-магазину. На початковому етапі немає жодних транзакційних даних: ні користувачів з історією, ні оцінок товарів. Стандартні методи колаборативної фільтрації у такому випадку незастосовні. Запропонована система вирішує цю проблему за допомогою аналізу текстових описів товарів: товари кластеризуються за схожістю описів, і користувач може отримати рекомендації з тематичного кластера на основі пошукового запиту.

Архітектура системи складається з трьох взаємодоповнюючих підсистем (див. рис. 2.1). Перша підсистема – рекомендації на основі популярності – використовується для нових користувачів без жодної історії взаємодій. Підсистема працює з агрегованими даними по всіх товарах та видає top-N найпопулярніших продуктів за кількістю оцінок. Друга підсистема –

колаборативна фільтрація з SVD – є основним механізмом персоналізованих рекомендацій після того, як накопичилася достатня кількість оцінок. Третя підсистема – кластеризація текстових описів – використовується для нових бізнесів та нових товарів.

Кожна підсистема використовує власні набори даних та алгоритми. Popularity-based та collaborative filtering працюють з даними Amazon Reviews: набором, що містить оцінки користувачів для товарів електроніки. Text clustering працює з даними Home Depot Product Search Relevance, що містить детальні текстові описи будівельних та господарських товарів. Це дозволяє продемонструвати роботу всіх трьох підходів на реальних промислових даних.

Технологічний стек системи включає мову Python та її екосистему бібліотек для аналізу даних та машинного навчання. Pandas використовується для обробки табличних даних, NumPy – для числових обчислень, SciPy – для роботи з розрідженими матрицями та SVD, scikit-learn – для TF-IDF векторизації та K-means кластеризації, matplotlib та seaborn – для візуалізації результатів. Середовищем розробки є Jupyter Notebook, що забезпечує інтерактивну роботу з даними.

Рисунок 2.1 – Загальна архітектура гібридної рекомендаційної системи

Перевагою гібридної архітектури є її гнучкість та робастність. Жоден окремий алгоритм не може ефективно працювати у всіх можливих сценаріях, тому об’єднання трьох підходів забезпечує максимальне покриття. Окрім того,

гібридна архітектура дозволяє поступово вдосконалювати систему: можна почати з popularity-based рекомендацій, потім додати text clustering, а коли накопичиться достатньо даних – запустити collaborative filtering.

2.2 Сценарії застосування та логіка перемикання

Ключовою архітектурною особливістю запропонованої системи є автоматичне перемикання між алгоритмами залежно від ситуації. Логіка перемикання (див. рис. 2.2) базується на оцінці доступності даних на трьох рівнях: рівень користувача (чи має він історію), рівень товару (чи має він оцінки) та рівень системи (чи має вона дані взагалі).

Рисунок 2.2 – Логіка автоматичного перемикання між підсистемами

Сценарій 1: Новий користувач, існуюча система з даними. Користувач відвідує сайт уперше. Система не має про нього інформації, але має дані про інших користувачів та оцінки товарів. Активується popularity-based підсистема: користувач отримує список найбільш популярних товарів сайту. Ця стратегія є оптимальною, оскільки популярні товари мають високу ймовірність сподобатися широкому колу користувачів.

Сценарій 2: Постійний користувач, існуюча система. Користувач уже здійснив декілька покупок або поставив оцінки товарам. Активується підсистема колаборативної фільтрації з SVD. На основі історії оцінок користувача та інших користувачів, система обчислює прогнозовані оцінки для товарів, яких користувач ще не бачив, і рекомендує товари з найвищими прогнозами. Це найбільш персоналізована форма рекомендацій.

Сценарій 3: Новий бізнес без даних. Інтернет-магазин щойно запустився і ще не має жодної історії покупок або оцінок. Стандартні методи рекомендацій незастосовні. Активується підсистема text clustering: товари кластеризуються за описами, і користувач отримує рекомендації товарів з того кластера, до якого належить його пошуковий запит або переглянутий товар.

Сценарій 4: Гібридне використання. У реальних виробничих системах часто застосовується комбінування. Наприклад, для постійного користувача основні рекомендації даються через collaborative filtering, але якщо у списку зустрічається новий товар без оцінок, його релевантність оцінюється через text clustering. Або: popularity-based рекомендації перемішуються з персоналізованими для забезпечення різноманітності.

Логіка перемикання може бути реалізована як простий ланцюжок умов або як більш складна правила-орієнтована система. У простому випадку: якщо у користувача 0 оцінок – використовується popularity-based; якщо у користувача 1+ оцінок – colaborative filtering; якщо це новий магазин без жодних оцінок – text clustering. Пороги переключення можуть бути налаштовані експериментально.

2.3 Підсистема рекомендацій на основі популярності

Підсистема рекомендацій на основі популярності реалізована найпростіше з трьох. Її задача – надати швидку та статистично обґрунтовану рекомендацію для нового користувача, який щойно зареєструвався або зайшов на сайт уперше. Як метрика популярності використовується кількість оцінок

товару: чим більше людей купило та оцінило товар, тим він популярніший і тим більша ймовірність, що новому користувачеві він також сподобається.

Алгоритм роботи підсистеми складається з декількох послідовних кроків. Перший крок – завантаження даних оцінок Amazon Reviews. Кожен запис у даних містить такі поля: userId (ідентифікатор користувача), productId (ідентифікатор товару), rating (оцінка від 1 до 5), timestamp (часова мітка). Другий крок – групування даних за productId та обчислення метрик популярності: кількість оцінок та середня оцінка.

Третій крок – фільтрація товарів за мінімальною кількістю оцінок. Це важливо для статистичної надійності: товар з лише 1 оцінкою 5.0 не є більш популярним, ніж товар з 1000 оцінок та середньою 4.5. Зазвичай застосовується поріг від 50 до 100 мінімальних оцінок. Четвертий крок – сортування товарів за спаданням кількості оцінок (або іншою метрикою популярності). П’ятий крок – повернення top-N товарів як рекомендацій (зазвичай N = 10).

Підсистема popularity-based має декілька важливих властивостей. По-перше, її обчислювальна складність є O(n), де n – кількість оцінок. Це означає, що навіть для великих наборів даних рекомендації обчислюються миттєво. По-друге, результат не залежить від користувача – усі нові користувачі отримують однакові рекомендації. По-третє, рекомендації є детермінованими: за умови незмінних даних результат буде однаковим.

Можливі модифікації базового підходу включають: зважування з урахуванням середньої оцінки (формула IMDb Top 250); врахування свіжості оцінок (нещодавні оцінки мають більшу вагу); популярність у конкретній категорії або сегменті; персоналізована популярність на основі демографічних даних (популярне серед користувачів зі схожою демографією); популярність у певному часовому проміжку (наприклад, «популярне цього тижня»).

2.4 Підсистема колаборативної фільтрації з SVD

Підсистема колаборативної фільтрації з сингулярним розкладанням матриці є основним механізмом персоналізованих рекомендацій у системі. Вона активується після того, як у користувача накопичилася певна кількість оцінок (хоча б одна). Підсистема працює з тим самим набором даних Amazon Reviews, але використовує не лише агреговані статистики, а й структуру взаємодій між користувачами та товарами.

Перший крок роботи підсистеми – побудова матриці оцінок користувач-товар. З вихідних даних формується таблиця, де рядки відповідають користувачам, стовпці – товарам, а у клітинках на перетині знаходяться оцінки. Невиставлені оцінки заповнюються нулями. Розмір такої матриці може досягати десятків мільйонів елементів, при цьому заповненими є лише декілька відсотків клітинок (матриця є розрідженою).

Другий крок – застосування SVD для розкладання матриці оцінок на низькорангове наближення. Використовується функція svds з бібліотеки SciPy, яка ефективно працює з розрідженими матрицями. Параметр k визначає кількість латентних факторів і обирається експериментально (зазвичай від 10 до 100). Менше значення k призводить до більшого спрощення, що зменшує шум, але може втратити важливі деталі. Більше значення k зберігає більше інформації, але збільшує обчислювальну складність.

Третій крок – відновлення повної матриці оцінок з трьох SVD-матриць. Прогнозовані оцінки обчислюються як добуток $U \cdot \Sigma \cdot V^T$. Результатом є щільна матриця, де для кожної пари (користувач, товар) є числове значення прогнозованої оцінки. Важливо зазначити, що ці прогнози не є точними оцінками – вони є наближеннями, що враховують латентні залежності між користувачами та товарами.

Четвертий крок – обчислення кореляції товарів. Для конкретного товару (наприклад, того, що користувач щойно купив або переглянув), знаходяться найбільш корельовані товари у просторі латентних факторів. Кореляція

обчислюється між векторами товарів у латентному просторі за допомогою коефіцієнта кореляції Пірсона або косинусної схожості.

П’ятий крок – формування топ-10 рекомендацій. З усіх товарів обираються 10 з найвищою кореляцією до товару-зразка, виключаючи сам товар-зразок. Ці товари рекомендуються користувачеві як «Інші клієнти, які купили цей товар, також купили...». Така стратегія використовується Amazon і є однією з найбільш ефективних форм item-to-item колаборативної фільтрації.

Алгоритм має декілька важливих переваг. По-перше, він масштабований: SVD на розріджених матрицях ефективно працює навіть з мільйонами користувачів та товарів. По-друге, він автоматично виявляє латентні залежності у даних без потреби у експертних правилах. По-третє, він забезпечує високу якість рекомендацій, що було підтверджено численними експериментами та практичним успіхом Netflix.

2.5 Підсистема кластеризації текстових описів

Підсистема кластеризації текстових описів є відповіддю на проблему повного холодного старту нового бізнесу. Коли у системі немає жодних даних про користувачів або їхні оцінки, єдиним джерелом інформації залишаються самі товари – їхні описи, характеристики, ключові слова. Аналіз текстових описів дозволяє групувати товари у тематичні кластери та надавати рекомендації на основі семантичної близькості.

Підсистема працює з даними Home Depot Product Search Relevance – набором, що містить інформацію про десятки тисяч товарів великого американського будівельного гіпермаркету. Кожен запис включає такі поля: product_uid (ідентифікатор товару), product_title (назва), product_description (детальний опис). Описи містять інформацію про призначення товару, його характеристики, способи використання.

Перший крок – попередня обробка тексту. Описи товарів проходять стандартний пайплайн обробки природної мови: переведення у нижній

регістр, видалення пунктуації та цифр, токенізація (розбиття на окремі слова), видалення стоп-слів (загальних слів типу «the», «a», «and», які не несуть специфічної інформації), лематизація або стемінг (приведення слів до базової форми).

Другий крок – векторизація TF-IDF. Очищені описи перетворюються у числові вектори за допомогою класу TfidfVectorizer бібліотеки scikit-learn. Для кожного унікального слова з усього корпусу описів обчислюється його TF-IDF вага у кожному документі. Результатом є розріджена матриця: рядки відповідають товарам, стовпці – словам, а значення – ваги TF-IDF. Зазвичай матриця має розмірність кілька тисяч на десятки тисяч.

Третій крок – кластеризація K-means. Алгоритм MiniBatchKMeans (більш ефективна модифікація K-means для великих наборів даних) групує товари у заданій кількості кластерів (наприклад, 10). Кожен товар отримує мітку кластера. Центроїди кластерів представляються як середні TF-IDF вектори товарів у кожному кластері і відображають «семантичний центр» кожної тематичної групи.

Четвертий крок – аналіз кластерів. Для кожного кластера визначаються найбільш характерні терміни – ті, що мають найвищі ваги у центроїді. Наприклад, у даних Home Depot перший кластер може характеризуватися словами «light watt volt led power fan bulb bulbs lighting home» – це товари освітлення. Інший кластер може бути присвячений сантехніці зі словами «pipe water valve drain plumbing» і так далі.

П’ятий крок – видача рекомендацій. Коли користувач вводить пошуковий запит або переглядає певний товар, його запит/опис також векторизується TF-IDF. Потім обчислюється відстань до центроїдів усіх кластерів, і обирається найближчий кластер. Рекомендуються товари з цього кластера, ранжовані за схожістю до запиту. Це дозволяє рекомендувати тематично пов’язані товари навіть без жодної інформації про користувача.

2.6 Інтеграція трьох підсистем

Інтеграція трьох підсистем у єдину рекомендаційну систему здійснюється на рівні диспетчера запитів, який вибирає відповідну підсистему залежно від контексту запиту. Коли система отримує запит на рекомендації, вона перевіряє наступне: чи є користувач у системі; чи має він історію оцінок; чи містить система достатньо даних для колаборативної фільтрації.

Діалог із системою може бути таким: користувач відвідує сайт уперше → popularity-based видає топ-10 популярних товарів; користувач переглядає товар X → збираються неявні дані; користувач купує товар Y та ставить оцінку → у системі з’явилася історія; наступне відвідування → SVD-collaborative filtering видає персоналізовані рекомендації на основі товару Y; користувач вводить пошуковий запит для категорії, де у нього мало історії → text clustering доповнює рекомендації.

Інтеграція також може бути реалізована як зважений ансамбль, де рекомендації з різних підсистем змішуються з різними вагами залежно від ситуації. Наприклад: 70% рекомендацій від колаборативної фільтрації + 20% від популярного + 10% від text clustering для забезпечення різноманітності. Конкретні ваги налаштовуються експериментально через A/B-тестування.

3 ОПИС РОБОТИ ПРОГРАМНОГО КОДУ

3.1 Середовище розробки та використані бібліотеки

Програмна реалізація рекомендаційної системи виконана мовою Python у середовищі Jupyter Notebook. Jupyter Notebook є інтерактивним середовищем для наукових обчислень, що дозволяє поєднувати код, текстові пояснення та візуалізації в одному документі. Це робить його ідеальним інструментом для дослідницької роботи з даними та машинного навчання.

Файл проєкту «Recommendation System - Paul.ipynb» містить послідовність обчислювальних комірок з кодом, який реалізує усі три підсистеми. Структура файла зрозуміла та документована: кожна підсистема відокремлена тематичним заголовком у форматі Markdown, що полегшує навігацію та розуміння логіки роботи.

Стек використаних бібліотек Python включає такі компоненти. Бібліотека pandas – основний інструмент для обробки табличних даних. Надає структуру даних DataFrame, методи завантаження CSV-файлів, групування, фільтрації, агрегації. У проєкті pandas використовується для завантаження даних оцінок Amazon та описів товарів Home Depot, побудови матриці користувач-товар, агрегації статистик популярності.

Бібліотека NumPy – фундаментальний пакет для наукових обчислень. Надає підтримку багатовимірних масивів (ndarray) та векторизованих математичних операцій. У проєкті використовується для роботи з матрицями оцінок та обчислення метрик.

Бібліотека SciPy – містить алгоритми для наукових та інженерних обчислень. Особливо важливим є модуль scipy.sparse для роботи з розрідженими матрицями (типовий випадок для рекомендаційних систем) та модуль scipy.sparse.linalg, що містить функцію svds – ефективну реалізацію сингулярного розкладання матриці для розріджених матриць.

Бібліотека scikit-learn – провідна бібліотека машинного навчання для Python. У проєкті використовуються класи TfidfVectorizer для векторизації текстів та MiniBatchKMeans для кластеризації. MiniBatchKMeans є більш ефективною модифікацією K-means для великих наборів даних, що обробляє дані мінібатчами замість усього набору одночасно.

Бібліотеки matplotlib та seaborn – інструменти візуалізації, що використовуються для побудови графіків розподілу оцінок, гістограм популярності, кластерних діаграм. NLTK (Natural Language Toolkit) – бібліотека обробки природної мови, що використовується для токенізації, видалення стоп-слів та стемінгу текстових описів товарів.

3.2 Завантаження та підготовка даних Amazon

Перший етап роботи системи – завантаження та підготовка даних оцінок Amazon. Використовується відкритий датасет, що містить оцінки користувачів для товарів категорії електроніки. Дані завантажуються з CSV-файла за допомогою функції pd.read_csv() з бібліотеки pandas (лістинг 3.1):

Лістинг 3.1 – Завантаження даних

```
import pandas as pd
import numpy as np


amazon_ratings = pd.read_csv('ratings_Beauty.csv')
amazon_ratings = amazon_ratings.dropna()
print(amazon_ratings.shape)
print(amazon_ratings.head())
```

Стандартна структура датасету Amazon Reviews включає чотири колонки: UserId (унікальний ідентифікатор користувача, рядок), ProductId (унікальний ідентифікатор товару, рядок), Rating (оцінка користувача від 1 до 5, число з плаваючою точкою), Timestamp (часова мітка оцінки, числове значення Unix timestamp). Після завантаження викликається dropna() для

видалення записів з пропущеними значеннями, що забезпечує цілісність даних.

Після завантаження виконується розвідувальний аналіз даних (EDA). Описова статистика отримується через метод describe(). Кількість унікальних користувачів та товарів обчислюється через amazon_ratings[’UserId’].nunique() та amazon_ratings[’ProductId’].nunique(). Розподіл оцінок візуалізується через гістограму (лістинг 3.2):

Лістинг 3.2 – Побудова гістограми

```
import matplotlib.pyplot as plt
import seaborn as sns


amazon_ratings['Rating'].value_counts().plot(kind='bar')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.title('Distribution of Ratings')
plt.show()
```

Для практичних обчислень, особливо для матричної факторизації, використовується підмножина даних. Розріджена матриця користувач-товар для повного датасету може мати розмірність кілька мільйонів на сотні тисяч, що вимагає значних обчислювальних ресурсів. Тому беруться лише перші N записів (наприклад, перші 10000 рядків) або застосовується фільтрація: вибираються лише ті користувачі, що мають мінімум K оцінок, та товари з мінімум M оцінками. Це підвищує щільність матриці та якість роботи алгоритмів.

3.3 Реалізація Popularity-based рекомендацій

Реалізація popularity-based підсистеми є найпростішою з трьох. Основна ідея – групування даних за товаром та підрахунок кількості оцінок для кожного (лістинг 3.3). Код використовує функціонал агрегації pandas.

Лістинг 3.3 – Групування даних за товаром та підрахунок кількості оцінок для кожного

```
# Підрахунок кількості оцінок для кожного товару
popular_products = pd.DataFrame(
    amazon_ratings.groupby('ProductId')['Rating'].count())

# Сортування за спаданням популярності
most_popular = popular_products.sort_values(
    'Rating', ascending=False)

# Виведення топ-10 найпопулярніших товарів
print(most_popular.head(10))
```

Метод groupby() агрегує дані за стовпцем ProductId, після чого функція count() підраховує кількість записів (оцінок) для кожного товару. Результат сортується методом sort_values() у спадному порядку. Перші 10 записів видаються як рекомендації найпопулярніших товарів.

Для візуалізації топ-30 найпопулярніших товарів використовується горизонтальна стовпчикова діаграма: most_popular.head(30).plot(kind='bar'). Це дозволяє графічно побачити розподіл популярності – зазвичай він має вигляд експоненційного спаду, типового для «довгого хвоста» у даних електронної комерції.

Розширена версія алгоритму може враховувати не лише кількість оцінок, а й середню оцінку. Це реалізується за допомогою декількох агрегацій: popular_products = amazon_ratings.groupby('ProductId').agg({'Rating': ['count', 'mean']}). Потім обчислюється комбінована метрика, наприклад, за формулою Байєсової середньої оцінки IMDb: $(v / (v + m)) \cdot R + (m / (v + m)) \cdot C$, де v – кількість оцінок, m – мінімальна кількість оцінок для топа, R – середня оцінка товару, C – середня оцінка по всіх товарах.

3.4 Реалізація колаборативної фільтрації з SVD

Колаборативна фільтрація з SVD є центральним алгоритмом системи. Перший крок – побудова матриці користувач-товар з оцінками. Це досягається за допомогою методу pivot_table() з бібліотеки pandas (лістинг 3.4):

Лістинг 3.4 – Побудова матриці користувач-товар з оцінками

```
# Беремо підмножину для практичних обчислень
amazon_ratings1 = amazon_ratings.head(10000)

# Побудова матриці користувач-товар
ratings_utility_matrix = amazon_ratings1.pivot_table(
    values='Rating',
    index='UserId',
    columns='ProductId',
    fill_value=0)

print(ratings_utility_matrix.shape)
```

Метод pivot_table() перетворює «довгий» формат даних (де кожен рядок – одна оцінка) у «широкий» формат матриці. Рядки відповідають користувачам, стовпці – товарам, а значення на перетині – оцінкам. Параметр fill_value=0 заповнює відсутні оцінки нулями. Розмірність отриманої матриці може бути, наприклад, 8000 користувачів на 2000 товарів.

Другий крок – транспонування матриці. Для item-to-item колаборативної фільтрації необхідно мати товари у рядках. Це дозволяє знаходити кореляцію між векторами товарів (лістинг 3.5):

Лістинг 3.5 – Транспонування матриці

```
# Транспонування: тепер рядки - товари
X = ratings_utility_matrix.T
print(X.shape)

# Збереження ідентифікаторів товарів
X1 = X
```

Третій крок – застосування SVD. Функція TruncatedSVD з scikit-learn виконує усічене сингулярне розкладання, що зберігає лише k найбільших сингулярних чисел (лістинг 3.6):

Лістинг 3.6 – Усічене сингулярне розкладання

```
from sklearn.decomposition import TruncatedSVD

# Створення моделі SVD з 10 латентними факторами
SVD = TruncatedSVD(n_components=10)
decomposed_matrix = SVD.fit_transform(X)
print(decomposed_matrix.shape)
```

Параметр n_components=10 означає, що зберігаються 10 латентних факторів. Це суттєво стискає простір ознак: замість тисяч стовпців з оцінками, кожен товар представляється вектором з 10 чисел. Цей низькорозмірний простір захоплює основні приховані залежності у даних, ігноруючи шум.

Четвертий крок – обчислення матриці кореляції між товарами у низькорозмірному просторі (лістинг 3.7):

Лістинг 3.7 – Обчислення матриці кореляції між товарами

```
import numpy as np

# Кореляційна матриця товарів
correlation_matrix = np.corrcoef(decomposed_matrix)
print(correlation_matrix.shape)
```

Функція np.corrcoef() обчислює коефіцієнти кореляції Пірсона між усіма парами рядків матриці. Результатом є симетрична матриця розмірністю (n_products × n_products), де елемент [i, j] показує кореляцію між товаром i та товаром j у латентному просторі.

П’ятий крок – видача рекомендацій. Для конкретного товару (наприклад, того, що користувач щойно придбав), знаходяться найбільш корельовані товари (лістинг 3.8):

Лістинг 3.8 – Видача рекомендацій

```
# Знайти індекс товару у матриці
product_names = list(X.index)
product_ID = product_names.index('B00000K135')


# Кореляції цього товару з усіма іншими
correlation_product_ID = correlation_matrix[product_ID]


# Топ-10 найбільш корельованих товарів
Recommend = list(X.index[correlation_product_ID > 0.90])
Recommend.remove('B00000K135')  # видаляємо сам товар
Recommend[0:9]  # топ-9 рекомендацій
```

Така стратегія є типовою item-to-item collaborative filtering, що використовується Amazon. Користувачеві, який купив товар X, рекомендуються товари, які мають високу кореляцію з X у латентному просторі. Інтерпретація: «Інші користувачі, що купили X, також купили...». Поріг кореляції 0.90 обраний експериментально – він забезпечує високу схожість при достатній кількості рекомендацій.

Схематично алгоритм видачі рекомендацій показано на рисунку 3.1.

3.5 Реалізація кластеризації текстових описів

Кластеризація текстових описів є третьою підсистемою, що вирішує проблему холодного старту нового бізнесу. Для демонстрації використовуються дані Home Depot Product Search Relevance, що містять детальні описи будівельних та господарських товарів. Перший крок – завантаження даних (лістинг 3.9).

Лістинг 3.9 – Завантаження даних

```
import pandas as pd

product_descriptions = pd.read_csv(
    'product_descriptions.csv')
print(product_descriptions.shape)
print(product_descriptions.head())
```

Датасет містить два основні поля: product_uid (унікальний ідентифікатор товару) та product_description (детальний текстовий опис). Описи можуть бути різної довжини – від декількох речень до повних технічних специфікацій. Після завантаження викликається dropna() для видалення записів без описів.

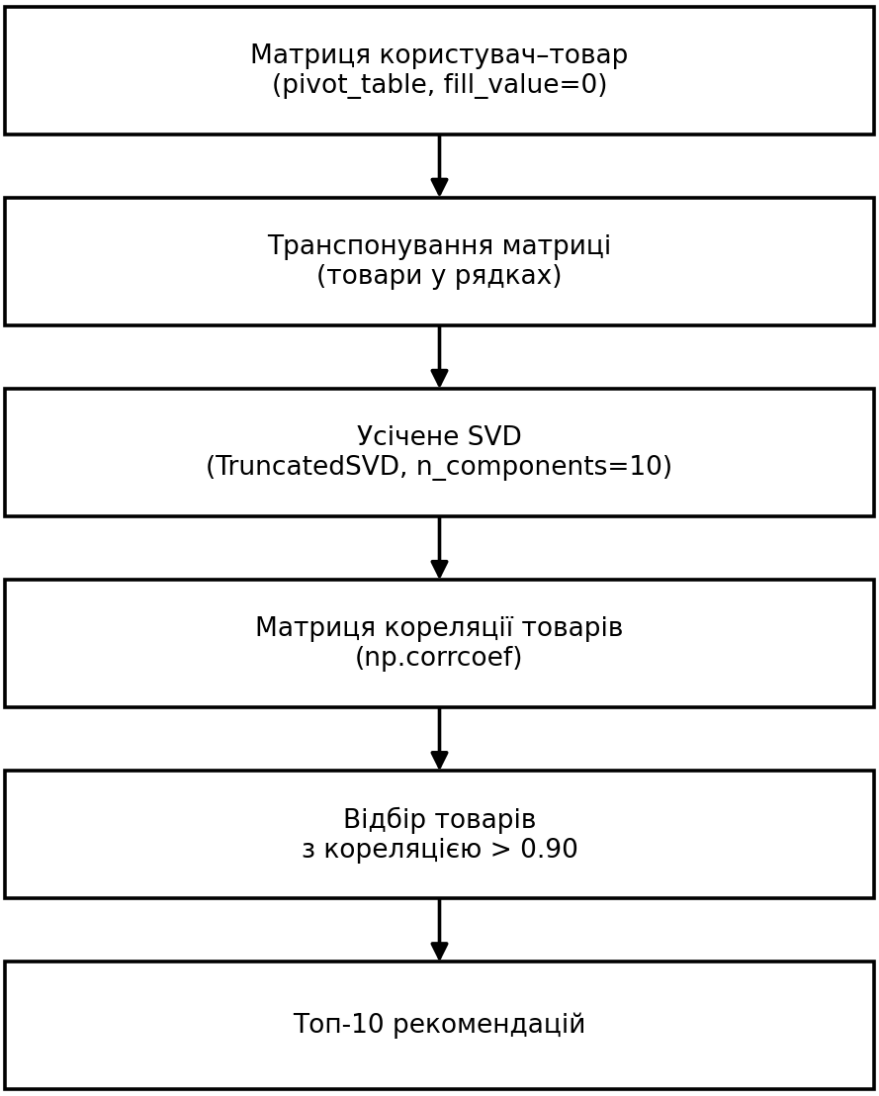

Рисунок 3.1 – Конвеєр колаборативної фільтрації з SVD

Другий крок – векторизація TF-IDF. Використовується клас TfidfVectorizer з бібліотеки scikit-learn (лістинг 3.10):

Лістинг 3.10 – Векторизація

```
from sklearn.feature_extraction.text import TfidfVectorizer

# Беремо підмножину для практичних обчислень
product_descriptions1 = product_descriptions.head(500)

# Створення TF-IDF векторизатора
vectorizer = TfidfVectorizer(stop_words='english')
X1 = vectorizer.fit_transform(
    product_descriptions1['product_description'])
print(X1.shape)
```

Параметр stop_words=’english’ вказує на автоматичне видалення англійських стоп-слів (the, a, and, of, to і т.д.), які не несуть специфічної інформації. Результатом методу fit_transform() є розріджена матриця TF-IDF: рядки – товари, стовпці – слова, значення – ваги TF-IDF. Атрибут vectorizer.get_feature_names_out() повертає список усіх слів-ознак.

Третій крок – кластеризація K-means. Використовується клас MiniBatchKMeans – більш ефективна модифікація для великих наборів даних (лістинг 3.11):

Лістинг 3.11 – Кластеризація методом K-means

```
from sklearn.cluster import MiniBatchKMeans

true_k = 10  # кількість кластерів
model = MiniBatchKMeans(n_clusters=true_k,
                        init='k-means++',
                        n_init=1)
model.fit(X1)
```

Параметр n_clusters=10 задає кількість кластерів, init='k-means++' – розумний метод ініціалізації центроїдів (на відміну від випадкової), що покращує збіжність та якість результату. Після виклику fit() модель готова до використання.

Четвертий крок – аналіз кластерів. Для кожного кластера виводяться найбільш характерні терміни (лістинг 3.12):

Лістинг 3.12 – Аналіз клстерів

```
# Топ-10 термінів кожного кластера
order_centroids = model.cluster_centers_.argsort()[:, ::-1]
terms = vectorizer.get_feature_names_out()

for i in range(true_k):
    print(f'Cluster {i}:')
    for ind in order_centroids[i, :10]:
        print(f'  {terms[ind]}')
    print()
```

Атрибут cluster_centers_ містить координати центроїдів у TF-IDF просторі. Метод argsort()[:, ::-1] сортує індекси термінів за спаданням ваги у центроїді. Для кожного кластера виводяться перші 10 найбільш характерних термінів. У результаті виконання коду на даних Home Depot можна побачити такі кластери: один, що характеризується словами «light watt volt led power fan bulb bulbs lighting home» (товари освітлення); інший зі словами «water pipe drain valve faucet» (сантехніка) тощо.

П'ятий крок – функція рекомендацій за запитом (лістинг 3.13):

Лістинг 3.13 – Видача рекомендацій

```
def show_recommendations(product):
    Y = vectorizer.transform([product])
    prediction = model.predict(Y)
    print_cluster(prediction[0])

show_recommendations('cutting tool')
```

```
show_recommendations('spray paint')
show_recommendations('steel drill')
```

Функція приймає текстовий запит, перетворює його у TF-IDF вектор через vectorizer.transform(), визначає найближчий кластер через model.predict() і виводить характерні товари цього кластера. Схематично весь алгоритм показано на рисунку 3.2.

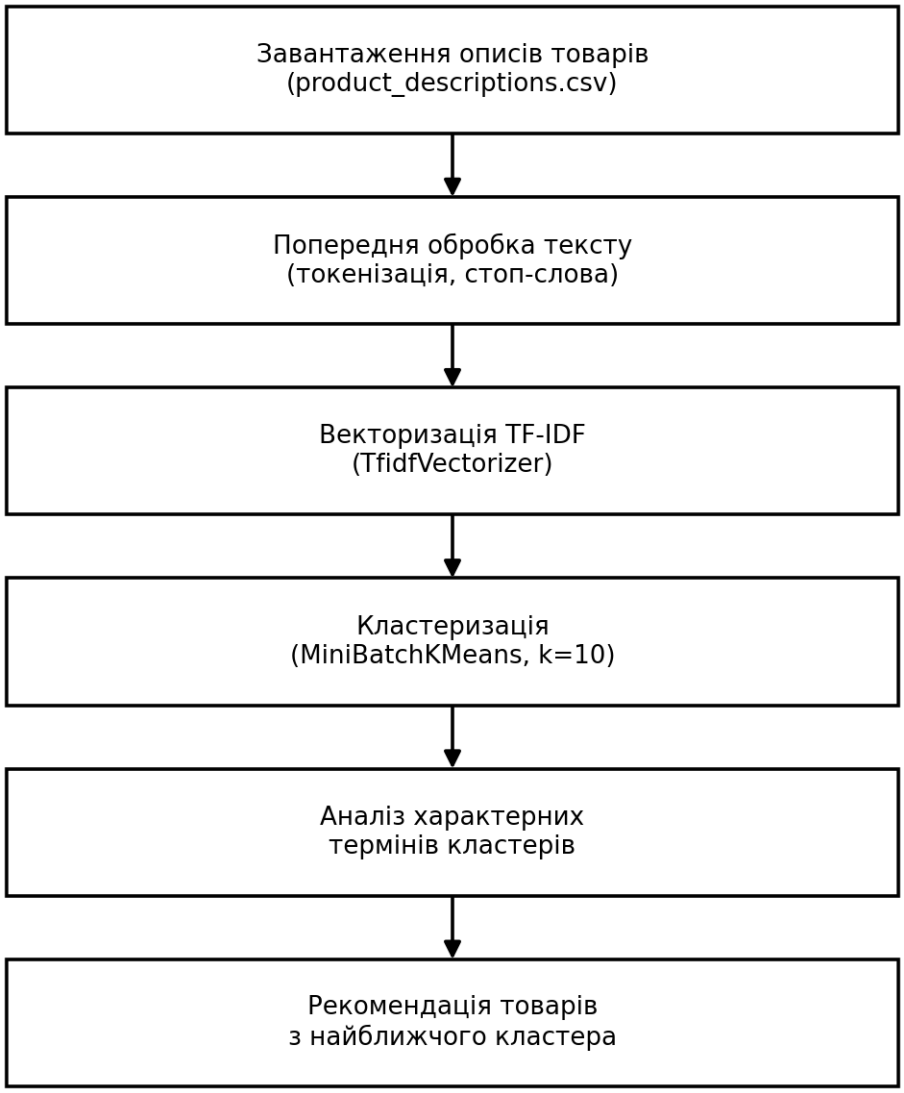

Рисунок 3.2 – Конвеєр кластеризації описів товарів (TF-IDF + K-means)

Таким чином, користувач може шукати товари за загальною темою, навіть якщо точна назва йому невідома.

3.6 Результати та аналіз роботи системи

Перевірка роботи всіх трьох підсистем демонструє ефективність гібридного підходу. Для popularity-based підсистеми результатом є список товарів з найвищою кількістю оцінок. Як правило, перші позиції займають товари популярних брендів з кількома тисячами оцінок, що відображає природну концентрацію уваги покупців на проміжку відомих продуктів.

Для колаборативної фільтрації з SVD результатом для конкретного товару (наприклад, з ID «B00000K135») є список з 10 товарів, які мають найвищу кореляцію у латентному просторі. Якщо ID-зразок належить до косметики, рекомендації будуть з тієї ж категорії, бренду або типу продукту, тобто система автоматично виявляє концептуальну схожість без явних правил.

Для кластеризації текстів результат залежить від запиту користувача. Запит «cutting tool» класифікується у кластер інструментів, що містить пили, ножиці, різальні полотна. Запит «spray paint» потрапляє у кластер фарб та малярних товарів. Запит «steel drill» – у кластер свердл та електроінструментів. Таким чином, навіть без жодних оцінок чи історії покупок система здатна надавати тематично релевантні рекомендації.

Аналіз якості рекомендацій усіх трьох підсистем показує, що кожна з них найкраще працює у своєму цільовому сценарії. Popularity-based є оптимальним для нових користувачів, оскільки популярні товари мають високу ймовірність сподобатися широкому колу. Collaborative filtering з SVD забезпечує найбільш персоналізовані рекомендації, коли є достатня історія. Text clustering ефективно вирішує повний холодний старт через семантичний аналіз товарів.

Обчислювальна складність трьох підсистем суттєво різниться. Popularity-based має складність $O(n)$, де n – кількість оцінок. SVD має $O(\min(m, n)^2 \cdot \max(m, n))$ для повного розкладання, але для розріджених матриць з усіченим SVD складність значно нижча. K-means має складність $O(n \cdot k \cdot i \cdot d)$, де n – кількість документів, k – кількість кластерів, i – кількість

ітерацій, d – розмірність простору. У промислових застосуваннях вибір алгоритму також враховує його обчислювальну складність.

Запропонована гібридна архітектура має ряд практичних переваг порівняно з використанням окремих алгоритмів. По-перше, вона забезпечує безперервний цикл рекомендацій від першого відвідування до повної персоналізації. По-друге, вона є робастною до проблеми холодного старту. По-третє, вона дозволяє поступово вдосконалювати систему: можна почати з простих popularity-based рекомендацій, а потім поступово додавати більш складні алгоритми у міру накопичення даних.

Обмеженнями розглянутої системи є: SVD-based collaborative filtering чутливий до розрідженості матриці і може мати знижену якість при дуже малій кількості оцінок; text clustering залежить від якості описів товарів – описи неінформативні або занадто короткі будуть погано кластеризуватися; popularity-based не персоналізована та схильна до ефекту «популярний стає популярнішим».

Напрямки подальшого розвитку системи включають: інтеграцію методів глибокого навчання (нейронної колаборативної фільтрації, автоенкодерів); врахування контексту (час доби, тип пристрою, геолокація); інкорпорацію неявного зворотного зв’язку (час перегляду, кліки) разом з явними оцінками; реалізацію A/B-тестування для оптимізації порогів перемикання між алгоритмами; додавання механізмів забезпечення різноманітності рекомендацій (diversity-aware recommendations).

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Аналіз небезпеки і шкідливості при розробці програмного забезпечення

Організація робочого місця розробника ПЗ впливає на його працездатність.

У своїй діяльності розробник використовує комп’ютер, пристрої збереження інформації, а тому є необхідність забезпечення зручного доступу до всіх технічних засобів. Тому в даному розділі докладніше розглянемо відомості про систему ергономічних норм і принципів організації робочого місця, на котрому проводяться роботи зі створення модуля збору статистики.

Під робочим місцем розуміється зона, оснащена необхідними технічними засобами, у якій відбувається трудова діяльність виконавця або групи виконавців, які спільно виконують одну роботу або операцію.

Організація робочого місця полягає у виконанні заходів, які забезпечують безпечний і раціональний трудовий процес і ефективне використання знарядь та предметів праці, що підвищує продуктивність праці і знижує стомлюваність працівника.

Організація робочого місця залежить від характеру розв'язуваних задач і особливостей предметно-просторового оточення, що визначають робоче положення тіла і можливість пауз для відпочинку, типи і способи засобів відображення і керування, необхідність у засобах захисту, спецодягу, простору для налагодження і ремонту устаткування.

Одним з компонентів діяльності на робочому місці є робочі рухи. Їхня раціональна організація створює умови для зниження стомлення, резерви для підвищеної працездатності. Просторові характеристики руху оператора визначаються траєкторіями руху і розмірами моторного поля (зони досяжності).

При організації робочого місця необхідно забезпечити нормальні умови огляду. Зону огляду описує кут, вершина якого знаходиться в центрі ока, а сторони складають границі, в яких людина при фіксованому положенні голови й ока добре розрізняє їхнє місцезнаходження.

У горизонтальній площині цей кут складає 300 – 400. При організації робочого місця кут огляду можна взяти 500 – 600, включаючи зону менш ясного огляду. Допустимий кут огляду по горизонталі 900. У вертикальній площині оптимальний кут огляду 100 вверх і 300 вниз від лінії погляду, а допустимий 300 вгору і 400 вниз від лінії погляду.

Щоб зберегти нормальну гостроту зору, робочу поверхню розташовують від очей на відстані від 0,3 м до 0,75 м. Робочі меблі повинні бути зручними для виконання робочих операцій. В даному випадку робочий стіл є основним устаткуванням. Особливо важливе значення має висота столу, його конструкція, яка повинна передбачати шухляди для розміщення інструментів, документації.

Важливе значення має конструкція робочих крісел. Погано підібрані крісла можуть бути причиною надмірної стомлюваності.

Нахил і висота крісла повинні регулюватися відповідно до висоти робочої поверхні і росту працюючого. Рекомендована ширина крісла 370 – 400 мм, глибина 370 – 420 мм, висота спинки 370 – 1000 мм від рівня крісла. Для розміщення ніг необхідно передбачити вільний простір під робочою площиною [11].

Праця людини, що протікає в умовах надмірного нервово-емоційного напруження, довготривалих статичних навантажень, обмеженої рухової активності призводить до неврозів, відхилень у психіці, захворювань опорно-рухового апарату, серцево-судинної системи тощо. Комп'ютери, телебачення, системи зв'язку та інші засоби, що використовують досягнення радіоелектроніки, є генераторами цілої низки електромагнітних випромінювань, вплив яких на організм людини ще не зовсім вивчений.

З широким впровадженням автоматизації та комп'ютеризації виникла потреба врахування психологічних можливостей людини, таких як швидкість реакції, особливості пам'яті та уваги, емоційний стан та ін. Поява операторської діяльності призвела до суттєвих змін у фаховій структурі праці. Зменшились фізична важкість праці, ризик виробничого травматизму, однак разом з тим, на працюючу людину посилюється вплив нових, раніше не відомих чи мало вивчених несприятливих виробничих факторів фізичного, хімічного і особливо психофізіологічного характеру.

Проте, розвиток сучасної обчислювальної техніки відбувається не лише у бік поктащення її технічних параметрів, але також звертається увага безпеку використання цієї техніки людиною шляхом зменшення потужності випромінювачів, зменшенням рівня випромінювання з моніторів, зменшення напруг живлення, покращення ергономічних характеристик.

Таким чином, в розділі з охорони праці виконано огляд питань безпечної роботи при створенні сайту та встановлено, що умови такої роботи відповідають вимогам з охорони праці, які застосовуються в галузі інформаційних технологій.

4.2 Інформаційно-психологічні небезпеки

Сучасні реалії постіндустріального суспільства, зумовлені значним ростом інформації, відкривають ще одну сферу життєдіяльності людини – інформаційну. Сучасні засоби комунікації і обробки інформації створили принципово нові умови існування людини, що зумовило появу грандіозного проекту об'єднання національних інформаційних і телекомунікаційних структур в глобальну інформаційну інфраструктуру.

Життєдіяльність людини реалізується одночасно зі світом природи і у специфічному для людського суспільства інформаційному середовищі, що має свої закономірності розвитку і функціонування. Інформаційна сфера стає такою ж важливою складовою суспільного життя, як економічна, виробнича,

побутова, політична, військова та ін. Нові інформаційні технології, засоби масової комунікації багатократно підсилили можливості впливу на свідомість і підсвідомість як окремої людини, так і на великі групи людей та населення країни загалом.

Інформаційна сфера – сукупність таких елементів:

– об'єкти інформаційної взаємодії чи впливу;

– особисто інформація, призначена для використання суб'єктами інформаційної сфери;

– інформаційна інфраструктура, що забезпечує можливість здійснення обміну інформацією між суб'єктами;

– суспільні відносини, що складаються у зв'язку з формуванням, переданням, розповсюдженням і збереженням інформації.

Особистість, активний соціальний суб'єкт, його психіка піддаються безпосередньому впливу інформаційних чинників (передумов, що чинять опір чи утруднюють формування і функціонування адекватної інформаційно-орієнтуючої основи суспільної поведінки людини (життєдіяльності у суспільстві)), які трансформуються, через його поведінку, діяльність (бездіяльність), здійснюють деструктивний, дисфункційний вплив на його життєдіяльність.

До основних загроз інформаційно-психологічної безпеки відносять можливість настання негативних наслідків для суб'єктів, що піддаються інформаційно-психологічному впливу, які виражаються в таких формах:

– нанесення шкоди здоров'ю людини;

– блокування на неусвідомленому рівні волі, волевиявлення людини, штучне прививття їй синдрому залежності;

– втрата здатності до політичної, культурної, моральної самоіндентифікації людини;

– маніпуляція суспільною свідомістю;

– руйнування єдиного інформаційного і духовного простору України, традиційних устроїв суспільства і суспільної моральності, а також порушення інших життєво важливих інтересів особистості, суспільства, держави.

Наприклад, культ жорстокості, насильства, порнографії, розбещеності тощо, які пропагують у засобах масової інформації, друкованих виданнях, комп'ютерних іграх, мережі Інтернет веде до неусвідомленого бажання у підлітків і молоді, а також дорослих з нестійкою психікою, копіювати запропоновані моделі поведінки. Цей вид пропаганди знижує рівень порогових обмежень і правових заборон, що поряд з іншими умовами відкриває шлях для багатьох правопорушень. Це своєю чергою наносить непоправну шкоду не тільки окремій особистості, але й суттєві збитки національним інтересам країни.

Отже, джерелом інформаційно-психологічної небезпеки є та частина інформаційного середовища, яка через визначені причини неадекватно відображає реалії, вводить в оману людину, засліплює її ілюзією.

Інформаційно-психологічні загрози зумовлені розробкою, виготовленням, розповсюдженням та використанням суб'єктами негативних інформаційно-психологічних впливів, спеціальних засобів і методів такого впливу.

Концепція інформаційно-психологічної безпеки.

Сучасне розуміння безпеки в контексті врахування відношення інтересів особистості, суспільства і держави висуває завдання розгляду нового аспекту цієї проблеми – безпеки в інформаційній сфері життєдіяльності людини, тобто інформаційно-психологічної безпеки.

В інформаційному середовищі, що є складовим системним утворенням, виділяється процесуальна складова як найбільш динамічна і змінна її частина – інформаційно-комунікативні процеси, які активно впливають на індивідуальну, групову і суспільну психологію (індивідуальну, групову, масову свідомість). Маніпулюючи станом інформаційного середовища, змінюється стан духовної сфери суспільства, деформація і деструктивні зміни

якої у формі психоемоційної і соціальної напруженості, спотворених норм і неадекватних соціальних стереотипів і установок, оманливих і неприродних орієнтацій та цінностей. Це своєю чергою впливає на стан і процеси у всіх основних сферах суспільного життя, в тому числі політичній і економічній.

Вперше у пострадянському просторі про проблему інформаційно-психологічної безпеки було зазначено в листопаді 1995 р. на науково-практичній конференції, організованій Інститутом психології Російської академії наук. На цій та подальших конференціях було розкрито роль знання технологій інформаційно-психологічного впливу, метою якого є маніпуляція, для вироблення напрямів реформування психологічного захисту особистості і особистої інформаційно-психологічної безпеки.

Інформаційно-психологічну безпеку особистості визначають такими основними причинами.

Зростання тиску інформаційного середовища визначає необхідність формування нових механізмів та засобів виживання людини як особистості й активного соціального суб'єкта у сучасному суспільстві.

Взаємодія психіки людини з інформаційним середовищем відрізняється якісною специфікою і не має аналогів у комунікації інших біологічних, технічних, соціальних і соціотехнічних структур.

Основною і центральною "мішенню" інформаційного впливу є людина, її психіка.

Отже, інформаційно-психологічну безпеку можливо розглядати як стан захищеності особистості, різних соціальних груп і об'єднань людей від дій, впливів, які здатні проти їхньої волі і бажання змінити психічні стани та психологічні характеристики людини, модифікувати її поведінку і обмежувати свободу вибору, зумовило потребу переосмислення інформаційної взаємодії, а також деяких інших соціально-психологічних процесів і явищ у сучасному суспільстві.

Інформаційно-психологічна безпека – стан захищеності окремих осіб чи груп осіб від негативних інформаційно-психологічних впливів і пов'язаних з

цим інших життєво важливих інтересів особистості, суспільства, держави в інформаційному середовищі.

Негативний інформаційно-психологічний вплив – процес зміни психічних станів і характеристик людей під впливом інформаційно-комунікативних процесів як динамічного компонента інформаційного середовища. Цей вплив спрямований на людину чи групу осіб (у тому числі без їхньої згоди) з метою примусу до визначеної поведінки, оцінки ситуації, керування та корекції індивідуальної та колективної свідомості. Він здійснюється з використанням спеціальних засобів і методів впливу на психіку людини, унаслідок чого він приводить до негативних наслідків для особистості, суспільства і держави.

Спеціальні засоби впливу – технічні і програмні засоби, що використовують для використання з метою негативного інформаційно-психологічного впливу на людину чи групу людей.

Спеціальні методи впливу – послідовність прийомів впливу на психіку людини, використання яких приводить до негативних наслідків для особистості, суспільства та держави.

Головним об'єктом забезпечення інформаційно-психологічної безпеки в інформаційному середовищі у сфері індивідуальної безпеки є усвідомлення інформації, здатність людини адекватно сприймати навколишню дійсність, своє місце в зовнішньому світі, формувати відповідно до свого життєвого досвіду визначені переконання і приймати стосовно них рішення.

Інформаційно-психологічна безпека має спиратися на стандарти інформаційно-психологічної безпеки – затверджені у визначеному порядку інформаційно-психологічного впливу, який не викликає негативних наслідків для психіки людини.

ВИСНОВКИ

У кваліфікаційній роботі магістра проведено комплексне дослідження принципів побудови рекомендаційних систем для електронної комерції з використанням методів машинного навчання. Дослідження охоплює як теоретичні основи галузі, так і практичну реалізацію гібридної рекомендаційної системи, що поєднує три різних алгоритми для різних сценаріїв застосування.

По-перше, проведено систематичний теоретичний аналіз рекомендаційних систем як підгалузі машинного навчання та систем фільтрації інформації. Розглянуто історичний розвиток галузі від ранніх систем 1990-х років до сучасних промислових рішень, що генерують значну частку доходу таких компаній, як Amazon, Netflix, Spotify. Описано формальну постановку задачі рекомендацій як задачу заповнення розрідженої матриці оцінок.

По-друге, проаналізовано основні класи рекомендаційних систем: popularity-based, collaborative filtering, content-based, hybrid та context-aware. Для кожного класу описано основні алгоритми, переваги та обмеження. Особлива увага приділена методам колаборативної фільтрації, зокрема, model-based підходам на основі матричної факторизації.

По-третє, детально розглянуто математичний апарат сингулярного розкладання матриці (SVD) як ключового інструменту матричної факторизації у рекомендаційних системах. Описано теорію SVD, його властивості та особливості застосування для розріджених матриць оцінок. Розглянуто такі модифікації як Funk SVD, SVD++ та їхні переваги.

По-четверте, проаналізовано методи аналізу текстового вмісту: векторизація TF-IDF та кластеризація K-means. Описано теоретичні основи кожного методу, особливості реалізації та практичні аспекти застосування для рекомендаційних систем у сценаріях повного холодного старту.

По-п’яте, розглянуто проблему холодного старту у всіх її варіантах (новий користувач, новий товар, нова система) та проаналізовано підходи до її вирішення. Показано, що жоден окремий алгоритм не може ефективно працювати у всіх можливих сценаріях, що обґрунтовує необхідність гібридного підходу.

По-шосте, описано архітектуру практично реалізованої гібридної рекомендаційної системи, яка автоматично обирає алгоритм залежно від доступності даних. Система складається з трьох підсистем: popularity-based для нових користувачів, collaborative filtering з SVD для постійних користувачів з історією, text clustering для нових бізнесів без жодних транзакційних даних.

По-сьоме, проаналізовано програмну реалізацію системи мовою Python з використанням бібліотек pandas, NumPy, SciPy, scikit-learn та matplotlib. Розглянуто кожен з трьох алгоритмів окремо: побудова матриці користувач-товар, застосування TruncatedSVD для зменшення розмірності, обчислення кореляційної матриці товарів, побудова item-to-item рекомендацій, TF-IDF векторизація описів, кластеризація K-means.

По-восьме, проаналізовано результати роботи кожної з трьох підсистем на реальних даних Amazon Reviews та Home Depot Products. Показано, що кожна підсистема ефективно працює у своєму цільовому сценарії. Виявлено практичні переваги гібридного підходу: робастність до проблеми холодного старту, безперервний цикл рекомендацій, можливість поступового вдосконалення системи.

Наукове значення роботи полягає у систематичному дослідженні принципів побудови гібридних рекомендаційних систем з автоматичним перемиканням алгоритмів. Описаний підхід може бути застосований не лише в електронній комерції, а й у інших галузях, де виникає задача рекомендацій: стримінгові сервіси, освітні платформи, новинні агрегатори, соціальні мережі.

Практичне значення результатів полягає у демонстрації архітектурного шаблону, придатного для впровадження у реальних промислових системах.

Принципи, описані у роботі, можуть бути використані при розробці нових e-commerce платформ або при модернізації існуючих рекомендаційних систем. Гнучкість гібридного підходу дозволяє адаптувати його до конкретних бізнес-потреб та обмежень.

Перспективні напрямки подальших досліджень включають: інтеграцію методів глибокого навчання (нейронна колаборативна фільтрація, графові нейронні мережі для рекомендацій); врахування контексту користувача (час, місце, пристрій) у запропонованій архітектурі; розробку механізмів забезпечення різноманітності та новизни рекомендацій; дослідження впливу інтерпретованості рекомендацій на довіру користувачів; A/B-тестування реалізованої системи у виробничому середовищі для емпіричної оцінки ефективності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Курчак Ю. Р. Моделювання та аналіз системи рекомендацій на веб-сайті на основі теорії графів та статистичних алгоритмів : робота на здобуття кваліфікаційного ступеня магістра, спец. 122 – комп'ютерні науки / Ю. Р. Курчак ; наук. кер. І. О. Боднарчук. – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2025. – 83 с.

2. Багрій М. Т. Розробка рекомендаційної системи для інтернет-магазину книг : робота на здобуття кваліфікаційного ступеня бакалавра, спец. 122 – комп'ютерні науки / М. Т. Багрій ; наук. кер. С. В. Марценко. – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2025. – 64 с.

3. Небесний Р. М. Рекомендаційна система формування команд виконавців з відповідними фаховими компетентностями : дис. ... д-ра філософії : 122 / Р. М. Небесний. – Тернопіль, 2023. – 253 с.

4. Ржеуський А., Кунанець Н., Стахів М. Рекомендаційна система інформаційного обслуговування користувачів бібліотек // Матеріали V науково-технічної конференції «Інформаційні моделі, системи та технології», 1–2 лютого 2018 року. – Тернопіль : ТНТУ, 2018.

5. Бідюк П. І., Коршевнюк Л. О. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень. Київ : Наукова думка, 2010. 340 с.

6. Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. 2nd ed. Springer, 2015. 1003 p.

7. Aggarwal C. C. Recommender Systems: The Textbook. Springer, 2016. 498 p.

8. Falk K. Practical Recommender Systems. Manning Publications, 2019. 432 p.

9. Sarwar B., Karypis G., Konstan J., Riedl J. Item-Based Collaborative Filtering Recommendation Algorithms. Proceedings of the 10th International Conference on World Wide Web. 2001. P. 285–295.

10. Koren Y., Bell R., Volinsky C. Matrix Factorization Techniques for Recommender Systems. IEEE Computer. 2009. Vol. 42, No. 8. P. 30–37.

11. Bennett J., Lanning S. The Netflix Prize. Proceedings of KDD Cup and Workshop. 2007. P. 3–6.

12. Linden G., Smith B., York J. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing. 2003. Vol. 7, No. 1. P. 76–80.

13. Goldberg D., Nichols D., Oki B. M., Terry D. Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM. 1992. Vol. 35, No. 12. P. 61–70.

14. Pedregosa F. et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011. Vol. 12. P. 2825–2830.

15. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. 2nd ed. O'Reilly Media, 2017. 544 p.

16. Salton G., McGill M. J. Introduction to Modern Information Retrieval. McGraw-Hill, 1983. 448 p.

17. Manning C. D., Raghavan P., Schütze H. Introduction to Information Retrieval. Cambridge University Press, 2008. 482 p.

18. Arthur D., Vassilvitskii S. k-means++: The Advantages of Careful Seeding. Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms. 2007. P. 1027–1035.

19. He X., Liao L., Zhang H., Nie L., Hu X., Chua T.-S. Neural Collaborative Filtering. Proceedings of the 26th International Conference on World Wide Web. 2017. P. 173–182.

20. Su X., Khoshgoftaar T. M. A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence. 2009. Article ID 421425. 19 p.

21. Burke R. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction. 2002. Vol. 12, No. 4. P. 331–370.

22. Zhang S., Yao L., Sun A., Tay Y. Deep Learning Based Recommender System: A Survey and New Perspectives. ACM Computing Surveys. 2019. Vol. 52, No. 1. P. 1–38.

23. Paul R. Python-Ecommerce-recommendation-system-using-machine-learning. GitHub repository. URL: https://github.com/RudrenduPaul/Python-Ecommerce-recommendation-system-using-machine-learning (дата звернення: 03.06.2026).

24. Amazon Product Reviews Dataset. Kaggle. URL: https://www.kaggle.com/datasets (дата звернення: 03.06.2026).

25. Home Depot Product Search Relevance Dataset. Kaggle. URL: https://www.kaggle.com/c/home-depot-product-search-relevance (дата звернення: 03.06.2026).

26. Scikit-learn Documentation. URL: https://scikit-learn.org/stable/documentation.html (дата звернення: 03.06.2026).

27. SciPy Documentation: Sparse SVD. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.linalg.svds.html (дата звернення: 03.06.2026).

28. Стручок, В. С., Стручок, О. С., & Мудра, Д. В. (2017). Навчальний посібник до написання розділу дипломного проекту та дипломної роботи ″Безпека в надзвичайних ситуаціях ″для студентів всіх спец. денної, заочної (дистанційної) та екстернатної форм навчання.

29. Стручок, В. С. (2022). Техноекологія та цивільна безпека. Частина "Цивільна безпека". Навчальний посібник.

30. Жидецький, В. Ц., Джигирей, В. С., & Мельников, О. В. (2000). Основи охорони праці. Львів: Афіша, 350, 132-136.

31. Навакатікян О. О., Кальниш В. В., & Стрюков С. М. (1997). Охорона праці користувачів комп'ютерних відеодисплейних терміналів. О. Навакатікян.