

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Системотехніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА

### Пояснювальна записка

другий (магістерський)  
(рівень вищої освіти)

ГЮІК.50 2840.006 ПЗ  
(позначення документа)

Дослідження та використання методів  
розпізнавання зображень  
для рекомендаційних систем  
(тема)

Виконав: здобувач групи ІТПм-20-1  
спеціальності 122 – Комп'ютерні науки  
(код і повна назва спеціальності)

освітньої програми Освітньо-професійна  
«Інформаційні технології проектування»  
(повна назва освітньої програми)

Міронова К. В.  
(прізвище, ініціали)

Керівник проф. Колесник Л. В.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри системотехніки

\_\_\_\_\_

(підпис)

\_\_\_\_\_


(прізвище, ініціали)

2021 р.

*Кваліфікаційна робота оформлена у відповідності до вимог діючих стандартів та методичних вказівок.*

*Матеріали кваліфікаційної роботи не містять відомостей, що заборонені для опублікування у відкритих виданнях.*

*Попередній захист проведено 13 грудня 2021 року.*

*Керівник кваліфікаційної роботи*  *Л.В. Колесник*

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_  
Кафедра \_\_\_\_\_ Системотехніки \_\_\_\_\_  
Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_  
Спеціальність \_\_\_\_\_ 122 Комп'ютерні науки \_\_\_\_\_  
(код і повна назва)  
Освітня програма \_\_\_\_\_ Інформаційні технології проектування \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри

\_\_\_\_\_ (підпис)  
«\_\_\_\_\_» \_\_\_\_\_ 20 21  
р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачові \_\_\_\_\_ Міроновій Катерині Вадимівні \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та використання методів розпізнавання зображень для рекомендаційних систем  
затверджена наказом по університету від «08» листопада 20 21 р. № \_\_\_\_\_ 1663 Ст
2. Термін подання здобувачем роботи \_\_\_\_\_ 18 грудня 2021р.
3. Вихідні дані до роботи \_\_\_\_\_ Функція: Дослідження та використання методів розпізнавання зображень для рекомендаційних систем. Форма діалогу: веб-сервіс. Перелік використовуваних програмних засобів: ОС Windows, інтегроване середовище розробки PyCharm, Python, Tensorflow, Flask framework, PostgreSQL. Технічне забезпечення: комп'ютер з веб-браузером Google Chrome версії вище 60, Safari версії вище 12, Firefox версії вище за 60 незалежно від операційної системи
4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ. 4.2 Поняття рекомендаційної системи. 4.3 Загальна класифікація рекомендаційних систем. 4.4 Класифікація рекомендаційних систем за методами фільтрації. 4.5 Оцінка якості рекомендаційних систем. 4.6 Проблеми рекомендаційних систем. 4.7 Порівняння методів надання рекомендацій. 4.8 Розпізнавання зображень. 4.9 Використання розпізнавання зображень в рекомендаційних системах. 4.10 Постановка задачі. 4.11 Дослідження методів розв'язання задачі. 4.12 Метод спільного метричного навчання. 4.13 Метод знаходження візуального байесівського персоналізованого рейтингу. 4.14 Знаходження косинусної міри подібності. 4.15 Нейронна мережа для визначення візуальних характеристик зображень. 4.16 Удосконалення методу надання рекомендацій. 4.17 Опис тестових наборів даних. 4.18 Опис методики оцінювання результатів експерименту. 4.19 Результати досліджень. 4.20 UML-проекування. 4.21 Опис

структури бази даних. 4.22 Обґрунтування вибору інструментів. 4.23 Опис реалізації додатку. 4.24 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 5.1 Загальна схема згорткової нейронної мережі. 5.2 Результати досліджень 5.3 Діаграма прецедентів. 5.4 Діаграма послідовностей отримання рекомендацій. 5.5 Діаграма послідовностей додавання фільму в «Переглянути пізніше». 5.6 Проектування бази даних. 5.7 Архітектура додатку. 5.8 Схема алгоритму запропонованої рекомендаційної системи

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	01.09.2021	Виконано
2	Аналіз завдання, пошук літератури та аналогів з теми дипломної роботи	02.09.2021 – 12.09.2021	Виконано
3	Опрацювання літератури та аналіз об'єкту дослідження	13.09.2021 – 18.09.2021	Виконано
4	Постановка та формалізація задачі	19.09.2021 – 20.09.2021	Виконано
5	Дослідження методів та алгоритмів вирішення задачі	21.09.2021 – 09.10.2021	Виконано
6	Розробка гібридного алгоритму системи	10.10.2021 – 24.10.2021	Виконано
7	Розробка та тестування програмної реалізації	25.10.2021 – 06.11.2021	Виконано
8	Аналіз результатів, отриманих за допомогою програмного засобу	07.11.2021 – 10.11.2021	Виконано
9	Оформлення пояснювальної записки	11.11.2021 – 04.12.2021	Виконано
10	Оформлення графічної частини та презентаційних матеріалів комп'ютерного захисту	05.12.2021 – 16.12.2021	Виконано
11	Представлення дипломної роботи в деканат	17.12.2021	Виконано

Дата видачі завдання 1 вересня 2021 р.

Здобувач \_\_\_\_\_

(підпис)

Керівник роботи \_\_\_\_\_ проф. Колесник Л. В.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи: 107 с., 8 табл., 18 рис., 3 додатки, 41 джерело інформації.

ГІБРИДНИЙ МЕТОД, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, КОНТЕНТНА ФІЛЬТРАЦІЯ, КОСИНУСНА МІРА ПОДІБНОСТІ, РЕКОМЕНДАЦІЙНА СИСТЕМА, РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, СПІЛЬНА ФІЛЬТРАЦІЯ, CNN, PYTHON, VGG19

Об'єкт дослідження – процес формування персональних рекомендацій продуктів на базі вподобань користувача і технологій розпізнавання зображень.

Предмет дослідження – методи та моделі формування персональних рекомендацій із застосуванням підходів до розпізнавання зображень.

Мета роботи – підвищення ефективності роботи рекомендаційних систем за рахунок врахування візуальних ознак зображень продуктів. Для досягнення мети необхідно виконати такі завдання:

- здійснити порівняльний аналіз існуючих методів створення рекомендацій та розпізнавання зображень;
- розробити гібридний алгоритм вирішення поставленої задачі;
- спроектувати та розробити прототип рекомендаційної системи;
- проаналізувати ефективність роботи алгоритму.

Методи дослідження – теоретичні (абстрагування, порівняння, аналіз літератури) та емпіричні (опис та моделювання) методи, системний підхід.

Наукова новизна результатів полягає у застосуванні розпізнавання зображень для формування гібридного методу для формування персональних рекомендацій.

В результаті проведених досліджень вирішено задачу формування рекомендацій на базі попередньої взаємодії користувача з продуктами системи та їх текстовими та візуальними характеристиками, спроектована та побудована рекомендаційна система з використанням згорткових нейронних мереж.

Запропонована технологія може бути застосована у будь-яких сервісах, що використовують рекомендації: у сфері електронної комерції, стримінгових сервісах, порталах новин, соціальних мережах та ін.

## ABSTRACT

Explanatory note to the master's qualification work: 107 pages, 8 tables, 18 figures, 3 additions, 41 sources.

CNN, COLLABORATIVE FILTERING, CONTENT FILTRATION, CONVOLUTIONAL NEURAL NETWORK, COSINE SIMILARITY, HYBRID METHOD, IMAGE RECOGNITION, PYTHON, RECOMMENDATION SYSTEM, VGG19

Object of research is the process of forming personal product recommendations based on user preferences and image recognition technologies.

The subject of research is methods and models for the formation of personal recommendations with the use of approaches to image recognition.

The purpose of the research is to improve the efficiency of recommendation systems by taking into account the visual features of product images. To achieve the purpose, we need to accomplish following tasks:

- carry out a comparative analysis of existing methods of creating recommendations and image recognition;
- to develop a hybrid algorithm for solving the problem;
- design and develop a prototype of a recommendation system;
- analyze the efficiency of the algorithm.

Methods of research – theoretical (abstraction, comparison, analysis of the literature) and empirical (description and modeling) methods, system approach.

The scientific novelty of the results is the use of image recognition to form a hybrid method for the formation of personal recommendations.

As a result of the conducted researches the problem of formation of recommendations on the basis of preliminary interaction of the user with products of system and their text and visual characteristics is solved, the recommendation system with based on convolutional neural networks is designed and constructed.

The proposed technology can be used in any services that use the recommendations: in the field of e-commerce, streaming services, news portals, social networks, etc.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	9
Вступ.....	10
1 Аналіз предметної області.....	12
1.1 Поняття рекомендаційної системи .....	12
1.2 Загальна класифікація рекомендаційних систем.....	14
1.3 Класифікація рекомендаційних систем за методами фільтрації .....	16
1.3.1 Методи, засновані на вмісті .....	16
1.3.2 Методи, засновані на знаннях.....	17
1.3.3 Спільна фільтрація .....	18
1.3.4 Гібридна фільтрація .....	20
1.4 Оцінка якості рекомендаційних систем .....	21
1.5 Проблеми рекомендаційних систем .....	25
1.6 Порівняння методів надання рекомендацій.....	26
1.7 Розпізнавання зображень .....	29
1.7.1 Згорткові нейронні мережі .....	30
1.7.2 Операція згортки .....	31
1.7.3 Структура згорткової нейронної мережі.....	32
1.7.4 Додаткові техніки .....	34
1.7.5 Переваги та недоліки згорткових нейронних мереж.....	35
1.8 Використання розпізнавання зображень в рекомендаційних системах .....	36
1.9 Постановка задачі .....	39
2.1 Дослідження методів розв'язання задачі.....	40
2.1.1 Дослідження методів розпізнавання зображень .....	40
2.1.2 Дослідження методів надання рекомендацій .....	43
2.2 Метод спільного метричного навчання.....	44
2.3 Метод знаходження візуального байєсівського персоналізованого рейтингу .....	49
2.4 Знаходження косинусної міри подібності.....	52
2.5 Нейронна мережа для визначення візуальних характеристик зображень..	54
2.6 Удосконалення методу надання рекомендацій з використанням розпізнавання зображень.....	56
3 Результати експериментальних досліджень.....	61
3.1 Опис тестових наборів даних .....	61

3.2	Опис методики оцінювання результатів експерименту .....	63
3.3	Результати досліджень .....	65
4	Проектування та розробка рекомендаційної системи .....	67
4.1	UML-проектування.....	67
4.2	Проектування бази даних .....	70
4.3	Обґрунтування вибору інструментів .....	72
4.4	Архітектура додатку.....	74
	Висновки .....	75
	Перелік джерел посилання .....	76
	Додаток А. Графічний матеріал кваліфікаційної роботи .....	80
	Додаток Б. Текст програми .....	89
	Додаток В. Відомість кваліфікаційної роботи.....	106

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

РС – рекомендаційна система.

BPR – Bayesian Personalized Ranking – байєсівське персональне ранжування – алгоритм рекомендації товарів на базі неявного зворотного зв'язку та байєсівського аналізу.

CML – Collaborative Metric Learning – спільне метричне навчання.

CNN – Convolutional Neural Network – згорткова нейронна мережа.

ERD – Entity-Relationship Diagram – діаграма «сутність-зв'язок».

FM – Factorization Machine – факторизаційна машина.

MAE – Mean Absolute Error – середня абсолютна похибка.

MF – Matrix Factorization – матрична факторизація.

MSE – Mean Squared Error – середньоквадратична похибка.

nDCG – Normalized discounted cumulative gain – нормалізований дисконтований сукупний виграш.

SVD – Singular Value Decomposition – Сингулярний розклад матриці – розкладання прямокутної дійсної або комплексної матриці для приведення її до канонічного виду, що має широке застосування при вирішенні багатьох прикладних завдань.

RBM – Restricted Boltzmann Machine – Обмежена машина Больцмана – генеративна стохастична нейронна мережа, яка здатна навчатися розподіляти ймовірності за набором вхідних даних.

RMSE – Root Mean Squared Error – корінь із середньоквадратичної похибки.

VBPR – Visual Bayesian Personalized Ranking – візуальне байєсівське персональне ранжування.

## ВСТУП

У зв'язку з безперервним зростанням інформації на онлайн-платформах, користувачам доводиться стикаються з надзвичайно великою кількістю статей, відео, музики або зображень, що призводить до проблеми інформаційного навантаження. Для відвідувачів сайтів стає все складніше орієнтуватися в них навіть із використанням фільтрів пошуку, щоб знаходити саме ті матеріали, що їх цікавлять. У той же час для компаній на конкурентному ринку стало все важче пропонувати саме ті товари чи послуги, які задовольняють потреби клієнтів. Вирішенням цієї проблеми може стати персоналізація контенту: врахування смаків конкретного споживача при пропонуванні йому вмісту веб-сервісів. Тому в сучасному світі зростає інтерес до використання та вдосконалення рекомендаційних систем.

Рекомендаційна система – це інформаційна система, головною метою якої є надання користувачеві персоналізованих списків елементів інформації на базі його вподобань та характеристик продуктів.

Наразі існують два основних підходи до побудови рекомендаційних систем: використання методів, заснованих на вмісті, та спільна фільтрація. Окрім згаданих, іноді використовуються методи, засновані на знаннях, демографічні та гібридні, що поєднують переваги декількох методів.

Всі ці технології мають свої переваги та недоліки, проте існують труднощі, з якими стикаються всі методи – проблема холодного старту і проблема бульбашки фільтрів. Перша з них виникає в ситуації, коли в системі відсутні дані про переваги користувача, наприклад, при появі нового користувача в системі. Ця проблема може виникнути також у випадку, якщо характеристики об'єктів предметної області складні та у відвідувачів недостатньо взаємодії з ними, наприклад, у разі продажу нерухомості або автомобілів, купівля яких у середньому відбувається рідше, ніж одягу або меблів. Проблема бульбашки фільтрів полягає в рекомендації клієнтам товарів, які були схожі на ті, які йому сподобалися. Таким чином навколо відвідувача штучно створюється середовище, де можуть бути відсутні товари з інших категорій, які потенційно могли б бути йому цікавими.

У загальному випадку списки рекомендацій будуються на налаштуваннях користувача, особливостях товарів, історії його взаємодій із вмістом онлайн-

платформ у вигляді покупок, переглядів, лайків або оцінок продуктів у диференційованій шкалі. Зазвичай ці дані мають текстовий вигляд, однак для підвищення обізнаності системи про смаки користувача пропонується використовувати розпізнавання зображень.

Розпізнавання зображень – підрозділ технологій комп'ютерного зору, який дозволяє ідентифікувати сутності на зображенні та відносити їх до певних категорій. Дана технологія використовується в таких сферах життя: медицина, виробництво, електронна комерція, транспорт, соціальні мережі та ін.

Метою даної роботи є дослідження застосування розпізнавання зображень у існуючих методах та системах, а також підвищення ефективності рекомендаційних систем за рахунок створення нового вдосконаленого підходу для надання точніших рекомендацій.

Виходячи з цієї мети були визначені завдання дослідження:

- розгляд існуючих методів створення рекомендацій та розпізнавання зображень;
- порівняльний аналіз обраних алгоритмів надання рекомендацій;
- постановка формалізованої задачі надання формування рекомендацій на базі гібридного підходу;
- розробка ефективного алгоритму вирішення поставленої задачі з урахуванням переваг вибраних методів;
- проектування та реалізація прототипу рекомендаційної системи;
- аналіз ефективності роботи алгоритму.

Об'єктом дослідження є процес формування персональних рекомендацій контенту на інтернет-платформах, а також технології розпізнавання і обробки зображень.

Предмет дослідження – методи та моделі формування персональних рекомендацій із застосуванням підходів до розпізнавання зображень.

Для досягнення мети роботи використовується комплекс з теоретичних (абстрагування, порівняння, аналіз технічної літератури та результатів дослідження) та емпіричних (експеримент, опис та моделювання) методів.

Матеріали роботи опубліковані в тезах Всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих учених «Комп'ютерно-інтегровані технології автоматизації технологічних процесів на транспорті та у виробництві».

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Поняття рекомендаційної системи

У сучасному світі з постійно зростаючими обсягами онлайн-інформації проблема надлишку вибору для користувачів мережі стала особливо гострою. Інтернет-сервіси щодня пропонують десятки нових статей, відео- і аудіоматеріалів, а також тисячі варіантів різноманітних товарів. Ухвалення навіть простих рішень в таких умовах стає завданням, що вимагає комплексного підходу.

Впоратись з цією ситуацією допомогла персоналізація контенту. Персоналізація – це адаптація рекомендацій для користувача, заснована на знаннях про нього (наприклад, на його вподобаннях або поведінці) [1]. Вона дозволила клієнтам ефективніше взаємодіяти з веб-ресурсами, приймати якісніші рішення, а також посприяла зростанню і розвитку онлайн-бізнесу.

Рекомендаційні системи (РС) – це інформаційні системи, які допомагають користувачеві зробити вибір продукту, представляючи персоналізовані рекомендації щодо асортименту продуктів, які будуть цікаві користувачеві в тій чи іншій ситуації (наприклад, при перегляді відео, читанні новин або виборі місця для відпочинку). На відміну від традиційної системи підтримки прийняття рішень, які переважно використовуються фахівцями, рекомендаційні системи призначені для підтримки споживачів в процесі прийняття рішень [1].

Серед найбільш відомих онлайн-ресурсів, які використовують РС, варто відзначити наступні:

- Netflix – потоковий сервіс за передплатою, що надає можливість дивитися фільми і серіали онлайн (в тому числі і власного виробництва), а також використовує технологію матричної факторизації (SVD) та обмежених машин Больцмана (RBM) для рекомендацій відеоконтенту (з персоналізованими обкладинками) на основі історії переглядів, оцінок фільмів, характеристик фільму (жанр, акторський склад, рік виходу та інше), даних про час доби, пристрої, і часу перегляду контенту користувачем; в даний момент в системі тестуються трейлери, створені за допомогою штучного інтелекту по вподобаннях глядача;

- Amazon – найбільша в світі платформа електронної комерції та відомий постачальник хмарних послуг, що продає майже всі категорії продуктів: електроніка, одяг, товари для дому, програмне забезпечення та ін. та

використовує власний метод спільної фільтрації за елементами (Item-To-Item Collaborative Filtering) [2], який полягає в зіставленні кожного з придбаних і оцінених користувачем товарів з аналогічними елементами, а потім об'єднанні схожих товарів в список рекомендацій;

– YouTube – онлайн платформа поширення відео, яка налічує понад 2.29 млн. глядачів, і використовує дві глибокі нейронні мережі для генерації кандидатів на основі історії переглядів і ранжування результатів (відео присвоюється оцінка на основі релевантності ключових слів (заголовка, опису та ін.) І показників залученості (час перегляду, кількість лайків, коментарів), таким чином отримуючи можливість обробляти значну кількість відеозаписів) [3];

– Facebook – найбільша соціальна мережа, що має близько 2.85 млн. Активних користувачів на місяць, яка використовує РС із спільною фільтрацією для рекомендацій фото, відео, товарів, груп, подій, сторінок, ігор та ін. контенту, а також Apache Giraph для обробки великих масивів інформації; факторами ранжування для постів є відносини (оцінюється, чи часто користувач взаємодіє з джерелом інформації), тип контенту (фото, відео і т. д.), популярність (кількість лайків, репостів, реакцій і коментарів), давність розміщення;

– Spotify – сервіс потокової передачі аудіоконтенту, що надає доступ до музичних композицій, аудіокниг і подкастів, відома завдяки персоналізованому списку Параметри «Discover Weekly», який використовує спільну фільтрацію для аналізу моделі поведінки слухачів (на основі додавання до плейлиста або перегляду сторінки виконавця), обробку природного мови для аналізу тексту і згорткову нейронну мережу (CNN) для знаходження подібних пісень за розпізнаними характеристикам (гармонія, гучність, темп, тривалість і іншим).

– TikTok – платформа поширення коротких мобільних відео, швидко завоювала популярність по всьому світу завдяки своїй РС, яка використовує як дані про дії користувача (перегляди, лайки), так і дані про відео (підписи, звуки, хештеги), а також налаштування пристрою або облікового запису;

– Google News – персоналізований портал новин від Google, який знаходить актуальні новини за тематикою, що цікавить читача, використовуючи історію перегляду статей, пошукові запити у Google Search та YouTube, а також профілі інтересів користувачів та дані їх місцезнаходження у алгоритмах фільтрації параметрів новин (байесівський аналіз попередніх вподобань) та спільної фільтрації [4].

Charu C. Aggarwal в книзі «Recommender Systems: The Textbook» [5] сформулював види проблем, які вирішуються рекомендаційними системами:

- проблема прогнозу або проблема заповнення матриці (прогноз рейтингу для комбінації користувач-продукт відповідає матриці вподобань  $m \times n$  для  $m$  користувачів та  $n$  продуктів, яка має відсутні значення, які необхідно заповнити алгоритмом навчання);
- проблема ранжування або проблема топ- $k$  рекомендацій (визначення переліку продуктів, які можуть вподобатись користувачеві).

Метою рекомендаційних систем з боку відвідувачів онлайн-ресурсу є отримання більш якісного клієнтського досвіду, а з боку підприємців – підвищення лояльності користувачів, збільшення кількості нових клієнтів та повернення старих, надання більш чіткого уявлення про їх потреби, збільшення обсягів продажу і прибутку відповідного веб-сервісу. Таким чином було встановлено.

Серед операційних і технічних цілей виділяють такі:

- актуальність (необхідно рекомендувати саме ті продукти, що цікавлять користувача);
- новизна (популярні товари швидше за все відомі користувачеві, тому або вони йому не сподобалися, або він вже використовував їх, отже необхідно надавати більше нових, ще не бачених клієнтом, продуктів);
- несподіваність (окрім очевидних рекомендацій доцільно додавати нові товари, розширюючи коло інтересів користувача та відкриваючи його приховані вподобання);
- збільшення різноманітності рекомендацій (при наданні переліку дуже схожих продуктів існуватиме більший ризик того, що всі вони не зможуть задовольнити бажанням користувача, аніж список різних товарів).

## 1.2 Загальна класифікація рекомендаційних систем

РС можна класифікувати за наступними критеріями:

- спеціалізація: товари з онлайн-магазинів (Amazon, eBay, Rozetka), туризм (Booking.com, Expedia, TripAdvisor, Airbnb), музика (Spotify, Apple Music, YouTube Music, Deezer), відео (Netflix, YouTube, HBO Max), зображення

(Instagram, Pinterest, Unsplash), новини (The New York Times, CNN, BBC News), інші користувачі (Facebook, Twitter, LinkedIn);

- мета рекомендації: створення великої бази активних користувачів, надання інформації, допомога, навчання;

- ступінь персоналізації: неперсоналізовані РС надають ідентичні рекомендації всім користувачам системи, частково персоналізовані ділять всіх користувачів на групи за певними критеріями (за місцезнаходженням, віком, національністю і т. д.), а потім пропонують продукти для груп, а персоналізовані РС пропонують товари, адаптуючись під інтереси конкретного користувача (крім цього існує ефемерна персоналізація, яка вживає тільки дані поточної сесії і постійна, яка зберігає дані про користувача і між сесіями, але вимагає ідентифікації клієнта);

- спосіб передачі інформації: надання рекомендацій можливо за допомогою push-технології (коли дані надходять користувачеві без його взаємодії з веб-сайтами в даний момент (наприклад, розсилка на електронну пошту)), або через pull-технологію, коли процес рекомендації ініціюється користувачем;

- дані про клієнта: для пропозиції товарів системам потрібно зовнішні дані про користувача, такі як явні вподобання (explicit feedback) (зворотній зв'язок, який вимагає зусиль від користувача: оцінювання за диференційованою шкалою, складання списків улюблених продуктів, лайки і дизлайки) або неявні (implicit feedback) (зберігання історії переглядів, налаштувань повідомлень, часу перегляду матеріалу на сторінці);

- методи рекомендації: ручні (списки, зазвичай складені експертами або іншими користувачами), статистичні (узагальнення думок про продукт), методи фільтрації інформації: засновані на вмісті (content-based), засновані на знаннях (knowledge-based), спільна фільтрація (collaborative filtering) і гібридні (hybrid);

- врахування змін смаків у часі [6]: загальні (такі, що використовують глобальну взаємодію між користувачем та товаром і відображають довгострокові інтереси користувача), тимчасові (такі, що поділяють час на сегменти та моделюють поведінку користувача та продукту в них) та послідовні рекомендаційні завдання (ті, що передбачають вподобання користувача на основі його останніх дій).

### 1.3 Класифікація рекомендаційних систем за методами фільтрації

Методи фільтрації в рекомендаційних системах є алгоритмами, які використовуються для визначення списку бажаних користувачем продуктів. Ідея, що лежить в основі двох основних методів, полягає в тому, товари з аналогічними характеристиками будуть оцінюватися користувачами однаково, і що користувачі із подібними вподобаннями будуть оцінювати продукти однаково. Перший підхід називається заснованим на вмісті (content-based filtering). Він базується на визначенні подібності користувачів або предметів за певними ознаками (наприклад, для фільмів це можуть бути жанри, список акторів, режисер або країна походження). Друга техніка називається спільною фільтрацією (collaborative filtering), вона визначає відношення користувача до товарів за допомогою частково заповненою матриці рейтингу (корисності). Гібридні методи поєднують в собі кілька підходів для формування більш точних рекомендацій та усунення недоліків окремо взятих методів.

#### 1.3.1 Методи, засновані на вмісті

Суть даних методів полягає в отриманні точних характеристик кожного продукту та створення «профілів» у вигляді векторів для товарів та користувачів. Характеристики продукту залежать від предметної області системи рекомендацій: так, для одягу це буде колір, матеріал, тип одягу, фірма-виробник, а для аудіо – жанр, виконавець, час створення, музикальні інструменти, що використовуються, умови виконання та інші. Профіль користувача будується на базі його історії взаємодії з товарами.

У подальшому пропонуватися будуть ті об'єкти, які мають найбільшу подібність або із самим користувачем, або із іншими об'єктами, які йому сподобалися. Для визначення міри подібності між користувачами та предметами використовуються такі методи, як косинус подібності або коефіцієнт Жаккара.

Методи, засновані на вмісті, включають такі фази:

– попередня обробка даних та вилучення характеристик (наприклад, для обробки даних текстових продуктів букви переводяться в нижній регістр, видаляються варіанти одного слова, стоп-слова та символи, які не відносяться до алфавіту, а також визначаються словосполучення);

- навчання моделі на основі вподобань користувачів та характеристик продуктів (модель зазвичай базується на алгоритмах класифікації або регресії: найвнутому класифікаторі Байеса, k-найближчих сусідів або дерев ухваленень рішень);
- використання навченої моделі для фільтрації і прогнозування рекомендацій в режимі реального часу.

Для створення рекомендацій на основі вмісту також можуть бути використані методи машинного навчання, такі як Random forest, XGBoost та інші. Вони дозволяють робити більш точний прогноз за наявності зовнішніх факторів, що не належать безпосередньо товарам чи користувачам: зміни погодних умов чи стану на ринку.

Перевагами методів, заснованих на вмісті, є надання більш точних рекомендацій в умовах недостатньої рейтингової інформації для певних продуктів, оскільки нові продукти можуть мати схожі ознаки із тими продуктами, рейтинг яких є визначеним користувачами системи. Недоліками цього виду методів є відсутність можливості надання несподіваних рекомендацій продуктів (наприклад, якщо клієнт ніколи не мав контакту з товарами, що описуються певним набором ключових слів, то вони ніколи не будуть йому запропоновані) та неефективність надання рекомендацій для нових відвідувачів сервісу (через використання у методах прогнозу історії рейтингів товарів, наданих цією людиною). Також проблемою даних методів є складність створення «профілів» у деяких видів продуктів.

Виходячи з вищесказаного, можна зробити висновок, що методи, засновані на вмісті, краще використовувати у рекомендаційних системах текстових товарів: статей або новин, аніж музики чи відеоматеріалів.

### 1.3.2 Методи, засновані на знаннях

У випадку складної системи [7], продукти якої містять багато характеристик і складні відношення між ними, доцільним є використання методів, заснованих на знаннях. До таких сфер відносяться нерухомість, автомобілі, фінансові послуги, предмети розкоші та підтримка прийняття рішень щодо здоров'я. Дані методи не використовують оцінки користувачів для формування рекомендацій. Процес пропонування заснований на базі знань, які містять дані про функції і правила, які

використовуються під час пошуку. Явна специфікація вимог призводить до більшого контролю користувачів над процесом рекомендацій.

Рекомендаційні системи, засновані на знаннях поділяються на системи на основі обмежень (ті, що надають ліміти до характеристик товарів) та системи на основі випадків (ті, що використовують конкретні продукти в ролі прикладів для знаходження подібних).

Дані методи тісно пов'язані з методами фільтрації, заснованими на вмісті, проте вони намагаються вирішити проблему холодного старту використовуючи активні специфікації користувачів або їх потреби та інтереси.

### 1.3.3 Спільна фільтрація

Підхід спільної фільтрації заснований на знаходженні подібності між смаками та вподобаннями користувачів онлайн-сервісів. Основою алгоритму є пропонування клієнтові продуктів, що сподобались іншим клієнтам у минулому, тому для надання обґрунтованих рекомендацій необхідна достатня кількість користувачів та оцінок продуктів відвідувачами.

Залежно від способу отримання та пропонування товарів системи спільної фільтрації поділяються на активні та пасивні. Активні системи представляють сервіс як засіб зв'язку, в якому користувачі мають можливість надавати рекомендації друзям або колегам. В даному випадку користувач повинен бути усвідомленим про вподобання інших користувачів, а це в свою чергу суттєво впливає на масштабованість. Через цей недолік більшість сучасних онлайн-ресурсів використовують пасивні системи спільної фільтрації. В них дані про смаки клієнтів збираються і обробляються автоматично, тому спілкування відвідувачів не є необхідністю.

На базі явних та неявних вподобань користувача будується матриця корисності (матриця користувач-продукт) (табл. 1.1).

Таблиця 1.1 – Матриця корисності

	Продукт 1	Продукт 2	Продукт 3	Продукт 4
Користувач А	1	2	3	5
Користувач Б	2	4		1
Користувач В	1	5		

Приклад відображає просту матрицю, в якій вподобання користувачів вимірюються в шкалі 1-5. Випадок, в якому користувачеві найбільше сподобався продукт відповідає цифрі 5, найменше – 1. Порожні клітинки (missing values) означають відсутність взаємодії користувача з окремим продуктом.

Аналіз матриці корисності дозволяє виявити подібність між різними продуктами (подібність типу продукт-продукт) та подібність вподобань різних клієнтів (подібність типу користувач-користувач) за відповідними оцінками продуктів. Таким чином за допомогою даної подібності з'являється можливість визначити рейтинг неоціненого продукту для рекомендації відвідувачу. Основною проблемою даного підходу є розрідженість матриці, оскільки у більшості випадків користувач не може охопити весь асортимент продуктів.

У підході спільної фільтрації зазвичай використовуються два типи методів: методи на основі пам'яті та методи на основі моделі.

Методи на основі пам'яті – це одні з найперших методів, в яких рейтинги між користувачами та продуктами прогнозуються на базі їх сусідства. Для визначення подібності в них використовуються статистичні методи, наприклад коефіцієнт кореляції Пірсона або косинус подібності. Заповнення порожніх клітинок матриці корисності відбувається на базі користувачів або продуктів.

Алгоритми на основі користувачів визначають групу з  $k$  користувачів, оцінки товарів яких схожі на відповідні у цільового користувача  $A$  (подібність рядків матриці), та вираховують рейтинги для неспостережених продуктів шляхом середнього зваженого оцінок членів цієї групи.

Алгоритми на основі продуктів визначають множину товарів  $S$ , оцінених користувачем  $A$ , які найбільше схожі на цільовий продукт  $B$  (подібність стовпців матриці). Прогноз рейтингу нового продукту знаходиться за допомогою середнього зваженого оцінок цих товарів.

Перевагами методів на основі пам'яті є простота інтерпретації та реалізації, а недоліками – складність знаходження прогнозу у випадку розрідженості даних і складність розрахунків.

Методи на основі моделі використовують моделі машинного навчання та інтелектуального аналізу даних для заповнення порожніх клітинок матриці користувач-продукт.

Прикладами таких моделей можуть слугувати дерева ухвалення рішень, моделі, засновані на правилах, мережі Байєса або моделі прихованих факторів.

Одним з найпопулярніших методів на основі моделі є матрична факторизація. Він виявляє приховані ознаки при множенні двох різних типів сутностей. Його перевагами є зниження розмірності, прискорення обчислень, вирішення проблеми розрідженості, а недоліком – недостатня інтерпретованість.

Дані методи є ефективними для даних з високою розрідженістю, а також можуть надати швидке рішення за наявності попередньо навченої моделі.

#### 1.3.4 Гібридна фільтрація

За наявності великої кількості різної інформації про вподобання користувача з'являється гнучкість при виборі методу пропонування продуктів. В даних випадках доречно використовувати гібридні рекомендаційні системи для отримання більш точних результатів та подолання обмежень окремих підходів.

Існують три головні підходи до створення таких систем:

- дизайн ансамблю (об'єднання прогнозів готових алгоритмів для створення єдиного рейтингу шляхом обчислення середньозваженого або наданням виходу одного алгоритму на вхід іншого);
- монолітний дизайн (система такого дизайну є неподільною та не має чіткої межі між складовими методами (наприклад, заснованими на вмісті або спільною фільтрацією));
- змішані системи (кожен з методів, включених в систему, представлений у вигляді чорного ящика, а система на виході надає множину рекомендацій окремих алгоритмів).

За методом гібридизації рекомендаційні системи поділяються на такі [8]:

- зважений – результати окремих методів поєднуються у спільне значення шляхом зваженого середнього, при чому ваги різних підходів коригуються під час на основі оцінки якості прогнозів;
- перемикання – обрання прогнозуючого методу системою в залежності від обставин, наприклад, для подолання проблеми холодного старту;
- змішаний – надання рекомендацій за допомогою різних підходів одночасно, що підходить для комплексних продуктів (наприклад, для програми телепередач);
- поєднання функцій – поєднання підходів шляхом створення їх єдиного представлення і застосування єдиного алгоритму (наприклад, можна припустити,

що спільні дані – це додаткові характеристики продуктів і використати методи, заснованих на вмісті);

- каскад – уточнення кожним наступним методом системи прогнозу попереднього;

- розширення функцій – використання виходу одного методу (у вигляді рейтинг чи класифікація продукту) як вхідні дані до наступного;

- мета-рівень – використання моделі, згенерованої одним методом, як вхідні дані до іншого, на відміну від розширення функцій, входом для наступного методу є не характеристики продукту, згенеровані попереднім методом, а вся модель.

Чотири методи гібридизації не чутливі до порядку: зважений, змішаний, перемикання та поєднання функцій. Такі методи гібридизації, як зважений, перемикання, каскад та розширення функцій представляють дизайн ансамблю, а поєднання функцій і мета-рівень – монолітний.

#### 1.4 Оцінка якості рекомендаційних систем

Для оцінки рекомендаційних систем важливо правильно обрати метрики оцінки якості, оскільки неправильна система оцінювання якості може призвести до суттєвої переоцінки або недооцінки справжньої точності алгоритму чи моделі.

Існують три основних підходи до оцінювання рекомендаційних систем, такі як дослідження користувача, онлайн та офлайн оцінювання.

Дослідження користувача представляє собою тестові сесії, в яких користувачів просять виконати певні задачі в системі, надаючи зворотній зв'язок до і після взаємодії з нею. Таким чином можна зібрати детальну інформацію про окремі компоненти сервісу та протестувати деталі його роботи, проте відповіді користувачів не завжди можуть бути коректні через нерепрезентативність вибірки або обізнаність клієнтів у роботі програми.

Онлайн оцінювання проводиться реальними користувачами комерційної системи. Даний вид оцінювання є більш об'єктивним, оскільки клієнти взаємодіють з сервісом в реальних умовах. Таким чином можна порівнювати довгострокову ефективність різних алгоритмів, попередньо поділивши відвідувачів на підгрупи. Однією з найпопулярніших метрик, що використовуються для онлайн оцінювання є коефіцієнт конверсії (conversion rate).

Він дозволяє визначити частоту, з якою клієнт натискає на рекомендований продукт. Недоліками цього виду є необхідність наявності великої кількості користувачів, що призводить до неможливості даного оцінювання при запуску системи, а також відсутність відкритого доступу до систем.

Офлайн оцінювання проводиться на історичних наборах даних (наприклад, рейтингах). Воно не потребує великої кількості користувачів та має стандартизовані фреймворки (наприклад, перехресне затвердження) та метрики оцінки якості (наприклад, точність).

Рекомендаційні системи досягти таких цілей: покриття, точності, різноманітності (diversity), неочікуваності (serendipity), новизни, надійності та масштабованості. Для визначення ступеня їх деяких з них досягнення можна використовувати метрики оцінки.

Метрики точності є одними із фундаментальних метрик оцінки, вони поділяються на статистичні метрики та метрики підтримки прийняття рішень.

Статистична точність надає оцінку прогнозу у числовому вигляді шляхом порівняння прогнозованого рейтингу з фактичним рейтингом користувача. Нехай похибка прогнозу  $e_{ui} = \hat{p}_{ui} - p_{ui}$  для  $u$  користувачів та  $i$  продуктів, де  $p_{ui}$  – реальне значення рейтингу, а  $\hat{p}_{ui}$  – прогнозоване. Загальна похибка визначається як середнє значення для всіх  $n$  похибок. Найбільш розповсюдженими є середня абсолютна похибка (mean absolute error, MAE) (1.1), середньоквадратична похибка (mean squared error, MSE) (1.2) та корінь із середньоквадратичної похибки (root mean squared error, RMSE) (1.3):

$$MAE = \frac{1}{n} \sum_{(ui)}^n |e_{ui}|, \quad (1.1)$$

$$MSE = \frac{1}{n} \sum_{(ui)}^n e_{ui}^2, \quad (1.2)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{(ui)}^n e_{ui}^2}. \quad (1.3)$$

Точність підтримки прийняття рішень використовується для оцінки рекомендацій із бінарними значеннями (перегляд товару, додавання в списки,

лайки). Продукти в рекомендаційних системах, які відповідають вподобанням користувача, відносяться до позитивного класу, а які не відповідають – до негативного. Таким чином відповідно до значення прогнозу товар може бути істинно позитивного (якщо вірно визначено, що продукт подобається користувачеві), істинно негативного (якщо вірно визначено, що продукт не подобається користувачеві), хибно позитивного (якщо помилково визначено, що продукт сподобається користувачеві) та хибно негативного (якщо помилково визначено, що продукт не сподобається користувачеві) класів (табл. 1.2).

В статистиці хибно негативний клас має назву помилки першого роду, а хибно позитивний – помилки другого роду. В залежності від довжини списку рекомендованих товарів змінюється відношення кількості товарів у класах false positive та false negative.

Таблиця 1.2 – Матриця помилок

		Прогнозований клас	
		+	–
Дійсний клас	+	TP (True positive)	FN (False negative)
	–	FP (False positive)	TN (True negative)

Дану криву компромісу можна визначити за допомогою метрик precision-recall (влучність-повнота) або ROC-кривої. Precision (1.4) визначає частину істинно позитивних продуктів серед усіх, які були визначені позитивними, а recall (1.5) показує, яку частку істинно позитивних продуктів з усіх об'єктів позитивного класу знайшов алгоритм.

$$Precision = \frac{TP}{TP + FP}, \quad (1.4)$$

$$Recall = \frac{TP}{TP + FN}. \quad (1.5)$$

Recall відображає можливість знаходження даного класу алгоритмом, тоді як precision – можливість виділення його серед інших класів. F-метрика комбінує

властивості precision та recall в єдиній метриці (1.6), що дозволяє точніше знаходити оптимальний розмір списку рекомендацій

$$F_{\beta} = (1 + \beta^2) * \frac{Precision * Recall}{(\beta^2 * Precision) + Recall} \quad (1.6)$$

де  $\beta$  – вага precision в метриці.

Метрики ранжування  $N$  продуктів порівнюють список рекомендованих продуктів із списком усіх продуктів, базуючись на його релевантності. Дані метрики використовують припущення, що високорелевантні продукти є більш корисними, ніж низькорелевантні та високорелевантні продукти є більш корисними, якщо вони мають вищі ранги. Таким чином усувається недолік рівних ваг помилок на низькорангових та високорангових продуктах у RMSE.

Сукупний виграш (Cumulative Gain, CG) є сумою значень релевантності всіх результатів списку рекомендацій (1.7).

$$CG_p = \sum_{i=1}^p rel_i, \quad (1.7)$$

де  $rel_i$  – релевантність елемента списку з позицією  $i$ .

Головним недоліком CG є відсутність залежності між порядком елементів в списку. Дисконтований сукупний виграш (Discounted cumulative gain, DCG) штрафует значення релевантності продуктів логарифмічно пропорційно їх положенню у списку (1.8)

$$DCG_p = \sum_{i=1}^p \frac{2^{rel_i} - 1}{\log_2(i + 1)}. \quad (1.8)$$

Для оцінки ефективності рекомендаційних систем, що можуть надавати різні за довжиною списки рекомендацій, використовується нормалізований дисконтований сукупний виграш (Normalized discounted cumulative gain, nDCG) (1.9). Для його розрахунку використовується ідеалізований дисконтований сукупний виграш (Ideal discounted cumulative gain, IDCG) (1.10)

$$nDCG_p = \frac{DCG_p}{IDCG_p}, \quad (1.9)$$

$$IDCG_p = \sum_{i=1}^{|REL_p|} \frac{2^{rel_i} - 1}{\log_2(i + 1)}, \quad (1.10)$$

де  $REL_p$  – список рекомендованих продуктів, впорядкований за їх релевантністю.

Хоча влучність і повнота часто використовуються для оцінки рекомендаційних систем, вони не завжди відображають ефективність з точки зору кінцевого користувача, оскільки у більшості випадків кінцевий користувач бачить лише невелику підмножину позицій з найвищим рейтингом.  $nDCG$  враховує важливість продуктів високого рангу тому краще підходить для розрізнення високорангових та низькорангових товарів.

### 1.5 Проблеми рекомендаційних систем

Головними проблемами, з якими стикаються рекомендаційні системи, є проблема холодного старту і бульбашки фільтрів [9].

Проблема холодного старту – це проблема створення рекомендацій для нового товару, з яким ще не взаємодіяли користувачі, або для нових користувачів, дані про вподобання яких відсутні. Способами її вирішення є вилучення інформації з інших джерел: опитування при реєстрації клієнта або отримання даних про продукт із його змісту, а також поєднання методів, заснованих на вмісті, з спільною фільтрацією. Іноді проблема холодного старту має циклічний характер (наприклад, якщо з часом змінюються смаки користувачів чи властивості продуктів). Дана ситуація має назву постійного холодного старту (continuous cold-start problem, CoCoS). Прогнозувати подібні зміни і адаптувати до них алгоритм рекомендацій допомагають методи машинного навчання.

Якщо рекомендаційна система пропонує користувачеві продукти лише на основі його попереднього досвіду, то вона створює інформаційне середовище з однотипних товарів. Дана проблема має назву бульбашки фільтрів. В цьому випадку клієнт може втратити можливість дізнатися про корисні товари або втратити інтерес до системи в цілому. Вирішенням даної проблеми є врахування

додаткових метрик оцінки якості рекомендаційної системи, такі як покриття, точності, різноманітності (diversity), неочікуваності (serendipity) і новизни.

Окрім вищезгаданих, рекомендаційні алгоритми стикаються з такими труднощами: синонімія, шилінг-атаки, конфіденційність, обмежений аналіз вмісту та надмірна спеціалізація, проблема сірої вівці, розрідженість, масштабованість, проблема затримки, проблеми оцінки та доступності онлайн-даних, усвідомлення контексту [10].

### 1.6 Порівняння методів надання рекомендацій

Окрім розглянутих глобальних проблем рекомендаційних систем, кожен їх вид може стикатися з певними лише йому властивими труднощами.

Наприклад, оскільки спільна фільтрація заснована на збігу оцінок користувачами продуктів, то у випадку, якщо таких оцінок мало (користувачі недостатньо переглядали продукт, або продуктів занадто багато щодо користувачів, як у випадку з системою новин), то матриця буде розріджена і визначити смаки користувачів стає складніше. Ця проблема може бути частково вирішена завдяки методу singular value decomposition (SVD), який може зменшити розмірність простору, де відбувається порівняння. Виходячи з цього можна зробити висновок, що чисті методи спільної фільтрації краще підходять для тих предметних областей, в яких велика кількість користувачів може взаємодіяти з відносно невеликою статичною множиною продуктів, щільно заповнюючи матрицю корисності.

Також даному методу складно виявити смаки користувача, якщо вони не є схожими на смаки інших користувачів (проблема сірої вівці).

Перевагою даного методу над методами, заснованими на змісті, є можливість виходити за рамки списку переглянутих продуктів і пропонувати користувачеві щось нове, що користується популярністю у колі клієнтів зі схожими інтересами. Тому для методу спільної фільтрації проблема бульбашки фільтрів стоїть менш гостро.

Проблема методів, заснованих на змісті, полягає в необхідності наявності точної та коректної інформації про продукти, оскільки в багатьох випадках самі продукти являють собою дані, непрозорі для системи. Так, при рекомендації фільмів система може аналізувати опис, список акторів або режисера, але не сам

відеоконтент, тому стає залежною від цієї метаінформації, яка в свою чергу є вторинною і не може повністю описати відеоматеріал.

Обидва підходи можуть відчувати труднощі у передбаченні продуктів, з якими користувач вже знайомий з інших джерел. У деяких предметних областях проблема стає досить складною, оскільки, наприклад, серед списку новин на одну тему, деякі статті можуть надати читачеві нову корисну інформацію.

Системи, що ґрунтуються на знаннях, не стикаються з проблемами холодного старту або розрідженості матриць, оскільки не використовують накопичену статистичну інформацію для прогнозу смаків клієнтів. Однак вони потребують придбання знань каталогу (знань про самі продукти та їх характеристики), функціональних знань (способи зіставити потреби користувача з відповідним товаром) та знань користувача (демографічна інформація або конкретні питання користувачеві). Такі системи можуть давати настільки великі рекомендації, наскільки дозволяє їхня база знань.

Короткий перелік усіх переваг та недоліків, виявлених в процесі порівняння методів та аналізу літератури [11] надання рекомендацій описаний у табл. 1.3.

Таблиця 1.3 – Переваги і недоліки методів надання рекомендацій

Метод рекомендації	Переваги	Недоліки
Гібридний	– Дозволяє вирішити проблеми окремих методів.	– Додаткові налаштування для комбінації моделей, необґрунтовані пропозиції.
Заснований на вмісті	– Немає необхідності у знанні предметної області, – якість рекомендацій зростає з часом, – прозорість рекомендацій, – може використовувати неявний зворотній зв'язок.	– Проблема холодного старту для нових користувачів, – якість залежить від накопичених історичних даних, – складність зміни смаків з часом, проблема бульбашки фільтрів, – складність рекомендацій мультимедійного контенту.

Продовження таблиці 1.3

Спільна фільтрація	<ul style="list-style-type: none"> <li>– Пропонування нового крос-жанрового контенту,</li> <li>– незалежність від контенту,</li> <li>– якість рекомендацій зростає з часом,</li> <li>– може використовувати неявний зворотній зв'язок.</li> </ul>	<ul style="list-style-type: none"> <li>– Проблема холодного старту для нових користувачів,</li> <li>– проблема холодного старту для нових продуктів,</li> <li>– проблема «сірої вівці»,</li> <li>– якість залежить від накопичених історичних даних,</li> <li>– погана масштабованість,</li> <li>– складність зміни смаків з часом, більше рекомендуються популярні продукти.</li> </ul>
Заснований на знаннях	<ul style="list-style-type: none"> <li>– Відсутність проблем холодного старту,</li> <li>– чутливість до змін смаків,</li> <li>– може включати не лише характеристики продуктів,</li> <li>– може відображати потреби користувача в продуктах.</li> </ul>	<ul style="list-style-type: none"> <li>– Статичні рекомендації,</li> <li>– необхідне внесення і обробка знань.</li> </ul>

Також дані методи були проаналізовані за переліком критеріїв, серед яких є точність надання рекомендацій (перелік товарів повинен якомога краще співпадати з побажаннями користувача), швидкість пропонування (оскільки найчастіше рекомендаційні системи являють собою веб-сервіси, то час виконання запиту повинен бути досить малим), можливість комбінування з розпізнаванням зображень або наявність технологій, що її використовують, та вартість обчислення (даний критерій впливатиме на властивості технічного забезпечення, яке повинні мати компанії для використання методів). Для критеріїв «Точність», «Швидкість» та «Вартість обчислень» буде виставлена оцінка за десятибальною шкалою (де 10 – найкращий результат, 1 – найгірший), а критерій «Розпізнавання

зображень» може мати значення «так» або «ні». Результати аналізу описано в табл. 1.4.

Таблиця 1.4 – Аналіз методів надання рекомендацій за обраними критеріями

Метод рекомендації	Точність	Швидкість	Розпізнавання зображень	Вартість обчислень
Заснований на вмісті	8	5	так	5
Заснований на знаннях	6	8	ні	8
Спільна фільтрація	9	4	так	4
Гібридний	9	3	так	3

Для подальшого дослідження та удосконалення обрано гібридну модель, що складатиметься з методів спільної фільтрації та заснованих на вмісті, нейтралізуючи недоліки і доповнюючи один одного.

### 1.7 Розпізнавання зображень

Зображення продуктів можуть відігравати надзвичайно важливу роль для рекомендаційних завдань у таких областях, як мода, ресторанний та готельний бізнес, а також на платформах, пов'язаних із зображеннями, таких як Flickr та Instagram. Їх можна використовувати для підвищення привабливості рекомендованих елементів на додаток до точності. Для додання зображень у рекомендаційні методи необхідно спочатку визначити їх характеристики за допомогою розпізнавання зображень.

Розпізнавання зображень – підрозділ технологій комп'ютерного зору, який дозволяє ідентифікувати сутності на зображенні та відносити їх до певних категорій. Дана технологія використовується в таких сферах життя, як медицина, виробництво, електронна комерція, транспорт, освіта, соціальні мережі, сільське господарство, інтернет речей та ін. Серед сервісів, що використовують розпізнавання зображень, можна виділити такі:

- Im2calories – додаток від Google, що обчислює кількість калорій на фотографії їжі;
- FaceApp – додаток, який розпізнає обличчя на фото та дозволяє накладати візуальні ефекти, наприклад, змінювати вік, зачіску, наносити макіяж та ін;

– Google Lens – потужний додаток, що надає корисну інформацію за об'єктом на фото, може знаходити інформацію про тварин чи рослин чи шукати предмети у онлайн-магазинах.

Під час розпізнавання зображень виконуються такі завдання:

- виявлення об'єктів (виділення об'єктів, розташованих на зображенні);
- ідентифікація зображень (визначення природи всього зображення або об'єктів на ньому);
- класифікація зображень (визначення категорії, до якої відноситься зображення, при чому воно може мати лише одну категорію).

Існують два основних підходи до розпізнавання зображень: традиційний підхід комп'ютерного зору та підхід машинного навчання. Традиційний підхід являє собою послідовність фільтрації, сегментації, виділення ознак і класифікації на основі правил. Однак він вимагає налаштування великої кількості параметрів та потребує багато інженерного часу.

Підхід глибокого навчання навпаки, в останні роки досяг великого прогресу у точності, адаптивності та продуктивності завдяки появі потужного обладнання та використанню графічних процесорів. Даний вид розпізнавання зображень знаходить приховану інформацію із навчальних екземплярів і потребує лише знання інструменту машинного навчання та значно менше навчальних зразків, проте всі вони повинні бути анотовані як позитивні або негативні. Загальний алгоритм побудови нейронної мережі для розпізнавання зображень має такі фази: вибір і попередня обробка навчальних даних, вибір топології, вибір характеристик, вибір параметрів навчання, навчання, перевірка якості, корегування та вербалізація.

Серед нейронних мереж, що використовуються для розпізнавання зображень, важливе місце посідають згорткові нейронні мережі, оскільки вони надають точніші результати більш високої точності, дозволяють зменшити вимоги до обчислювальної потужності і обробляти зображення великого розміру.

### 1.7.1 Згорткові нейронні мережі

Згорткові нейронні мережі (Convolutional neural networks, CNN, ConvNet), є спеціалізованим типом штучних нейронних мереж, які мають глибоку архітектуру прямого поширення і дивовижну здатність до узагальнення, вони можуть вивчати

високоабстраговані особливості об'єктів, особливо просторових даних, і можуть їх ідентифікувати більш ефективно в порівнянні з іншими мережами з повнозв'язними шарами [12]. Хоча дана технологія була запропонована французьким вченим Яном ЛеКуном ще у 1989 р, вона набула поширення після перемоги AlexNet на конкурсі ImageNet у 2012 р [13] значно знизивши рекорд помилок класифікації з 26.2% до 15.3%.

Після досягнутого успіху моделі CNN почали застосовуватися для задач класифікації відео та зображень, розпізнавання обличчя, виявлення об'єктів, аналізу документів, обробки природної мови, аналізу медичних зображень, прогнозу фінансових часових рядів і у безлічі інших сфер. Сьогодні CNN широко використовуються у найбільших високотехнологічних світових компаній, таких, як Google, Amazon, Facebook та інших.

### 1.7.2 Операція згортки

Архітектура даного типу нейронних мереж заснована на використанні в деяких шарах математичної операції під назвою згортка замість звичайного множення матриць. Згортка – це спеціалізований вид лінійної операції над двома функціями (наприклад,  $f$  і  $g$ ), результатом якої є інтеграл  $h$ , що виражає величину накладання однієї функції ( $f$ ) у міру її переміщення над іншою функцією ( $g$ ). Офіційно вона описується за формулами 1.11 або 1.12 [14]:

$$h(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau, \quad (1.11)$$

$$h = f * g. \quad (1.12)$$

Згорткові нейронні мережі зазвичай працюють з двовимірною операцією згортки, яка зображена на рисунку 1.1.

Відповідно до термінології CNN аргументи згортки мають такі назви: перший аргумент (функція  $f$ ) – вхідний, другий (функція  $g$ ) – ядро або фільтр, а вихід іноді називають картою ознак. У програмах машинного навчання вхід зазвичай є багатовимірним масивом (тензором) даних, а ядро – це зазвичай тензор параметрів, адаптованих алгоритмом навчання [15].

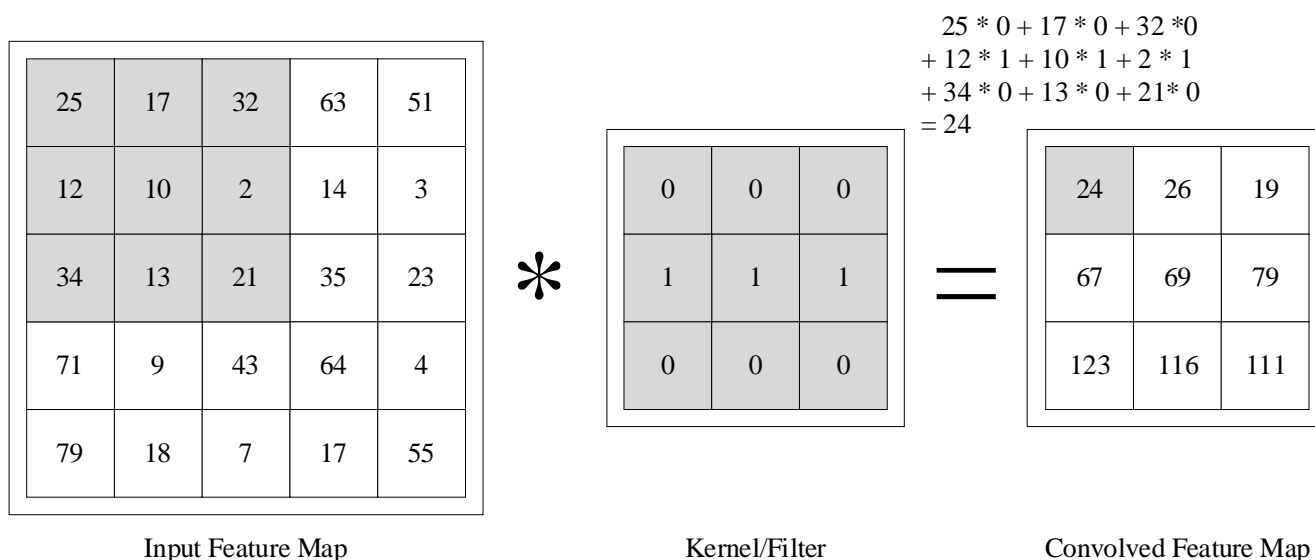


Рисунок 1.1 – Ілюстрація операції згортки

Ядро, переміщуючись над двовимірним зображенням, виконує поелементну операцію множення над тими даними, над якими воно розташоване, а потім обчислює значення нового пікселя як знаходить суму отриманих добутків. Таким чином на виході створюється матриця, розмір якої безпосередньо залежить від вибору розміру ядра, а елементи якої відповідають деякому фрагменту вхідного зображення.

### 1.7.3 Структура згорткової нейронної мережі

Модель глибокої CNN складається з кінцевого набору шарів виділення ознак, які можуть вивчати різні особливості вхідних даних (наприклад, зображення) з кількома рівнями абстракції. Початкові шари вивчають і витягують ознаки високого рівня (з нижчою абстракцією), а більш глибокі шари вивчають і витягують ознаки низького рівня (з вищою абстракцією).

Зазвичай робота шару виділення ознак в CNN складається з трьох етапів (рис. 1.2) [15]. На першому етапі паралельно виконуються декілька операцій згортки для отримання набору лінійних активацій. На другому етапі (етапі детектора або активації) кожна лінійна активація проходить через нелінійну функцію активації, якою можуть бути сигмоїд, tanh, ReLU (або її підвиди: ReLU з шумом, нещільна, параметрична та ін.). Останній етап використовує функцію агрегації (pooling function) для подальшої модифікації вихідних даних шару.

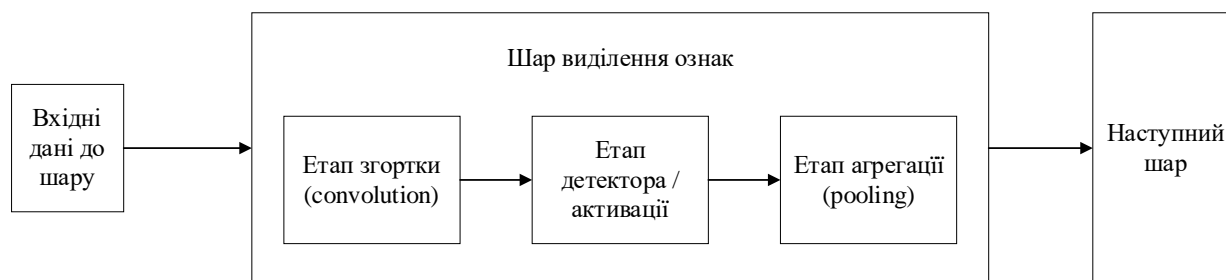


Рисунок 1.2 – Склад шару згорткової нейронної мережі

Етап детектора або активації виконується після кожного етапу згортки з метою підвищення нелінійних властивостей моделі та нейронної мережі в цілому, і управління нелінійним відображенням входів і виходів шарів. Важливою особливістю функції активації є те, що вона повинна бути диференційованою, щоб уможливити зворотне поширення помилки для навчання моделі. У минулому найчастіше на цьому етапі застосовувались гіперболічний тангенс або сигмоїдна функція, але дослідження [16] виявили, що ReLU значно збільшує ефективність обчислень та швидкість навчання, не надаючи суттєвих втрат в точності. Сьогодні на етапі активації застосовується ReLU (функція 1.13) до всіх вхідних значень. Відповідно до неї, обсяг залишається колишнім, якщо  $x > 0$ , інакше осікаються непотрібні деталі в каналі шляхом заміни на 0:

$$f(x)_{ReLU} = \max(0, x). \quad (1.13)$$

Шари агрегації використовуються для формування підвибірки карт ознак, створених за операції згортки, тобто у ньому виконується стиснення карт ознак. Ці шари дозволяють, поступово зменшуючи розмір зображення, знижувати кількість параметрів та обчислень в мережі, а також протидіють перенавчанню.

При зменшенні карт ознак даний шар завжди зберігає найбільш домінуючі об'єкти (або інформацію) на кожному етапі пулу. Операція агрегації виконується шляхом визначення розміру об'єднаної області та кроку операції, подібно до операції згортки. Існують різні типи методів агрегації, які використовуються на різних рівнях агрегації, наприклад, максимізаційне агрегування, мінімізаційне агрегування, усереднювальне агрегування, агрегування із закритим доступом, агрегування дерев тощо. Максимізаційне агрегування є найпопулярнішим і найчастіше використовуваним методом агрегування.

Під час агрегації вхідне зображення ділиться на набір прямокутників, що не перетинаються, а результат отримується шляхом запису відповідного значення з групи (так, за максимізаційного агрегування – максимального значення). Вважається, що положення ознаки серед інших є важливішим за її точні координати, тому даний шар може знижувати загальну продуктивність CNN.

Один або декілька останніх шарів згорткової нейронної мережі завжди є повнозв'язними (fully connected), де кожен нейрон всередині шару пов'язаний з кожним нейроном з попереднього шару. В них створюється N-мірний вектор з вхідних даних, де N – число класів, з яких необхідно обрати потрібний. Кожне число в цьому векторі представляє ймовірність конкретного класу. Повнозв'язний шар мережі CNN проходить через власний процес зворотного розповсюдження для визначення найбільш точних ваг. Кожен нейрон отримує ваги, які визначають пріоритет найбільш підходящого класу. Нарешті, нейрони «голосують» за кожен клас, і переможець цього голосування – рішення про класифікацію. Як правило, останній повнозв'язний шар містить функцію активації softmax, яка виводить значення ймовірності від 0 до 1 для кожної з класифікаційних міток, які модель намагається передбачити.

Рисунок 1.3 ілюструє структуру згорткової нейронної мережі.

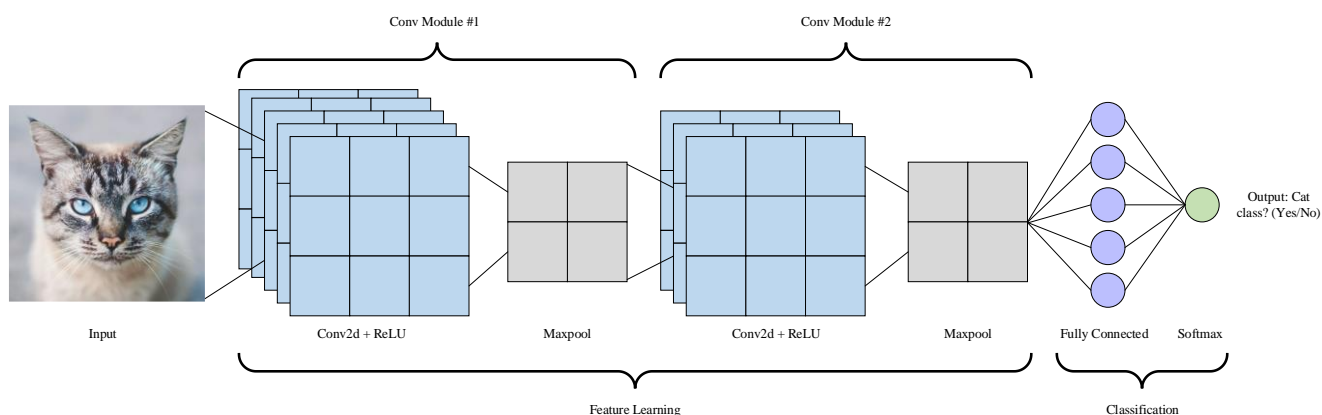


Рисунок 1.3 – Загальна схема згорткової нейронної мережі

#### 1.7.4 Додаткові техніки

З метою зменшення обчислювальної вартості іноді використовують stride (крок). Суть методу полягає у тому, що ядро переміщається уздовж вхідної

матриці з кроком, відмінним від одиниці, пропускаючи деякі дані. Таким чином, матриця ознак додатково зменшується в розмірі. Для розрахунку кількості вихідних значень після операції згортки можна використовувати формулу 1.14:

$$n_{out} = \text{floor} \left( \frac{n_{in} - f}{s} \right) + 1, \quad (1.14)$$

де  $n_{in}$  – кількість вхідних пікселів;

$f$  – кількість пікселів у фільтрі (ядрі);

$s$  – страйд.

Для акцентування додаткової уваги на ознаках, що знаходяться біля меж зображення, або для збільшення вхідного зображення, може бути використана padding (паддінг, підкладка). Її використання впливає на збільшення розміру вихідної карти ознак.

Існують три варіанти встановлення паддінгу:

– same convolution: до результату згортки додається рамка з «помилковими» пікселями, значення яких зазвичай дорівнює нулю (такий паддінг називають «нульовим» (zero padding));

– valid convolution (дійсна згортка): ядро може знаходитися лише в тих позиціях, де воно повністю вміщується на зображенні, проте розмір виходу зменшується на кожному шарі в залежності від величини ядра;

– full convolution (повна згортка): до виходу додається нульовий паддінг так, щоб кожен піксель було пройдено  $k$  разів у будь-якому напрямку, в результаті шириною зображення становить  $m + k - 1$ .

Найбільш точними зазвичай є системи, в яких використовується same чи valid convolution.

### 1.7.5 Переваги та недоліки згорткових нейронних мереж

Детально розглянувши технологію виділення ознак та класифікації зображення за допомогою згорткових нейронних мереж можна виділити їх сильні та слабкі сторони. Перевагами CNN є:

– значна точність розпізнавання зображень;

- чітка залежність виходу моделі від ознак оскільки навчання шарів вилучення ознак та класифікації проводяться разом;
- використання методу зворотного поширення помилки для навчання;
- зручність паралельних обчислень та навчання на графічних процесорах;
- простота реалізації порівняно з іншими нейронними мережами.

Недоліком CNN в свою чергу є велика кількість змінних параметрів мережі, які необхідно визначати емпірично через відсутність чітких рекомендацій до створення згорткових нейронних мереж. До цих параметрів можна віднести кількість шарів, розмірність ядра згортки для кожного з шарів, крок зсуву ядра при обробці шару, функція по зменшенню розмірності та інші.

Суттєвим недоліком CNN сприйнятливість навіть до незначних спотворень вхідних даних, тобто, якщо до зображення навмисно додають шум, то мережі помітно втрачають точність класифікації. Зображення із незначними змінами мають назву змагальних прикладів (*adversarial examples*).

Також слід вказати, що CNN можуть розпізнавати зображення в термінах кластерів пікселів, які розташовані в чітких шаблонах, і не розуміють їх як компоненти, які присутні в зображенні. Зображення, які візуалізуються CNN, не мають жодного внутрішнього уявлення компонентів та їх зв'язків частина-ціле, що ускладнює класифікацію у випадку зміни системи координат зображення.

Незначними недоліками є уповільнення через використання агрегування, необхідність наявності потужного графічного процесора для складних мереж та великої вибірки даних для навчання.

## 1.8 Використання розпізнавання зображень в рекомендаційних системах

Традиційно алгоритми, які використовуються в рекомендаційних системах, засновані на текстових даних. Однак зображення відіграють вирішальну роль у наданні рекомендацій у таких сферах, як мода, готельний та ресторанний бізнес, а також для платформ, пов'язаних з зображеннями, таких як Instagram або Pinterest.

Однак варіанти використання методів розпізнавання зображень для створення рекомендацій добре розкриті в сучасних дослідженнях .

Так, наприклад в [17] описується рішення задачі пошуку картин з урахуванням не тільки їхньої семантики, а й переваг користувача. Автори статті запропонували рішення у вигляді подвійної моделі глибокої нейронної мережі,

яка перетворює зображення та інтереси користувача в єдиний семантичний простір. Оскільки мережа обробляє уявлення і про зображення, і про користувачів, дані гібридні уявлення дозволяють приймати рішення про рекомендації. Більше того, для навчання мережі використовується порівняльне глибоке навчання (Comparative deep learning – CDL): вивчаються відносні відстані між двома зображеннями: «позитивним» – яке зацікавило користувача та «негативним» – яке користувач проігнорував. Таким чином у процесі навчання використовуються три підмережі: дві для зображень та одна для користувача. Порівняно із Social embedding Image Distance Learning (SIDL), цей підхід забезпечує вигравш у середньому на 42,58% за точністю та 46,50% за влучністю і повнотою.

Серед досліджень комбінації методів розпізнавання об'єктів та складання рекомендацій сфері електронної комерції одним з найбільш важливих досліджень стало [18], в якому був представлений підхід на базі матричної факторизації, суттю якого є включення візуальних ознак продуктів до системи прогнозу інтересів користувачів для подолання проблеми холодного старту. Даний метод використовує Bayesian Personalized Ranking (BPR) із стохастичним градієнтним підйомом для навчання моделі із використанням візуальних ознак продуктів.

Велике значення має дослідження візуальної інформації про ресторани [19], а саме фотографій у блогах, оскільки надає приклад гібридної рекомендаційної системи з використанням методів, заснованих на вмісті та спільної фільтрації.

Серед застосувань розпізнавання зображень у сфері туризму особливо виділяється дослідження R. Kitamura та T. Itoh [20]. Запропонована ними система складається з чотирьох компонентів: виділення ключових слів, візуалізація графу, користувальницький інтерфейс для вибору цілей подорожей та запит туристичних місць. Для виділення ключових слів використовується Microsoft Computer Vision API, що визначає на зображеннях об'єкти, живі істоти, пейзаж та події і присвоює їм теги з величиною впевненості від 0 до 1. На їх основі будується граф, вузли якого відповідають векторам впевненості для ключового слова в зображеннях, а дуги поєднують вектори у випадку, якщо внутрішніх добуток між векторами є більшим, за встановлене значення. Взаємодія із графом відбувається за допомогою запитів та візуального інтерфейсу, відображає кластери фотографій і ключових слів на базі графу. Ключове слово, що відповідає вибраному вузлу,

використовується як слово запиту для пошуку туристичного місця, який формується через Google Places API.

Особливу увагу методам розпізнавання зображень та наданню рекомендацій на їх основі у сфері одягу та моди. Так, наприклад, Yujie Lin [21] запропонував власний тип архітектури нейронної мережі для моди, який складається з двох кодерів (для верхньої частини образу і для нижньої), які визначають візуальні характеристики із зображень. Потім, завдяки використанню механізму взаємної уваги (mutual attention mechanism), вони трансформуються в приховані уявлення, після чого узгоджувальний декодер, який являє собою багатосаровий перцептрон (MLP), оцінює оцінку відповідності між верхньою частиною і нижньою. Декодер генерації являє собою блокований рекурентний блок, який використовується для перекладу комбінації прихованого представлення верхньої частини та прихованого представлення нижньої частини в послідовність слів у вигляді коментарів.

Окрім описаних сфер, рекомендація з використанням візуальних ознак зображень є актуальною для соціальних мереж, орієнтованих на контент, а не на користувача. У своєму дослідженні, Xue Geng [22] для вирішення проблеми розрідженості даних та різноманітності мультимедійного контенту пропонує нову структуру глибокого навчання у вигляді деревоподібної ієрархії, де листами є користувачі та зображення. Алгоритм використовує мережу соціального кураторства, вхідними даними якої є посилання користувач-продукт, а вихідними – вектори прихованих відображень користувачів і продуктів. При цьому вектори пов'язаних користувачів і товарів є ближчі, за непов'язані. В ієрархії кожна функція, яка не є листом, кодує інформацію про взаємодії користувачів і продукти. Створена глибока модель може компактно та ефективно вивчати репрезентативні функції, щоб виявити слабкі кореляції між зображеннями та користувачами на місці надзвичайно розріджених зв'язків і надзвичайно різноманітних зображень завдяки своїй глибинній структурі.

Розглянуті методи мають свої переваги, такі як часткове подолання проблеми холодного старту та ефективність для великих розмірів даних, проте мають і свої проблеми, наприклад труднощі оптимізації або складність адаптації до зміни смаків користувачів з часом.

## 1.9 Постановка задачі

Нехай  $U = \{u_1, u_2, \dots, u_n\}$  – множина користувачів системи, а  $V = \{v_1, v_2, \dots, v_m\}$  – множина наявних продуктів в системі, які можуть бути запропоновані користувачеві. Кількість користувачів  $n$  та кількість продуктів  $m$  в системі можуть бути достатньо великими (декілька тисяч або мільйонів). Кожен продукт  $v_j$  представлений у вигляді зображення  $v_j^i$  і вектора текстових характеристик  $v_j^t = \{v_1^t, v_2^t, \dots, v_k^t\}$ .

Нехай  $h$  – функція, яка відображає корисність деякого продукту  $v$  для деякого користувача  $u$  [23]:

$$h: U \times V \rightarrow R, \quad (1.15)$$

де  $R$  – повністю впорядкована множина (наприклад, цілі невід’ємні чи дійсні числа в певному діапазоні).

Тоді, для кожного користувача  $u \in U$  необхідно знайти такий продукт  $v' \in V$ , який би дозволив отримати максимальну корисність для користувача. Математично задачу знаходження найбільш релевантного продукту можна описати у вигляді:

$$\forall u \in U, v'_u = \arg \max_{v \in V} h(u, v). \quad (1.16)$$

У простих рекомендаційних системах, що базуються на явному зворотному зв'язку, функція корисності  $h$  представлена у вигляді оцінки користувачем певного продукту за детермінованою шкалою. Проте значення функції  $h$  може не лише визначатися кінцевими користувачами, а й розраховуватися програмно.

Тому задачею дослідження є виявлення такої функції корисності, яка б виявилася ефективнішою за наявні аналоги.

## 2 РОЗРОБКА МЕТОДУ РЕКОМЕНДАЦІЙ З ВИКОРИСТАННЯМ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ

### 2.1 Дослідження методів розв'язання задачі

#### 2.1.1 Дослідження методів розпізнавання зображень

За останній час була представлена велика кількість варіантів архітектури згорткових нейронних мереж. Архітектура моделі є критичним фактором для підвищення продуктивності додатків і сервісів на її базі. З 1989 року до сьогодні в архітектурі CNN були внесені різноманітні модифікації: структурне переформування, регуляризація, оптимізація параметрів тощо. Однак ключове підвищення продуктивності CNN відбулося значною мірою завдяки реорганізації процесорних блоків, а також розробці нових блоків. Зокрема, найновіші розробки в архітектурі CNN були виконані на використанні глибиних нейронних мереж.

Для огляду і дослідження було обрано декілька видів архітектури, що навчалися на тестовому наборі даних ImageNet: AlexNet, VGG, GoogLeNet, Inception-V3 та ResNet. У табл. 2.1 наведено їх короткий опис [24].

Таблиця 2.1 – Коротка характеристика CNN архітектур

Назва	Функціональні відмінності	Глибина	Відсоток помилок	Розмір вхідних даних	Рік
AlexNet	Використання Dropout, ReLU	8	16.4	227×227×3	2012
VGG	Збільшена глибина, малий розмір фільтра	16,19	7.3	224×224×3	2014
GoogLe-Net	Збільшена глибина, концепція блоку, інший розмір фільтра, концепція конкатенації	22	6.7	224×224×3	2015
Inception-V3	Використовує малий розмір фільтрів, краще представлення функцій	48	3.5	229×229×3	2015

Продовження табл. 2.1

ResNet	Надійний проти перенавчання завдяки пропускним посиленням на основі відображення симетрії	152	3.57	224×224×3	2016
--------	---	-----	------	-----------	------

AlexNet – архітектура, яка стала переможцем в 2012 році в ImageNet Large Scale Visual Recognition Challenge (ILSVRC 2012) [13], здійснивши прорив у галузях розпізнавання та класифікації зображень. Порівняно з LeNet, першою згортковою нейронною мережею, AlexNet має більшу глибину та реалізує декілька стратегій оптимізації параметрів. Зокрема, архітектура AlexNet складається з восьми шарів, включаючи п'ять згорткових шарів з ядрами розміром  $11 \times 11$  для першого шару,  $5 \times 5$  для другого шару та  $3 \times 3$  для інших шарів, за якими слідують три повністю повнозв'язні шари. Великий розмір фільтрів на початкових шарах значно покращив продуктивність попередніх мереж. Функція активації ReLU використовується на виході кожного згорткового та повнозв'язного шару, за винятком останнього повнозв'язного шару, який передає свій вихід у функцію softmax для остаточної класифікації. AlexNet має значне значення для всіх останніх поколінь CNN, оскільки розпочав інноваційну еру використання згорткових нейронних мереж у програмних додатках.

Після поширення технології згорткових нейронних мереж, К. Simonyan та А. Zisserman запропонували простий та ефективний принцип проектування CNN для широкомасштабного розпізнавання зображень. VGG має багато версій, таких як VGG-11, VGG-13, VGG-16 і VGG-19, основна відмінність між якими полягає в кількості шарів. На відміну від попередніх нейронних мереж, у шарах згортки VGG використовується ядро розміром  $3 \times 3$  замість  $5 \times 5$ ,  $7 \times 7$  або  $9 \times 9$ . Причиною такого рішення є експериментальне підтвердження однакової ефективності фільтрів малого і великого розміру, а також зменшення кількості параметрів та обчислювальної складності. Архітектура VGG отримала значущі результати для проблем локалізації та класифікації зображень і стала відомою завдяки своїй збільшеній глибині, однорідній топології та простоті. Однак обчислювальні витрати VGG є надмірними через використання близько 140 мільйонів параметрів, що є її основним недоліком.

GoogLeNet [25], також відома як Inception-V1 – це архітектура глибокої нейронної мережі, запропонована Google, також відома як Inception-v1, яка

отримала перемогу у ILSVRC 2014 року. Для досягнення високої точності та зниження витрат на обчислення в даному підході була запропонована концепція початкового блоку або модуля (inception block). Цей блок поєднує багатомасштабні згорткові перетворення, використовуючи функції злиття, перетворення та розбиття для вилучення ознак. Початковий блок у GoogLeNet розроблено шляхом поєднання таких шарів:

- шару згортки  $1 \times 1$ ;
- послідовно з'єднаних шару згортки  $1 \times 1$  та шару згортки  $3 \times 3$ ;
- послідовно з'єднаних шару згортки  $1 \times 1$  та шару згортки  $5 \times 5$ ;
- послідовно з'єднаних шару максимізаційного агрегування  $3 \times 3$  та шару згортки  $1 \times 1$ .

Глибина GoogLeNet становить 22 шари, серед яких 9 – початкові блоки. Використання початкових блоків дозволило зменшити кількість параметрів так, що їх в 12 разів менше, ніж у AlexNet з 8 шарами.

Недоліками GoogLeNet є гетерогенна топологія, яка вимагає адаптації одного модуля до іншого, а також стискання представлення, що може призводити до втрати цінної інформації.

Christian Szegedy запропонував нову версію мережі Inception під назвою Inception-V3 [26]. Додаткова мінімізація витрат обчислення в ній виконується за рахунок використання асиметричних малих фільтрів ( $1 \times 5$  та  $1 \times 7$ ) замість великих ( $5 \times 5$  та  $7 \times 7$ ), а також завдяки доданню шару згортки  $1 \times 1$  до фільтрів великого розміру.

Архітектура ResNet (Residual Network) стала відомою завдяки перемозі у ILSVRC в 2015 [27]. Існують п'ять версій архітектури ResNet для ImageNet: ResNet-18, ResNet-34, ResNet-50, ResNet-101 і ResNet-152. Де  $x$  у ResNet- $x$  відповідає кількості шарів.

Проблема деградації виникає у глибоких нейронних мереж з великою кількістю шарів, коли спочатку точність значно збільшується, а потім різко погіршується. Для вирішення цієї проблеми було запропоноване рішення, яке базується на залишкових блоках (residual blocks), які реалізують з'єднання швидкого доступу (shortcut connection) суть яких полягає у пропуску одного або кількох шарів та виконання зіставлення ідентифікаторів. Таким чином ResNet дозволяє покращити точність із збільшенням глибини.

Провівши порівняння згорткових нейронних мереж, можна зробити висновок, що збільшення шарів мережі не завжди призводить до збільшення точності результатів, тому пропонується обрати для реалізації мережу VGG19, яка має відносно невелику кількість шарів, і при цьому має достатню точність.

### 2.1.2 Дослідження методів надання рекомендацій

Виходячи з того, що для реалізації обрано методи спільної фільтрації, то для аналізу відібрано декілька методів цього типу, а саме FM (Factorization Machine), IBR (Image-based Recommendation), CML (Collaborative Metric Learning) та VBPR (Visual Bayesian Personalized Ranking). Важливою особливістю, з урахуванням якої вибиралися методи порівняння і дослідження, є можливість обробляти візуальну інформацію чи вектори параметрів товарів.

Факторизаційна машина (FM) — це загальна модель навчання з вчителем, яка відображає довільні реальні ознаки в низьковимірний латентний факторний простір і може застосовуватися для різноманітних завдань прогнозування: регресії, класифікації та формування рейтингу. Перевагою FM є її точність в оцінці параметрів моделі за дуже розріджених даних і лінійна складність навчання, що дозволяє масштабувати її до дуже великих наборів даних — ці характеристики роблять FM моделями, які ідеально підходять для вирішення реальних проблем рекомендацій. У моделях FM взаємодії користувача з продуктами мають вигляд кортежів векторів дійсних значень і числових цільових змінних. Цей метод може бути пристосований до використання характеристик зображень [28].

Рекомендації, засновані на зображеннях (IBR) – модель, запропонована J. McAuley [29], яка вивчає візуальний простір і отримує стилістично подібні елементи до зображення запиту. Потім прогнозування виконується пошуком найближчого сусіда у вивченому візуальному просторі.

Спільне метричне навчання (CML) – модель, яка була винайдена Cheng-Kang Hsieh у 2017р. [30], яка використовує спільну метрику користувача-продукту, для формування рекомендацій не лише на базі уподобань користувачів, але й використовуючи подібність користувача-користувача та продукту-продукту, а також надає набір методів регуляризації, об'єднання функцій і навчання. Переходячи від явного зворотного зв'язку до неявного зворотного зв'язку, фокус

спільної фільтрації в даній моделі перемістився з оцінки певного набору рейтингів до фіксації відносних уподобань користувачів щодо різних елементів.

Візуальний байєсівський персоналізований рейтинг (VBPR) – підхід, розроблений R. Не та J. McAuley у 2015 році [18]. Суттю якого є створення моделей, які включають візуальні функції до завдання персоналізованого рейтингування на наборах даних неявного зворотного зв'язку для подолання проблеми холодного старту та підвищення якості рекомендацій. Методологічно візуальні характеристики продуктів виділяються за допомогою глибинної нейронної мережі, поверх якої встановлено додатковий шар, який розкриває як візуальні, так і приховані розміри, які відповідають вподобанням користувачів. Для отримання рекомендацій використовується комбінація матричної факторизації, візуальних характеристик та процедури навчання на основі байєсівського персоналізованого рейтингу (BPR).

Виходячи з аналізів ефективності методів у дослідженнях [18], [30], визначено, що методи спільного матричного навчання та знаходження візуального байєсівського рейтингу є найбільш ефективними. Тому ці методи обрано для побудови удосконаленого алгоритму гібридної фільтрації.

## 2.2 Метод спільного метричного навчання

Визначення відстані між даними є одним із ключових підходів серед фундаментальних алгоритмів машинного навчання, таких як k-найближчих сусідів, k-середніх та методу опорних векторів (Support Vector Machine, SVM). Алгоритми навчання використовують відстань для виявлення важливих взаємозв'язків між даними у різних задачах: класифікації зображень, пошуку інформації або прогнозування функції білків.

Нехай  $X = \{x_1, x_2, \dots, x_n\}$  – сукупність даних у вхідному просторі  $R^m$ . Метрика є математичним записом відстані між будь-якими точками у метричному просторі, що має вигляд функції  $d(x_i, x_j)$ . Прикладами метрик можуть бути Евклідова метрика, Мангеттенська метрика, дискретна відстань або відстань Махаланобіса.

Залежно від набору даних, метричне навчання вирішує одну з двох задач:

- навчання з учителем (якщо дані мають інформацію про належність до певного класу, як у задачі класифікації, мета в даному випадку – вивчити метрику

відстані, яка розташовує близько точки одного класу і далеко – різних);

– слабо контрольоване навчання (якщо дані представлені у вигляді кортежів по 2-4 точки, наприклад, як подібні (позитивні) (2.1) або неподібні (негативні) пари (2.2), у такому разі мета – вивчити метрику відстані, яка призначає менші відстані між подібними парами, а великі – між неподібними на основі їх характеристик).

$$S = \{(x_i, x_j) \mid x_i \text{ та } x_j \text{ вважаються подібними}\}, \quad (2.1)$$

$$D = \{(x_i, x_j) \mid x_i \text{ та } x_j \text{ вважаються неподібними}\}. \quad (2.2)$$

Метрика відстані мусить мати такі характеристики:

- невід'ємність:  $d(x_i, x_j) \geq 0$  та  $d(x_i, x_j) = 0$  якщо  $x_i = x_j$ ,
- нерівність трикутника:  $d(x_i, x_j) \leq d(x_i, x_k) + d(x_j, x_k)$ ,
- симетричність:  $d(x_i, x_j) = d(x_j, x_i)$ .

Особливе значення має нерівність трикутника, оскільки вона сприяє процесу поширення подібності – поширенню інформації з пар, подібність яких відома, до тих, зв'язки яких невідомі. Так, за наявності інформації « $x_i$  подібний до обох точок  $x_j$  та  $x_k$ », метрика не тільки зближить зазначені пари  $(x_i, x_j)$  та  $(x_i, x_k)$ , а й наблизить до них  $(x_j, x_k)$ .

Оригінальний підхід метричного навчання використовує метрику Махаланобіса (2.3), за якої задача оптимізації має вигляд (2.4)

$$d_A(x_i, x_j) = \sqrt{(x_i - x_j)^T A (x_i - x_j)}, \quad (2.3)$$

$$\min_A \sum_{(x_i, x_j) \in S} d_A(x_i, x_j)^2, \quad (2.4)$$

$$\text{за умови } \sum_{(x_i, x_j) \in D} d_A(x_i, x_j)^2 \geq 1 \text{ та } A \geq 0,$$

де  $A \in R^{m \times n}$  – позитивна напіввизначена матриця (зворотня до коваріаційної).

Поняття поширення подібності тісно пов'язане із спільною фільтрацією. Під час спільної фільтрації розглядається матриця корисності, що містить зв'язки між парами (користувач, предмет). Метричне навчання завдяки поширенню подібності має можливість виявлення нових зв'язків між користувачами та предметами.

Принцип роботи методу спільного метричного навчання [30] полягає у поєднанні технології метричного навчання із спільною фільтрацією для надання списку з  $k$  найкращих товарів за явними та неявними даними про вподобання користувачів. Ідея даного підходу полягає в моделюванні спостережуваного неявного зворотного зв'язку у вигляді набору позитивних пар користувач-продукт  $S$  та навчанні спільної метрики для користувача  $i$  продукту для кодування їх відносин.

Нехай кожен користувач (user) представлений в системі вектором користувача  $u_i \in R^r$ , а кожен продукт (item) – вектором продукту  $v_j \in R^r$ . В процесі навчання евклідова відстань (2.5) залежить від відношення користувача щодо різних продуктів, тобто продукт, який сподобався користувачеві, буде знаходитися ближче до нього, ніж інші продукти, які йому не сподобалися.

$$d(i, j) = \|u_i - v_j\|. \quad (2.5)$$

Нерівність трикутника сприяє об'єднанню користувачів, яким подобаються одні й ті ж продукти та продукти, які подобаються декільком користувачам. Таким чином найближче до окремого користувача будуть знаходитися продукти, що сподобались йому в минулому та продукти, що сподобалися іншим користувачам із схожим смаком.

Функція втрати формально позначається як:

$$L_m(d) = \sum_{(i,j) \in S} \sum_{(i,k) \notin S} w_{ij} [m + d(i, j)^2 - d(i, k)^2]_+, \quad (2.6)$$

де  $i$  – користувач;

$j$  – продукт, що сподобався користувачеві;

$k$  – продукт, що не сподобався користувачеві;

$[z]_+ = \max(z, 0)$  – стандартна завісна втрата;

$w_{ij}$  – вага втрати рангу;

$m > 0$  – розмір безпечної відстані.

На рис. 2.1 зображено градієнти, що виникають через описану завісну функцію втрати. Градієнти елементів, які подобаються користувачеві, рухаються до нього, зменшуючи радіус. Предмети, що не подобаються користувачеві, проте

які потрапили в його радіус, рухаються назовні доки не вийдуть за безпечну відстань.

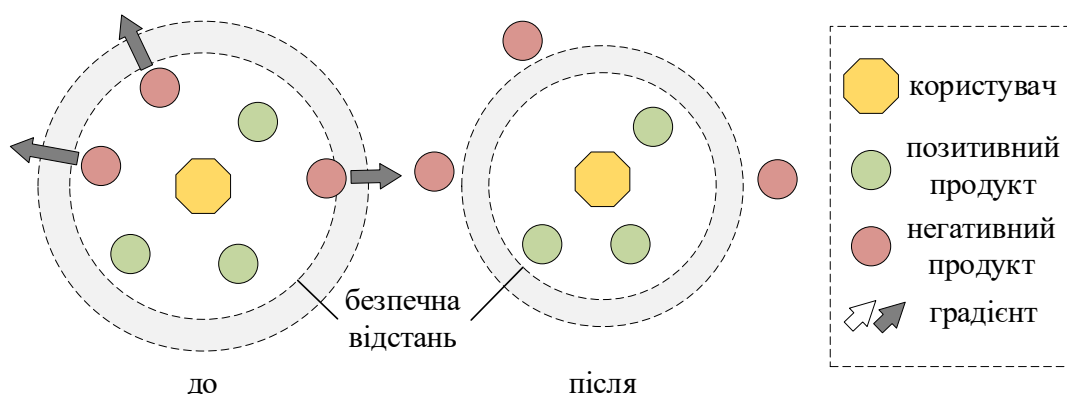


Рисунок 2.1 – Робота градієнтів у спільному метричному навчанні

Для персоналізації продуктів з низьким рангом для користувача використовується рангова схема зважування з назвою зважена попарна втрата приблизного рангу (Weighted Approximate-Rank Pairwise, WARP), запропонована J. Weston [31]. За наявної метрики  $d$  позначимо  $J$  як загальну кількість продуктів, а  $rank_d(i, j)$  – ранг продукту  $j$  в рекомендаціях користувача  $i$ . Штрафування позитивних продуктів з низьким рангом відбувається за формулою:

$$w_{ij} = \log(rank_d(i, j) + 1). \quad (2.7)$$

Для зменшення часу та складності обчислень використовується паралельна процедура випробування, яка відбирає негативні продукти доки не знайде ті, що знаходиться в радіусі користувача. Вона складається з таких кроків:

- для кожної пари  $(i, j)$  паралельно відбираються негативні продукти  $U$  (кількість продуктів зазвичай обмежується 10 або 20 для збереження часу), для яких розраховується завісна функція втрати (2.6);

- нехай  $M$  – кількість негативних продуктів в радіусі користувача в  $U$  зразках, тоді  $rank_d(i, j)$  апроксимується як  $\left\lfloor \frac{J \times M}{U} \right\rfloor$ .

Спільне матричне навчання при створенні рекомендацій може використовувати характеристики продуктів у вигляді як тестових даних (теги, текстовий опис), так і пікселі зображення. Нехай  $x_j \in R^m$  – це  $m$ -вимірний вектор характеристик продукту  $j$ ,  $f$  – функція трансформації, яка проектує  $x_j$  у спільний

простір користувач-продукт. Оскільки  $f(x_j)$  повинна відповідати характеристикам продукта  $j$ , то вона штрафується, якщо розташування  $j$  у просторі  $(v_j)$  суттєво відрізняється від  $f(x_j)$ . Функція L2-втрати має вигляд:

$$L_f(\theta, v_*) = \sum_j \|f(x_j, \theta) - v_j\|^2. \quad (2.8)$$

Дана функція змінюється під час навчання одночасно з мінімізацією метричної втрати. У підході для відображення даної функції обрано багатошаровий перцептрон (multi-layer perceptron, MLP) через його репрезентативну здатність та легкість навчання.

Для запобігання перенавчання використовується коваріаційна регуляризація [32], яка окрім зменшення кореляції між активаціями в нейронній мережі може бути корисною для декореляції розмірів у метричному навчанні. Нехай  $y^n$  – прихований вектор користувача чи продукта,  $n$  – індекси об'єкта в пакеті розміру  $N$ . Матриця коваріацій знаходиться за формулою 2.9,  $\mu_j$  за – 2.10, а втрата для регуляризації  $L_c$  – за 2.11.

$$C_{ij} = \frac{1}{N} \sum_n (y_i^n - \mu_i)(y_j^n - \mu_j), \quad (2.9)$$

$$\mu_i = \frac{1}{N} \sum_n y_i^n, \quad (2.10)$$

$$L_c = \frac{1}{N} (\|C\|_f - \|\text{diag}(C)\|_2^2). \quad (2.11)$$

Загальна цільова функція колаборативного метричного навчання може бути сформована як

$$\min_{\theta, u_*, v_*} L_m + \lambda_f L_f + \lambda_c L_c, \quad (2.12)$$

$$\text{за умови } \|u_*\|^2 \leq 1 \text{ та } \|v_*\|^2 \leq 1,$$

де  $\lambda_f, \lambda_c$  – гіперпараметри, що контролюють вагу кожної частини втрати.

Процедура навчання даного методу складається з таких кроків:

- відбір  $N$  позитивних пар з  $S$ ;
- відбір  $U$  негативних продуктів для кожної пари та апроксимація рангу;
- збереження негативного продукта  $k$  для кожної пари, який максимізує завісну функцію втрати;
- розрахунок градієнтів та оновлення параметрів за допомогою AdaGrad;
- цензор норми  $u_*$  та  $v_*$  для  $y' = \frac{y}{\max(\|y\|, 1)}$
- повторення до зближення.

Наприкінці,  $k$  продуктів з найвищим рангом, рекомендовані користувачеві, відповідатимуть його  $K$ -найближчим в отриманому просторі.

Завдяки поєднанню метричного навчання, матричної факторизації та векторів ознак, даний алгоритм став одним з найшвидших та найточніших сучасних алгоритмів спільної фільтрації і може використовуватися для широкого кола рекомендаційних завдань.

### 2.3 Метод знаходження візуального байєсівського персоналізованого рейтингу

Метою цього методу [18] також є комбінування інформації про явні та неявні уподобання користувача з візуальними характеристиками продуктів для зниження впливу проблеми холодного старту, пояснення рекомендацій з погляду візуальних особливостей продуктів, а в результаті – складання персоналізованих рейтингів, які більш точно відповідають перевагам користувачів.

В основі даного методу лежить підхід матричної факторизації (MF), який пов'язує користувачів і продукти шляхом знаходження прихованих вимірювань таких, що користувачі та продукти, що їх високо оцінюють, представляються схожим чином. Однак класична MF стикається з труднощами при врахуванні нерозглянутих продуктів і використанні неявного зворотного зв'язку, тому даний метод є модифікацією парних методів, ідея яких полягає у сприйнятті позитивного зв'язку трохи кращим, ніж нерозглянутого. Rendle St.[33] запропонував структуру узагальненого байєсівського персоналізованого рейтингу (BPR), яка з MF як основний предиктор (BPRMF) використовується як базовий метод для додавання візуальних характеристик.

Вилучення візуальної інформації в VBPR проводиться за допомогою глибоких нейронних конволюційних мереж (Deep CNN), оскільки вони справляються як із завданнями виділення об'єктів на зображенні, так і визначення візуального стилю.

Нехай  $U$  – множина користувачів системи, а  $I$  – множина товарів, при чому кожен користувач  $u$  асоціюється з  $I_u^+$  – множиною продуктів, позитивний зворотній зв'язок користувача  $u$  про які відомий. Також кожен продукт  $i \in I$  характеризується одним зображенням. Метою методу є генерація персоналізованого рейтингу для користувача  $u$ , який складається з товарів, зв'язок з якими для нього невідомий ( $I/I_u^+$ ).

Предиктор вподобань, побудований на базі MF з використанням неявного зворотного зв'язку, формує модель для прогнозування відношення користувача  $u$  щодо продукту  $i$  за формулою:

$$\hat{x}_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u^T \gamma_i + \theta_u^T \theta_i, \quad (2.13)$$

де  $\alpha$  – глобальне зміщення (скаляр),

$\beta_u, \beta_i$  – зміщення користувача  $u$ , продукту  $i$  (скаляр),

$\gamma_u, \gamma_i$  – приховані фактори користувача  $u$ , продукту  $i$  (вектори  $K \times 1$ ),

$\theta_u, \theta_i$  – візуальні фактори користувача  $u$ , продукту  $i$  (вектори  $D \times 1$ ).

Добуток  $\gamma_u^T \gamma_i$  кодує «сумісність» між користувачем  $u$  та продуктом  $i$ , тобто ступінь, до якої приховані «уподобання» користувача узгоджені з «властивостями» продуктів, тоді як добуток  $\theta_u^T \theta_i$  моделює візуальну взаємодію між  $u$  та  $i$ , тобто ступінь, до якої користувача  $u$  приваблює кожен із  $D$  візуальних вимірів.

Для того, щоб не перевантажувати систему векторами візуальних характеристик безпосередньо з CNN (які найчастіше є 4096-мірними) та не втрачати інформацію методами зниження розмірності (такими, як PCA), запропоновано навчання ядра вектора, яке лінійно трансформує високорозмірний вектор ознак у простір візуальних ознак значно меншої розмірності. Отримані значення прогнозованих розмірів можна інтерпретувати як ступінь, з якою продукт демонструє певний аспект візуальної характеристики. Зниження розмірності виконується за формулою:

$$\theta_i = E f_i, \quad (2.14)$$

де  $E$  – матриця  $D \times F$  перетворення вектора ознак ( $F$ -розмірного) у візуальний простір (розмірності  $D$ ),

$f_i$  – оригінальний вектор ознак для продукту  $i$ .

Для моделювання загальної думки користувачів щодо вигляду візуального елемента було введено вектор візуальне зміщення  $\beta'^T f_i$ , що є добутком вектора візуального зміщення  $\beta'$  на вектор ознак  $f_i$ . У підсумку модель прогнозу має вигляд:

$$\hat{x}_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u^T \gamma_i + \theta_u^T (E f_i) + \beta'^T f_i. \quad (2.15)$$

Тренування моделі проводиться за допомогою байєсівського персоналізованого ранжування (BPR) – попарної системи оптимізації рейтингу, яка використовує стохастичний градієнтний підйом як процедуру навчання. Набір даних  $D_s$  представляє собою триплети типу  $(u, i, j)$ , де  $u$  відповідає користувачеві,  $i$  – продукту, про який  $u$  надав позитивний зворотній зв'язок,  $j$  – продукту, який ще переглянутий користувачем

$$D_s = \{(u, i, j) \mid u \in U \wedge i \in I_u^+ \wedge j \in I \setminus I_u^+\}. \quad (2.16)$$

Нехай  $\Theta$  – параметризований вектор, а  $\hat{x}_{uij}(\Theta)$  – довільна функція  $\Theta$ , що параметризує відношення між компонентами триpletу  $(u, i, j)$ . Для персонального ранжування використовується такий критерій оптимізації:

$$\sum_{(u,i,j) \in D_s} \ln \sigma(\hat{x}_{uij}) - \lambda_{\Theta} \|\Theta\|^2, \quad (2.17)$$

де  $\lambda_{\Theta}$  - гіперпараметр регуляризації, специфічний для моделі,

$\sigma$  – логістична (сигмовидна) функція:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.18)$$

При використанні матричної факторизації (BPRMF) як предиктора вподобань,  $\hat{x}_{uij}$  визначається за формулою 2.19, де  $\hat{x}_{u,i}$  та  $\hat{x}_{u,j}$  розраховуються за формулою 2.15

$$\hat{x}_{uij} = \hat{x}_{u,i} - \hat{x}_{u,j}. \quad (2.19)$$

Навчання BPRMF проводиться за допомогою стохастичного градієнтного підйому. Спочатку триплет  $(u, i, j)$  обирається з  $D_S$ , а потім алгоритм навчання оновлює параметри наступним чином:

$$\Theta \leftarrow \Theta + \eta \cdot (\sigma(-\hat{x}_{uij}) \frac{d\hat{x}_{uij}}{d\Theta} - \lambda_{\Theta} \Theta), \quad (2.20)$$

де  $\eta$  – темп навчання.

У VBPR в процесі навчання оновлюються два набори параметрів: невізуальні параметри та нещодавно введені візуальні параметри. Невізуальні параметри оновлюються аналогічно до моделі BPRMF, тоді як візуальні за формулами:

$$\begin{aligned} \Theta_u &\leftarrow \Theta_u + \eta \cdot (\sigma(-\hat{x}_{uij}) E(f_i - f_j) - \lambda_{\Theta} \Theta_u), \\ \beta' &\leftarrow \beta' + \eta \cdot (\sigma(-\hat{x}_{uij}) (f_i - f_j) - \lambda_{\beta} \beta'), \\ E &\leftarrow E + \eta \cdot (\sigma(-\hat{x}_{uij}) \Theta_u (f_i - f_j)^T - \lambda_E E). \end{aligned} \quad (2.21)$$

де  $\lambda_E$  – гіперпараметр для упорядкування матриці  $E$ .

Експериментальні результати, наведені у [18] свідчать, що завдяки використанню візуальних характеристик із зображень продуктів, масштабування та базуючись на матричній факторизації, даний алгоритм значно перевищує найсучасніші методи надання рекомендацій та полегшує проблему холодного старту.

#### 2.4 Знаходження косинусної міри подібності

Навчання подібності [34] – це вимірювання подібності між парою зображень або об'єктів, що найчастіше застосовується у завданнях, пов'язаних з класифікацією та регресією. Прикладами програм, що використовують даний

підхід, є розпізнавання рукописного тексту, ідентифікація обличчя, пошукові системи, перевірка підпису тощо. Мета навчання подібності полягає в тому, щоб вивчити функцію подібності, яка знаходить оптимальне співвідношення між двома спорідненими або подібними об'єктами кількісним чином. Зазвичай навчання подібності передбачає надання пари зображень або текстових векторів як вхідних даних і виявлення, наскільки вони схожі один на одного. Результатом може бути невід'ємна оцінка подібності від 0 (якщо об'єкти зовсім не схожі один на одного) до 1 (якщо є ідентичними).

Косинусна міра подібності [35] вимірює подібність двох векторів внутрішнього простору продукту. Вона визначається, як косинус кута між двома векторами і з'ясовує, чи вказують два вектори приблизно в одному напрямку (рис. 2.2).

Нехай  $x = (x_1, x_2, \dots, x_n)$  та  $y = (y_1, y_2, \dots, y_n)$  – два вектори характеристик об'єктів, подібність яких необхідно визначити. Тоді сама косинусна міра подібності визначається як відношення скалярного добутку векторів  $x$  та  $y$  до добутку їх норм (довжин). При цьому, чим більше значення норм векторів, тим менше буде значення косинусної міри подібності.

$$\text{similarity}(x, y) = \cos(\theta) = \frac{x \cdot y}{\|x\| \cdot \|y\|}, \quad (2.22)$$

де  $\|x\|$  – Евклідова норма вектора  $x$ , що визначається як

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}. \quad (2.23)$$

Косинусна міра подібності є ефективним методом визначення схожих об'єктів, оскільки навіть якщо два подібних вектори знаходяться далеко один від одного за евклідовою відстанню через розмір (наприклад, якщо в одному тексті певне слово зустрічається значно частіше, ніж в іншому), вони все одно можуть мати менший кут між ними. Чим менший кут, тим вище подібність (рис. 2.2).

Для знаходження міри подібності текстових даних вони представляються у вигляді векторів частоти термінів – частоти певних слів (наприклад, ключових слів) або фраз в документі. Зазвичай частотні вектори є дуже розрідженими і традиційні методи вимірювання відстані погано працюють для таких даних, однак

оскільки косинусна міра подібності використовує для розрахунку лише ненульові елементи, то має низьку складність обчислення.

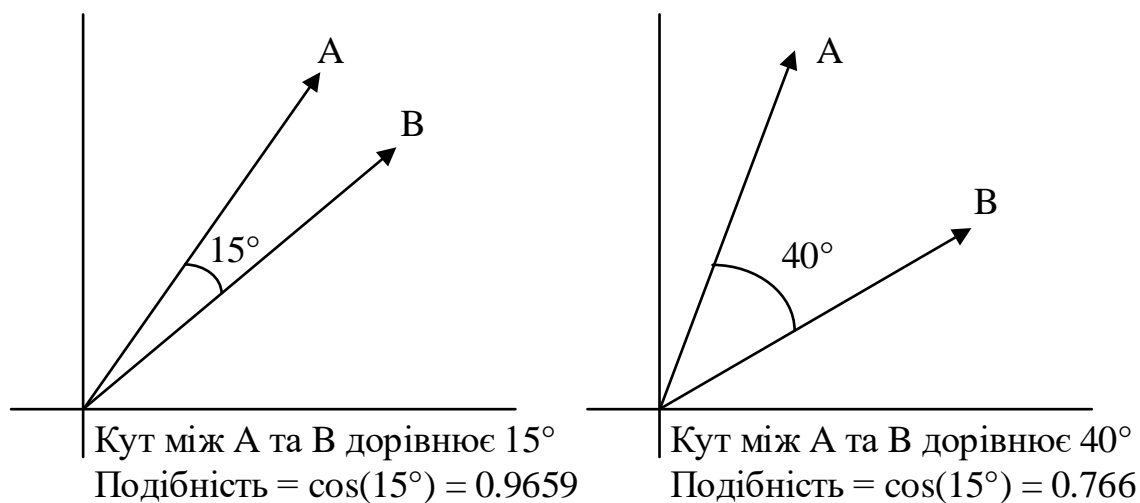


Рисунок 2.2 – Векторне зображення косинусної подібності

Для розрахунку косинусної міри подібності між зображеннями необхідно спочатку знайти їх векторні представлення за допомогою технології глибокого навчання, наприклад через CNN.

## 2.5 Нейронна мережа для визначення візуальних характеристик зображень

Для розпізнавання візуальних характеристик зображень продуктів в рекомендаційних системах була обрана нейронна мережа VGG19. VGG19 – це одна з моделей VGG, яка була опублікована у 2014 році К. Simonyan та А. Zisserman в Оксфорді [36]. Дана нейронна мережа складається з 19 послідовних шарів (з яких 16 шарів згортки, 3 повнозв'язних шарів, 5 шарів maxpool і 1 шар soft-max) (рис. 2.3). Відповідно до функціональних особливостей, мережа ділиться на дві частини: частину вилучення однак (від першого шару згортки до останнього maxpool шару) та класифікуючу (повнозв'язні шари та soft-max).

Для виявлення характеристик зображень використана VGG19, ваги якої попередньо навчені на базі зображень ImageNet. ImageNet складається з 14,197,122 зображень, класифікованих на 1000 класів (наприклад, клавіатура, миша, олівець або види тварин) за ієрархією WordNet.

На вхід нейронної мережі подається RGB зображення розміром  $224 \times 224$ , яке можна розглядати як масив пікселів  $224 \times 224 \times 3$ . Потім зображення

попередньо оброблюється відніманням середнього значення RGB, обчисленого на всій навчальній вибірці, з кожного пікселя.

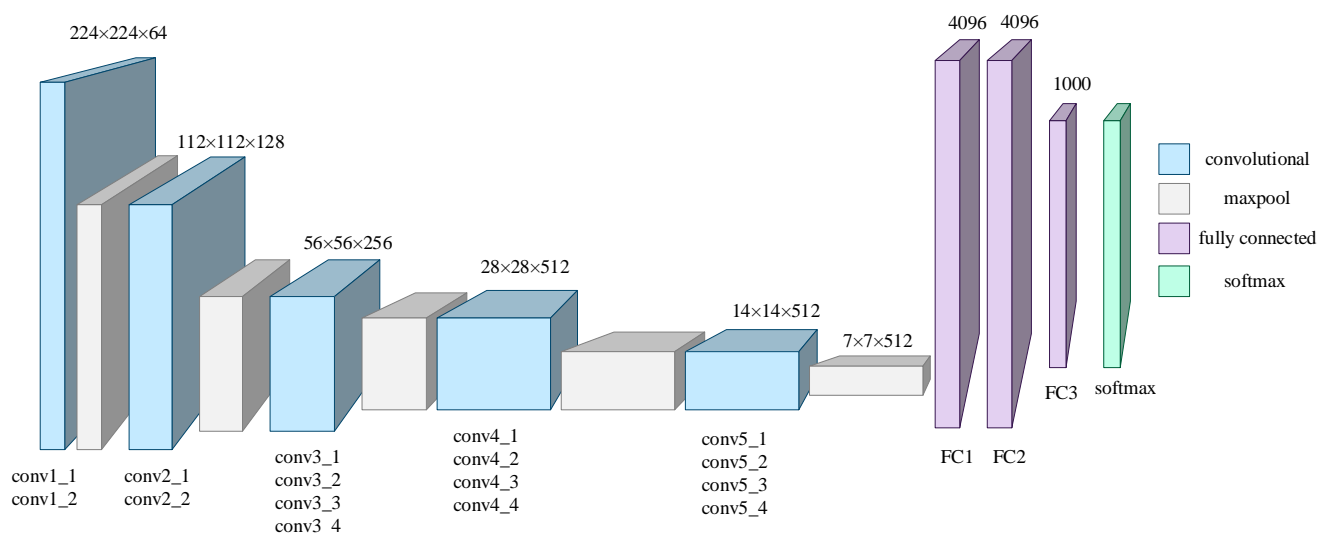


Рисунок 2.3 – Архітектура мережі VGG19

Операція згортки виконується за допомогою ядра  $3 \times 3$  з розміром кроку 1 піксель, що дозволяє охопити всі сторони зображення. Просторове об'єднання здійснюється п'ятьма шарами maxpool, які слідуєть за деякими шарами згортки. Maxpool виконується із ядром, розмір якого становить  $2 \times 2$  пікселя з кроком 2. Ширина кожної рівня згортки (кількість каналів) досить мала, вона починається з 64 у першому шарі, а потім збільшується в 2 рази після кожного шару maxpool, поки не досягне 512.

За блоками згорткових шарів знаходяться три повнозв'язні шари, перші два з яких мають по 4096 каналів кожен, а третій виконує класифікацію ILSVRC і містить 1000 каналів (один для кожного класу). Останнім шаром є шар soft-max.

В результаті можна отримати візуальні характеристики, що у початкових шарах мережі визначають багато дрібних деталей у зображенні, і кожен наступний шар надає інформацію про більш загальні характеристики зображення. Модель абстрагує ознаки із зображення на більш загальні поняття, які можна використовувати для класифікації у останніх шарах.

Відповідно до проведеного аналізу точності [36] можна зробити висновок, що VGG19 є найбільш точною та ефективною як серед інших мереж типу VGG, так і порівняно з GoogLeNet, MSRA, Clarifai, Zeiler & Fergus, OverFeat та мережею Krizhevsky.

## 2.6 Удосконалення методу надання рекомендацій з використанням розпізнавання зображень

З метою підвищення точності рекомендацій та подолання проблеми холодного старту було запропоновано використання гібридної рекомендаційної системи, яка поєднує переваги методів, заснованих на вмісті, та спільної фільтрації, а також дозволяє враховувати візуальні характеристики зображень продуктів при визначенні вподобань користувача. Алгоритм на вищому рівні абстракції зображено на рис. 2.4.

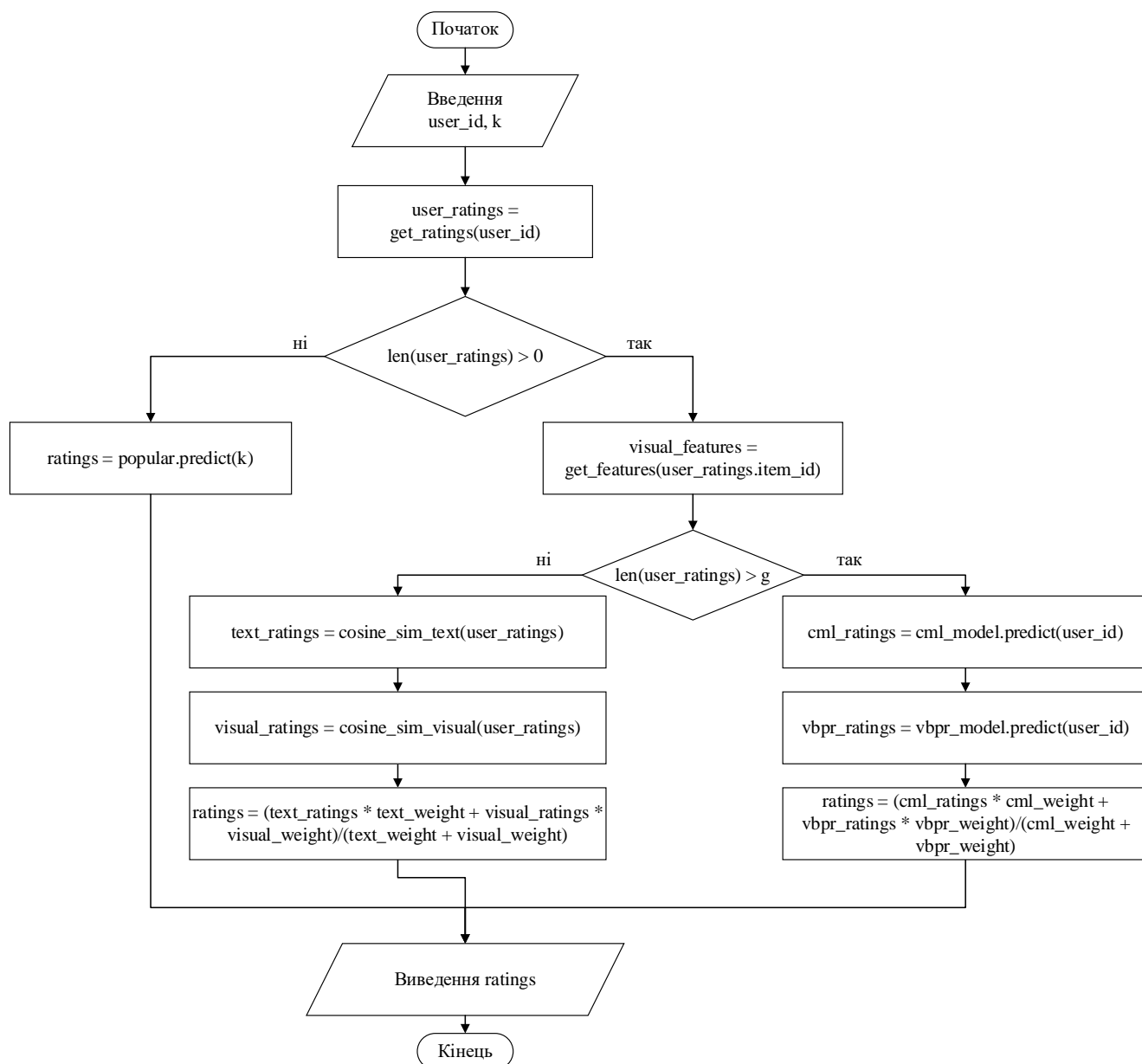


Рисунок 2.4 – Алгоритм запропонованої рекомендаційної системи

Ідея розробленого підходу полягає в застосуванні різних методів гібридизації для подолання проблеми холодного старту.

Метод перемикання застосовується для вибору алгоритму надання рекомендацій відповідно до кількості оцінених користувачем продуктів. Так, якщо користувач вперше зайшов у систему і не має оцінених товарів, йому буде запропоновано переглянути список із найпопулярніших товарів в системі, а якщо він має декілька (менше п'яти) оцінених товарів – застосовуватиметься фільтрація за змістом. Якщо ж користувач активно взаємодіяв із системою і має більше п'яти оцінених товарів – він отримає рекомендацію за допомогою методів спільної фільтрації.

Виконання методів, заснованих на вмісті, та спільної фільтрації в свою чергу використовує зважений метод гібридизації: для методів за вмістом спочатку розраховується подібність продуктів на базі текстової інформації, а потім – на базі візуальної; для спільної фільтрації поєднуються рейтинги, отримані методом спільного метричного навчання та знаходження візуального байєсівського персоналізованого рейтингу.

Перевагами даного методу є підвищена точність надання рекомендації як для користувачів із великою кількістю оцінених продуктів, так і для нових користувачів, що переглянули лише декілька з них. Недоліками є велика вартість обчислень через використання методів спільної фільтрації.

Вхідними даними до алгоритму є дані про оцінки користувачами продуктів системи, характеристики товарів, зображення товарів.

Запропонований метод можна умовно поділити на такі етапи:

- визначення алгоритму надання рекомендацій;
- знаходження рекомендацій за популярністю;
- знаходження візуальних характеристик продуктів;
- знаходження рекомендацій методами, заснованими на вмісті;
- знаходження рекомендацій методами спільної фільтрації.

На першому етапі застосовується аналіз кількості оцінок користувача, для цього значення кількості порівнюється із коефіцієнтом *rate*, який обирається емпірично шляхом аналізу предметної області рекомендаційної системи.

На етапі знаходження рекомендацій за популярністю визначається перелік товарів, що мають не менше  $R_n$  оцінок та середній рейтинг яких становить більше, ніж  $R_{min}$ , які розраховуються за формулами 2.24 та 2.25.

$$R_n = \frac{1}{10} n_r, \quad (2.24)$$

$$R_{min} = \frac{2}{n_r} \sum_{i=1}^{n_r} r_i, \quad (2.25)$$

де  $n_r = |R|$  – потужність множини оцінок продуктів,

$r_i$  – оцінка продукту користувачем.

На етапі отримання візуальних характеристик спочатку визначається модель, що використовує нейронну мережу VGG19 із вагами, створеними при її навчанні на базі зображень Imagenet, та повертає візуальні характеристики у вигляді вектору  $1 \times 4096$ , визначені у другому повнозв'язному шарі. Потім завантажуються зображення кожного продукту, конвертується до формату RGB, приводиться до масиву розміром  $224 \times 224$  та подається на вхід моделі. Результатом цього етапу є матриця  $N \times 4096$ , де  $N$  – кількість продуктів в системі.

На етапі рекомендацій методами, заснованими на вмісті, спочатку проводиться обробка текстових даних і утворення векторів, які характеризують продукти. Із множини характеристик виділяються ті, за якими буде розраховуватися схожість між продуктами. Оскільки більшість характеристик представляють собою текстові дані, то проводиться розробка якісного показника оцінки особливостей продуктів. Для цього текстова інформація позбавляється зайвих пробілів та дублікатів. Потім характеристики кожного продукту поєднуються у єдиний вектор і створюється матриця кількості маркерів (вживань кожного слова або словосполучення у векторі продукту). Для визначення схожості між зображеннями також використовується косинусна міра, проте на її вхід подаються векторні характеристики зображень, отримані за допомогою VGG19. Для розрахунку спільного рейтингу застосовується зважена сума рейтингів:

$$r_{cbi} = \frac{w_{tx} r_{tx} + w_{im} r_{im}}{w_{tx} + w_{im}}, \quad (2.26)$$

де  $w_{tx}$ ,  $w_{im}$  – коефіцієнти текстових та візуальних подібностей відповідно,  
 $r_{tx}$ ,  $r_{im}$  – значення текстових та візуальних подібностей.

Оскільки методи, засновані на вмісті, застосовуються у випадку, коли користувач оцінив декілька продуктів, тоді рейтинг  $r_{cb_i}$  знаходиться для кожного продукту окремо, а потім за формулою 2.27 визначається загальний рейтинг з урахуванням оцінок, наданих користувачем до товарів, подібні до яких знаходяться цим підходом

$$r_{cb} = \frac{r_i * r_{cb_i}}{\sum_{i=1}^{n_r} r_i}. \quad (2.27)$$

На етапі знаходження рекомендацій методами спільної фільтрації спочатку визначаються рейтинги за допомогою методу спільного метричного навчання, потім за візуальним байєсівським персоналізованим рейтингом, а після цього отримується зважена сума рейтингів за формулою 2.28.

$$r_{cf} = \frac{w_{cml}r_{cml} + w_{vbpr}r_{vbpr}}{w_{cml} + w_{vbpr}}, \quad (2.28)$$

де  $w_{cml}$ ,  $w_{vbpr}$  – коефіцієнти для рейтингів методів CML та VBPR відповідно,

$r_{cml}$ ,  $r_{vbpr}$  – значення рейтингів за CML та VBPR.

Для методів спільної фільтрації також проводиться попередня обробка даних: обирається навчальна та тестова виборки, визначаються продукти, неоцінені користувачами. Оскільки для навчання моделей зазвичай використовують великі виборки, то для оптимальної роботи з ними визначається значення таких параметрів, як кількість епох – проходжень набору даних через нейронну мережу, розмір і кількість батчів – частин набору даних, на яких відбувається навчання, а також кількість ітерацій – кількість батчів, необхідну для завершення епохи. Потім виконується визначення моделі, її параметрів та функції втрат. Після цього починається етап тренування моделей: з навчальної вибірки у випадковому порядку обираються користувачі, для яких проводиться підбір рекомендацій із застосуванням описаних параметрів.

Для оптимізації алгоритму і простоти роботи після попередньої обробки можуть зберігатися у текстовому вигляді такі дані: вектори характеристик

зображень, вектори текстових характеристик продуктів, косинусні міри подібності між продуктами.

Наприкінці користувач у будь-якому випадку отримує перелік товарів, проте він базується на його вподобаннях лише в тому разі, якщо клієнт попередньо оцінив якийсь продукт. При цьому за основу можуть братися як характеристики продуктів, так і статистична інформація про оцінки користувача.

## 3 РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

### 3.1 Опис тестових наборів даних

Для проведення тестування та оцінки ефективності запропонованого методу обрано набір даних MovieLens 100K. Цей набір даних розроблено проектом GroupLens в університеті Міннесоти. Дані були зібрані через веб-сайт MovieLens протягом семи місяців з 19 вересня 1997 року по 22 квітня 1998 року. Цей набір даних складається з 100 000 оцінок за шкалою від одного до п'яти від 943 користувачів до 1682 фільмів. При цьому кожен користувач оцінив щонайменше 20 фільмів.

Набір даних складається з файлів `u.data`, `u.info`, `u.item`, `u.genre`, `u.user`, `u.occupation`, різних варіантів розподілів на тестову і навчальну виборки та скриптів, що виконують цей розподіл.

Файл `u.info` містить загальну інформацію про кількість користувачів, фільмів та оцінок.

У файлі `u.data` знаходиться загальний набір даних, що представляє собою кортежі з ідентифікатора користувача, ідентифікатора фільму, рейтингу та часу, у форматі кількості секунд з 01.01.1970. Ідентифікатори користувачів та фільмів є послідовними числами, що починаються з одиниці.

Файл `u.item` містить детальну інформацію про фільми, таку як ідентифікатор, назву, дату випуску, дату виходу у відеопрокати, посилання на `imdb`, а також 19 полів жанрів (`unknown`, `Action`, `Adventure`, `Animation`, `Children's`, `Comedy`, `Crime`, `Documentary`, `Drama`, `Fantasy`, `Film-Noir`, `Horror`, `Musical`, `Mystery`, `Romance`, `Sci-Fi`, `Thriller`, `War`, `Western`), що містять значення 1 якщо фільм відповідає жанру та 0 – якщо ні. Перелік жанрів також продубльовано у файлі `u.genre`.

Файл `u.user` містить демографічну інформацію про користувачів, таку як ідентифікатор, вік, стать, рід занять, поштовий код. Всі можливі роди занять можна знайти у файлі `u.occupation`.

Окрім даних, що були надані у MovieLens було додатково вилучено посилання на постери, ім'я режисера та трьох головних акторів, текстовий опису фільму та перелік ключових слів шляхом парсингу сайту `imdb.com`. Ці дані необхідні для виконання фільтрації методами, заснованими на вмісті.



шкалі від одного до п'яти. Розподіл балів за взаємодію користувача з системою відображено в таблиці 3.1.

Таблиця 3.1 – Коефіцієнти для обробки неявного зворотного зв'язку

Дія користувача	Кількість балів
Перегляд сторінки фільму	1.2
Перехід за посиланням на imdb	1.5
Додання до списку «Переглянути пізніше»	2.3
Видалення із списку «Переглянути пізніше»	-2.3

Таким чином, якщо користувач перегляне сторінку фільму, перейде за посиланням до imdb та додасть фільм до списку «Переглянути пізніше», то він зможе отримати  $1.2 + 1.5 + 2.3 = 5$  балів рейтингової оцінки фільму. Також, у випадку, якщо користувач сам оцінив фільм за шкалою в системі, його неявні дані будуть замінені на явну оцінку.

### 3.2 Опис методики оцінювання результатів експерименту

Для проведення тестування ефективності вибірка даних була поділена на у співвідношенні 80%/20%, де 80% становить навчальна вибірка, а 20% - тестова.

відповідно. Таким чином на базі 80% випадково обраних оцінок продуктів відбувається навчання алгоритму, а потім виконується порівняння спрогнозованих оцінок з тестовими 20%. Даний вид поділу дозволяє отримати інформацію щодо релевантності отриманих оцінок та ефективності алгоритму в цілому.

Серед розглянутих метрик для вимірювання якості запропонованої системи обрано Precision (формула 1.4), Recall (формула 1.5) та nDCG (формула 1.9). Окрім них для оцінки будуть використані Mean average precision (MAP) та середній взаємний рейтинг (Mean reciprocal rank, MRR).

Точність (precision) і повнота (recall) — це показники оцінки, які зазвичай використовуються в налаштуваннях класифікації. У контексті рекомендаційних використовуються такі метрики, як Precision@K і Recall@K, оскільки найчастіше надається користувачеві надається перелік рекомендацій обмеженого розміру K. K найкращих продуктів визначаються шляхом обмеження кількості відповідних

продуктів між першими  $k$  елементами [37]. Так, для розрахунку Precision@K та Recall@K застосовуються формули 3.1 та 3.2:

$$\text{Precision@K} = P@K = \frac{\sum_{k=1}^K \text{rel}^{\text{true}}}{K} = \frac{|\text{релевантні серед } K \text{ кращих}|}{K}, \quad (3.1)$$

$$\text{Recall@K} = R@K = \frac{\sum_{k=1}^K \text{rel}^{\text{true}}}{|\text{rel}^{\text{true}}|} = \frac{|\text{релевантні серед } K \text{ кращих}|}{|\text{релевантні}|}. \quad (3.2)$$

Однак у списку рекомендованих продуктів перший рекомендований пункт завжди є найважливішим, потім другий є другим за важливістю і так далі. В цьому є недолік Precision@K – вона не враховує порядок продуктів у переліку.

Середня точність (Average precision, AP) може бути використана для визначення рейтингу, використовуючи значення precision від 1 до  $K$ . Для її розрахунку необхідно знайти суму Precision@K за індексами релевантних елементів. Це показано в наступній формулі:

$$\text{AP@K} = \frac{\sum_{k=1}^K \text{rel}^{\text{true}}(\pi^{-1}(k)) * P@k}{K}, \quad (3.3)$$

де  $\pi^{-1}(k)$  – продукт, що опинився на позиції  $k$  в результаті перестановки  $\pi$  ( $\pi^{-1}(k)$  відповідає продукту з найбільшою релевантністю  $\text{rel}$ , а  $\pi^{-1}(K)$  – з найменшою).

Mean average precision є однією з найпопулярніших метрик оцінки якості рекомендаційних систем, оскільки вона дозволяє оцінити значення середньої точності не лише для окремого користувача, а для їх множини (що відповідає реальній роботі системи). Для розрахунку MAP@K необхідно знайти значення AP@K для кожного користувача та поділити на кількість користувачів (3.4)

$$\text{MAP@K} = \frac{\sum_{j=1}^N \text{AP@K}_j}{N}, \quad (3.4)$$

де  $N$  – кількість користувачів системи.

Середній взаємний ранг (3.6) визначається як середнє значення взаємного рангу RR@K для всіх користувачів. Взаємний ранг дорівнює 1, поділений на

значення рангц першого релевантного пункту у списку рекомендацій (3.5). MRR може мати значення у діапазоні [0,1] та враховує позиції продуктів, проте головним недоліком цієї метрики можна визначити врахування лише першого релевантного продукту

$$RR@K = \frac{1}{\min \{k \in [1 \dots K]: rel^{true}(\pi^{-1}(k)) = 1\}}, \quad (3.5)$$

$$MRR@K = \frac{\sum_{j=1}^N RR@K_j}{N}. \quad (3.6)$$

### 3.3 Результати досліджень

Для оцінки ефективності запропонованого гібридного алгоритму виконується його порівняння з іншими описаними алгоритмами: методами спільного метричного навчання та знаходження візуального байєсівського персоналізованого рейтингу (дані підходи є складовими гібридного методу), а також із методами, що не використовують візуальні ознаки: методом нейронної матричної факторизації та байєсівського персоналізованого ранжування.

Всі гіперпараметри були підібрані таким чином, щоб надати найкращі значення ефективності для обраного тестового набору даних. Серед варіантів розміру списку рекомендацій обрано значення 5 та 10. Результати порівняння наведені у табл. 3.2.

Таблиця 3.2 – Порівняння якості рекомендаційних систем

	CML	VBPR	NeuMF	BPR	Hybrid
Precision@5	0.3525	0.3574	0.3042	0.3121	<b>0.3602</b>
Recall@5	0.1068	0.1092	0.0947	0.0997	<b>0.1152</b>
Precision@10	0.3172	0.3257	0.2601	0.2563	<b>0.3215</b>
Recall@10	0.1918	0.2000	0.1494	0.1569	<b>0.2101</b>
MAP	0.2437	0.2412	0.2021	0.2357	<b>0.2519</b>
MRR	0.5691	0.5978	0.5280	0.5905	<b>0.6001</b>
nDCG	0.5818	0.5850	0.5299	0.5770	<b>0.5891</b>
nDCG@5	0.3777	0.3781	0.3370	0.3701	<b>0.3802</b>
nDCG@10	0.3609	0.3679	0.2881	0.3565	<b>0.3637</b>

Аналізуючи значення, отримані у таблиці, можна зробити наступні висновки: по-перше, в середньому алгоритми, що використовують візуальні характеристики зображень є ефективнішими ніж ті, що орієнтуються лише на оцінки користувачів; по-друге, гібридний алгоритм перевищив незначно показники алгоритмів CML та VBPR, та суттєво – MF та BPR.

Для більш детального аналізу обрано метрики Precision@K та Recall@K, залежність яких від довжини списку рекомендацій K була розглянута для всіх реалізованих методів (рис. 3.3). Можна помітити, що запропонований підхід надає кращі результати як за точністю, так і за повнотою для всіх значень K, що демонструє ефективність гібридного алгоритму (табл. 3.3).

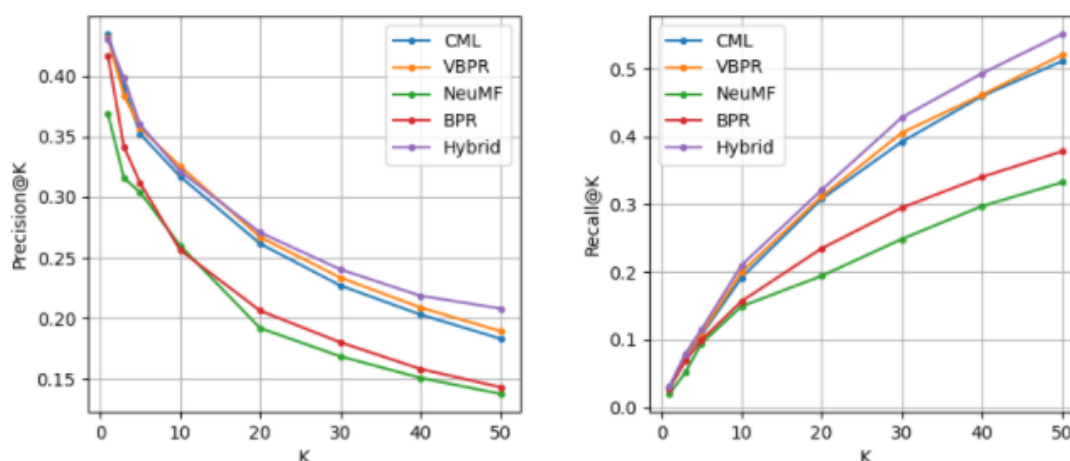


Рисунок 3.3 – Графіки залежності Precision@K та Recall@K від розміру списку рекомендацій K

Таблиця 3.3 – Порівняння часу підбору рекомендацій

	CML	VBPR	NeuMF	BPR	Hybrid
Час підбору, с	0.3525	0.3202	0.2142	0.3493	0.3517

Таким чином встановлено, що використання запропонованого алгоритму є доцільним, оскільки існує невеликий приріст у значеннях метрик оцінки якості. Недоліком алгоритму є те, що він програє методам у часі підбору рекомендацій, проте час створення списку рекомендацій для спільної фільтрації є досить великим, і приріст часу у відсотковому розрахунку щодо інших методів є незначним.

## 4 ПРОЕКТУВАННЯ ТА РОЗРОБКА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

### 4.1 UML-проектування

Уніфікована мова моделювання (UML) — це візуальна мова для моделювання та обміну інформацією про системи за допомогою діаграм і допоміжного тексту [38]. UML може бути застосована до багатьох різних областей і може охоплювати й моделювати все: від організації компанії до бізнес-процесів до розподіленого корпоративного програмного забезпечення. Вона є способом фіксації та вираження стосунків, поведінки та ідей високого рівня в простій для розуміння та ефективній; майже все в ній має графічне представлення [39].

Моделювання варіантів використання (use case) — це спеціалізований тип структурного моделювання, пов'язаний з моделюванням функціональності систем, який зазвичай застосовується під час встановлення вимог до того, що повинна робити система. Моделювання варіантів використання починається на початку проекту і триває протягом усього процесу розробки системи.

Основними елементами діаграми варіантів використання є актори та варіанти використання. Актор — зовнішня до системи сутність, яка взаємодіє з нею, це може бути людина-користувач або інша система. Актори відповідають на питання про те, хто і що взаємодіє з системою і відображуються на схемі у вигляді піктограми «фігурка». Варіант використання визначає функціональну вимогу, яка описується як послідовність кроків, які включають дії, що виконуються системою, і взаємодію між системою та акторами. Варіанти використання відповідають на питання про те, як учасники взаємодіють із системою, і описують дії, які виконує система. На діаграмі має вигляд овалу із назвою дії всередині.

Діаграма варіантів використання зображена на рис. 4.1. Основними акторами, що є неавторизований користувач, зареєстрований авторизований користувач та адміністратор (останні два актори успадковуються від неавторизованого). Всі користувачі системи можуть взаємодіяти з системою таким чином:

- зареєструватися;
- авторизуватися;

- переглянути список популярних фільмів;
- переглянути детальну інформацію про фільм;
- знайти фільм за назвою;
- перейти за посиланням до imdb сторінки фільму.

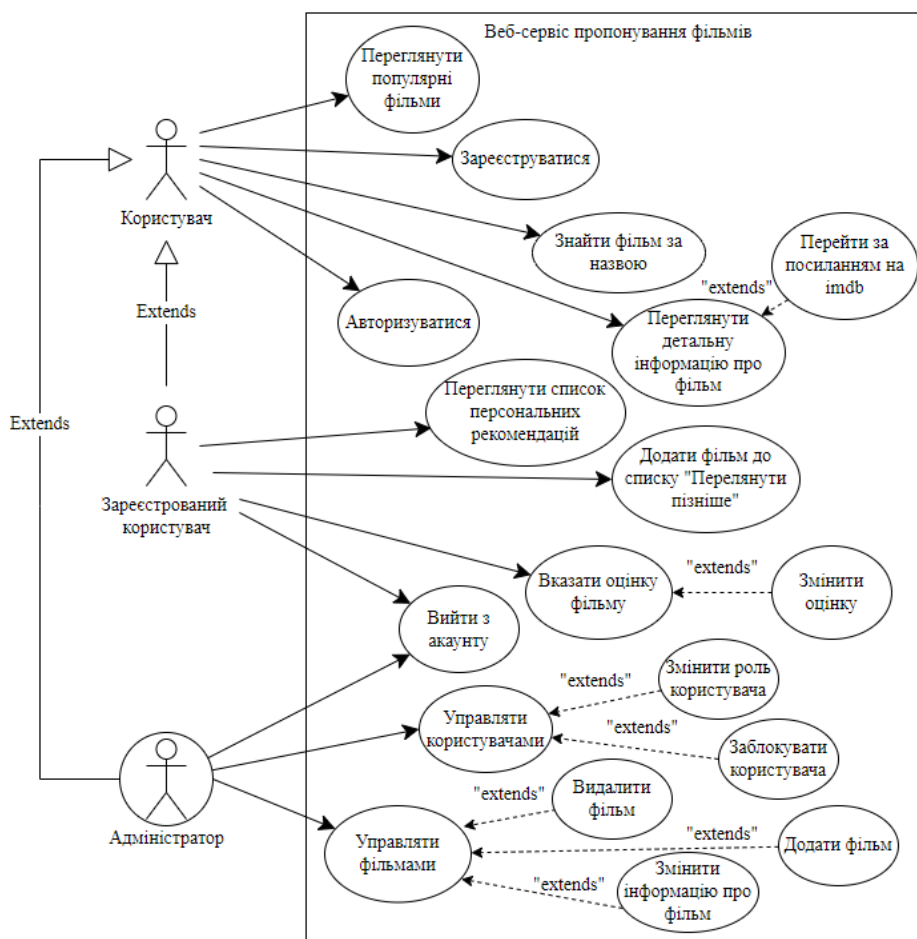


Рисунок 4.1 – Діаграма варіантів використання

Діаграми послідовності зображують динамічну поведінку елементів, які складають систему, при їх взаємодії з часом. Даний вид діаграм дозволяє встановити послідовність дій, які необхідно виконати користувачам та елементам системи для досягнення мети окремого варіанту використання. Вони організовані вздовж двох осей: горизонтальна вісь показує елементи, які беруть участь у взаємодії, а вертикальна вісь представляє час. Дані діаграми складаються з ролей класів, конкретних об'єктів, лінії життя та активацій. Класи та об'єкти позначаються прямокутниками із назвою всередині; лінія життя – вертикальна пунктирна лінія від елемента, що представляє існування впродовж часу; активація, зображається у вигляді високого тонкого прямокутника на лінії життя,

представляє період, протягом якого елемент виконує операцію (верхня частина прямокутника вирівнюється за часом початку, а нижня — за часом завершення).

Діаграми послідовності основних процесів системи, таких як отримання списку рекомендацій, додання фільму до списку «Переглянути пізніше» та виставлення оцінки фільму користувачем зображено на рис. 4.2 – 4.4.

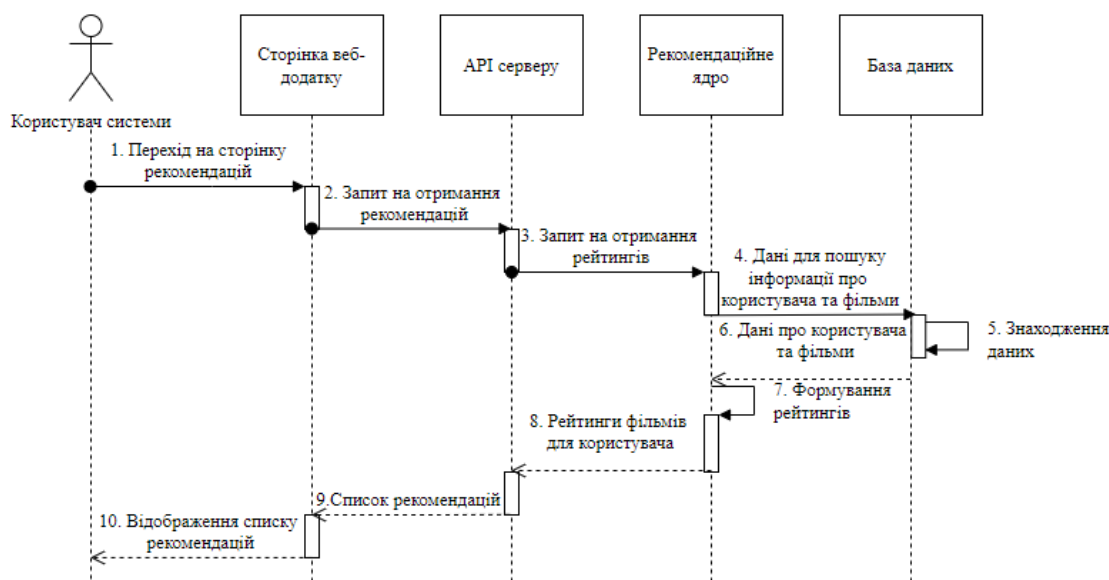


Рисунок 4.2 – Діаграма послідовності отримання списку рекомендацій

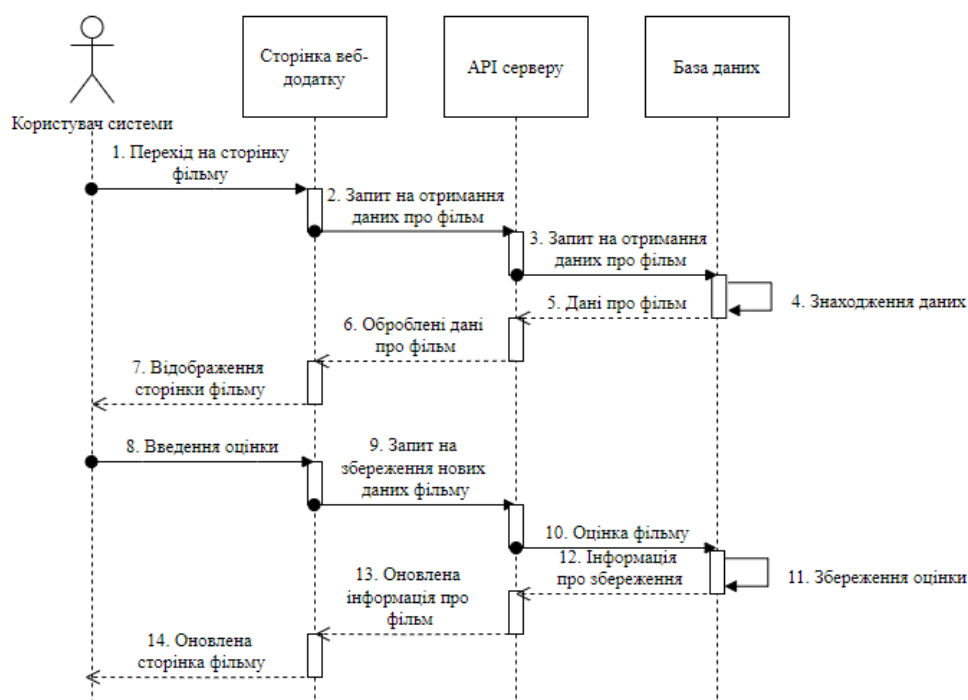


Рисунок 4.3 – Діаграма послідовності додання фільму до списку «Переглянути пізніше»

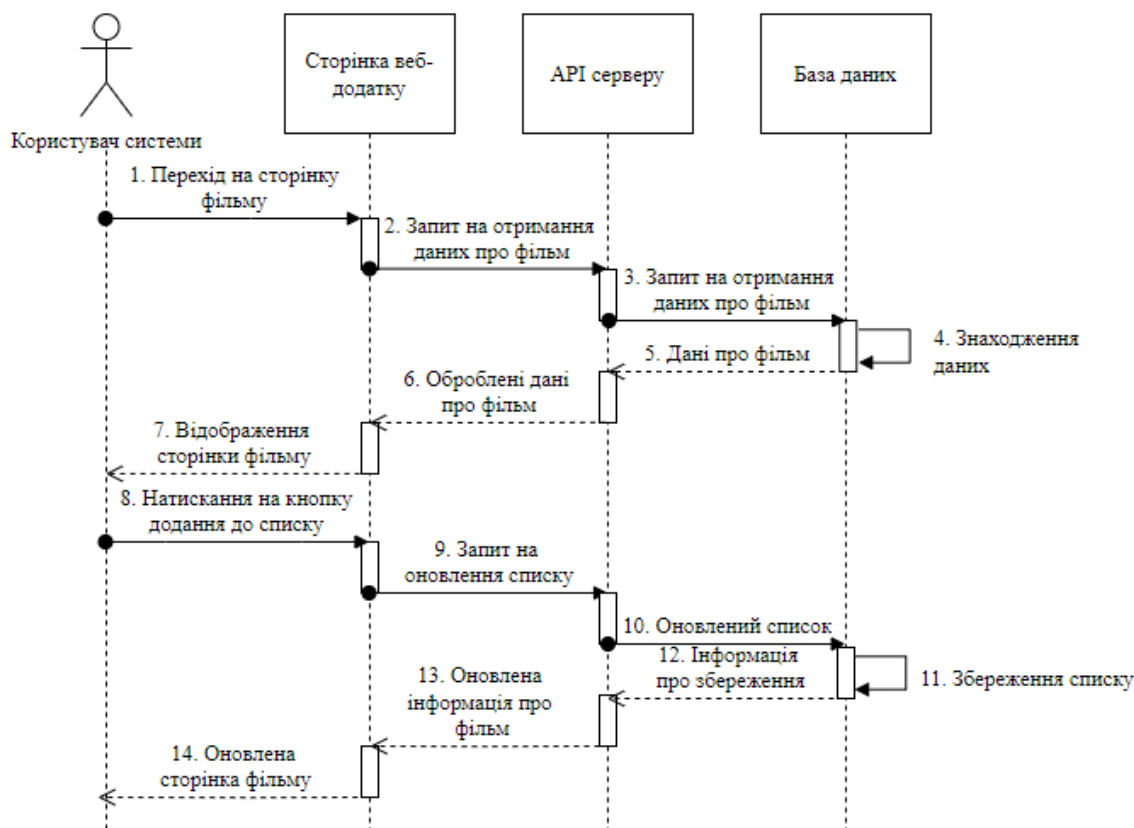


Рисунок 4.4 – Діаграма послідовності оцінювання фільму

## 4.2 Проектування бази даних

Для проектування системи було розроблено Entity-Relationship Diagram (ERD), яка використовується для специфікації інформаційних вимог, а також для уточнення інформації про систему із замовниками. ERD оперує такими поняттями, як об'єкт (сутність), зв'язок та атрибут.

Об'єкт – це важливе для системи поняття, яке проектується чи аналізується, та інформація про яке повинна бути збережена.

Зв'язок – іменована асоціація між сутностями, що має значення для предметної області. Зв'язки завжди асоціюються із подіями, що відбуваються в об'єктах, які пов'язуються.

Атрибут – це характеристика сутності, значима для предметної області і призначена для ідентифікації, класифікації, кваліфікації, кількісної характеристики або вираження стану сутності.

Під час проектування системи було розроблено діаграму «сутність-зв'язок» із головними сутностями, на яких базується робота системи на логічному (рис. 4.5) та фізичному рівні (рис. 4.6).

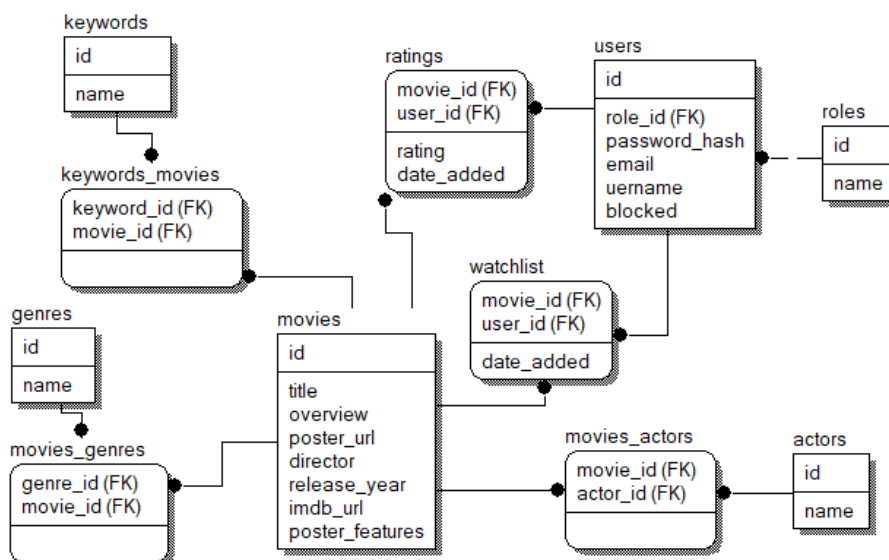


Рисунок 4.5 – Логічна схема бази даних

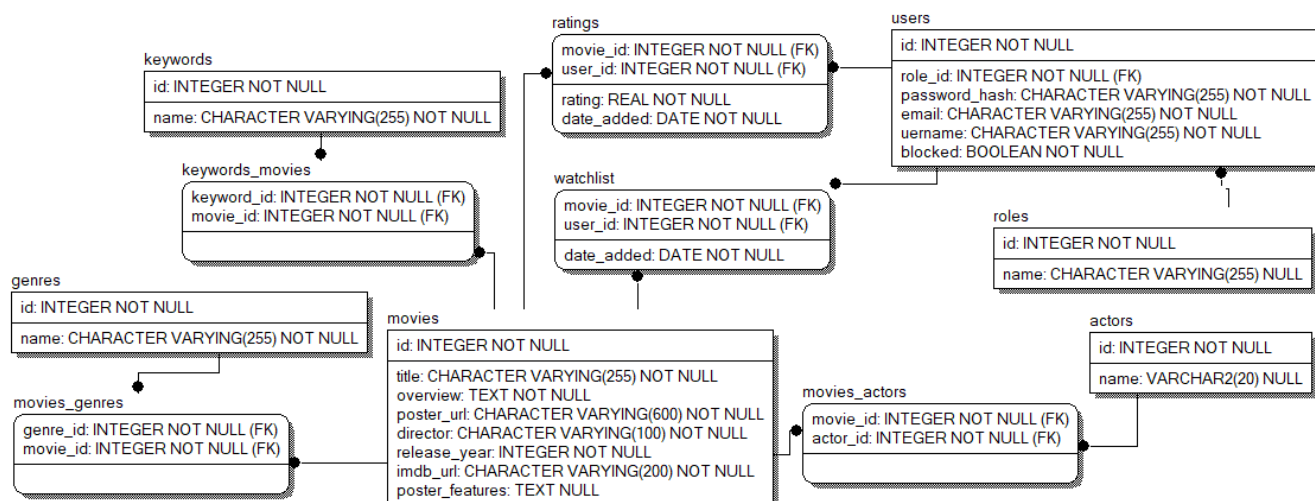


Рисунок 4.6 – Фізична схема бази даних

На даних діаграмах спроектовано такі сутності:

- «movies» зберігає інформацію про фільми, а саме унікальний ідентифікатор, назву, рік створення, опис, посилання на сторінку imdb та посилання на зображення постеру, а також ім'я режисера;
- «users» містить інформацію про користувачів: унікальний ідентифікатор, ідентифікатор ролі, закодований пароль, ім'я, поштову адресу та флаг блокування;
- «genres» – таблиця з ідентифікатором і назвою жанрів фільмів;
- «actors» – таблиця з ідентифікатором та іменами акторів;
- «keywords» – таблиця з ідентифікатором і текстів ключових слів;

- «roles» – таблиця з ідентифікатором і назвою ролей користувачів;
- «ratings» містить ідентифікатор фільму, ідентифікатор користувача, а також оцінку фільму користувачем і дату оцінювання;
- «watchlist» містить ідентифікатор фільму, ідентифікатор користувача і дату додавання фільму до списку «Переглянути пізніше» користувачем.

Додатково сгереновано таблиці, що розкривають відношення «багато-до-багатьох» для таблиць movies та keywords, movies та genres, movies та actors.

### 4.3 Обґрунтування вибору інструментів

Для програмної реалізації алгоритму формування рекомендацій з використанням розпізнавання зображень було використано мову програмування Python, Tensorflow, Keras, Pandas, Numpy, Scipy, Sklearn, PIL та matplotlib.

Python [40] — це високорівнева, універсальна і дуже популярна мова програмування. Мова програмування Python використовується у веб-розробці (завдяки фреймворкам Django та Flask), додатках машинного навчання, а також для розробки передових технологій в індустрії програмного забезпечення.

Tensorflow [41] – програмна платформа для чисельних обчислень на основі графіків потоків даних, розроблена як інтерфейс для вираження та реалізації алгоритмів машинного навчання, а здебільшого – глибоких нейронних мереж. TensorFlow був розроблений з урахуванням портативності, що дозволяє виконувати ці обчислювальні графіки в широкому діапазоні середовищ та апаратних платформ.

Keras — це високорівневий гнучкий API для створення, навчання, оцінки та запуску нейронних мереж, написаний на Python і здатний працювати разом з TensorFlow.

NumPy (Numerical Python) – одна із найвідоміших бібліотек для обчислень на Python. Вона надає структури даних, алгоритми та функціонал, необхідний для більшості наукових додатків, що включають числові дані в Python. NumPy також використовується це як контейнер для даних, які передаються між алгоритмами та бібліотеками. Серед іншого, дана бібліотека містить:

- швидкий та ефективний багатовимірний об'єкт масиву ndarray;
- функції для виконання поелементних обчислень з масивами або математичні операції між масивами;

- інструменти для читання та запису масивів даних на диск;
- операції лінійної алгебри, перетворення Фур'є та генерація випадкових чисел.

Pandas надає високорівневі структури даних і функції, призначені для швидкої та легкої та роботи зі структурованими або табличними даними. Основними об'єктами в pandas є DataFrame, таблична, орієнтована на стовпці структура даних з мітками як рядків, так і стовпців, і Series, одновимірний позначений об'єкт масиву.

Matplotlib — найпопулярніша бібліотека Python для створення графіків та інших двовимірних візуалізацій даних.

Для реалізації клієнтської та серверної частин також використано мову Python, а також фреймворк Flask, HTML, CSS, Bootstrap.

Flask — мікрофреймворк для створення веб-додатків на мові програмування Python, який використовує набір інструментів Werkzeug, та шаблонізатор Jinja2. Мікрофреймворк є мінімалістичним каркасом веб-додатку, який має лише базові можливості. Перевагами Flask є наявність серверу розробки, використання RESTful запитів, підтримка модульного тестування, файлів cookie та велика документація.

Bootstrap — це безкоштовна платформа, що має набір багаторазових фрагментів коду, написаних на HTML, CSS та JavaScript для створення адаптивних веб-додатків для різних розмірів екранів: від смартфона до персонального комп'ютера.

У якості системи управління базами даних обрана PostgreSQL. PostgreSQL — це безкоштовна розширена реляційна база даних корпоративного класу з відкритим кодом, яка підтримує як реляційні (SQL), так і нереляційні (JSON) запити. Оскільки PostgreSQL є об'єктно-реляційною СУБД, то вона підтримує складні структури та широкий спектр вбудованих і визначених користувачем типів даних. PostgreSQL пропонує своїм користувачам величезну кількість функцій, що допомагають розробникам створювати стійкі та безпечні програми, керувати великими об'ємами даних та виконувати безліч інших корисних операцій. І у підсумку дана СУБД забезпечує велику ємність і високу цілісність даних.

#### 4.4 Архітектура додатку

Для розробки запропоноване використання архітектури типу MVC («модель-представлення-контролер») (рис. 4.7) – множині архітектурних принципів, яка підтримує сувору ізоляцію між окремими частинами програми. У концепції MVC програма поділяється на три компоненти: модель, представлення і контролер.

Модель є відображає дані предметної області в програмі, відповідає за структуру додатку та реагує на запити контролера, змінюючи свій стан та сповіщаючи його. Відповідно до структури додатків з використанням Flask, модель може бути представлена блоками моделей об'єктів та бізнес-логіки, а також до цієї частини можна віднести ядро формування рекомендацій.

Роллю представлення є взаємодія з користувачем та надання йому візуальної інформації. За обраної структури, відображення буде мати вигляд шаблонів сторінок.

Контролер поєднує представлення та модель за певною логікою. Контролер отримує дані користувача через представлення і потім взаємодіє з моделлю для виконання специфічних дій, передаючи результати назад представленням. Як контролери можуть виступати відображення у Flask.

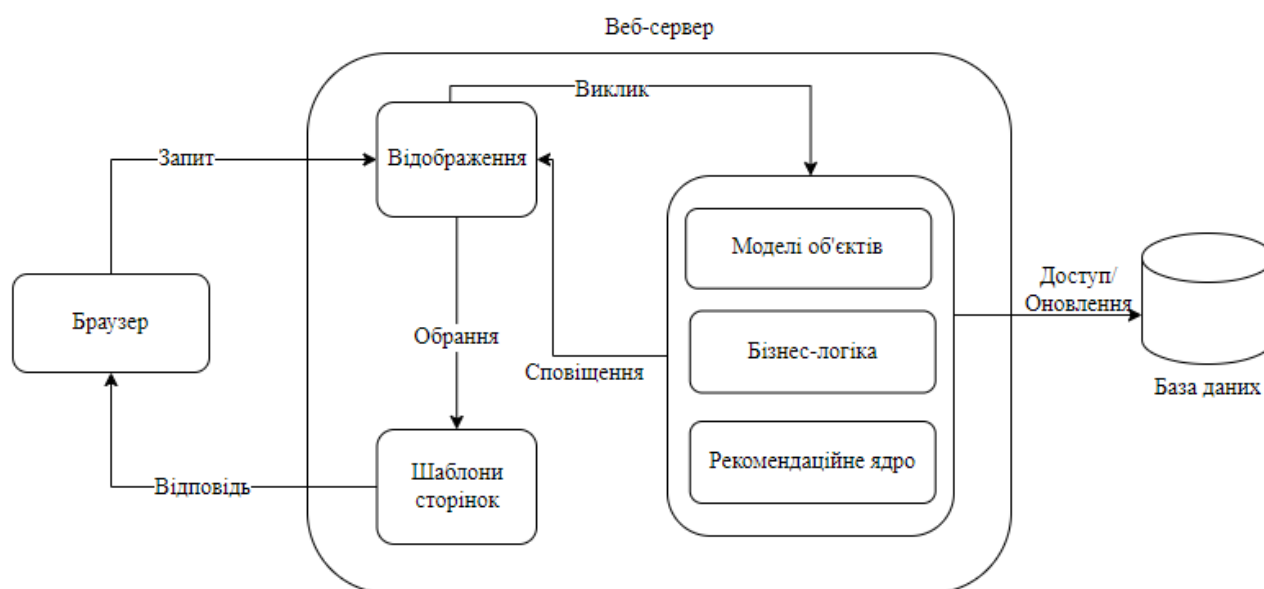


Рисунок 4.7 – Архітектура додатку

## ВИСНОВКИ

Метою кваліфікаційної роботи стало підвищення ефективності роботи рекомендаційних систем за рахунок врахування візуальних ознак зображень продуктів та створення удосконаленого алгоритму створення рекомендацій на базі розглянутих методів із урахуванням їх переваг.

Для досягнення цієї мети в першому розділі була проаналізована предметна область рекомендаційних систем: визначено поняття рекомендаційної системи, сфери використання та розглянуто найбільш відомі приклади, а також описано класифікацію та методи оцінки якості алгоритмів надання рекомендацій. Також в ньому було вивчено тему розпізнавання зображень, зокрема проведений огляд структури та основних понять згорткових нейронних мереж. Окрім зазначеного, було розглянуто сучасні наукові праці щодо використання методів розпізнавання зображень для підвищення точності і якості надання рекомендацій.

У другому розділі було розглянуто та проаналізовано різні методи розв'язання задач створення рекомендацій та варіантів архітектури згорткових нейронних мереж, обрано та детально описано методи спільного метричного навчання та знаходження візуального байєсівського рейтингу, а також згорткова нейронна мережа VGG19 – складові частини гібридного алгоритму. Після детального опису було запропоновано алгоритм удосконаленого методу формування рекомендацій.

У третьому розділі описано результати проведеного дослідження, яке становило порівняння існуючих підходів із запропонованим на тестовому наборі даних MovieLens та виявлено, що новий гібридний підхід є ефективнішим за існуючі, проте він потребує додаткової оптимізації через складність обчислень.

У четвертому розділі було спроектовано веб-сервіс рекомендаційної системи: розроблена функціональна модель системи у вигляді діаграми варіантів використання, описана взаємодія між елементами програми для досягнення необхідного функціоналу у діаграмі послідовностей, а також створена логічна та фізична схеми бази даних.

Результати дослідження представлено на Всеукраїнській науково-практичній конференції здобувачів вищої освіти і молодих учених «Комп'ютерно-інтегровані технології автоматизації технологічних процесів на транспорті та у виробництві».

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Knotzer N. Product recommendations in e-commerce retailing applications. Frankfurt am Main : P. Lang, 2008. 202 p.
2. Linden G., Smith B., York J. Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Computing. 2003. Т. 7, № 1. С. 76–80. URL: <https://doi.org/10.1109/mic.2003.1167344> (дата звернення: 03.11.2021).
3. Covington P., Adams J., Sargin E. Deep Neural Networks for YouTube Recommendations. RecSys '16: Tenth ACM Conference on Recommender Systems, м. Boston Massachusetts USA. New York, NY, USA, 2016. URL: <https://doi.org/10.1145/2959100.2959190> (дата звернення: 03.11.2021).
4. Google news personalization: Scalable online collaborative filtering / A. S. Das та ін. the 16th international conference, м. Banff, Alberta, Canada, 8–12 трав. 2007 р. New York, New York, USA, 2007. URL: <https://doi.org/10.1145/1242572.1242610> (дата звернення: 03.11.2021).
5. Aggarwal C. C. Recommender Systems: The Textbook. Springer, 2018. 519 с. URL: <https://doi.org/10.1007/978-3-319-29659-3> (дата звернення: 25.09.2021).
6. Research commentary on recommendations with side information: A survey and research directions / Z. Sun та ін. Electronic Commerce Research and Applications. 2019. Т. 37. С. 100879. URL: <https://doi.org/10.1016/j.elerap.2019.100879> (дата звернення: 10.12.2021).
7. A rough set based algebraic approach to modelling complex systems / D. Sitnikov та ін. International Journal of Design & Nature and Ecodynamics. 2018. Т. 13, № 3. С. 324–329. URL: <https://doi.org/10.2495/dne-v13-n3-324-329> (дата звернення: 10.12.2021).
8. Burke R. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction. 2002. Т. 12, № 4. С. 331–370. URL: <https://doi.org/10.1023/a:1021240730564> (дата звернення: 03.11.2021).
9. Meleshko Y. ПРОБЛЕМИ СУЧАСНИХ РЕКОМЕНДАЦІЙНИХ СИСТЕМ ТА МЕТОДИ ЇХ РІШЕННЯ. Системи управління, навігації та зв'язку. Збірник наукових праць. 2018. Т. 4, № 50. С. 120–124. URL: <https://doi.org/10.26906/sunz.2018.4.120> (дата звернення: 03.11.2021).
10. Khusro S., Ali Z., Ullah I. Recommender Systems: Issues, Challenges, and Research Opportunities. Lecture Notes in Electrical Engineering. Singapore, 2016. С.

1179–1189. URL: [https://doi.org/10.1007/978-981-10-0557-2\\_112](https://doi.org/10.1007/978-981-10-0557-2_112) (дата звернення: 03.11.2021).

11. Comparative Study on Approaches of Recommendation Systems / К. Al Fararni та ін. *Embedded Systems and Artificial Intelligence*. Singapore, 2020. С. 753–764. URL: [https://doi.org/10.1007/978-981-15-0947-6\\_72](https://doi.org/10.1007/978-981-15-0947-6_72) (дата звернення: 10.12.2021).

12. Fundamental Concepts of Convolutional Neural Network / A. Ghosh та ін. *Intelligent Systems Reference Library*. Cham, 2019. С. 519–567. URL: [https://doi.org/10.1007/978-3-030-32644-9\\_36](https://doi.org/10.1007/978-3-030-32644-9_36) (дата звернення: 10.12.2021).

13. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 2017. Т. 60, № 6. С. 84–90. URL: <https://doi.org/10.1145/3065386> (дата звернення: 10.12.2021).

14. Wang, H., and Raj, B. On the origin of deep learning. 2017. URL: <https://arxiv.org/abs/1702.07800> (дата звернення : 03.11.2021).

15. Bengio Y., Courville A., Goodfellow I. *Deep Learning*. MIT Press, 2016. 800 с.

16. Nair, V., Hinton, G. E. Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010. С. 807-814.

17. Comparative Deep Learning of Hybrid Representations for Image Recommendations / C. Lei та ін. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), м. Las Vegas, NV, USA, 27–30 черв. 2016 р. 2016. URL: <https://doi.org/10.1109/cvpr.2016.279> (дата звернення: 03.11.2021).

18. He R., McAuley J. VBPR: Visual Bayesian Personalized Ranking from implicit feedback. *AAAI'16: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, м. Phoenix, Arizona, 12 лют. 2016 р. 2016. С. 144–150. URL: <http://arxiv.org/abs/1510.01784> (дата звернення: 29.09.2021).

19. Chu W.-T., Tsai Y.-L. A hybrid recommendation system considering visual information for predicting favorite restaurants. *World Wide Web*. 2017. Т. 20, № 6. С. 1313–1331. URL: <https://doi.org/10.1007/s11280-017-0437-1> (дата звернення: 05.11.2021).

20. Kitamura R., Itoh T. Tourist Spot Recommendation Applying Generic Object Recognition with Travel Photos. 2018 22nd International Conference Information

Visualisation (IV), м. Fisciano, Italy, 10–13 лип. 2018 р. 2018. URL: <https://doi.org/10.1109/iv.2018.00011> (дата звернення: 10.12.2021).

21. Explainable Outfit Recommendation with Joint Outfit Matching and Comment Generation / Y. Lin та ін. IEEE Transactions on Knowledge and Data Engineering. 2020. Т. 32, № 8. С. 1502–1516. URL: <https://doi.org/10.1109/tkde.2019.2906190> (дата звернення: 10.12.2021).

22. Learning Image and User Features for Recommendation in Social Networks / X. Geng та ін. 2015 IEEE International Conference on Computer Vision (ICCV), м. Santiago, Chile, 7–13 груд. 2015 р. 2015. URL: <https://doi.org/10.1109/iccv.2015.486> (дата звернення: 10.12.2021).

23. Adomavicius G., Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering. 2005. Т. 17, № 6. С. 734–749. URL: <https://doi.org/10.1109/tkde.2005.99> (дата звернення: 10.12.2021).

24. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions / L. Alzubaidi та ін. Journal of Big Data. 2021. Т. 8, № 1. URL: <https://doi.org/10.1186/s40537-021-00444-8> (дата звернення: 10.12.2021).

25. Going deeper with convolutions / C. Szegedy та ін. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), м. Boston, MA, USA, 7–12 черв. 2015 р. 2015. URL: <https://doi.org/10.1109/cvpr.2015.7298594> (дата звернення: 10.12.2021).

26. Rethinking the Inception Architecture for Computer Vision / C. Szegedy та ін. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), м. Las Vegas, NV, USA, 27–30 черв. 2016 р. 2016. URL: <https://doi.org/10.1109/cvpr.2016.308> (дата звернення: 10.12.2021).

27. Deep Residual Learning for Image Recognition / К. Хе та ін. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), м. Las Vegas, NV, USA, 27–30 черв. 2016 р. 2016. URL: <https://doi.org/10.1109/cvpr.2016.90> (дата звернення: 10.12.2021).

28. M. Kula Metadata Embeddings for User and Item Cold-start Recommendations. URL: <http://arxiv.org/abs/1507.08439> (дата звернення: 10.12.2021).

29. Image-Based Recommendations on Styles and Substitutes / J. McAuley та ін. SIGIR '15: The 38th International ACM SIGIR conference on research and

development in Information Retrieval, м. Santiago Chile. New York, NY, USA, 2015. URL: <https://doi.org/10.1145/2766462.2767755> (дата звернення: 10.12.2021)

30. Collaborative Metric Learning / С.-К. Hsieh та ін. WWW '17: 26th International World Wide Web Conference, м. Perth Australia. Republic and Canton of Geneva, Switzerland, 2017. URL: <https://doi.org/10.1145/3038912.3052639> (дата звернення: 10.12.2021).

31. Weston J., Bengio S., Usunier N. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine Learning*. 2010. Т. 81, № 1. С. 21–35. URL: <https://doi.org/10.1007/s10994-010-5198-3> (дата звернення: 10.12.2021).

32. Reducing Overfitting in Deep Convolutional Neural Networks Using Redundancy Regularizer / В. Wu та ін. *Artificial Neural Networks and Machine Learning – ICANN 2017*. Cham, 2017. С. 49–55. URL: [https://doi.org/10.1007/978-3-319-68612-7\\_6](https://doi.org/10.1007/978-3-319-68612-7_6) (дата звернення: 10.12.2021).

33. S. Rendle, C. Freudenthaler, Z. Gantner, і L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, м. Arlington, Virginia, черв. 2009р. 2009. С. 452–461. URL: <http://arxiv.org/abs/1510.01784> (дата звернення: 10.12.2021).

34. *Trends in Deep Learning Methodologies*. Elsevier, 2021. URL: <https://doi.org/10.1016/c2019-0-04635-3> (дата звернення: 10.12.2021).

35. *Data Mining*. Elsevier, 2012. URL: <https://doi.org/10.1016/c2009-0-61819-5> (дата звернення: 10.12.2021).

36. K. Simonyan, A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition.*, квіт 2015, URL: <http://arxiv.org/abs/1409.1556> (дата звернення: 07.12.2021).

37. Falk K. *Practical Recommender Systems*. Manning Publications, 2019. 432 с.

38. Alhir S. S. *Learning UML*. Beijing, 2003. 234 с.

39. Pilonе D., Pitman N. *UML 2.0 in a Nutshell*. O'Reilly Media, Inc., 2005. 234 с.

40. McKinney W. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, 2017. 550 с.

41. Hope T., Resheff Y. S., Lieder I. *Learning TensorFlow: A Guide to Building Deep Learning Systems* / ред. N. Tache. Beijing : O'Reilly Media, 2017. 242 с.