

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Аліна Савченко

« _____ » _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНОВАЛЬНА ЗАПИСКА)

ВИПУСНИКА ОСВІТНЬОГО СТУПЕНЯ МАГІСТР
ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ
«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ»

**Тема: «Інтелектуальний портал з рекомендованою системою підбору
кінофільмів»**

Виконавець:

Дар'я МАЛА

Керівник:

к.пед.н., доцент Юрій СІНЬКО

Нормоконтролер:

к.т.н., доцент Олена ТОЛСТІКОВА

КИЇВ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій
Кафедра комп'ютерних інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Аліна Савченко
« ____ » _____ 2023 р.

ЗАВДАННЯ на виконання кваліфікаційної роботи Малої Дар'ї Юріївни (ПІБ випускника)

1. Тема роботи «Інтелектуальний портал з рекомендованою системою підбору кінофільмів» затверджена наказом ректора від № 1976/ст від 29.09.2023р.
2. Термін виконання роботи: з 02 жовтня 2023 року по 31 грудня 2023 року.
3. Вихідні дані до роботи: колаборативного методу для створення рекомендаційної системи, WordPress.
4. Зміст пояснювальної записки: проаналізувати предметну область рекомендаційних систем, спроектувати систему за допомогою мови моделювання UML, розробити програмне забезпечення веб-системи для інтерактивної роботи із рекомендаційною системою, використати сучасні методології для розгортання веб-системи без виділеного серверу.
5. Перелік обов'язкового графічного (ілюстративного) матеріалу:
 1. Схема роботи нейронної мережі.
 2. Схема роботи колаборативної фільтрації.
 3. Діаграма прецедентів (спрощена).
 4. Діаграма класів (спрощена)
 5. Використані технології.
 6. Головна сторінка веб-додатку.
 7. Інтеракція з моделлю рекомендаційної системи.

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1	Аналіз літератури та джерел за темою кваліфікаційної роботи.	02.10.2023р. – 06.10.2023р.	
2	Розробка та затвердження плану роботи.	17.10.2023р. – 30.10.2023р.	
3	Розробка архітектури та загальної структури системи	31.10.2023р.- 14.11.2023р.	
4	Порівняльний аналіз існуючих систем	15.11.2023р.- 20.11.2023р.	
5	Програмна реалізація системи	16.11.2023р.- 20.11.2023р.	
6	Опис функціональних можливостей	22.11.2023р.- 08.12.2023р.	
7	Розробка застосунку	09.12.2023р.- 18.12.2023р.	
8	Висновки та оформлення пояснювальної записки.	19.12.2023р. – 22.12.2023р.	

7. Дата видачі завдання: 02.10.2023 р.

Керівник кваліфікаційної роботи _____ **Юрій СІНЬКО**
(підпис керівника)

Завдання прийняв до виконання _____ **Дар'я МАЛА**
(підпис випускника)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи на тему: «Інтелектуальний портал з рекомендованою системою підбору кінофільмів» містить: 88 сторінок, 20 рисунків, 1 таблицю, 32 інформаційних джерела.

Об'єкт дослідження: інтелектуальні рекомендаційні системи.

Предмет дослідження: технології побудови рекомендаційної системи.

Метою даної роботи є реалізація колаборативного методу для створення рекомендаційної системи (РС), спроможної надавати рекомендації щодо фільмів для користувачів.

Для досягнення цієї мети були визначені такі **завдання:**

- провести аналіз основних методів розробки рекомендаційних систем;
- створити рекомендаційну систему на основі нейронної мережі, використовуючи метод колаборативної фільтрації;
- зібрати та обробити дані для тренування моделі;
- провести експерименти для тренування моделі;
- оцінити якість результатів;
- створити інтерфейс взаємодії з рекомендаційною системою.

Методи дослідження, технічні та програмні засоби: розробка, порівняльний аналіз, синтез, абстрагування, узагальнення, обробка літературних джерел.

Дана кваліфікаційна робота присвячена розробці рекомендаційної системи фільмів на основі тегів та вподобань.

DNN, РЕКОМЕНДАЦІЙНІ СИСТЕМИ, РЕКОМЕНДАЦІЇ, ПРОГНОЗУВАННЯ, ІНТЕРНЕТ-МАРКЕТИНГ, КОЛЛАБОРАТИВНА ФІЛЬТРАЦІЯ, КОНТЕНТНА ФІЛЬТРАЦІЯ, ВЕБ-ДОДАТОК

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	6
ВСТУП.....	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1. Content base системи	21
1.2. Рекомендаційні системи, засновані на спільній фільтрації	27
1.3. Гібридні рекомендаційні системи	34
РОЗДІЛ 2. ВИБІР ТА ОБГРУНТУВАННЯ ПРОГРАМНИХ ТА АПАРАТНИХ ЗАСОБІВ РЕЛІЗАЦІЇ	22
2.1. Аналіз існуючих апаратних засобів	22
2.2. Методи машинного навчання	48
2.3. Архітектура інтелектуальної системи.....	52
2.4. Засоби для створення моделі рекомендаційної системи	58
РОЗДІЛ 3. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ В ПРОГРАМНОМУ СЕРЕДОВИЩІ	37
3.1. Архітектура рекомендаційної системи	37
3.2. Архітектура нейронної мережі	70
3.3. Процес навчання нейронної мережі	72
3.4. Створення сайту	79
ВИСНОВКИ.....	84
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	87

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

<i>РС</i>	–	Рекомендаційна Система
<i>КФ</i>	–	Колаборативна Фільтрація
<i>CF</i>	–	Collaborative Filtering
<i>CBF</i>	–	Content-Based Filtering
<i>ANN</i>	–	Artificial Neural Networks
<i>DNN</i>	–	Deep Neural Networks
<i>RHM</i>	–	рекурентні нейронні мережі
<i>SHM</i>	–	штучні нейронні мережі
<i>LSTM</i>		Long short– term memory

ВСТУП

Актуальність. У нашому суспільстві інформація стала ключовим та дорогоцінним ресурсом. Обсяги доступної інформації зростають надто швидко, перевищуючи можливості людського сприйняття та обробки. Задача виділення важливого та цікавого від надлишкового стає надто важкою, особливо в умовах величезної кількості інформації в Інтернеті. Люди часто втрачаються в цьому потоці та важко знаходять те, що їм справді необхідно.

Це твердження особливо актуальне для новинних агентств, інтернет-магазинів, платформ розміщення рекламних оголошень та інших організацій. Всі ці структури зацікавлені в отриманні більше інформації про своїх користувачів, з метою задоволення їхніх інтересів і збільшення власного прибутку та популярності. З ростом кількості подій у повсякденному житті людини важко встигнути слідкувати за всім, що може бути цікавим.

У такій ситуації вирішенням проблеми може бути створення спеціалізованої системи, яка аналізує та відображає об'єкти, відповідно до індивідуальних інтересів користувача. Така система пропонуватиме об'єкти на підставі дій користувача, допомагаючи ефективно використовувати час і уникати непотрібного пошуку. Для веб-платформ це відкриває можливість надавати персоналізований контент своїм користувачам. В онлайн середовищі, насиченому безліччю можливостей, важливо використовувати фільтрацію, встановлювати пріоритети та ефективно постачати відповідну інформацію, з метою спрощення проблеми інформаційного перевантаження, що може бути проблемою для численних користувачів Інтернету. Проблему цю успішно вирішують рекомендаційні системи, які вміло переглядають обсяги динамічної інформації для забезпечення персоналізованого вмісту та послуг користувачам.

Рекомендаційні системи – це фільтруючі системи інформації, які ефективно вирішують проблему перевантаження інформацією [1], виділяючи ключовий контент з обширної маси динамічно генерованої інформації, з

урахуванням вподобань, інтересів чи попередньої поведінки користувача. Рекомендаційна система може передбачати вибір користувача, здатність визначити, чи буде конкретний користувач зацікавлений у певному товарі, на основі його профілю.

Рекомендаційні системи є вигідними як для постачальників послуг, так і для користувачів. Вони спрощують процес пошуку і відбору товарів в інтернет-магазинах. Ці системи також підтверджено покращують якість прийняття рішень. У наукових бібліотеках рекомендаційні системи допомагають користувачам розширювати область пошуку поза каталогами.

Систему рекомендацій визначають як стратегію прийняття рішень для користувачів у складних інформаційних середовищах. Крім того, систему рекомендацій охарактеризовано як інструмент в електронній комерції, який допомагає користувачам шукати в записах знань, пов'язаних із їхніми інтересами та уподобаннями. Система рекомендацій також визначається як засіб підтримки та посилення соціального процесу за допомогою рекомендацій інших осіб для прийняття рішень у випадках, коли власних знань чи досвіду недостатньо. Системи рекомендацій успішно вирішують проблему перевантаження інформацією, що зазвичай виникає для користувачів, надаючи їм персоналізований та ексклюзивний контент та рекомендації щодо обслуговування. В останні роки було розроблено різні підходи до побудови систем рекомендацій, включаючи спільну фільтрацію, фільтрацію на основі вмісту та гібридні методи. Техніка колаборативної фільтрації є найбільш зрілою та широко використовуваною.

Даний проект присвячено дослідженню та розробці методів побудови рекомендаційних систем. Основною метою роботи є створення рекомендаційної системи для фільмів, що ґрунтується на аналізі тегів та особистих вподобань користувачів. Для досягнення цієї мети виконано обширний аналіз існуючих підходів до розробки систем, які здатні передбачати

та пропонувати рекомендації. Крім того, розроблено додаток із вбудованою системою рекомендацій, який є актуальним та інноваційним.

Метою даної роботи є реалізація колаборативного методу для створення рекомендаційної системи (РС), спроможної надавати рекомендації щодо фільмів для користувачів.

Для досягнення цієї мети були визначені такі **завдання**:

- провести аналіз основних методів розробки рекомендаційних систем;
- створити рекомендаційну систему на основі нейронної мережі, використовуючи метод колаборативної фільтрації;
- зібрати та обробити дані для тренування моделі;
- провести експерименти для тренування моделі;
- оцінити якість результатів;
- створити інтерфейс взаємодії з рекомендаційною системою.

Об'єкт дослідження: інтелектуальні рекомендаційні системи.

Предмет дослідження: технології побудови рекомендаційної системи.

Розроблений продукт може використовуватися як швидка та уніфікована платформа для передбачення рекомендацій щодо фільмів. Завдяки використанню веб-технологій для побудови рекомендаційної системи, досягнута універсальність та зручність використання на будь-якому пристрої, необхідності в адаптації під конкретну операційну систему відсутнє (браузер та його віртуальна машина виступають як цільова універсальна операційна система та комп'ютер).

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Аналіз даних у сфері інтелектуального використання – це процес виявлення інформації, яка може бути корисною в обширних наборах даних. Для досягнення цієї мети в інтелектуальному аналізі даних використовується математичний аналіз для визначення закономірностей і тенденцій, що присутні у даних. Зазвичай такі закономірності важко виявити за допомогою традиційного огляду даних через їх складні зв'язки або великий обсяг інформації.

Знання, отримане в результаті інтелектуального аналізу даних, повинне бути нетривіальним і раніше невідомим. Нетривіальність передбачає, що ці знання не можна легко виявити шляхом звичайного візуального аналізу. Дані повинні висвітлювати зв'язки між властивостями бізнес-об'єктів, передбачати значення одних ознак на основі інших тощо. Знання, виявлені в результаті, повинні бути застосовні до нових об'єктів.

Практична корисність отриманих знань визначається можливістю їх використання у процесі прийняття рішень та оптимізації функціонування підприємства.

Знайдені закономірності та тренди можна об'єднати в модель інтелектуального аналізу даних. Ці моделі можуть бути застосовані до конкретних сценаріїв, таких як:

1. Прогнозування: Це включає в себе використання моделей інтелектуального аналізу даних для оцінки продажів, прогнозування навантаження на сервер або прогнозування часу простою сервера. Наприклад, можна використовувати дані про минулі продажі для розробки моделі, яка

Кафедра КІТ				НАУ 23 01 37 000 ПЗ			
Розроб.	Мала Д.Ю.			РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	Літ.	Арк.	Аркушів
Керівник	Сінько Ю.І.					10	30
Н.Контр.	Толстікова О.В.				ТП-215М - 122		

передбачає майбутні тенденції та сприятиме ефективному управлінню запасами.

Кафедра КІТ

НАУ 23 01 37 000 ПЗ

<i>Розроб.</i>	<i>Мала Д.Ю.</i>			РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	Літ.	Арк.	Аркушів
<i>Керівник</i>	<i>Сінько Ю.І.</i>					10	30
<i>Н.Контр.</i>	<i>Толстікова О.В.</i>				ТП-215М - 122		

2. Ризик та ймовірність: Це включає в себе використання моделей для оцінки ризиків та ймовірностей. Наприклад, можна визначити найбільш відповідних замовників для цільової розсилки, використовуючи аналіз даних про попередні взаємодії та поведінку клієнтів. Також, можна використовувати дані для визначення точки рівноваги для ризикованих сценаріїв, призначення ймовірностей діагнозів або інших результатів в області, де важливо оцінювати ймовірність виникнення певних подій.

3. Рекомендації: Моделі інтелектуального аналізу даних можуть визначати продукти, які мають високу ймовірність бути продані разом, та створювати персоналізовані рекомендації для клієнтів. Наприклад, аналіз покупок покупців може виявити шаблони спільних покупок, на основі яких можна рекомендувати інші товари.

4. Пошук послідовностей: Моделі інтелектуального аналізу даних можуть вивчати вибір замовників під час покупок та прогнозувати наступні можливі події. Наприклад, аналіз послідовностей покупок може виявити тенденції та передбачити майбутні вибори клієнтів.

5. Групування: Моделі інтелектуального аналізу даних можуть поділяти замовників або події на кластери пов'язаних елементів та аналізувати та прогнозувати спільні риси. Наприклад, групування клієнтів за їхнім споживацьким звичаєм може допомогти у персоналізованому обслуговуванні та маркетингу.

Щодо задач інтелектуального аналізу даних, думок щодо того, які саме задачі до нього відносити, може бути різною. Однак більшість авторитетних джерел включають такі задачі, як класифікація, кластеризація, прогнозування, асоціація, візуалізація, аналіз і виявлення відхилень, оцінювання, аналіз зв'язків та підбивання підсумків.

Класифікація (Classification):

– Опис: Класифікація є однією з найпоширеніших інтелектуальних задач аналізу даних. У цьому випадку розв'язуються завдання ідентифікації ознак, що характеризують групи об'єктів у досліджуваному наборі даних,

відомі як класи. На основі цих ознак новий об'єкт може бути віднесений до конкретного класу.

Методи розв'язання:

- Найближчий сусід (Nearest Neighbor)
- k-Найближчий сусід (k-Nearest Neighbor)
- Байєсівські мережі (Bayesian Networks)
- Індукція дерев рішень
- Нейронні мережі (Neural Networks)

Кластеризація (Clustering):

Опис: Кластеризація є логічним продовженням класифікації. Ця задача більш складна, оскільки класи об'єктів спочатку не визначені. Результатом кластеризації є розбиття об'єктів на групи.

Методи розв'язання:

- Навчання "без вчителя" з використанням самоорганізованих карт Кохонена (Self-Organizing Maps).

Асоціація (Associations):

Опис: У цьому типі задачі виявляються закономірності між подіями, які відбуваються одночасно в наборі даних. Відмінність полягає в тому, що пошук асоціативних правил здійснюється між різними подіями, які мають спільний виникнення.

Алгоритм розв'язання:

- Алгоритм Аргіогі – найбільш відомий алгоритм для пошуку асоціативних правил.

Послідовність (Sequence) або Послідовна асоціація (Sequential Association):

Опис: Задача послідовності дозволяє виявити тимчасові закономірності між транзакціями в наборі даних. Вона схожа на задачу асоціації, але встановлює зв'язки не між подіями, що відбуваються одночасно, а між подіями, пов'язаними в часі.

Правило послідовності: Після події X через певний час відбудеться подія Y.

Приклад: Після покупки квартири мешканці в 60% випадків протягом двох тижнів купують холодильник, а протягом двох місяців в 50% випадків купується телевізор.

Прогнозування (Forecasting):

Опис: У цій задачі на основі історичних даних прогнозуються пропущені або майбутні значення цільових чисельних показників. Використовуються методи математичної статистики, нейронні мережі тощо.

Визначення відхилень або викидів (Deviation Detection):

Опис: Мета цієї задачі - виявлення та аналіз даних, які найбільш відрізняються від загальної множини даних. Це включає виявлення нехарактерних шаблонів.

Оцінювання (оцінка):

Опис: Задача оцінювання полягає в передбаченні неперервних значень ознаки на основі інших ознак та аналізі історичних даних.

Аналіз зв'язків (Link Analysis):

Опис: Ця задача включає в себе пошук залежностей та взаємозв'язків в наборі даних, що може виявити ключові зв'язки між об'єктами або подіями.

Ці задачі інтелектуального аналізу даних використовуються в різних галузях, включаючи маркетинг та менеджмент, для управління циклом роботи з клієнтом та прийняття управлінських рішень.

Візуалізація (Visualization, Graph Mining):

Опис: Візуалізація створює графічний образ аналізованих даних, використовуючи графічні методи для виявлення закономірностей у даних. Приклад методу - подання даних у 2-D та 3-D вимірах.

Підбивання підсумків (Summarization):

Опис: Мета цієї задачі - опис конкретних груп об'єктів з аналізованого набору даних. Включає в себе формування коротких, але інформативних висновків.

Побудова моделі інтелектуального аналізу даних:

Опис: Це частина масштабного процесу, що включає усі завдання від формулювання питань щодо даних до розгортання моделі у робочому середовищі.

Шість базових кроків:

- Постановка задачі: Чітке визначення проблеми, визначення галузі проблеми та метрик оцінки моделі.
- Підготовка даних: Збір та підготовка даних для подальшого використання.
- Вивчення даних: Детальний аналіз вхідних даних, виявлення особливостей та паттернів.
- Побудова моделей: Розробка та навчання моделей на основі вивчених даних.
- Дослідження та перевірка моделей: Оцінка та валідація ефективності створених моделей.
- Впровадження та розгортання: Впровадження успішних моделей у робоче середовище та моніторинг їхньої ефективності.

Об'єднання та очищення даних:

Опис: Другим кроком у процесі інтелектуального аналізу даних є об'єднання та очищення даних, визначених на етапі "Постановка задачі". Це важливий етап, який передбачає оптимізацію та підготовку даних для подальшого аналізу.

Перегляд підготовлених даних:

Опис: Третім кроком є ретельний огляд підготовлених даних для забезпечення правильного розуміння їх структури та особливостей. Для прийняття правильних рішень під час моделювання важливо глибоко розуміти дані.

Методи дослідження даних включають:

- Розрахунок мінімальних та максимальних значень.

- Обчислення середнього та стандартного відхилення.
- Вивчення розподілу даних.

Приклад: Максимальні, мінімальні та середні значення можуть вказувати на неспр REPRESENTATION лентативність вибірки даних, що вимагає збалансованих даних або переосмислення припущень.

Стандартне відхилення та інші характеристики розподілу можуть надати важливу інформацію про стабільність та точність результатів.

Незвичайні розподіли можуть свідчити про потребу в додаткових даних або попереджати про можливі спотворення.

Використання інструментів:

Для аналізу та усунення проблем використовуються інструменти, такі як Служби SQL Server Data Quality Services та профілі даних у службах Integration Services.

Ці кроки допомагають глибше зрозуміти характеристики даних і готувати їх до подальшого використання у моделюванні та аналізі.

Побудова моделей інтелектуального аналізу даних:

Опис: Четвертим кроком є побудова моделей інтелектуального аналізу даних. Знання, отримані під час попередніх етапів, використовуються для визначення та створення моделей. Модель є контейнером, що задає структуру вхідних даних, прогнозований атрибут та параметри, що впливають на алгоритм обробки даних.

Навчання моделі: Процес застосування математичного алгоритму до даних у структурі з метою виявлення закономірностей. Виявлені закономірності залежать від вибору навчальних даних, обраного алгоритму та його параметрів.

Приклад: Використання навчання для прогнозування продажів на основі історичних даних.

Дослідження та перевірка моделей:

Опис: П'ятим кроком є дослідження та перевірка побудованих моделей. Оцінка якості та характеристик моделей може бути виконана за допомогою

різних методів, включаючи статистичну вірогідність, розбиття даних на навчальні та перевіркові набори, а також звернення до фахівців для оцінки результатів моделі.

Методи оцінки моделей:

– Використання заходів статистичної вірогідності для виявлення проблем даних або моделі.

– Розбиття даних на навчальний та перевірковий набори для перевірки точності прогнозів.

– Звернення до фахівців для оцінки результатів моделі та їхньої цінності для бізнес-сценарію.

Приклад: Використання статистичних метрик, таких як точність, чутливість, специфічність для оцінки ефективності моделі.

Використання інструментів:

У методології інтелектуального аналізу даних використовуються різні методи для перевірки та оцінки моделей, інструменти для статистичного аналізу та визначення ефективності, а також залучення фахівців для детального огляду результатів моделі.

Ці кроки дозволяють вивчати, аналізувати та використовувати дані для прийняття рішень та вирішення конкретних проблем в різних галузях бізнесу.

Системи надання рекомендацій здобули значну наукову увагу в середині 90-х років минулого століття і з того часу пройшли значний розвиток. Протягом останнього десятиліття велика кількість теоретичних і прикладних досліджень була присвячена розробці та вдосконаленню систем надання рекомендацій. В даний момент такі системи можуть знаходити застосування в різних галузях та для розв'язання різноманітних задач, проте для ефективного та широкого їх використання в реальному часі необхідно подальше удосконалення.

Ці системи можуть бути використані у сферах, таких як туризм, фінансові послуги, вибір товарів, електронне навчання тощо. Незважаючи на досягнутий

прогрес, подальша оптимізація і розвиток методів та алгоритмів надання рекомендацій є ключовою задачею.

Існують різні підходи до оптимізації систем надання рекомендацій, такі як кластеризація, використання контекстної інформації, оптимізація показників (наприклад, розрідженість матриці переваг чи часу підрахунку рекомендацій) та використання гібридних методів. Як приклади сучасних систем рекомендацій можна вказати Amazon.com для книг, дисків і інших товарів, MovieLens для фільмів та VERSIFI Technologies для новин. У найпростішій формі, персоналізовані рекомендації представляють собою рейтингові списки предметів. При формуванні цих рейтингів рекомендаційні системи (РС) ставлять за мету передбачити, які продукти або послуги найбільш підходять для конкретного користувача, враховуючи його уподобання та обмеження. Для виконання цього завдання РС отримують від користувачів їхні уподобання, які можуть бути явно вираженими, наприклад, у вигляді рейтингів продуктів, або визначені на основі інтерпретації дій користувача [8].

Рекомендаційні системи - це програми, які ставлять за мету надавати користувачам рекомендації щодо різних продуктів чи сервісів на основі їхніх індивідуальних вподобань. У зв'язку зі зростанням обсягів інформації в Інтернеті, людям стає важко орієнтуватися в надмірі доступних варіантів. З іншого боку, власники онлайн-платформ зацікавлені в наданні персоналізованої інформації та реклами, що може значно збільшити їхній прибуток. Розробка та удосконалення рекомендаційних систем стали актуальним завданням.

Рекомендаційні системи широко використовуються в інтернет-комерції, де вони допомагають користувачам вибирати товари з урахуванням їхніх індивідуальних вподобань. Також вони знайомі в інших сферах, де важлива персоналізація контенту. Останнім часом інтерес до розробки та вдосконалення рекомендаційних систем значно зростають.

Рекомендаційні системи оцінюються за якістю їхніх рекомендацій, зазвичай використовуючи показники продуктивності, що враховують вузьке розуміння системного поведінки. Зазвичай, розробники оцінюють точність

рекомендацій за допомогою метрик, які вимірюють здатність системи точно відобразити вподобання користувачів. Точність вимірюється резервування частини даних для навчання та тестування системи, використання відомих вподобань та інших даних для створення рекомендацій, що дозволяє оцінити ефективність алгоритму(рис. 1.1).



Рис. 1.1. Структура рекомендаційної системи

Відомо, що метрики точності, такі як точність і відгук, надають обмежене та одномірне уявлення про продуктивність рекомендаційних систем. Є кілька обмежень в їхньому використанні. По-перше, ці метрики надають обмежену інформацію про поведінку об'єктів з невідомими уподобаннями. Тобто, якщо користувачі не виражають свої уподобання щодо певних об'єктів, ці метрики не можуть їх врахувати. По-друге, коли набір відомих уподобань є невеликим, значення точності і відгуку можуть бути низькими через малу ймовірність включення рекомендованих елементів в цей набір. Це особливо важливо, оскільки такі метрики не враховують контекст взаємодії з об'єктами, які користувачі не відзначили відомими уподобаннями.

Оскільки точність і відгук не надають повної інформації про взаємодію з об'єктами, що не входять в набір відомих уподобань, вони можуть призводити до неправильних висновків про відносну ефективність однієї системи порівняно з іншою.

Рекомендації формуються персонально для кожної особи, враховуючи її попередні дії на певному веб-ресурсі чи засновані на минулій активності. Також важливим елементом є поведінка попередніх учасників цього процесу. У випадку інтернет-магазинів це стає ключовою функцією, а для великих каталогів, подібних до Amazon, - одним з ефективних способів забезпечення якісної роботи. Система рекомендацій у даному випадку не є просто додатковою опцією, але й забезпечує зручність навігації користувача на веб-ресурсі. Якщо електронний каталог містить понад 20 000 товарів, то орієнтація стає дуже важкою, а коли товарів мільйони - ще складніше. Це може призвести до втрати потенційного покупця при взаємодії з подібним сайтом.

Вирішенням цього питання є використання віджету пошуку товарів, які візуально схожі на шуканий, або належать до однієї групи товарів, або є комплементарними (наприклад, пропонують вибрати сумочку до пари туфель). Такий підхід збільшує не лише кількість переглядів, але й позитивно впливає на конверсію.

Практика показує, що не лише онлайн-магазини використовують подібний метод. Подібні прийоми легко помітити на різних соціальних платформах, порталах про літературу, подорожі, новинних ресурсах та інтернет-магазинах. Ця методика дійсно користується великою популярністю. Рекомендаційні сервіси збирають різноманітну інформацію про особу, використовуючи декілька методів, які об'єднують всі системи.

Отже, перший тип - це явний збір даних, де користувач надає необхідну інформацію самостійно. Наприклад, рекомендаційні системи Google чи інших пошукових систем звертаються до користувачів із проханням оцінити різні елементи, скласти список улюблених або відповісти на кілька питань. У

випадку відмови від самостійної наданої інформації, застосовується інша методика.

Другий тип - це неявний збір даних, який можна порівняти із шпигунською місією, де дії учасника процесу реєструються програмою для подальшого аналізу та використання. Програма визначає покупки, оцінки на сайтах, збирає інформацію про перегляди та коментарі. Звісно, вибір цієї методики виникає з етичних проблем, оскільки захист персональних даних - один із ключових вимог користувачів щодо пошукових сайтів. Проте факт залишається фактом - такий вид стеження можливий, і відвідувачі сайтів не можуть перевірити, чи використовується подібний підхід.

Також існують різновиди рекомендаційних систем, які визначаються за застосовуваними підходами.

Збір відгуків здійснюється з різних причин, і не завжди він пов'язаний виключно з машинним навчанням. Кожна система навчання має свої унікальні вимоги до представлення даних для подальшого аналізу. Вибір конкретних даних для вивчення може суттєво вплинути на модель навчання, тому підготовка даних є ключовою частиною будь-якого проекту машинного навчання і часто стає найбільш трудомісткою.

У виборі конкретного типу фільтрації або комбінації методів велике значення мають складність проекту і обсяг фінансування. Наприклад, створення алгоритму для системи, яка аналізує тематичні пересічення блогів, може бути відносно простим завданням з помірними витратами. Проте більш масштабні та різноманітні проекти, наприклад, онлайн-магазини, можуть вимагати значних витрат, особливо якщо поставлена мета - збільшити конверсію на суттєві показники. Зазвичай в таких проектах важко обмежитися одним видом рекомендаційного алгоритму, і доводиться використовувати гібридну фільтрацію, що призводить до збільшення вартості і складності розробки на порядки.

Незалежно від цього, при розробці проекту, який надає користувачеві можливість вибирати конкретні об'єкти із загальної множини, важливо

враховувати стрімкий прогрес в галузі юзабіліті в усіх аспектах життя. Це охоплює від оптимізації сну за допомогою пристроїв, що аналізують всі процеси сну і надають рекомендації для його поліпшення, до автоматичного підбору повсякденних товарів з урахуванням поточних потреб користувача. Схеми представлені на рис. 1.2.

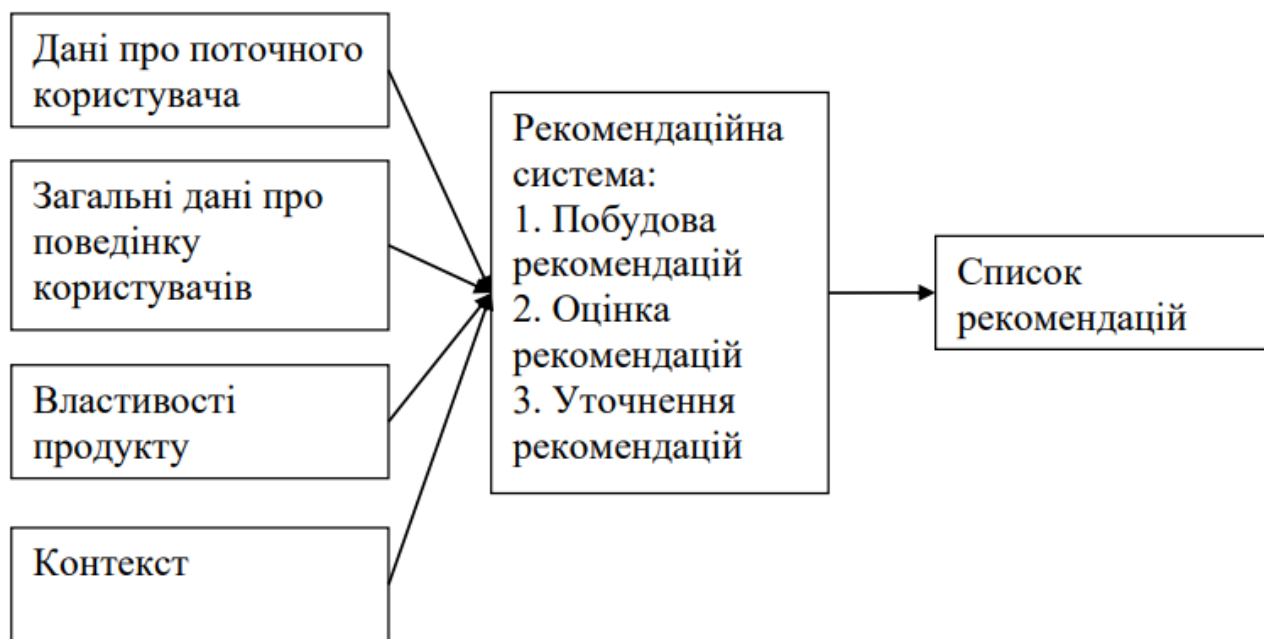


Рис. 1.2. Процес роботи рекомендаційних систем

На сьогоднішній день для створення рекомендаційних систем використовуються дві основні стратегії: фільтрація вмісту та колаборативна фільтрація. Важливо відзначити, що на практиці часто застосовують гібридні методи, які об'єднують переваги обох підходів. Класифікація рекомендаційних систем приведена на рис. 1.3.

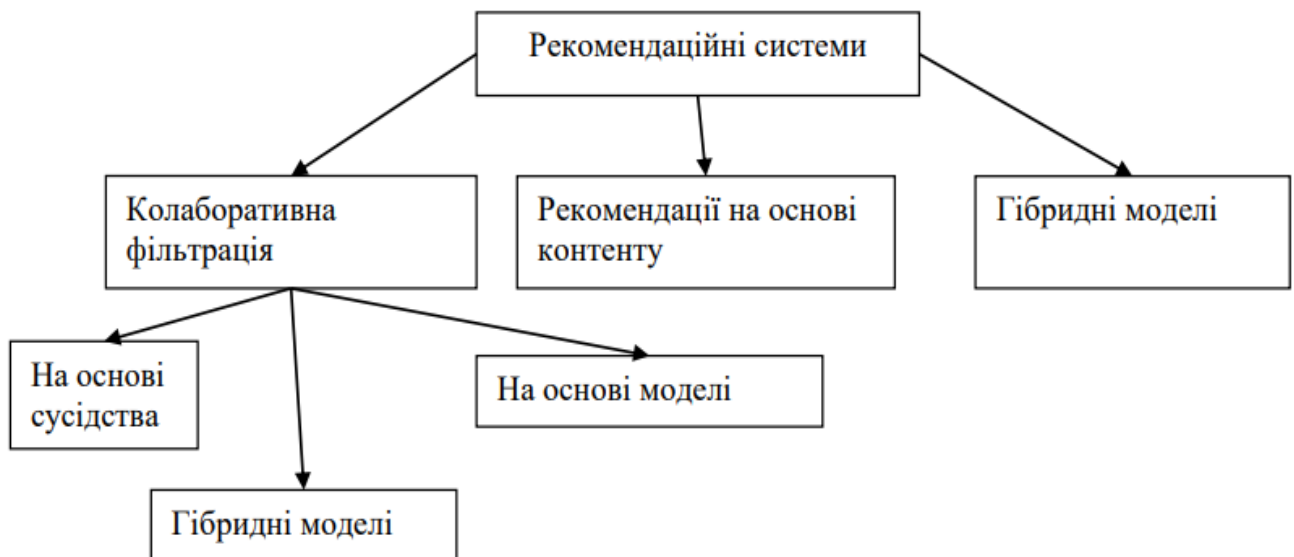


Рис. 1.3. Класифікація рекомендаційних систем

Під час розробки рекомендаційних систем розробники зазвичай стикаються з рядом проблем прогнозування:

- Розрідженість даних (більшість користувачів не надає оцінок товарам, що призводить до розрідженості матриці оцінок);
- Холодний старт (взаємодія з новими користувачами або товарами);
- Синонімія (визначення схожих товарів із різними назвами); [2]
- Шахрайство (цілеспрямоване завищення рейтингів певних товарів їх власниками);
- Розмаїття (нові або маловідомі товари мають низькі позиції в рейтинговому списку при великій вибірці);
- Білі ворони (користувачі з унікальними смаками, які важко обробляти, оскільки вони відхиляються від загальноприйнятих вподобань). [3]

Отже, видно, що проблеми створення систем прогнозування для інтернет-сервісів є значущими, і необхідно розглянути, як різні підходи до рекомендаційних систем вирішують ці труднощі та покращують якість рекомендацій.

В кожній рекомендаційній системі ми стикаємося з користувачем, якому пропонується ряд альтернатив, серед яких потрібно зробити вибір.

Користувачеві може бракувати досвіду та знань для самостійного відсіювання альтернатив, які не відповідають його потребам. Користувач, свідомо чи несвідомо, надає системі інформацію про свої уподобання, іноді навіть не будучи обізнаним з деякими альтернативами. Таким чином, рекомендаційна система виступає як механізм, який використовує певний тип фільтрації та наявні дані про потреби користувача для того, щоб запропонувати йому набір альтернатив, які ймовірно будуть йому найбільш корисними [12].

Для розробки рекомендаційних систем використовують чотири типи фільтрації:

- Заснована на контенті (Content-Based);
- Коллаборативна (Collaborative);
- Заснована на знаннях (Knowledge-Based);
- Гібридна (Hybrid) [19].

1.1. Content base системи

Тематична фільтрація формує рекомендації, виходячи зі спостережень за поведінкою користувача. Наприклад, цей підхід може використовувати історичну інформацію про перегляди (які блоги читає користувач і характеристики цих блогів). Якщо користувач регулярно читає статті про Linux або залишає коментарі в блогах з проектування програмного забезпечення, тематична фільтрація може використовувати цю інформацію для виявлення подібного контенту і пропозиції схожого контенту як рекомендованого для цього користувача (наприклад, статті в блогах про Linux або інші блоги з проектування програмного забезпечення). Визначення цього контенту може бути здійснено вручну або автоматично за допомогою інших методів визначення подібності [6].

Фільтрація вмісту, яка зображена на рис. 1.4, базується на створенні профілю користувача та профілю об'єкта. Цей підхід враховує параметри

об'єктів і їх відповідність перевагам користувача. Для досягнення цієї мети в рекомендаційних системах використовуються теги (ключові слова), які використовуються для опису об'єктів. Профіль користувача відображає оцінку певних тегів або їх сукупність.

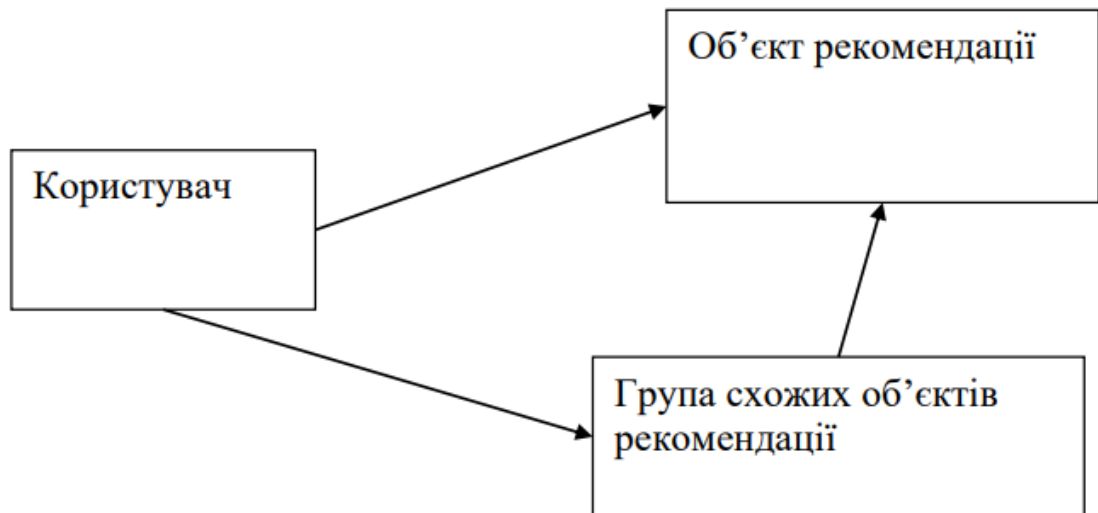


Рис. 1.4. Фільтрація вмісту

В рекомендаційних системах з фільтрацією вмісту функція задоволеності $h(u, s)$ користувача u деяким об'єктом s визначається на основі відомостей про задоволеність користувача об'єктами $s_i \in S$, які мають схожість з s . Наприклад, в музичному аудіо-стрімінговому веб-додатку рекомендаційна система визначає, що пов'язує різні музичні композиції, які користувач раніше високо оцінив. Потім система рекомендує користувачеві музичні композиції з максимальною відповідністю тим, які були високо оцінені користувачем раніше.

Профіль користувача в системі формується зі споживаного користувачем контенту у вигляді множини параметрів, що визначають об'єкт s . Параметри можуть включати ключові слова, які зазвичай мають ваги для кожного об'єкта.

На рис. 1.5 показана матриця характеристик, де кожен рядок представляє програму, а кожен стовпець - характеристику. Функції можуть включати

категорії (наприклад, освіта, повсякденне життя, здоров'я), видавця програми та інші. Для спрощення можна припустити, що ця матриця характеристик є двійковою: ненульове значення вказує на те, що додаток має цю характеристику

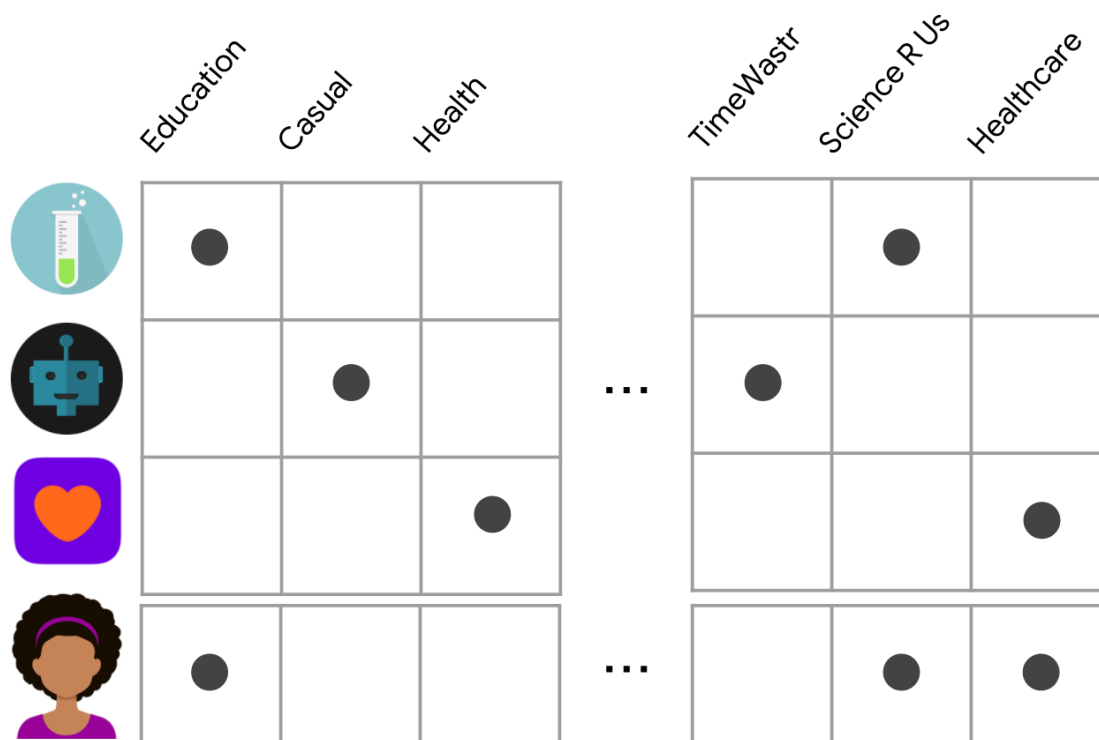


Рис. 1.5. Матриця даних користувача

Користувача представлено в одному і тому ж просторі функцій. Деякі функції, пов'язані з користувачем, можуть бути явно вказані ним, наприклад, обираючи "Розважальні програми" у своєму профілі. Інші функції можуть бути неявними на основі додатків, які він раніше встановив, наприклад, додаток, опублікований "Science".

Модель повинна рекомендувати предмети, що відповідають цьому користувачеві [4]. Спочатку потрібно обрати метрику подібності, наприклад, косинусну схожість. Потім систему слід налаштувати для оцінки кожного кандидата відповідно до цієї метрики подібності. Рекомендації стосуються тільки цього користувача, оскільки модель не використовує жодної інформації про інших користувачів.

Переваги фільтрації на основі вмісту включають:

— Модель не потребує даних про інших користувачів, оскільки рекомендації стосуються лише цього користувача. Це полегшує масштабування для великої кількості користувачів;

— Модель може враховувати конкретні інтереси користувача та рекомендувати ті ніші, які цікавлять дуже мало інших користувачів.

Недоліки фільтрації на основі вмісту включають:

— Оскільки представлення елементів частково розроблено вручну, ця техніка вимагає знань від розробника в галузі. Таким чином, модель може бути настільки ефективною, наскільки добре вручну створені функції;

— Модель може рекомендувати лише на основі існуючих інтересів користувача, не маючи можливості розширювати їх [5].

Ефективність фільтрації на основі вмісту та парадигма рекомендацій, що ґрунтуються на змісті:

1. Ознаки елементів та їх обробка:

- Опис: Ефективність фільтрації на основі вмісту залежить від розуміння вмісту елементів. Для цього використовуються методи вилучення ознак, такі як векторизація, TF-IDF, Word2Vec. Вони дозволяють представити елементи у вигляді числових векторів, що полегшує їх порівняння та аналіз.
- Приклад: Використання Word2Vec для векторизації текстових описів товарів.

2. Парадигма рекомендацій, що ґрунтуються на змісті:

- Опис: Парадигма рекомендацій, що ґрунтуються на змісті, має перевагу незалежності користувача. Модель не потребує даних про інших користувачів і може надавати персоналізовані рекомендації на основі інтересів конкретного користувача.
- Переваги:

- Незалежність користувача: Модель не вимагає даних про інших користувачів, спрощуючи масштабування для великої кількості користувачів.
- Специфічні рекомендації: Модель може рекомендувати нішові елементи, враховуючи конкретні інтереси користувача.
- Недоліки:
 - Залежність від характеристик: Потребує глибокого знання предметної області для визначення ефективних характеристик.
 - Обмеженість інтересів: Модель може обмежуватися існуючими інтересами користувача і не вміє радикально розширювати їх.
- Приклад: Рекомендація статей на основі аналізу текстового контенту.

3. Проблеми та недоліки:

- Опис: Content-based підхід має певні обмеження, такі як необхідність ручної розробки характеристик об'єктів та обмежена здатність до розширення інтересів користувачів при відсутності достатньої кількості оцінок.
- Недоліки:
 - Залежність від ручного вибору характеристик.
 - Обмежена здатність рекомендацій до нових інтересів користувача.
 - Необхідність достатньої кількості оцінок для точних рекомендацій.
- Приклад: Модель рекомендацій для музичних треків, враховуючи аналіз жанрів та текстового контенту.

Ці аспекти вказують на те, що, хоча content-based підхід має свої переваги, він також обмежений і може вимагати уваги до якості та обробки характеристик для досягнення найкращих результатів.

1.2. Рекомендаційні системи, засновані на спільній фільтрації

Щоб подолати деякі обмеження фільтрації на основі вмісту, спільна фільтрація використовує схожість між користувачами та елементами одночасно для створення рекомендацій [6]. Це дозволяє робити випадкові рекомендації; іншими словами, моделі колаборативної фільтрації можуть рекомендувати елемент користувачеві А на основі інтересів подібного користувача В. Крім того, вбудовування може бути вивчено автоматично, без потреби у ручній розробці функцій.

Колаборативна фільтрація (Рис. 1.6) використовує відомі переваги групи користувачів, до якої входить той користувач, для якого потрібно спрогнозувати рекомендації. Головна підстава такого підходу в тому, що користувачі однаково оцінили деякі об'єкти в минулому, найімовірніше, однаково оцінять інші об'єкти в майбутньому.

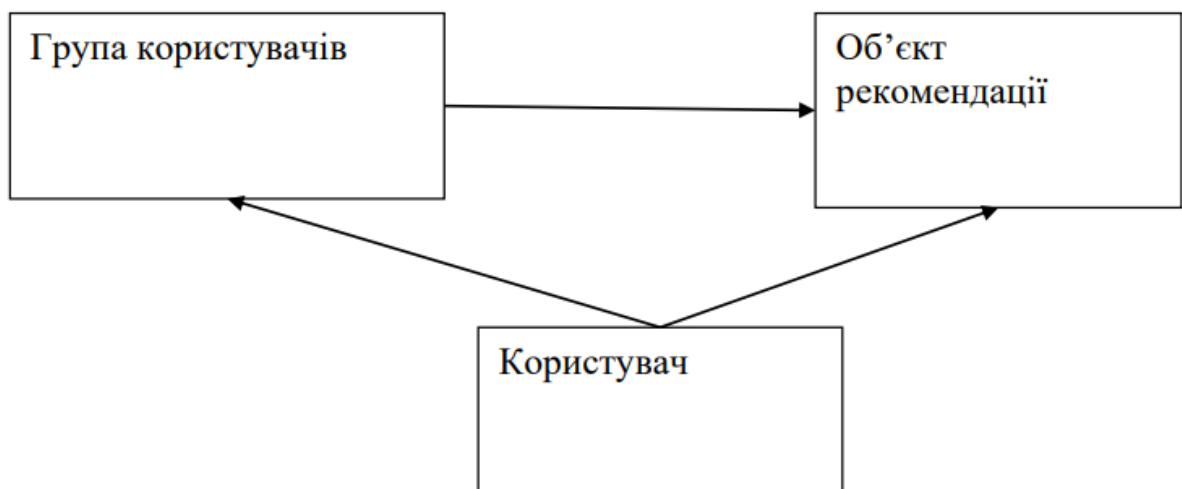


Рис. 1.6. Колаборативна фільтрація

Існує значна кількість систем рекомендаційного характеру, які базуються на взаємодії з користувачами та розроблені як в наукових, так і в комерційних сферах. Однією з таких систем є "Grundy system", де бібліотекар задає користувачам питання різної тематики для формування їхнього персоналізованого профілю і рекомендацій щодо книг. У випадку, якщо

користувач не проявляє зацікавленість пропонованою книгою, система активно досліджує причини цього вибору, щоб глибше зрозуміти користувача.

Інша система, на ім'я "Tapestry", пропонує користувачеві самостійно ранжувати інших користувачів на основі їхніх власних виборів та вподобань.

Важливо відзначити, що існують методи колаборативної фільтрації вмісту, які можна розділити на дві основні категорії: засновані на пам'яті і засновані на моделях.

Розглянемо систему рекомендацій для фільмів, де дані навчання складаються з матриці зворотного зв'язку, де:

- Кожен рядок представляє користувача;
- Кожен стовпець представляє елемент (фільм).

Відгуки про фільми поділяються на дві категорії:

- Явні: користувачі вказують, наскільки їм сподобався певний фільм, надаючи числовий рейтинг;
- Неявні: якщо користувач дивиться фільм, система припускає, що цей фільм їх зацікавив.

Матриця зворотного зв'язку є двійковою; значення 1 вказує на інтерес до фільму. Коли користувач відвідує свою домашню сторінку, система повинна рекомендувати фільми, враховуючи обидві:

- схожість із фільмами, які користувачу сподобались раніше;
- фільми, які сподобались іншим користувачам зі схожими інтересами.

Для ілюстрації розглянемо деякі характеристики фільмів, описані в таблиці 1.1:

Данні опису змісту фільмів

Фільм	Оцінка	Опис
Повернення Темного Лицаря	4	Бетмен намагається врятувати Готем-Сіті від ядерного знищення в цьому продовженні "Темного лицаря", створеному у всесвіті DC Comics.
Гаррі Поттер і філософський камінь	5	Осиротілий хлопчик виявляє, що є чарівником, і вступає до Школи чаклунства і чаклунства Хогwartса, де веде свою першу битву зі злим лордом Волдеморт.
Триплети Бельвіля	4	Коли професійного велосипедиста Чемпіона викрадають під час Тур де Франс, його бабусята собака із зайвою вагою вирушають за кордон, щоб врятувати його за допомогою тріолітніх джазових співаків.
Memento	5	Амнезіак відчайдушно прагне розкрити вбивство своєї дружини, татуюючи підказки на своєму тілі
Шрек	4	Милый людоїд та його віслук-помічник вирушили в місію, щоб врятувати принцесу Фіону, яку дракон утримує у своєму замку.

Кожному фільму ми призначаємо скаляр у діапазоні від -1 до 1, що описує, чи призначений фільм для дітей (з негативними значеннями) чи для дорослих (з позитивними значеннями). Припустимо, ми також присвоюємо скаляр кожному користувачеві в діапазоні від -1 до 1, що відображає інтерес користувача до дитячих фільмів (ближче до -1) або фільмів для дорослих (ближче до +1). Продукт вбудовування фільму та вбудовування користувача

повинен бути вищим (ближче до 1) для фільмів, які сподобались користувачеві (рис.1.7).



Рис. 1.7. Вікова шкала відповідності фільмів користувача

На рис. 1.8 кожен позначений фільм вказує на те, що його переглядав конкретний користувач. Переваги третього і четвертого користувачів легко пояснюються цією характеристикою: третій користувач віддає перевагу фільмам для дітей, тоді як четвертий користувач віддає перевагу фільмам для дорослих. Проте переваги першого і другого користувачів не можна повністю пояснити лише цією єдиною ознакою.

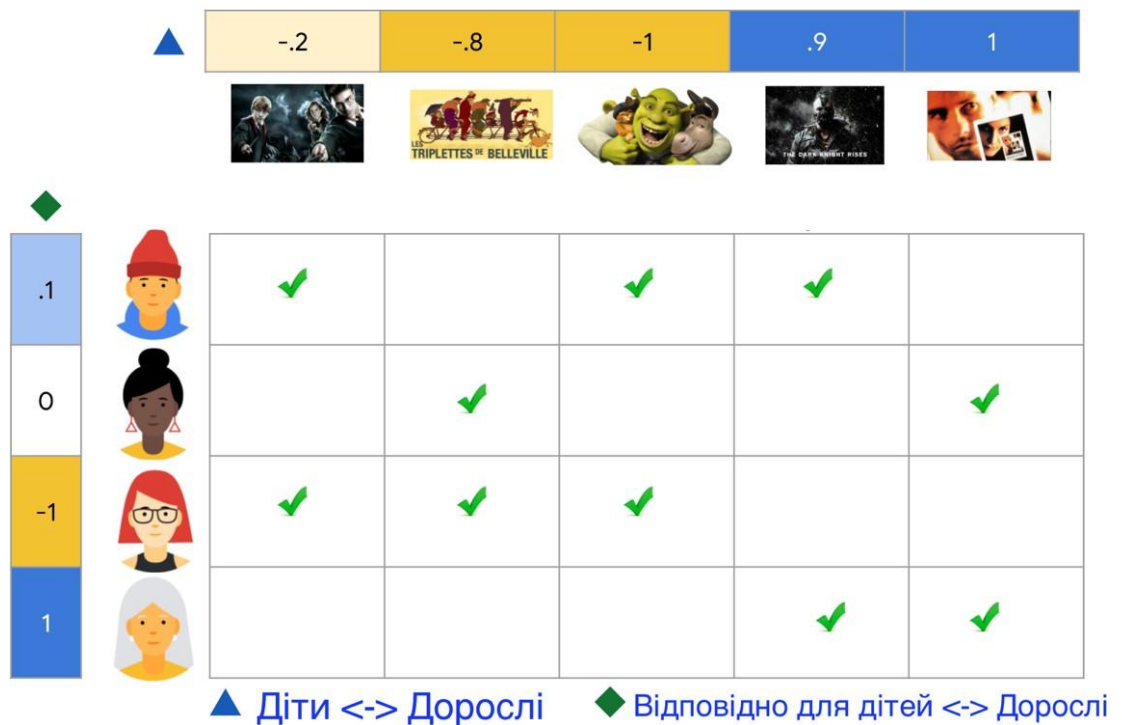


Рис. 1.8. Одно-вимірне представлення даних користувачів (1D)

Для пояснення вибірових переваг користувачів було виявлено, що однієї ознаки недостатньо. Для вирішення цієї проблеми було введено другий аспект: ступінь того, наскільки кожен фільм є блокбастером або арт-хаусом. З використанням другої характеристики ми тепер можемо представити кожен фільм у двовимірному просторі (рис. 1.9).

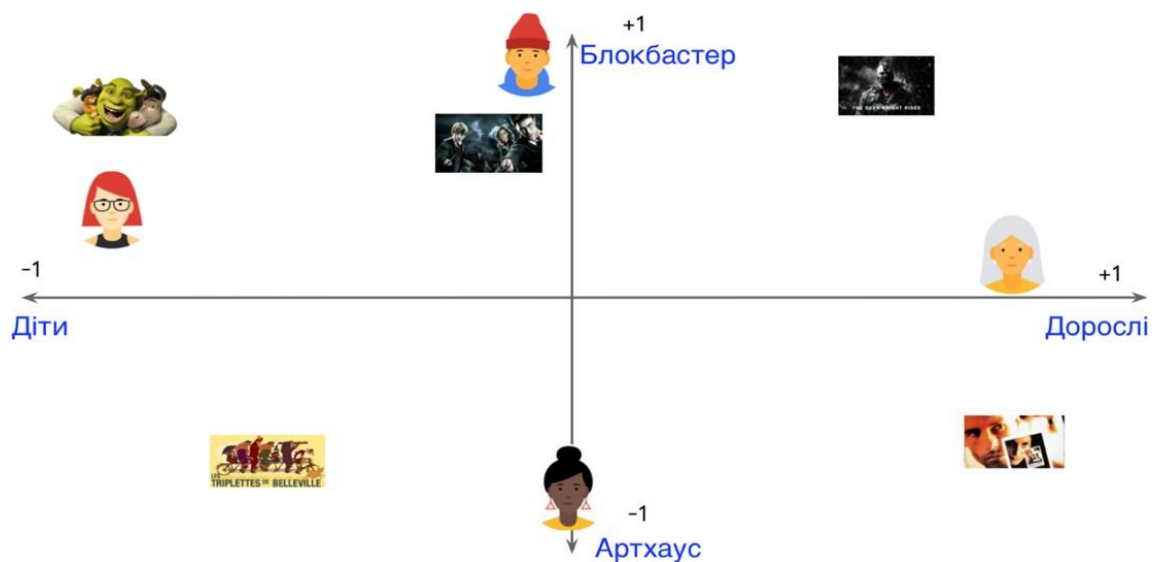
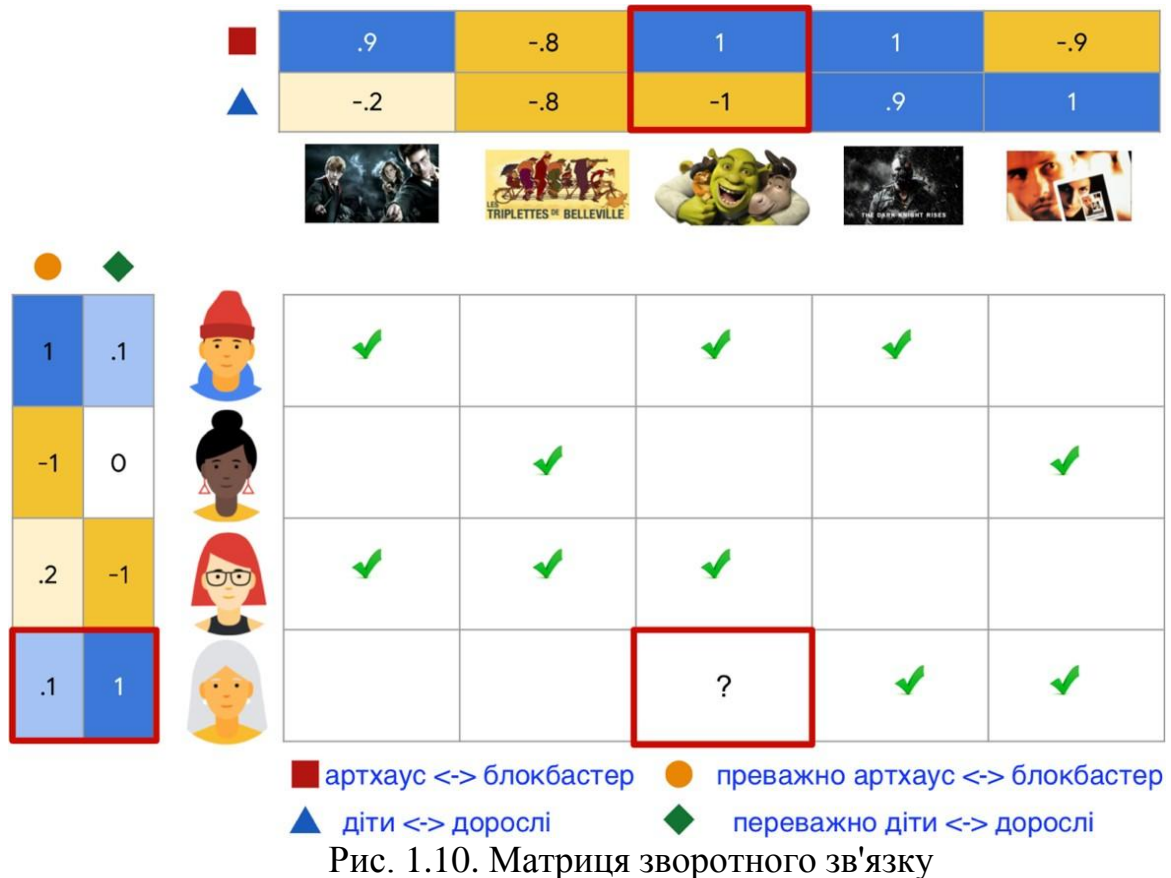


Рис. 1.9. Двовимірне представлення даних користувачів (2D)

Користувачі були розташовані в одному вбудовуванні для кращого пояснення матриці зворотного зв'язку (рис. 1.10): для кожної пари (користувач, елемент) ми б хотіли, щоб скалярний добуток вбудовування користувача та вбудовування елемента був близьким до 1, коли користувач переглядав фільм, і до 0 - у протилежному випадку.



Характер колаборативної фільтрації стає очевидним, коли модель вивчає вбудовування. Припустимо, що вектори вбудовування для фільмів є фіксованими. Тоді модель може вивчити вектор вбудовування користувачів, щоб найкраще пояснити їхні вподобання. Таким чином, вбудовування користувачів зі схожими уподобаннями буде близьким. Подібно, якщо вбудовування користувачів фіксоване, тоді ми можемо вивчити вбудовування фільмів, щоб найкраще пояснити матрицю зворотного зв'язку. В результаті вбудовування фільмів, які сподобалися подібним користувачам, буде близьким до їхніх вбудовувань.

Колаборативна фільтрація:

1. Опис методу:

- **Огляд:** Колаборативна фільтрація — це популярний метод у рекомендаційних системах, який не порівнює інтереси користувачів з характеристиками об'єктів, а здійснює пошук схожих

користувачів або об'єктів. Основна ідея полягає в тому, що схожі користувачі будуть мати схожі інтереси, і тому їм сподобаються однакові або подібні елементи.

- **Види:** Існують два основних підходи: User-Based та Item-Based. User-Based фокусується на пошуку схожих користувачів, тоді як Item-Based фокусується на схожості між елементами.

2. Етапи системи КФ:

- **Збір оцінок:** Початковий етап включає збір користувацьких оцінок для елементів, які служать основою для матриці оцінок.
- **Пошук сусідів:** Сусідів шукають шляхом обчислення подібності між користувачами або елементами. Це може включати косинусну схожість, Євклідову відстань тощо.
- **Прогнозування:** На основі знайдених сусідів виконується прогнозування рейтингу елементів для користувача. Методи агрегування використовуються для цього.
- **Рекомендації:** Найвище оцінені елементи, що не були оцінені користувачем, вважаються рекомендаціями для конкретного користувача.

3. Інтуїція та переваги:

- **Інтуїція:** Схожість в минулому призводить до схожих інтересів у майбутньому. Користувачі, які мали спільні вподобання в минулому, ймовірно, матимуть спільні вподобання в майбутньому.
- **Переваги:**
 - Не потрібно заздалегідь знати характеристики користувачів або елементів.
 - Масштабована для великої кількості користувачів.
 - Персоналізована рекомендація на основі взаємодії користувачів.

4. Недоліки:

- **Споживання ресурсів:** Може вимагати значних обчислювальних ресурсів при великій кількості користувачів та елементів.
- **Холодний старт:** Зазнає проблем при нових користувачах чи елементах без історії взаємодії.
- **Приклад:** Рекомендація фільмів на основі оцінок інших користувачів, які мали схожі вподобання в минулому.

1.3. Гібридні рекомендаційні системи

Гібридні підходи об'єднують колаборативну і контентну фільтрацію для підвищення ефективності та складності рекомендаційних систем. Ці підходи дозволяють поєднати результати колаборативної і контентної фільтрації, щоб потенційно збільшити точність рекомендацій. Гібридний підхід особливо корисний, коли виникає "холодний старт" через значну розрідженість даних. В цьому випадку гібридний підхід дозволяє спочатку враховувати результати контентної фільтрації, а потім поступово враховувати колаборативну фільтрацію зі зростанням доступних даних для конкретного користувача.

Так, комбінування різних типів рекомендаційних систем, таких як колаборативна фільтрація та фільтрація на основі вмісту, є розповсюдженою практикою для покращення точності та робастності рекомендацій. Цей підхід називається **гібридною рекомендаційною системою**. Вона дозволяє враховувати переваги обох методів та компенсувати їхні недоліки. Ось деякі переваги та можливості гібридних систем:

1. Компенсація недоліків:

- Гібридні системи можуть компенсувати недоліки одного методу іншим. Наприклад, коли у нового користувача немає історії взаємодії, фільтрація на основі вмісту може надати значущі рекомендації.

2. Покращення точності:

- Комбінування різних методів може призвести до більш точних рекомендацій, особливо якщо обидва методи доповнюють один одного в різних сценаріях.
- 3. Робастність:**
- Гібридні системи можуть бути більш робастними в умовах обмеженого обсягу даних або у випадку великої різноманітності користувачів та об'єктів.
- 4. Персоналізація:**
- Здатність до налаштування комбінацій методів дозволяє забезпечити персоналізовані рекомендації для різних категорій користувачів.
- 5. Управління холодним стартом:**
- В гібридних системах можливо використовувати фільтрацію на основі вмісту для управління холодним стартом, де немає достатньої інформації про користувача або елемент.

Гібридні рекомендаційні системи можуть використовувати різні стратегії комбінування, такі як вагове злиття рекомендацій, використання ансамблей моделей або переключення між методами в залежності від контексту. Такий підхід дозволяє створювати більш гнучкі та потужні системи рекомендацій, які враховують широкий спектр факторів та умов.

У рекомендаційних механізмах використовують різноманітні алгоритми, які можуть варіюватися в залежності від проблем, що вони призначені вирішувати, і від відносин, що присутні в даних. Багато з цих алгоритмів походять з галузі машинного навчання, яка спеціалізується на розробці алгоритмів для навчання, прогнозування і прийняття рішень.

Гібридні рекомендаційні системи використовують комбінації підходів фільтрації вмісту та колаборативної фільтрації, що дозволяє уникнути недоліків обох методів. Існують кілька основних варіантів комбінації різних методів:

– Комбінований підхід (Combination): Об'єднання як фільтрації вмісту, так і колаборативної фільтрації для отримання більш точних рекомендацій.

– Змінний підхід (Switching): Застосовує лише один метод рекомендацій в конкретних ситуаціях або для конкретних користувачів.

– Змішаний підхід (Mixed): Комбінує рекомендації з обох методів, пропонуючи їх у вигляді спільного списку.

– Навчання з посиленням (Reinforcement Learning): Використовує алгоритми машинного навчання для навчання системи на основі здійснених користувачем виборів і взаємодій.

Ці гібридні підходи дозволяють покращити якість рекомендацій і забезпечують більш гнучку адаптацію до різних сценаріїв та вимог користувачів.

Є кілька основних варіантів комбінування різних методів в рекомендаційних системах гібридного типу, вони указані на рис. 1.11.

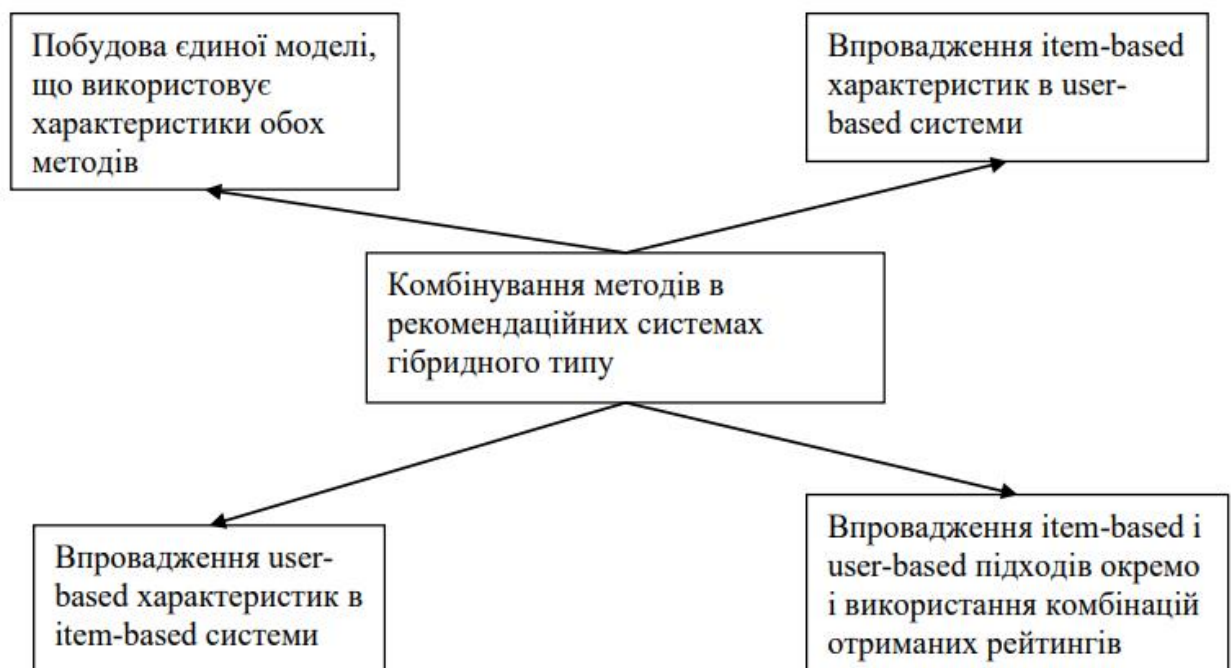


Рис. 1.11. Комбінування методів в рекомендаційних системах гібридного типу

Останнім часом цей підхід здобув велику популярність серед вчених. Основна ідея методу полягає в тому, щоб поєднати характеристики фільтрації вмісту та колаборативної фільтрації, такі як жанр музичної композиції та професія користувача, у єдиній рекомендаційній системі. Ця система використовує єдину модель, яка об'єднує особливості обох методів. Також, вона впроваджує характеристики, які визначені для об'єктів, у системи, орієнтовані на користувача, або навпаки. Окремо впроваджуються підходи, спрямовані на об'єкти та користувачів, і використовуються комбінації отриманих рейтингів.

До таких підходів відноситься уніфікований імовірнісний метод, який використовує латентно-семантичний аналіз. Гібридні рекомендаційні системи, які об'єднують колаборативні складові та включають дані фільтрації вмісту в профілі користувачів, дозволяють обчислювати схожість користувачів, замість оцінок загальних об'єктів. Цей підхід допомагає уникнути проблеми з недостатньою кількістю даних, що виникає при обмеженій кількості пар користувачів з великою кількістю об'єктів.

Гібридні рекомендаційні системи, також використовуючи зменшення розмірності профілів за допомогою латентно-семантичного аналізу, важливі для покращення продуктивності порівняно із системами, що використовують лише фільтрацію вмісту.

Ще один варіант гібридних систем передбачає роздільне впровадження підходів, спрямованих на об'єкти та користувачів. Це може включати комбінування оцінок від обох систем або використання тієї системи, яка найбільш підходить для конкретного випадку. Система, яка використовує рекомендаційну систему з мінімальним рівнем помилки, є прикладом такого підходу. Гібридні рекомендаційні системи також можуть доповнюватися методами, які використовують базу знань для вдосконалення якості рекомендацій та вирішення основних проблем більшості рекомендаційних систем, таких як холодний старт. Різні дослідження підтверджують перевагу

гібридних підходів над чистими item-based і user-based рекомендаційними системами.

РОЗДІЛ 2

ВИБІР ТА ОБГРУНТУВАННЯ ПРОГРАМНИХ ТА АПАРАТНИХ ЗАСОБІВ РЕЛІЗАЦІЇ

2.1. Аналіз існуючих апаратних засобів

Рекомендаційні системи вже успішно інтегровані в безліч веб-додатків, які активно використовуються мільйонами користувачів щоденно. Наведемо приклади найбільших ресурсів, які використовують рекомендаційні механізми:

LinkedIn: Бізнес-орієнтована соціальна мережа, яка використовує вбудований рекомендаційний механізм для пропозиції користувачеві зв'язків, вакансій, груп, та компаній, які можуть бути цікавими. Використовує технологію Apache Hadoop для спеціалізованої системи колаборативної фільтрації.

Amazon: Один з найбільших інтернет-майданчиків для торгівлі. Використовує рекомендації на основі контенту, зокрема, *item-to-item collaborative filtering*, де товари рекомендуються на основі подібності до інших придбаних користувачами.

Apple Music: Створює музичні станції рекомендованих пісень, враховуючи інтереси користувача та аналізуючи їхню поведінку. Використовує принципи спільної фільтрації.

Spotify: Використовує метадані пісень для генерації станцій з подібними властивостями, а також зворотний зв'язок від користувачів для уточнення рекомендацій. Використовує контент-орієнтований підхід.

Netflix: Використовує власну систему рекомендацій *CineMatch*, яка застосовується для персоналізованих рекомендацій фільмів. Netflix завжди

Кафедра КІТ				НАУ 23 01 37 000 ПЗ			
Розроб.	Мала Д.Ю.			РОЗДІЛ 2 ВИБІР ТА ОБГРУНТУВАННЯ ПРОГРАМНИХ ТА АПАРАТНИХ ЗАСОБІВ РЕЛІЗАЦІЇ	Літ.	Арк.	Аркушів
Керівник	Сінько Ю.І.					41	25
Н.Контр.	Толстікова О.В.				ТП-215М - 122		

ставить перед собою виклик користувачам перевершити показник середньоквадратичної помилки.

Кафедра КІТ

НАУ 23 01 37 000 ПЗ

<i>Розроб.</i>	<i>Мала Д.Ю.</i>			РОЗДІЛ 2 ВИБІР ТА ОБГРУНТУВАННЯ ПРОГРАМНИХ ТА АПАРАТНИХ ЗАСОБІВ РЕЛІЗАЦІЇ	Літ.	Арк.	Аркушів
<i>Керівник</i>	<i>Сінько Ю.І.</i>					41	25
<i>Н.Контр.</i>	<i>Толстікова О.В.</i>				ТП-215М - 122		

Ці приклади демонструють широкий спектр застосування рекомендаційних систем у великих інтернет-платформах та соціальних мережах для поліпшення користувацького досвіду та збільшення залучення.

Інформаційні технології досягли такого рівня розвитку, що тепер майже кожна компанія володіє власною рекомендаційною системою. Давайте приглянемося до того, як побудовані системи рекомендацій в інтернет-магазині Decathlon та на платформі YouTube.

Decathlon вирішила створити рекомендаційну систему товарів, ґрунтуючись на минулих покупках користувачів. Мета системи полягала у передбаченні наступного товару, який зацікавить конкретного користувача. Початково використовувалась система спільної фільтрації, реалізована через бібліотеку spark.ml. Цей пакет використовує алгоритм найменших квадратів (ALS), щоб розкрити приховані фактори матриці асоціації між користувачами та товарами.

Хоча спільна фільтрація є звичайним методом у рекомендаційних системах, виявилось, що її жорстка структура ускладнює використання корисних метаданих. Крім того, вона передбачає лінійну залежність між користувачами та товарами. Пізніші дослідження в області фреймворків глибокого навчання, таких як Netflix, YouTube та Spotify, показали, що вони випереджають класичні методи факторизації завдяки більш багатим нелінійним перетворенням вхідних даних та використанню корисних метаданих.

В новому підході Decathlon до глибокого навчання використовуються методи генерації тексту, в яких застосовуються моделі рекурентних нейронних мереж. Ці мережі здатні генерувати наступні елементи на основі попередніх, що було успішно використано для рекомендацій. Використані LSTM нейронні мережі разом із шаром самоуваги (Self-Attention).

Розглянемо принципи функціонування рекомендаційної системи на YouTube. Розробники YouTube сталики перед кількома важливими завданнями при створенні алгоритму:

1. **Розмаїття вмісту:** YouTube має величезну кількість відеороликів різної тематики, що ускладнює ефективний підбір рекомендацій.

2. **Динамічність обслуговування:** Кожну годину на платформу завантажуються сотні тисяч годин відео, тому система рекомендацій повинна бути гнучкою та динамічною.

3. **Зміна інтересів глядачів:** Інтереси глядачів є змінними, тому система повинна адаптуватися до їхніх змін.

4. **Оптимізація ресурсів:** Потрібно оптимізувати використання ресурсів для підбору рекомендацій, оскільки алгоритми підбору вимагають значних обчислювальних потужностей.

Система отримує мільйони відеороликів на вході і пропонує десятки відео на виході, які потрапляють користувачеві в розділ "Рекомендації". На рис. 2.1. можна побачити архітектуру рекомендаційної системи YouTube.

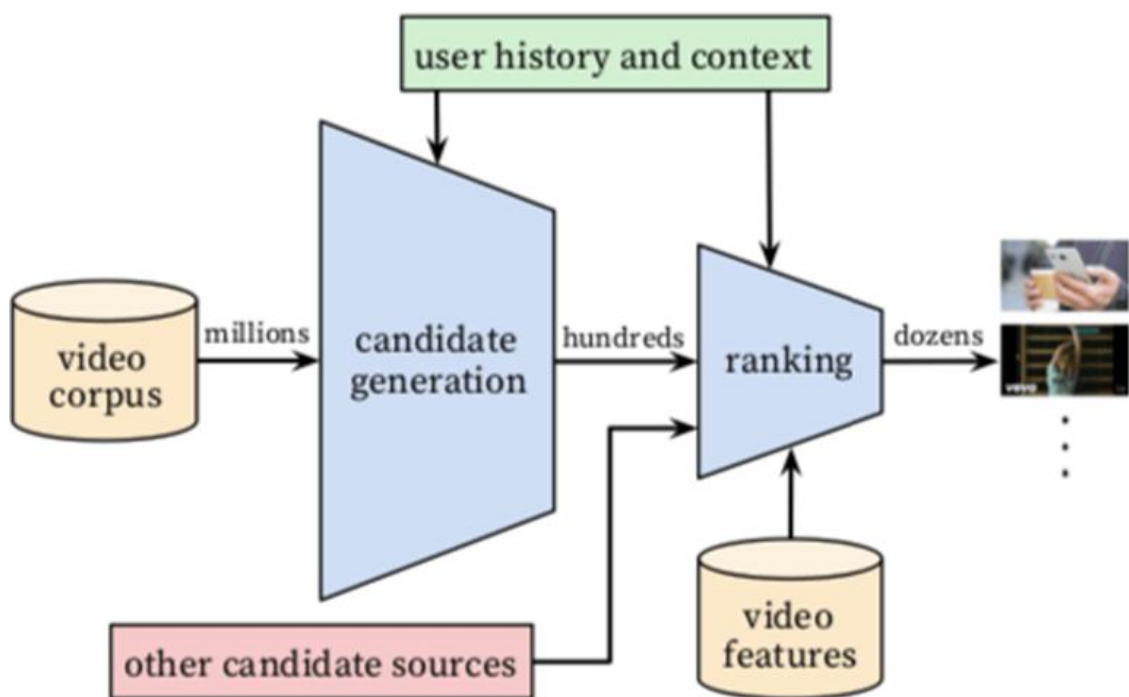


Рис. 2.1. Архітектура рекомендаційної системи YouTube.

Система рекомендацій на YouTube складається з двох згорткових нейронних мереж: "candidate generation" (генерація кандидатів) та "ranking" (ранжування). Перша мережа, оперуючи мільйонами відеороликів, відбирає сотні найбільш відповідних, тоді як друга нейромережа ранжує цю добірку в порядку цікавості для конкретного користувача. При формуванні вибірки

система бере до уваги всю історію перегляду користувача та контекст, який включає час доби, вік, стать та географічне розташування. Також використовується A/B тестування, де користувачеві показують різні вибірки для експерименту, і якщо якась виявляється більш перегляданою, система адаптується під неї через самонавчання.

Оцінка вибірки враховує не лише час перегляду, але й CTR (клік–по–відсоток) — кількість користувачів, які почали переглядати відеоролик в порівнянні з користувачами, які побачили відео в рекомендаціях. На етапі ранжування вибірка будується на основі показника очікуваного часу перегляду, і чим довше глядачі дивляться відео, тим вища його ймовірність потрапити у топ рекомендацій. YouTube не обмежується лише CTR, оскільки відео може привертати увагу просто через клікбейт. Метою навчання нейромережі ранжування є прогнозування часу перегляду відео.

Netflix – це стрімінговий сервіс, який надає можливість переглядати велику кількість фільмів та серіалів. Платформа пропонує користувачам список подібного контенту для кожного фільму чи серіалу, а також відображає відсоток подібності до інших творів. У Netflix існує різноманіття списків, таких як найпопулярніші фільми, найкращі за різними жанрами, а кожному користувачеві доступні персональні рекомендації. Зображення на рис. 2.2 та рис. 2.3 ілюструють різні списки фільмів та серіалів.

Недоліки Netflix включають обмежену кількість доступних фільмів та серіалів, а також відсутність можливості оцінювати чи залишати коментарі під фільмами.

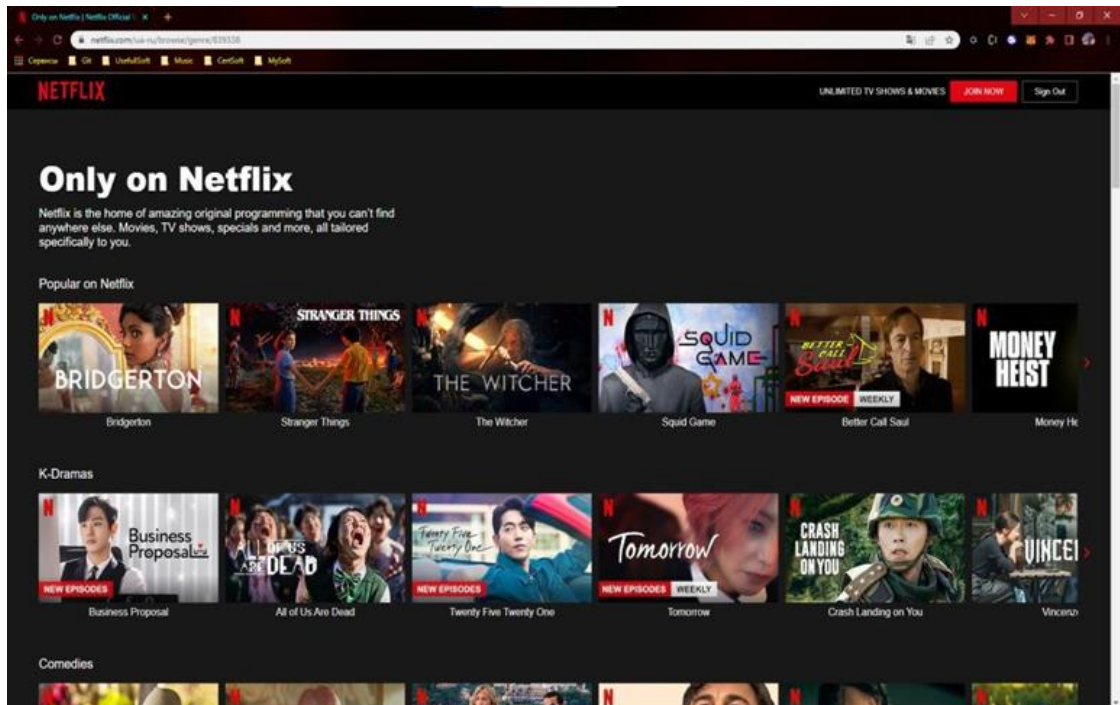


Рис. 2.2. Скріншот списку фільмів/серіалів рекомендацій на WEB-застосунку Netflix

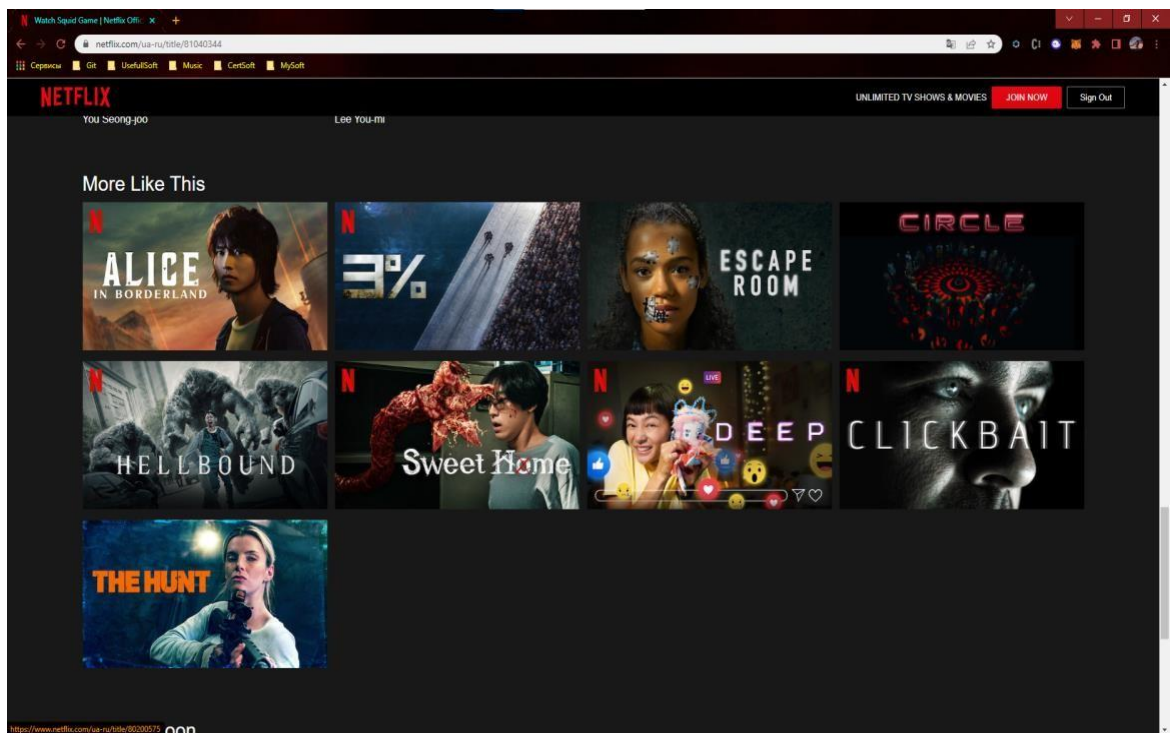


Рис. 2.3. Скріншот списку фільмів подібних до обраного на WEB-застосунку Netflix

IMDb (Internet Movie Database) - це онлайн база даних про фільми та серіали, яка включає інформацію про акторів, творчу команду, короткий опис сюжету, відгуки критиків та глядачів [5] (рис. 2.4 - 2.5).

Переваги IMDb включають:

- Доступ користувачів до сторінок фільмів, осіб, пов'язаних з фільмами та акторів;
- Можливість реєстрації користувачів для створення та редагування сторінок;
- Система оцінювання фільмів від 1 до 10, з урахуванням середньозважених оцінок;
- Фільтр захисту від підробок та піратства;
- Критичні огляди та рейтинги користувачів для кожного фільму;
- Список фільмів, які незабаром вийдуть у прокат;
- Можливість створення списку фільмів для перегляду;
- Багато різних списків кращих фільмів за різними характеристиками;
- Доступ до цікавої інформації про фільми та новини;
- Персональні рекомендації для зареєстрованих користувачів;
- Рекомендації на основі фільмів, які користувачі оцінили чи додали до списку спостереження;
- Рекомендації від інших користувачів із схожими вподобаннями, з окремим записом для кожної рекомендації.

Недолік IMDb включає те, що веб-застосунок доступний лише англійською мовою.

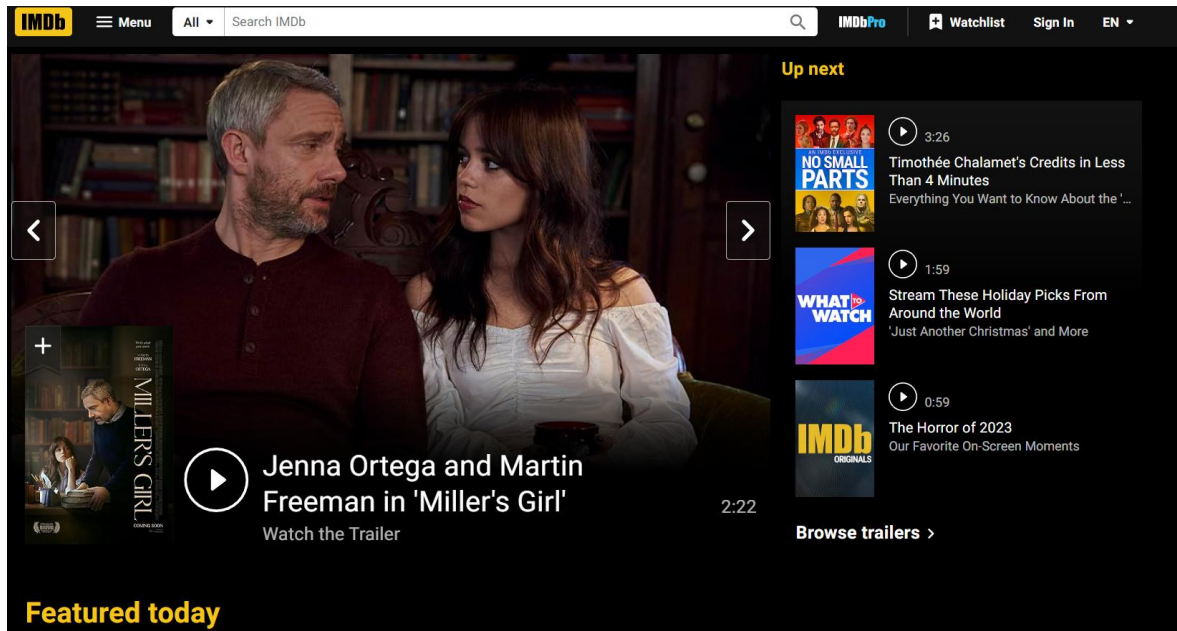


Рис. 2.4. Головна сторінка на WEB-застосунку IMDb

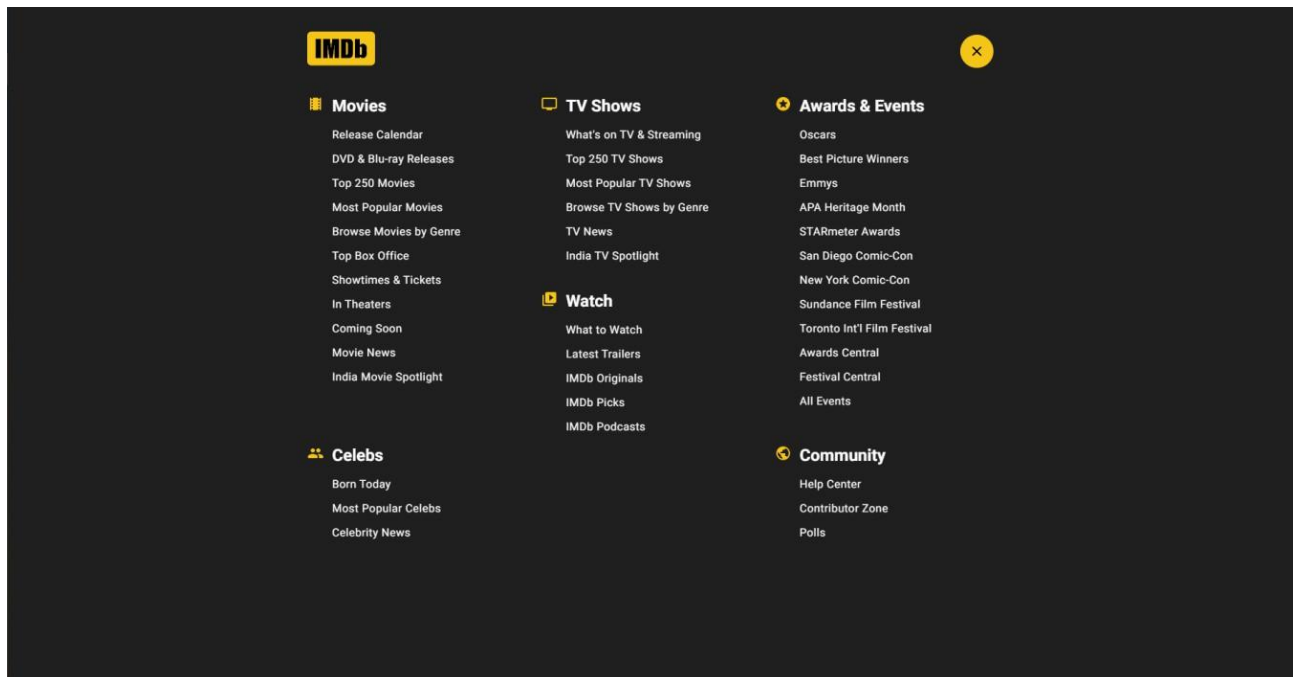


Рис. 2.5. Скріншот меню на WEB-застосунку IMDb

Metacritic - це платформа, що поєднує огляди музики, ігор та фільмів від провідних критиків. Сервіс розраховує середній рейтинг продуктів на основі зібраних оглядів, що активно використовується в індустрії комп'ютерних ігор та інших медійних сферах для прогнозування комерційного успіху нових випусків [7] (рис. 2.6).

Переваги Metacritic включають:

- Інформація про фільми, серіали та людей, пов'язаних з кіновиробництвом;
- Можливість залишати власні відгуки;
- Онлайн-покупка квитків в кіно;
- Список найкращих фільмів.

Недоліки Metacritic:

- Відсутність персональних рекомендацій для користувачів.

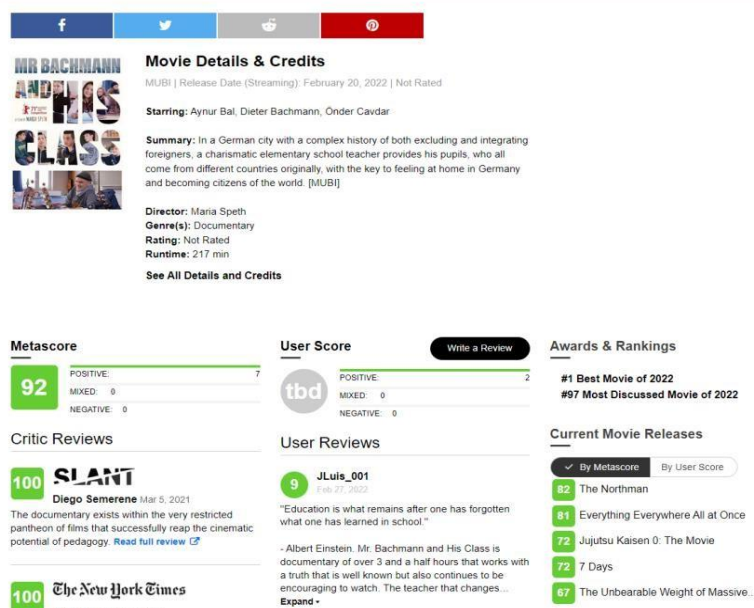


Рис. 2.6. Скріншот детальної інформації про фільм на WEB-застосунку Metacritic

Для реалізації рекомендаційної системи для пошуку фільмів та серіалів, яка задовольнятиме потреби користувача, можна використати такі компоненти та функціонал:

1. Авторизація та Створення Облікових Записів:

- Можливість реєстрації та авторизації користувачів для доступу до персоналізованих функцій.
- Кожен користувач отримує власний профіль, де відображаються його вибори та інші налаштування.

2. Встановлення Критеріїв Рекомендацій:

- Можливість користувача встановлювати свої уподобання та критерії для отримання персоналізованого списку рекомендацій.
- Врахування різних факторів, таких як жанр, рейтинги користувачів, актори, режисери тощо.

3. Перегляд Особистого Списку Рекомендацій:

- Зручний інтерфейс для перегляду рекомендацій, підготовлених системою на основі вказаних критеріїв.

4. Управління Списком Рекомендацій та Фільмами:

- Можливість додавати та видаляти фільми чи серіали до/зі списку рекомендацій.

5. Перегляд Схожих Фільмів/Серіалів:

- Функція, яка виводить користувачеві персоналізований список фільмів чи серіалів, схожих на його улюблені.

6. Аналіз та Оцінка:

- Врахування оцінок користувачів для покращення алгоритмів рекомендацій.
- Забезпечення можливості користувачам залишати відгуки та оцінки.

7. Веб-Застосунок:

- Реалізація веб-застосунка для легкого доступу та користування.

Ці функції можна реалізувати за допомогою мов програмування, таких як Python, JavaScript, та використовуючи фреймворки, наприклад, Django або Flask для серверної частини, та React або Vue.js для клієнтської частини. Управління базою даних може бути забезпечено за допомогою системи управління базами даних (наприклад, PostgreSQL або MySQL).

2.2. Методи машинного навчання

2.2.1. Навчання з вчителем

Навчання з вчителем – це клас задач, в якому модель вивчає закономірності між вхідними даними та цільовою змінною. У випадках, коли навчальні дані містять пари вхідних та відповідних цільових векторів, це називається проблемою навчання з вчителем.

Моделі, навчені на таких даних, використовуються для прогнозування на тестових наборах, де надаються лише вхідні дані. Вихідні дані моделі порівнюються з прихованими цільовими змінними для оцінки її навичок. Навчання полягає в пошуку гіпотез, які ефективно працюватимуть навіть на нових прикладах.

Існують два основних типи проблем у навчанні з вчителем: класифікація, яка передбачає мітку класу, і регресія, яка передбачає числове значення.

Класифікація: Модель передбачає мітку класу для вхідних даних.

Регресія: Модель передбачає числове значення для вхідних даних.

Ці задачі можуть включати одну або кілька вхідних змінних різних типів даних, таких як числові чи категоріальні. Наприклад, задачею класифікації може бути розпізнавання рукописних цифр за зображеннями, де вхідні – це пікселі, а вихідні – мітки класу. Задачею регресії може бути прогнозування цін на житло в Одесі, де вхідні – параметри околиць, а вихідні – ціни у гривнях.

Деякі алгоритми машинного навчання охарактеризовані як "контрольовані", оскільки вони призначені для вирішення завдань машинного навчання під керуванням. Серед популярних прикладів можна виділити дерева рішень, машини опорних векторів та інші.

Такі алгоритми отримали назву "контрольованими", оскільки вони вчать, роблячи передбачення на основі вхідних даних, і моделі піддаються контролю та коригуванню за допомогою алгоритму для забезпечення кращого передбачення очікуваних цільових результатів в навчальному наборі даних.

Термін "навчання під керівництвом" виник від ідеї, що цільові результати відповідають вказівкам викладача, який навчає систему машинного навчання.

Певні алгоритми можуть бути спеціально створені для класифікації, такі як логістична регресія, або для регресії, наприклад, лінійна регресія, в той час

як інші можуть застосовуватися до обох типів проблем з невеликими модифікаціями, наприклад, штучні нейронні мережі.

2.2.2. Навчання без вчителя

Навчання без нагляду - це категорія задач, які передбачають використання моделей для виявлення та вилучення зв'язків у наборі даних. На відміну від навчання з наглядом, де є визначені вихідні або цільові змінні, навчання без нагляду оперує лише вхідними даними без наявності учителя, який надає корекції моделі.

У цьому типі навчання відсутній інструктор чи викладач, і алгоритм повинен самостійно розуміти дані без зовнішнього керівництва. Існує багато видів навчання без нагляду, але дві основні проблеми, які часто виникають, - це групування (пошук груп у даних) і оцінка щільності (узагальнення розподілу даних).

- 1. Кластеризація:** Це завдання навчання без нагляду, яке передбачає виявлення груп у наборі даних. Наприклад, алгоритм k-середніх визначає k кластерів у вхідних прикладах даних.
- 2. Оцінка щільності:** Це завдання навчання без нагляду, яке полягає в узагальненні розподілу даних. Один із методів - оцінка щільності ядра, який використовує малі групи тісно пов'язаних вибірок для оцінки розподілу для нових точок у просторі проблеми.

Кластеризація є найпоширенішим завданням навчання без нагляду, і вона використовується для виявлення потенційно корисних кластерів вхідних прикладів даних. Наприклад, агент з нерухомості може використовувати цей підхід для виявлення різниці між "хорошими" і "поганими" місяцями з продажів, не отримуючи конкретних вказівок від вчителя.

Такі методи навчання без нагляду, як кластеризація та оцінка щільності, дозволяють виявити закономірності у наборі даних. Крім цього, використання додаткових методів, таких як візуалізація і проєкція, також може призвести до глибшого розуміння структури даних.

1. **Візуалізація:** Це завдання навчання без нагляду, яке передбачає створення графіків або діаграм для візуального представлення даних. Наприклад, матриця діаграм розсіювання може створити діаграму для кожної пари змінних у наборі даних, допомагаючи виявити можливі взаємозв'язки.
2. **Проекція:** Це завдання навчання без нагляду, яке передбачає зменшення розмірності даних, щоб створити їхнє представлення у просторі меншої вимірності. Наприклад, аналіз головних компонентів дозволяє підсумувати набір даних, видаливши лінійні залежності.

Ці методи важливі для виявлення груп схожих прикладів у даних, що називається кластеризацією, для оцінки розподілу даних у вхідному просторі, відомого як оцінка щільності, або для спроектування даних з високої вимірності до меншої з метою візуалізації. Ці методи допомагають розкрити приховані закономірності та забезпечують підтримку в прийнятті рішень на основі виявлених структур у даних.

2.2.3. Навчання з підкріпленням

Навчання з підкріпленням – це клас проблем, де агент вчиться діяти в середовищі, спираючись на зворотний зв'язок у вигляді числового сигналу винагороди. У такому навчанні, агент експериментує з різними діями та визначає їхню ефективність через отримані винагороди. Важливою особливістю є те, що агент не отримує конкретні інструкції щодо дій; замість цього він повинен вивчити оптимальні стратегії, максимізуючи винагороду.

У цьому контексті середовище не має фіксованого набору даних, і агент повинен досягти цілей, взаємодіючи з ним. Процес навчання заснований на циклі взаємодії агента з середовищем та зворотньому зв'язку на основі отриманих винагород.

Навчання з підкріпленням схоже на навчання з наглядом, але відмінність полягає в тому, що взаємодія з навколишнім середовищем не передбачає

жорстко заданих прикладів, і агент повинен самостійно дізнатися, які дії приводять до бажаних результатів.

Типовим прикладом задачі з підкріплення є гра, де агент намагається досягти високого рівня успіху, приймаючи послідовні дії та отримуючи винагороди чи покарання в залежності від результатів кожної дії.

Навчання з підкріпленням є ключовим методом в областях, де важко забезпечити точні та послідовні оцінки для великої кількості ситуацій, які потребують навчання. Наприклад, у грі людина може важко надавати докладні та послідовні оцінки багатьох позицій, але програму можна вивчити за допомогою винагороди чи покарання за виграш або програш. Це дозволяє програмі вивчати функцію оцінювання ймовірності виграшу з будь-якої позиції.

Одним із вражаючих прикладів успішного використання навчання з підкріпленням є AlphaGo від Google, яке змогло перемогти найкращого гравця світу в гру Go.

Деякі популярні алгоритми навчання з підкріпленням включають Q-навчання, навчання з тимчасовими відмінностями та глибоке навчання з підкріпленням. Ці методи дозволяють агентам вивчати оптимальні стратегії дій у варіативних середовищах, опираючись на винагороди та покарання.

2.3. Архітектура інтелектуальної системи

Зазвичай, система пропонує користувачеві через системний інтерфейс висловлювати свої оцінки для об'єктів з метою вдосконалення своєї моделі. Точність рекомендацій залежить від кількості наданих користувачем оцінок [2]. Однак цей метод має свій недолік у вигляді потреби у зусиллях від користувачів, які не завжди готові надавати достатньо інформації. Якщо дивитися на явний зворотний зв'язок, хоча він вимагає більше участі користувача, його все ж розглядають як джерело надійних даних. Це через те, що він не ґрунтується на видаленні власних уподобань користувача та

забезпечує прозорість у процесі рекомендацій, що призводить до кращого сприйняття рекомендацій та більшої впевненості в їх якості [7].

Прогнозування якісних рекомендацій включає отримання відгуку від користувача [8]. Фаза оновлення інформації бази даних фільмів користувача зображена на рис. 2.7. Система автоматично визначає уподобання користувача, враховуючи його дії, такі як історія покупок, навігація та тривалість перебування на певних веб-сторінках, переходи за посиланнями, зміст електронної пошти та інші. Неявний зворотний зв'язок полегшує завдання користувачів, виводячи уподобання з їхньої поведінки в системі. Хоча цей метод не вимагає активності від користувача, він менш точний. Також вказується, що неявні дані можуть бути об'єктивнішими, оскільки вони не піддаються впливу соціальної бажаності та не породжують проблем самовизначення чи потреби в підтримці іміджу перед іншими.

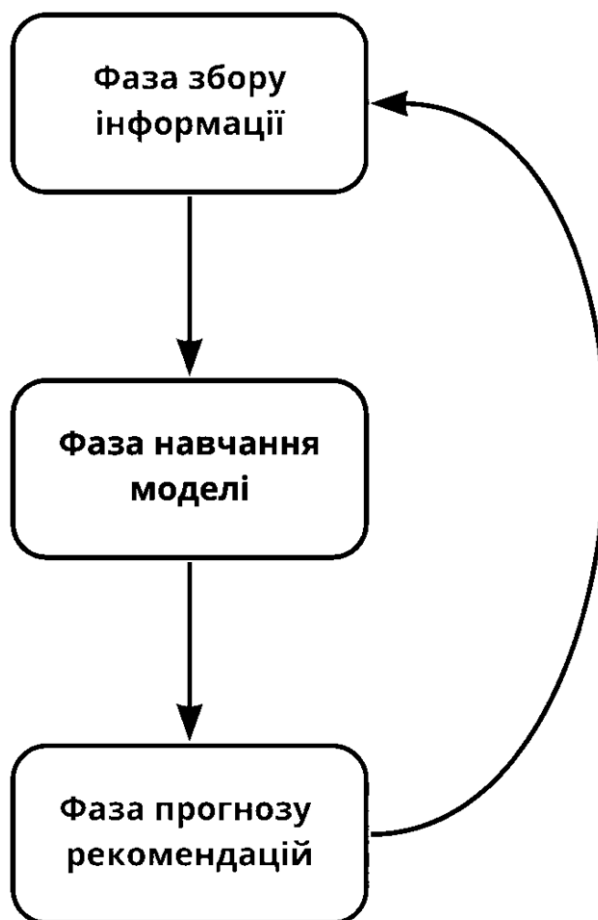


Рис. 2.7. Процес оновлення рекомендації користувачеві

Система надає користувачеві рекомендації або прогнозує, які предмети можуть бути йому цікаві [13]. Це може відбуватися безпосередньо на основі даних, зібраних на етапі збору інформації, який може бути оснований на пам'яті або моделі, або завдяки спостережуваній системою користувачів діяльності, яка визначає етапи рекомендацій.

Колаборативна фільтрація - це техніка прогнозування вмісту, яка не залежить від конкретної галузі і не може легко або адекватно описатися метаданими, такими як фільми та музика. Метод колаборативної фільтрації працює, будуючи базу даних (матрицю елементів користувача), що містить уподобання користувачів до різних елементів. Потім він встановлює відповідність між користувачами з подібними інтересами та уподобаннями, обчислюючи схожість між їхніми профілями для надання рекомендацій. Такі користувачі формують групу, відому як "сусідство". Користувач отримує рекомендації щодо елементів, які він раніше не оцінював, але які вже отримали позитивні оцінки від користувачів у його сусідстві. Рекомендації, створені методом колаборативної фільтрації, можуть бути як передбачення, так і конкретні рекомендації (рис. 2.8).


























				
				
				
				
				
				

Рис. 2.8. Процес колаборативної фільтрації

Загалом, рекомендаційна модель передбачає наявність заздалегідь підготовленої бази даних користувача, яка з кожним новим додаванням фільмів з оцінками буде оновлювати прогнози моделі для користувача використанням сервісу [15].

При першому використанні системи користувачем (тобто при холодному старті), система використовує штучно створену базу фільмів, яка включає популярні стрічки. Такий підхід дозволяє використовувати концепцію колаборативної фільтрації при відсутності відомостей про конкретного користувача. Під час оновлення бази фільмів користувача модель оновлюється та генерує рекомендації вже на основі нових оцінок.

База даних призначена для зберігання всієї довгострокової інформації системи, включаючи зареєстрованих користувачів, їхні оцінки фільмів,

особисті дані, а також довідники жанрів і років випуску. Крім того, існує тимчасова таблиця з рекомендаціями, куди записуються передбачені рейтинги для фільмів, які ще не були оцінені поточним користувачем (рис. 2.9).

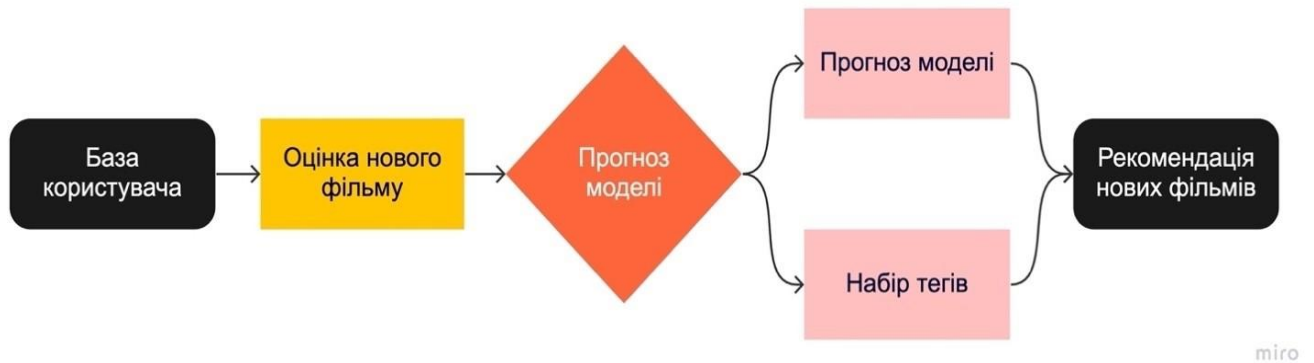


Рис. 2.9. Побудова прогнозу рекомендацій

Прогноз рекомендацій користувачеві здійснюється за допомогою набору тегів та попередньо натренованої моделі нейронної мережі. У випадку прогнозу рекомендацій фільмів, теги включають список жанрів, які відповідають кожному фільму в загальній базі даних фільмів. Теги та оцінка користувача розміщені в інтерфейсі користувача (рис. 2.10).

Процес отримання прогнозу рекомендацій передбачає, що користувач натискатиме на зірочки, вказуючи п'ятибальну шкалу оцінювання, де одна зірочка означає, що користувач оцінює фільм в один бал, а п'ять зірок відповідно вказують на оцінку в п'ять балів. Після того, як користувач визначить кількість балів для фільму, наступним кроком є підтвердження вибору користувача. Натисканням кнопки «Select» користувач ініціює процес оновлення прогнозу рекомендацій (рис. 2.10).

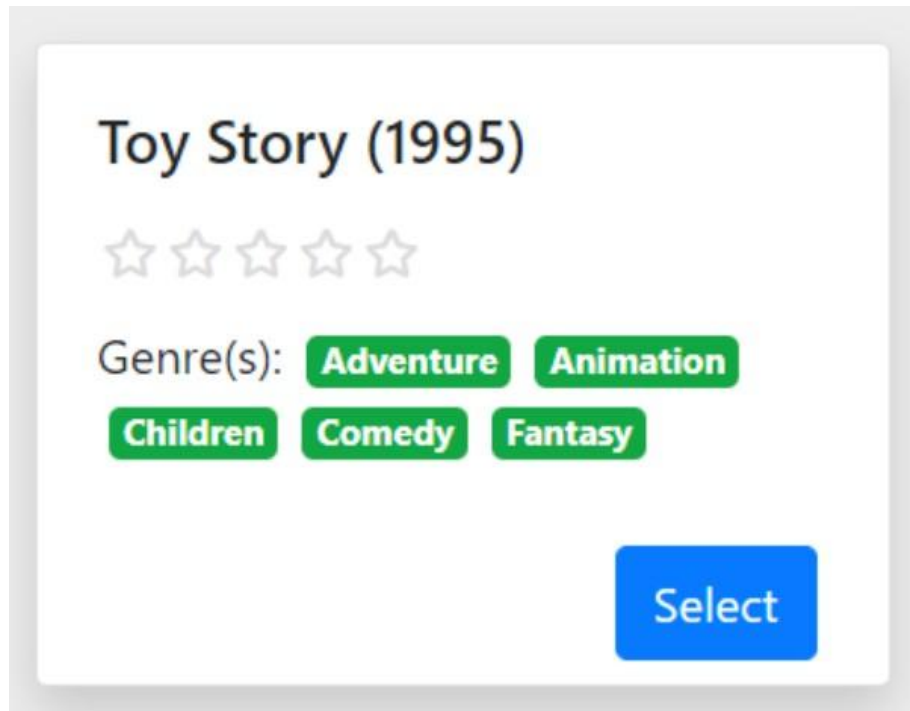


Рис. 2.10. Набір тегів та вподобань в інтерфейсі користувача

Даний вигляд користувацького інтерфейсу, дозволяє швидко орієнтуватися в жанрах до фільмів, та дає можливість користувачеві корегувати виставлені оцінки до фільмів.

Сучасна архітектура веб-розробки ґрунтується на клієнтському JavaScript, використанні багаторазових API та попередньо встановленій розмітці [17]. Ця архітектура розділяє функціональну відповідальність на три ключові компоненти:

- JavaScript обробляє всі динамічні функції.
- API (Application Program Interface) використовується для взаємодії з серверною стороною, такою як взаємодія із сторонніми сервісами чи використання спеціалізованих функцій.
- Розмітка веб-сайту представляє собою файли HTML (Hypertext Markup Language), які можна генерувати з вихідних файлів за допомогою статичних генераторів сайтів.

Використання методології JAMStack призводить до численних переваг:

- Збільшена швидкість завдяки передбаченій розмітці та активам, що постачаються через CDN (Content Delivery Network).

- Підвищена безпека завдяки відсутності прямого зв'язку із серверами чи базами даних.
- Краща масштабованість для розширення веб-системи за рахунок використання CDN.
- Зменшені витрати на зберігання веб-системи, оскільки хостинг статичних файлів розмітки є дешевшим.
- Швидкий і фокусований процес розробки, де розробники можуть повністю зосередитися на своїй роботі, не обмежуючись монолітною архітектурою.

Веб-сайти також можуть керуватися за допомогою систем управління контентом, часто відомих як CMS (Content Management System), які використовуються для централізованого управління контентом. Після оновлення в CMS запускається нова компіляція та розгортання сайту як статичних активів (static assets). Абстрактний робочий процес JAMStack зображений на рис. 2.11.

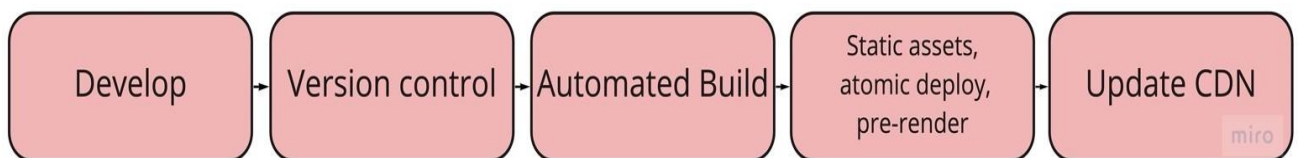


Рис. 2.11. Абстрактний процес JAMStack

2.4. Засоби для створення моделі рекомендаційної системи

Для розробки рекомендаційної системи на основі методу колаборативної фільтрації була обрана мова програмування Python. Ця мова має стрімкий розвиток і підтримку основних фреймворків для створення нейронних мереж. Python дозволяє як навчати модель нейронної мережі, так і створювати API для інтеграції системи прогнозу рекомендацій у веб-інтерфейс користувача.

Серед основних фреймворків для створення нейронних мереж на Python виділяються:

1. TensorFlow: Створений Google і написаний на Python і C++, TensorFlow є однією з найкращих відкритих бібліотек для чисельних обчислень. Використовується для складних проектів, таких як розпізнавання голосу, зображень і текстові додатки, такі як Google Translate.

2. PyTorch: Розроблений Facebook, PyTorch є конкурентом TensorFlow і використовується для швидкого і ефективного тренування моделей. Знайшовши застосування в таких компаніях, як Twitter, Salesforce і Оксфордський Університет.

3. Keras: Мінімалістична бібліотека, яка може запускатися поверх TensorFlow, Theano або CNTK. Вона спроектована для прискорення експериментів і підтримує широкий спектр шарів нейронних мереж.

Для реалізації рекомендаційної системи був обраний фреймворк Keras, який є зручним та ефективним для створення моделей нейронних мереж.

Штучна нейронна мережа (ANN) представляє собою систему взаємопов'язаних вузлів і вагованих зв'язків, яка інспірується архітектурою біологічного мозку. У ANN вузли називаються нейронами, а їхнє функціональне об'єднання утворює мережу, яка може вивчати задачі класифікації або регресії після навчання на достатній кількості даних[18].

Один з найпростіших випадків ANN - це модель перцептрон, яка зображена на рис. 2.7. Якщо деталізувати функцію активації ϕ як просту порогову функцію, вихід отримується шляхом підсумовування вхідних значень, взятих з урахуванням їхніх ваг та порівняння результату з деяким порогом θ_k . Формула виражає вихідну функцію. Модель перцептрон є лінійним класифікатором з простим та ефективним алгоритмом навчання. Однак, окрім простої порогової функції, використаної в моделі Perceptron, існують різні загальні варіанти функцій активації, такі як сигмоїдальна, гіперболічний тангенс або функції relu.

$$y_k = \begin{cases} 1, & \text{if } \sum x_i v_{ki} \geq \theta_k \\ 0, & \text{if } \sum x_i v_{ki} < \theta_k \end{cases}$$

Штучна нейронна мережа (ANN) може включати будь-яку кількість шарів. Ці шари класифікуються на три типи: вхідні, приховані та вихідні.

– Вхідні шари: Блоки вхідного шару реагують на дані, які вводяться в мережу. Кількість блоків в цьому шарі зазвичай відповідає кількості вхідних параметрів або ознак, які подаються на вхід мережі[25].

– Приховані шари: Блоки прихованих шарів отримують зважений результат від вхідних блоків. Ці шари виконують обчислення та вивчають складні залежності в даних. Кількість та конфігурація прихованих шарів може варіюватися в залежності від архітектури мережі.

– Вихідні шари: Блоки вихідного шару реагують на зважений вихідний результат від прихованих блоків. Вони генерують кінцевий вихід мережі, який може використовуватися для класифікації, регресії або інших завдань в залежності від природи задачі.

Така архітектура дозволяє нейронній мережі вивчати та узагальнювати складні залежності в даних, роблячи ANN потужним інструментом для різних завдань машинного навчання.

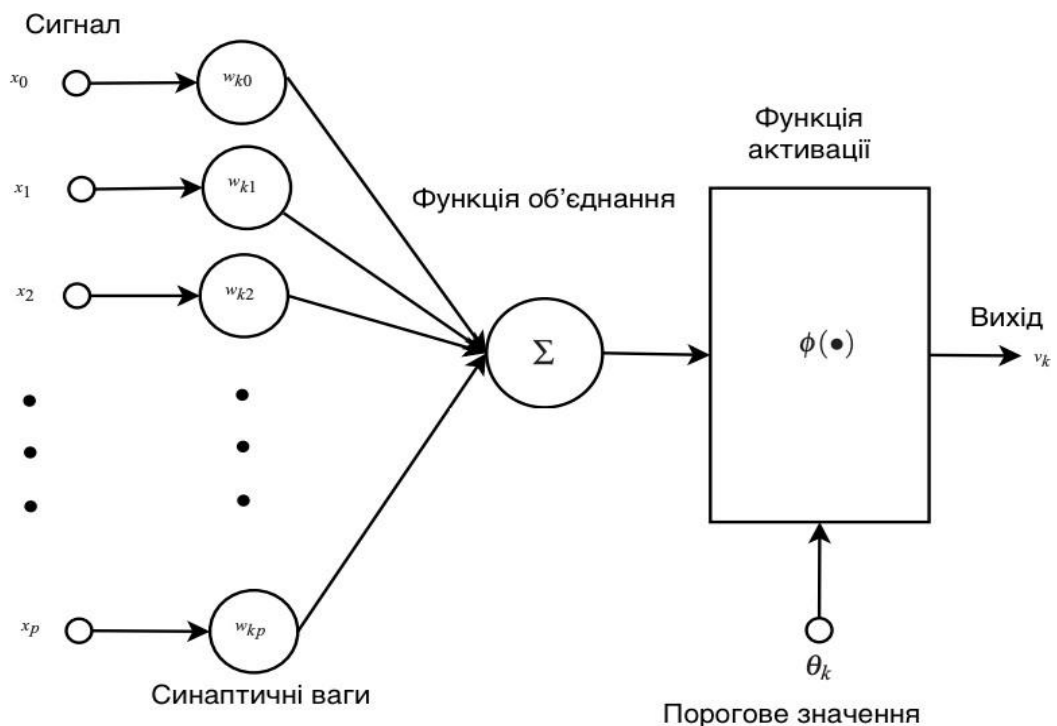


Рис. 2.12. Побудова перецептронну

Використовуючи нейрони як атомні функціональні одиниці, існує безліч можливих архітектур, що об'єднують їх у мережу. Однак найпоширеніший підхід полягає у використанні Штучних Нейронних Мереж (ANN) із зворотнім зв'язком. У цьому випадку сигнали суворо поширюються одним способом: від входу до виходу.

Основними перевагами ANN є те, що вони, залежно від функції активації, можуть виконувати завдання нелінійної класифікації. Додатково, завдяки своїй паралельній природі, вони можуть бути ефективними і навіть працювати в разі виходу з ладу частини мережі. Однак недоліком є важкість вибору ідеальної топології мережі для конкретної проблеми. Як тільки топологія визначена, вона виступає як нижня межа помилки класифікації. ANN відносяться до класу субсимволічних класифікаторів, що означає, що вони пропагують "чорну скриньку" підходу без надання семантики виведення знань[16].

ANN можна використовувати для побудови рекомендаційних систем, подібно до наївного байєсівського класифікатора. Однак результати експериментів не є однозначними щодо приросту продуктивності в порівнянні з іншими методами, такими як дерева прийняття рішень. У деяких випадках ANN та наївний байєсівський класифікатор демонструють аналогічну ефективність.

Наявна на рис. 2.13 порівняльна ілюстрація результатів експериментів різних моделей з базовою точністю RMSE стрімінгового сервісу Netflix. Визначена точність алгоритму Netflix становить 0.95, що еквівалентно наближено одному балу при оцінці фільмів за п'ятибальною системою.

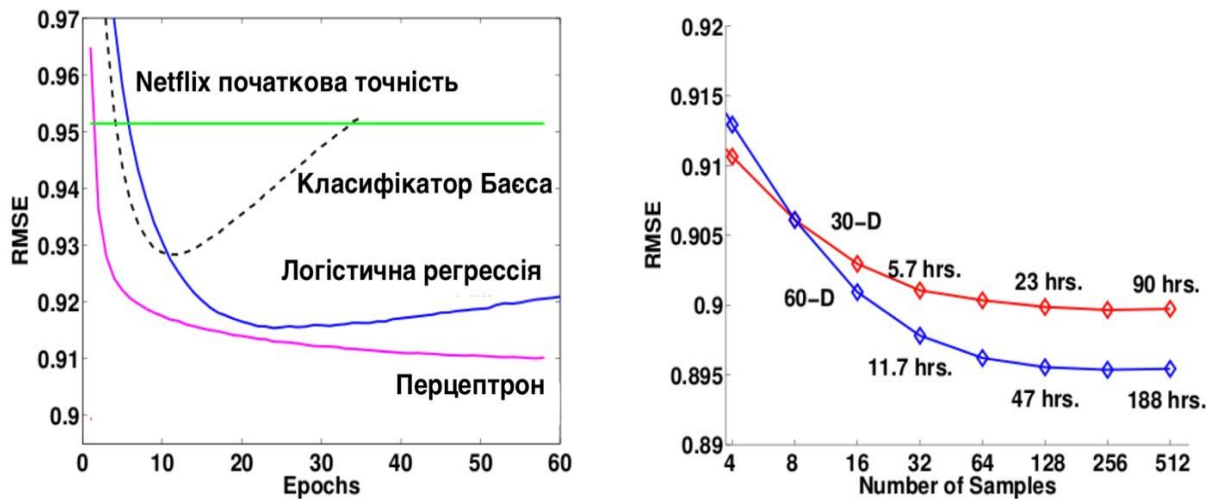


Рис. 2.13. Порівняння із початковою точністю Netflix

Після ретельного вивчення області та аналізу рекомендаційних систем та веб-розробки, було вирішено створювати програмний комплекс на основі колаборативної фільтрації. Для цього використовувався front-end фреймворк Node.js та бібліотека Keras для створення моделі нейронної мережі, при цьому Python використовувався як мова програмування. З основоположними принципами методології JAMStack приділялася значна увага для розробки та розгортання веб-систем[19].

Вибір колаборативної фільтрації базувався на можливості створення власної архітектури нейронної мережі та проведенні експериментів. Незважаючи на недоліки цього методу, було вирішено використовувати його разом із створеними евристичними та тегами (жанрами) для прогнозування рекомендацій.

Вибір фреймворку Keras для тренування нейронної мережі обумовлено його перевагами, такими як швидке прототипування, легкість використання, конфігуруємість, інтуїтивний інтерфейс та підтримка для навчання на різних апаратних платформах.

Бекенд програмного продукту реалізовано на Flask, що дозволяє ефективно створювати REST API для взаємодії з веб-сайтом. Фреймворк Node.js був обраний як оптимальний для створення легких та швидких веб-застосунків з урахуванням методології JAMStack.

Використання Node.js забезпечує високу швидкість та ефективну реалізацію веб-додатків.

JAMStack впроваджує новітній підхід до створення веб-сайтів, забезпечуючи покращену продуктивність, високий рівень безпеки, знижені витрати на масштабування та зручний досвід для розробників. Його основна перевага полягає в відсутності тісного зв'язку між клієнтом та веб-сервером, що забезпечує найкращу швидкість виконання.

У даному розділі було проведено дослідження засобів розробки рекомендаційних систем та нейронних мереж. Основна мета дослідження полягала у виборі підходів та інструментів, які оптимально підходять для створення рекомендаційної системи з використанням нейронних мереж.

В якості метрики точності було обрано RMSE (Root Mean Square Error), яка є загальноприйнятою метрикою для оцінки точності рекомендаційних систем. Ця метрика дозволяє виміряти відхилення прогнозованих значень від фактичних, виражених у квадратних одиницях, що робить її зручною для порівняння різних систем та методів.

На підставі проведеного дослідження був обрано програмний комплекс на базі колаборативної фільтрації з використанням нейронних мереж. Вибір колаборативної фільтрації обумовлений можливістю створення власної архітектури нейронної мережі та проведення експериментів для оптимізації результатів. Важливою перевагою є можливість використання тегів, зокрема жанрів, для побудови прогнозів рекомендацій [7].

Для реалізації нейронної мережі та тренування моделі було обрано бібліотеку Keras, яка відзначається швидкістю прототипування, легкістю використання та конфігуруємістю. Також, для створення бекенду програмного продукту використовується фреймворк Flask, що забезпечує ефективне створення REST API для взаємодії з веб-сайтом. Для розробки веб-інтерфейсу та використання методології JAMStack, вибраним є front-end фреймворк Node.js.

Обрана комбінація інструментів та підходів спрямована на створення ефективної та точної рекомендаційної системи з інтуїтивним інтерфейсом користувача.

.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ В ПРОГРАМНОМУ СЕРЕДОВИЩІ

3.1. Архітектура рекомендаційної системи

Штучні нейронні мережі є математичними моделями, а також їх програмні або апаратні реалізації, які спроектовані за принципом організації та функціонування біологічних нейронних мереж – мереж нервових клітин живого організму. Це поняття виникло для емуляції процесів, які відбуваються у мозку, та спроби моделювання цих процесів. Першим спробили створити такі моделі нейронних мереж Маккалока та Піттса. Згодом, після розробки алгоритмів навчання, отримані моделі стали використовувати у різноманітних практичних задачах, таких як прогнозування, розпізнавання образів, управління та інші.

Клас завдань, які можна вирішити за допомогою нейронних мереж, залежить від того, як мережа працює та як вона навчається. Під час роботи нейронна мережа приймає значення вхідних змінних і генерує значення вихідних змінних. Таким чином, мережу можна використовувати у випадках, коли ви маєте деяку відому інформацію і хочете отримати з неї поки що невідому інформацію.

Зазвичай нейронні мережі використовуються, коли точний характер зв'язків між входами та виходами невідомий, оскільки у випадку відомих зв'язків їх можна було б моделювати безпосередньо. Ще однією ключовою особливістю нейронних мереж є те, що залежність між входом і виходом розвивається під час навчання мережі.

Кафедра КІТ				НАУ 23 01 37 000 ПЗ			
Розроб.	Мала Д.Ю.			РОЗДІЛ 3 РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ В ПРОГРАМНОМУ СЕРЕДОВИЩІ	Літ.	Арк.	Аркушів
Керівник	Сінько Ю.І.					65	15
Н.Контр.	Толстікова О.В.				ТП-215М - 122		

При створенні штучної нейронної мережі ми використовуємо ту саму структуру, що і в біологічних нейронних мережах. Штучна мережа складається з нейронів, які взаємодіють між собою, проте це спрощена модель реальної біологічної нейронної мережі [3].

Повно зв'язкова нейронна мережа має особливість у тому, що кожен вузол, за винятком вхідних і вихідних, виступає як вхід та вихід одночасно, утворюючи прихований шар нейронів. Кожен нейрон наступного шару з'єднаний з усіма нейронами попереднього. Вхідні дані подаються з використанням ваг, які налаштовуються під час навчання і залишаються незмінними після цього процесу. Кожен нейрон має поріг активації, і після його перевищення він приймає одне з двох можливих значень: -1 або 1, або 0 або 1. Структуру багатозарової нейронної мережі можна розглядати, подивившись на рис. 3.1.

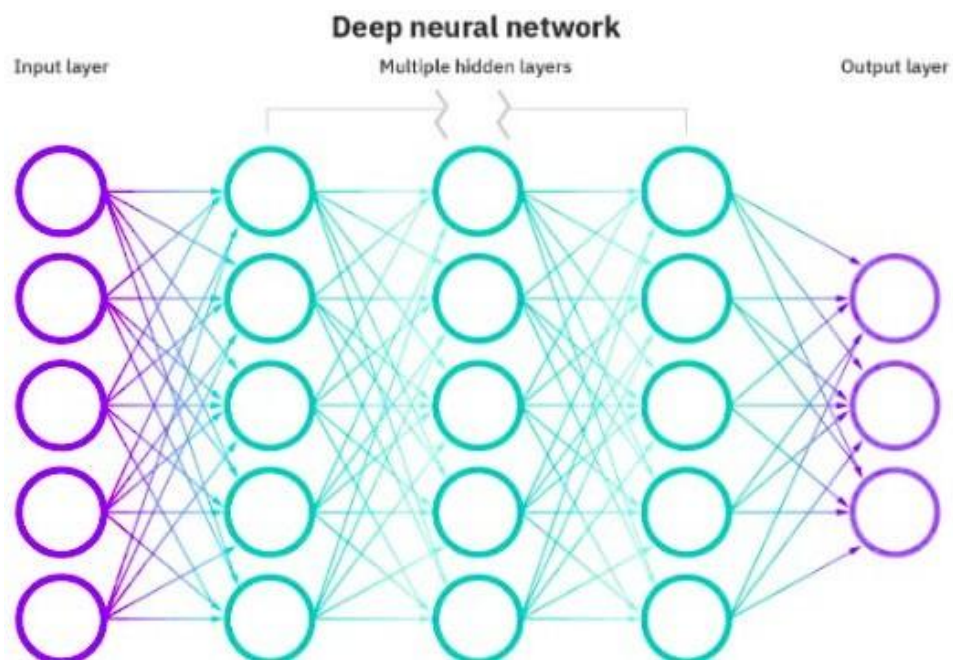


Рис. 3.1. Структура багатозарової нейронної мережі

Існують різні типи багатозарових нейронних мереж, серед яких можна виділити:

- 1. Монотонні нейронні мережі:**

1.1. Вони характеризуються тим, що кожен шар, крім останнього (вихідного), розділяється на два блоки: збуджуючий і гальмуючий.

1.2. Зв'язки між блоками також поділяються на гальмівні та збуджуючі.

1.3. Якщо від нейронів блоку до нейронів ведуть тільки збудливі зв'язки, то будь-який вихідний сигнал блоку є монотонною незнищувальною функцією будь-якого вихідного сигналу блоку.

1.4. Якщо ці зв'язки лише гальмують, то будь-який вихідний сигнал блоку є функцією, що не збільшується, будь-якого вихідного сигналу блоку.

1.5. Нейрони монотонних мереж потребують монотонної залежності вихідного сигналу нейрона від параметрів вхідних сигналів.

2. Мережі без зворотного зв'язку:

2.1. У таких мережах нейрони вхідного шару отримують вхідні сигнали, перетворюють їх і передають нейронам першого прихованого шару, і так далі аж до вихідного, який видає сигнали для інтерпретатора та користувача.

2.2. Кожен вихідний сигнал q -го шару може подаватися на вхід всім нейронам $(q+1)$ -го шару, хоча можливі інші варіанти з'єднання q -го шару з довільним іншим шаром.

Програмний комплекс для взаємодії з рекомендаційною системою складається з двох основних компонентів: фронтенду та моделі рекомендаційної системи, яка виконує прогнозування рекомендацій для користувачів і реалізована у бекенді через REST API. Основна увага приділена рекомендаційній системі, в той час як інтерактивна система користувача була розроблена з метою вивчення результатів навчання нейронної мережі.

Процес навчання нейронної мережі буде докладно розглянутий в подальшому в пояснювальній записці. Важливим функціональним елементом є можливість завантаження моделі, а сценарії та структура цього процесу будуть детально описані у подальших частинах звіту.

Відповідно до Netflix Prize, рекомендаційна система повинна досягати точності не менше 0,9525 RMSE на наборі даних MovieLens. Цей набір даних включає 100836 рейтингів та 3683 теги для 9742 фільмів. Ці дані були створені 610 користувачами з 29 березня 1996 року по 24 вересня 2018 року.

Програмний продукт, який взаємодіє з рекомендаційною системою, повинен відповідати наступним вимогам:

Інтерфейси користувача (User Interfaces):

Система повинна надавати повну навігацію та вибір інтерактивних дій за допомогою лише миші.

REST API:

Система повинна підтримувати взаємодію з веб-додатком через POST та GET запити.

Вимоги продуктивності (Performance Requirements):

Веб-сторінки, створені системою, повинні повністю завантажуватися за не більше ніж 5 секунд.

Генерація прогнозів рекомендаційної системи для користувача повинна відбуватися за не більше ніж 2 секунди після того, як користувач відправить оцінку фільму[14].

Загальна швидкість побудови прогнозів (інференс) рекомендаційною нейронною мережею повинна бути швидше за 10000 одиниць за 5 секунд.

Програмний комплекс для взаємодії з рекомендаційною системою включає два основні компоненти: фронтенд та модель рекомендацій, що працює у бекенді через REST API. Цей комплекс розроблено на базі колаборативної фільтрації, використовуючи фреймворк Node.js для фронтенду та бібліотеку Keras для моделі нейронної мережі мовою програмування Python. При цьому велика увага приділяється методології JAMStack для розробки та розгортання веб-систем.

Колаборативна фільтрація була обрана через можливість створення власної архітектури нейронної мережі та експериментування. Незважаючи на недоліки цього методу, вирішено використовувати його, додавши вручну створені евристики для уникнення його недоліків та використання тегів, зокрема жанрів, для прогнозування рекомендацій [21].

Фреймворк Keras обрано для тренування нейронної мережі через його переваги, такі як простота прототипування, легкість використання та конфігурування, а також підтримка навчання на декількох GPU.

Для взаємодії із веб-додатком використовується REST API, реалізоване на базі фреймворку Flask мовою Python. Фронтенд реалізовано на Node.js, який є оптимальним для створення легких та швидких веб-застосунків, особливо в рамках методології JAMStack.

JAMStack забезпечує високу продуктивність, безпеку та низькі витрати на масштабування веб-систем, а також полегшує роботу розробників, забезпечуючи відсутність щільного зв'язку між клієнтом та веб-сервером.

Таким чином, програмний комплекс використовує передові технології для створення ефективної та продуктивної рекомендаційної системи з інтерактивним веб-інтерфейсом.

Детальна діаграма прецедентів (або діаграма варіантів користування UseCase) мови проектування UML для описання моделі функціональних відношень між актором та прецедентами в системі представлена на рис. 3.2.

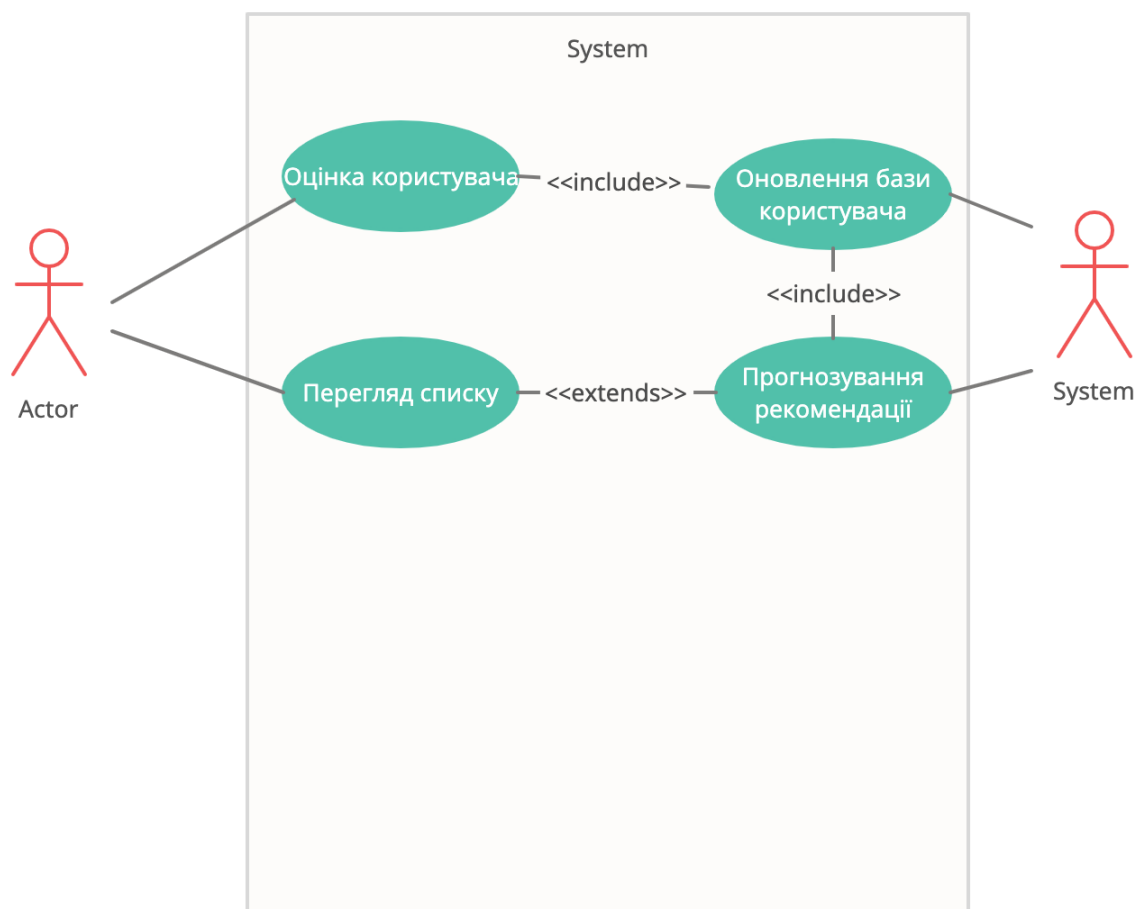


Рис. 3.2. Діаграма прецедентів веб-системи

Основною функцією, яку повинен виконувати дизайн рекомендаційної системи, є забезпечення інтуїтивно зрозумілого користування. З цією метою запропоновано використовувати список фільмів у вигляді блоків, де кожен блок дозволяє користувачам додавати оцінки за шкалою від 1 до 5. Кожна оцінка фільму сприяє формуванню загального рейтингу фільмів від користувачів на сервісі [11]. Дизайн інтерфейсу користувача веб-системи наведено на рис. 3.3. Вибір палітри кольорів здійснено з урахуванням розподілу: синій, зелений, жовтий.

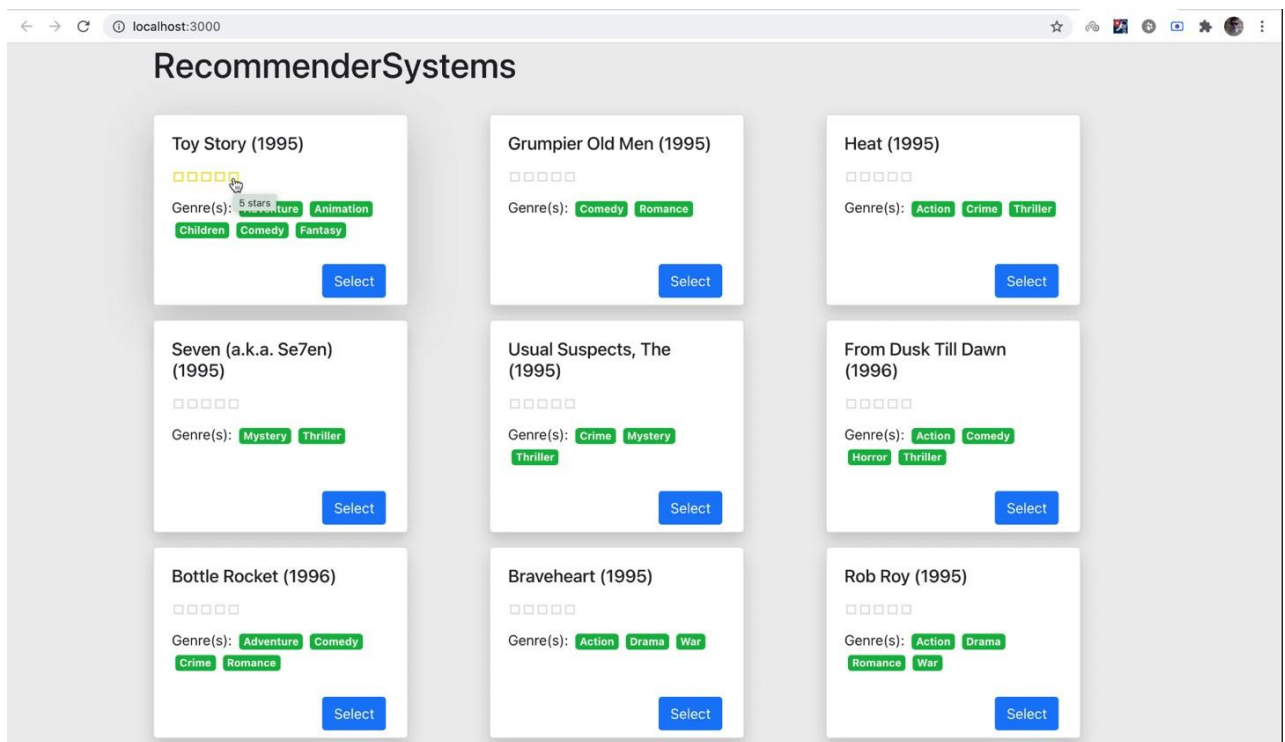


Рис. 3.3. Веб-інтерфейс користувача

Простий та зрозумілий дизайн спрямований на те, щоб зосередити увагу користувача лише на необхідному вмісті веб-сторінки рекомендаційної системи. Взаємодія користувача обмежена двома основними елементами: шкалою оцінки фільму та кнопкою "Select". Після того, як користувач оцінює фільм, список фільмів на головній сторінці оновлюється відповідно до зазначеної оцінки, яку користувач додав до фільму у наявному переліку [10].

Модель для отримання прогнозів рекомендаційної системи включає в себе структуру штучної нейронної мережі та набір евристик, які використовуються для додаткового фільтрування за наборами тегів, відповідно до визначених завдань. Загальна конфігурація рекомендаційної системи відображена на рис. 3.4.

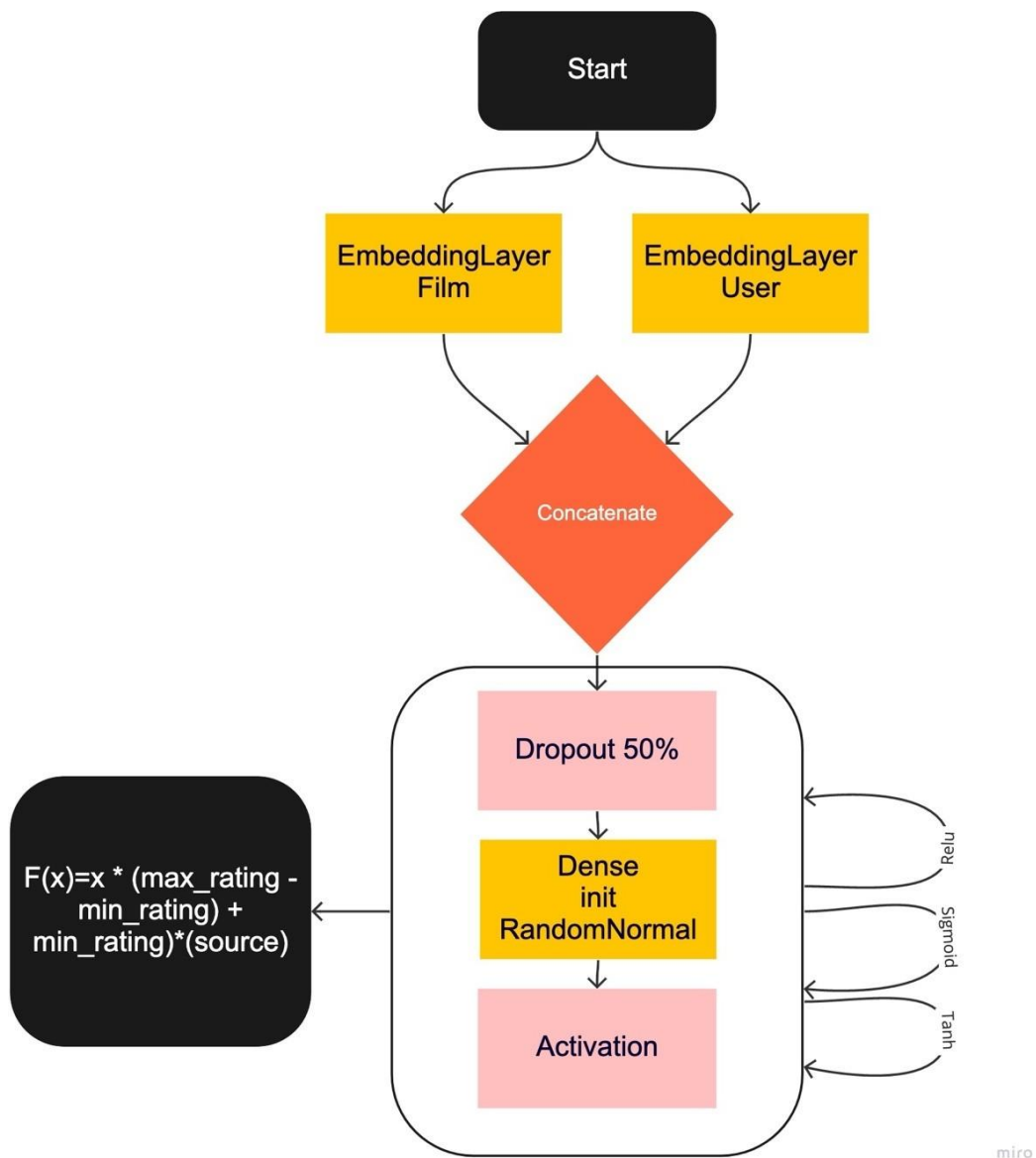


Рис. 3.4. Загальна архітектура рекомендаційної системи

3.2. Архітектура нейронної мережі

Модель нейронної мережі включає три шари активації, три повнозв'язні шари та механізм Dropout. Проблема перенавчання (overfitting) є однією з основних труднощів у глибоких нейронних мережах (Deep Neural Networks, DNN). Вона полягає в тому, що модель ефективно пояснює лише дані з

навчального набору, адаптуючись до навчальних прикладів, замість того, щоб навчатися класифікувати нові приклади, які не брали участі в навчанні, і втрачає здатність до узагальнення. Вирішенням цієї проблеми, яке виділяється своєю простотою і ефективністю, є метод Dropout («метод випадкового виключення» або просто «дропаут»). Суть концепції Dropout зображена на рис. 3.5.

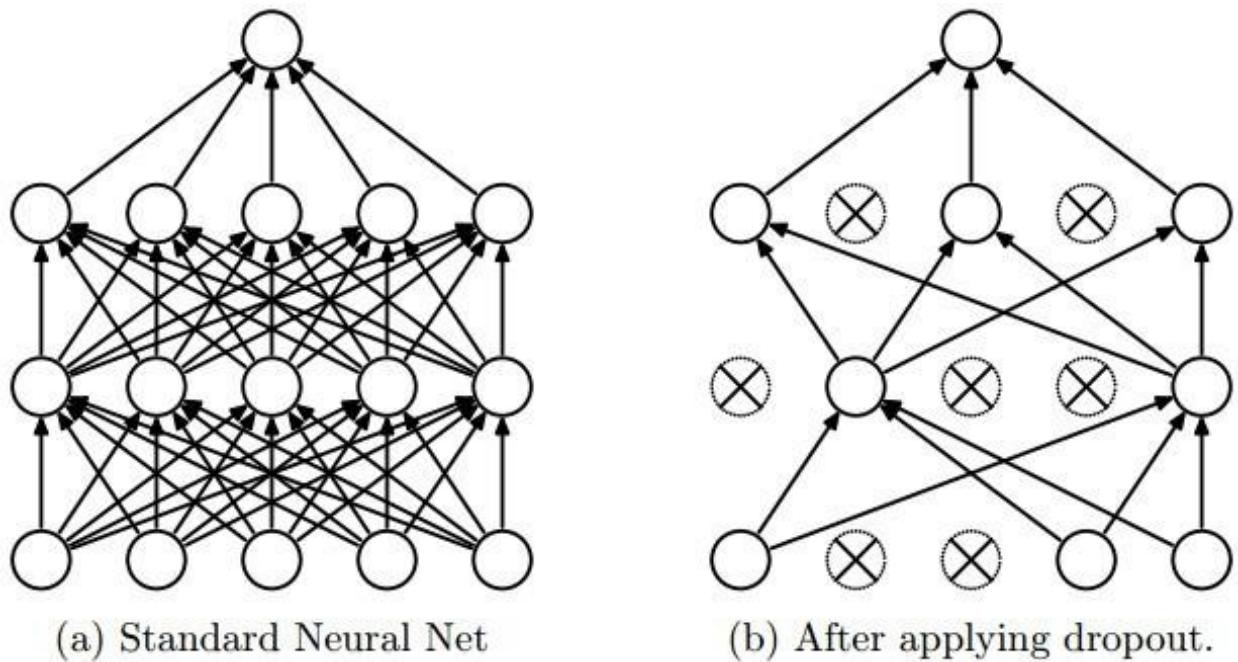


Рис. 3.5. Проріджування нейронної мережі

В ході ряду експериментів вдалося досягти оптимальних результатів, використовуючи три різні функції активації: сигмоїду, гіперболічний тангенс та функцію активації `relu`. Така унікальна архітектура є новаторським рішенням у створенні рекомендаційних систем і взагалі в дизайні архітектур нейронних мереж [8].

Використання фреймворку Keras, який був інтегрований у TensorFlow у 2020 році, дозволяє використовувати TensorBoard для графічного відображення часових рядів. Це включає можливість переглядати функцію витрат та загальну структуру графа нейронної мережі рекомендаційної системи (рис. 3.6).

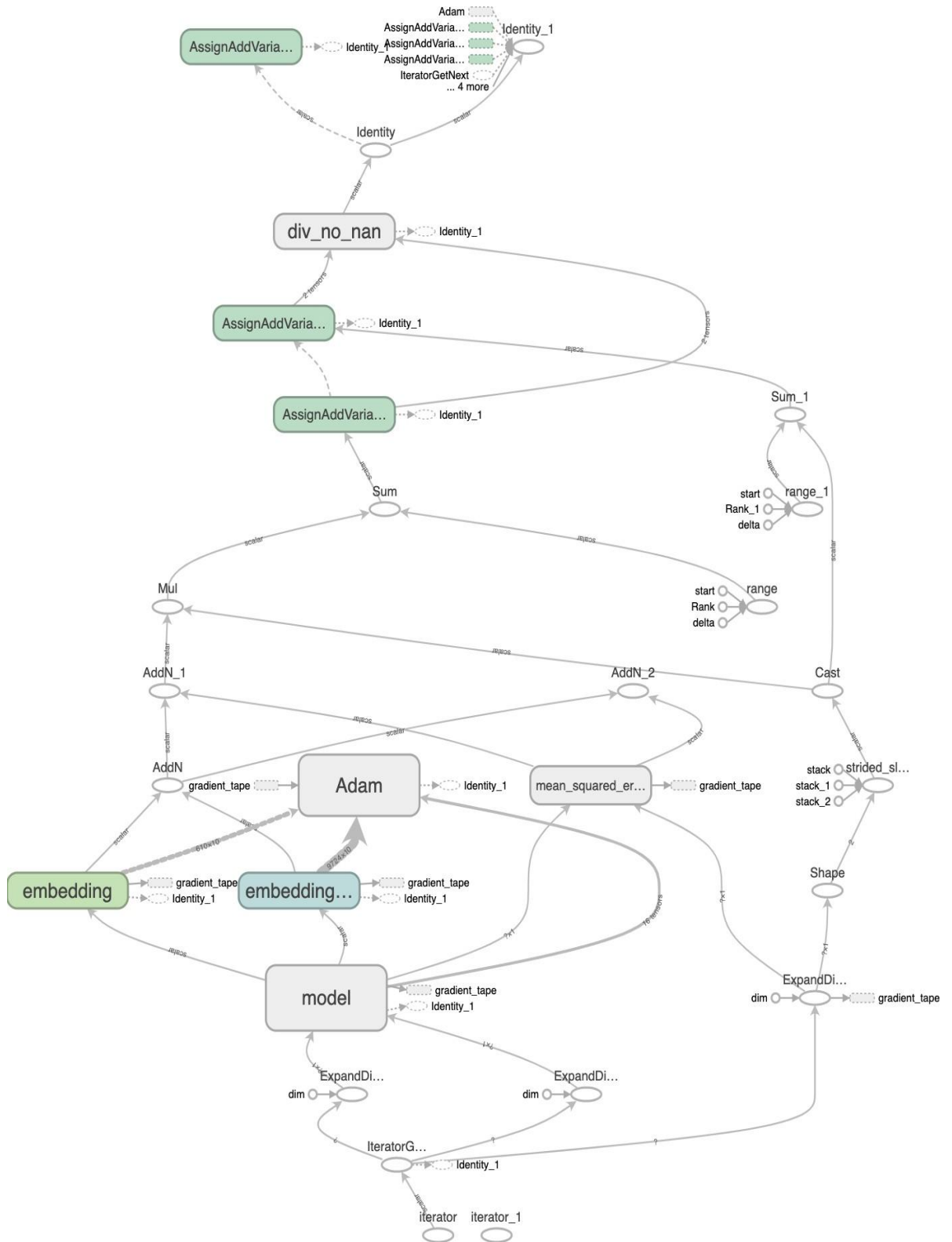


Рис. 3.6. TensorBoard

3.3. Процес навчання нейронної мережі

Під час навчання нейронної мережі проведено низку експериментів для визначення оптимальних гіперпараметрів моделі, налаштування розподілу

шарів та визначення оптимальної кількості епох для кожної ітерації тренування рекомендаційної системи. Вхідні дані були розділені на валідаційний та тренувальний набори в співвідношенні 30% до 70%. Результати точності моделі під час тренування та тестування відображені на рис. 3.7, який був побудований з використанням бібліотеки Matplotlib.

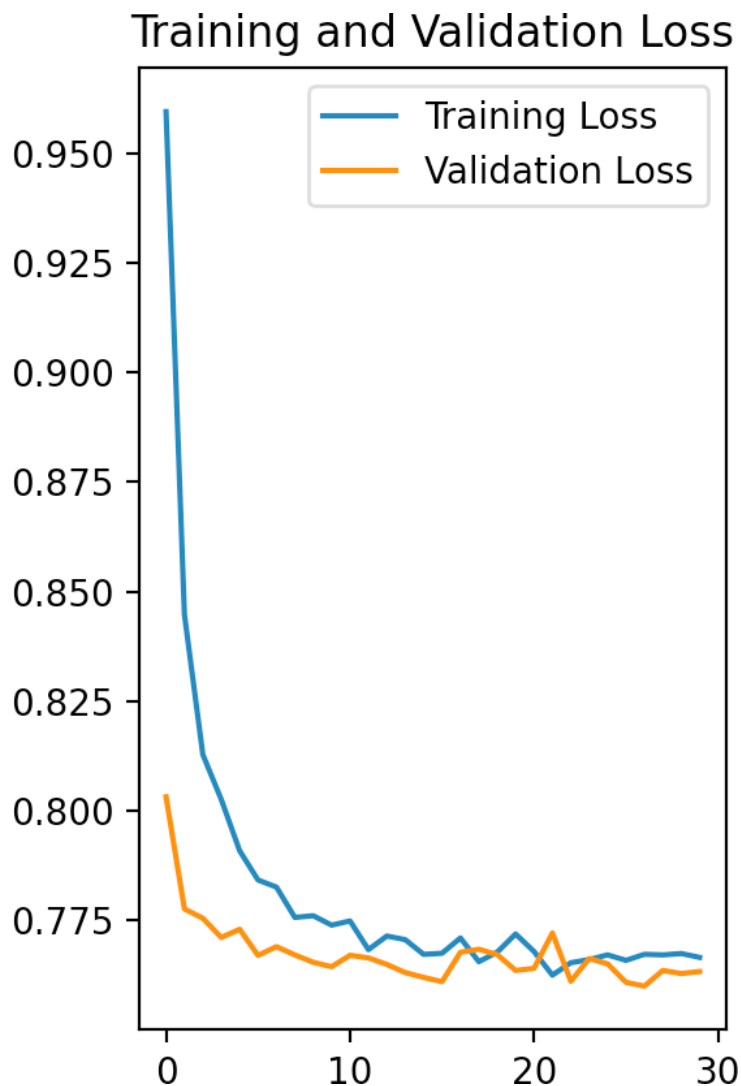


Рис. 3.7. Показник точності RMSE

Ще на першій епосі процесу навчання нейронної мережі були отримані результати RMSE, які виявилися кращими, ніж точність моделі Netflix у 2008 році. Після 30 епох навчання модель нейронної мережі продемонструвала точність прогнозів на рівні приблизно 0.7 RMSE.

Для забезпечення взаємодії в реальному часі між серверною частиною та користувачем використовувався HTTP протокол, який реалізує концепцію REST API. Сесію взаємодії можна побачити на рис. 3.8.

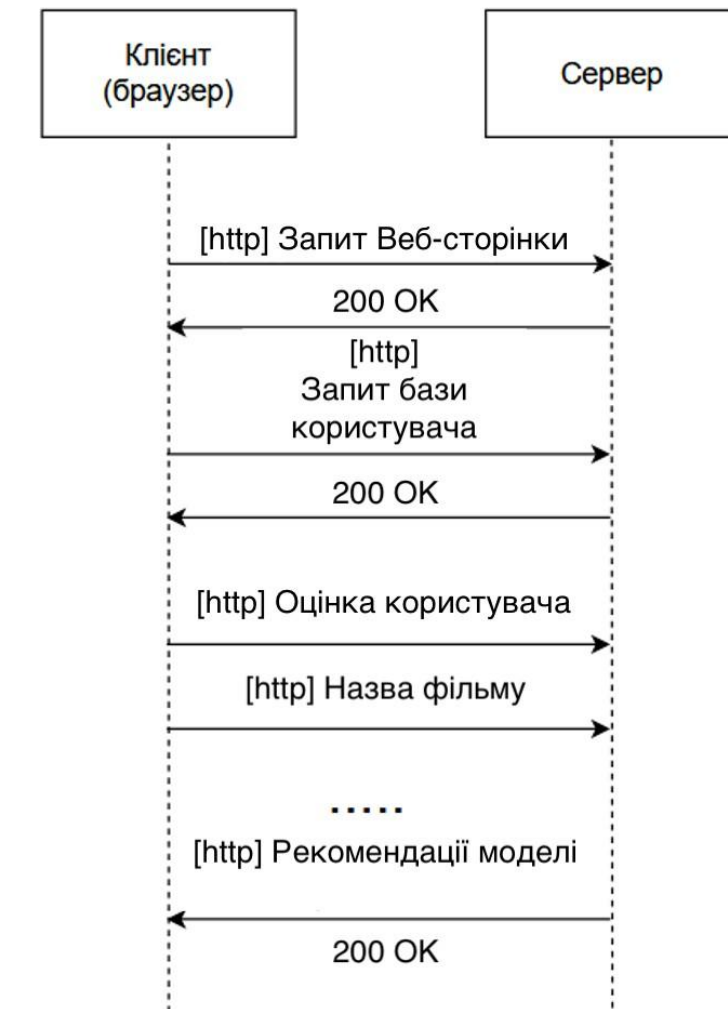


Рис. 3.8. Комунікація за допомогою HTTP

Для реалізації клієнт-серверної частини був використаний фреймворк Flask, який надає швидко та надійну можливість створення веб-додатків на мові програмування Python. В цьому розділі був розроблений алгоритм фільтрації та рекомендаційної системи, базованої на нейронній мережі, і подані їх описи та процес навчання. Модель рекомендаційної системи показала точність на рівні 0.7 RMSE. Також був спроектований інтерфейс веб-додатку для рекомендаційної системи, а також розглянута діаграма прецедентів та сценарії для кожного варіанту використання системи.

Рекомендаційна система має веб-інтерфейс і вимагає попереднього встановлення Node, Python 3.6 та згідно із вмістом requirements.txt [18]. Для відтворення результатів рекомендаційної системи рекомендується використовувати такі самі версії бібліотек, які вказані у файлі. Також, встановлення віртуального середовища (virtual environment) для ізоляції всіх необхідних залежностей є важливим для уникнення конфліктів між різними проектами та збереження порядку у середовищі Python. Щоб розгорнути додаток на локальному комп'ютері, слід виконати команди для встановлення Python пакетів та Node, які подані у відповідних командах.

Після виконання усіх кроків для встановлення та запуску серверної та клієнтської частин додатку, веб-інтерфейс стане доступним за адресою <http://localhost:3000/> для використання у браузері (рис. 3.9).

Package	Version
kiwisolver	1.3.1
matplotlib	3.3.4
numpy	1.19.5
oauthlib	3.1.0
opt-einsum	3.3.0
pandas	1.1.5
pip	21.1.1
protobuf	3.17.0
pyasn1	0.4.8
pyasn1-modules	0.2.8
pyarsing	2.4.7
python-dateutil	2.8.1
pytz	2021.1
requests	2.25.1
requests-oauthlib	1.3.0
rsa	4.7.2
scikit-learn	0.24.2
scipy	1.5.4
setuptools	56.2.0
six	1.15.0
sklearn	0.0
tensorboard	2.5.0
tensorboard-data-server	0.6.1
tensorboard-plugin-wit	1.8.0
tensorflow	2.5.0
tensorflow-estimator	2.5.0
termcolor	1.1.0
threadpoolctl	2.1.0
typing-extensions	3.7.4.3
urllib3	1.26.4
wheel	0.36.2
wrapt	1.12.1
zipp	3.4.1

Рис. 3.9. Список використаних бібліотек

Коли користувач переходить за адресою <http://localhost:3000/> у своєму браузері, він потрапляє на сторінку рекомендаційної системи та отримує можливість переглядати список фільмів і ставити оцінки новим фільмам, щоб отримати персоналізовані рекомендації. Для перегляду рекомендацій

користувачу потрібно оцінити фільм із запропонованого списку. Процес відправлення оцінки на сервер зображено на рис. 3.10.

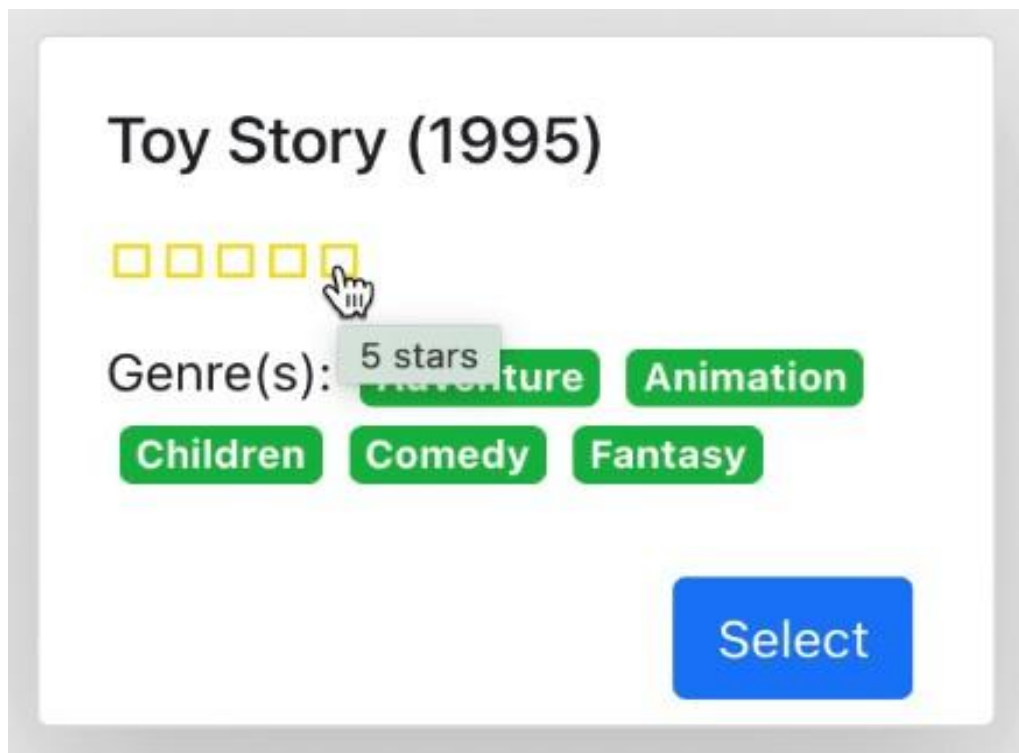


Рис. 3.10. Оцінка фільму

Для того щоб відправити оцінку фільму на сервер для обробки, користувач повинен вказати кількість зірок, яку він призначає фільму. Після цього, натисканням кнопки "SELECT", відбувається відправлення оцінки для подальшої обробки. Після відправлення оцінки порядок розташування фільмів у списку на сторінці веб-інтерфейсу змінюється, а рекомендовані фільми переміщуються знизу вгору списку. Область змінених рекомендацій відображена на рис. 3.11.

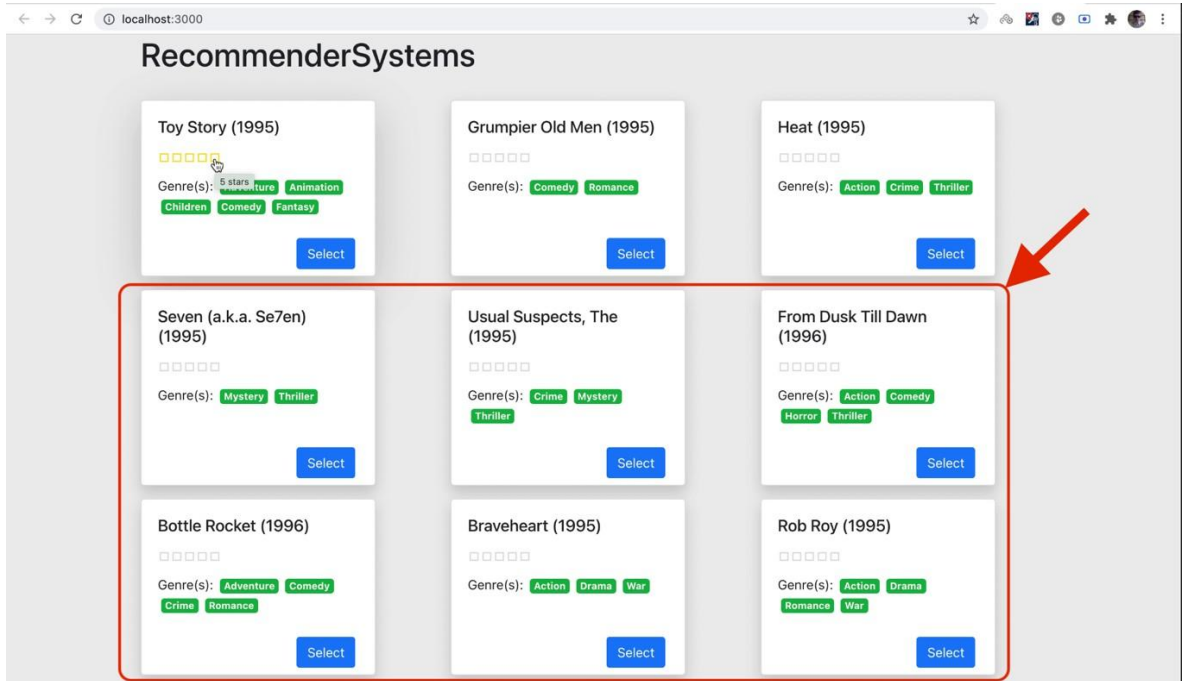


Рис. 3.11. Область зміни рекомендацій до фільмів

Прогнозовані рекомендації переміщуються лише серед фільмів, які користувач ще не оцінив. Фільми, які вже отримали оцінку від користувача, залишаються на своїх попередніх позиціях. Такий підхід до розташування блоків сприяє швидкій навігації серед раніше оцінених фільмів [11].

Перегляд навчання можна переглянути на рис. 3.12 - 3.14.

```

Epoch 38/50
744/744 [=====] - 43s 58ms/step - loss: 1.6336 - accuracy: 0.5002
Epoch 39/50
744/744 [=====] - 44s 59ms/step - loss: 1.6222 - accuracy: 0.5001
Epoch 40/50
744/744 [=====] - 44s 59ms/step - loss: 1.6357 - accuracy: 0.5002
Epoch 41/50
744/744 [=====] - 44s 59ms/step - loss: 1.6624 - accuracy: 0.4944
Epoch 42/50
744/744 [=====] - 43s 58ms/step - loss: 1.6475 - accuracy: 0.4990
Epoch 43/50
744/744 [=====] - 44s 59ms/step - loss: 1.6244 - accuracy: 0.5014
Epoch 44/50
744/744 [=====] - 44s 58ms/step - loss: 1.6373 - accuracy: 0.4979
Epoch 45/50
744/744 [=====] - 44s 59ms/step - loss: 1.6325 - accuracy: 0.4989
Epoch 46/50
744/744 [=====] - 44s 59ms/step - loss: 1.6758 - accuracy: 0.4950
Epoch 47/50
744/744 [=====] - 44s 59ms/step - loss: 1.6713 - accuracy: 0.4918
Epoch 48/50
744/744 [=====] - 44s 59ms/step - loss: 1.6492 - accuracy: 0.4986
Epoch 49/50
744/744 [=====] - 43s 58ms/step - loss: 1.6384 - accuracy: 0.5013
Epoch 50/50
744/744 [=====] - 43s 58ms/step - loss: 1.6303 - accuracy: 0.5008

```

Рис. 3.12. Навчання системи

Рис. 3.13. Результат на тестовій вибірці

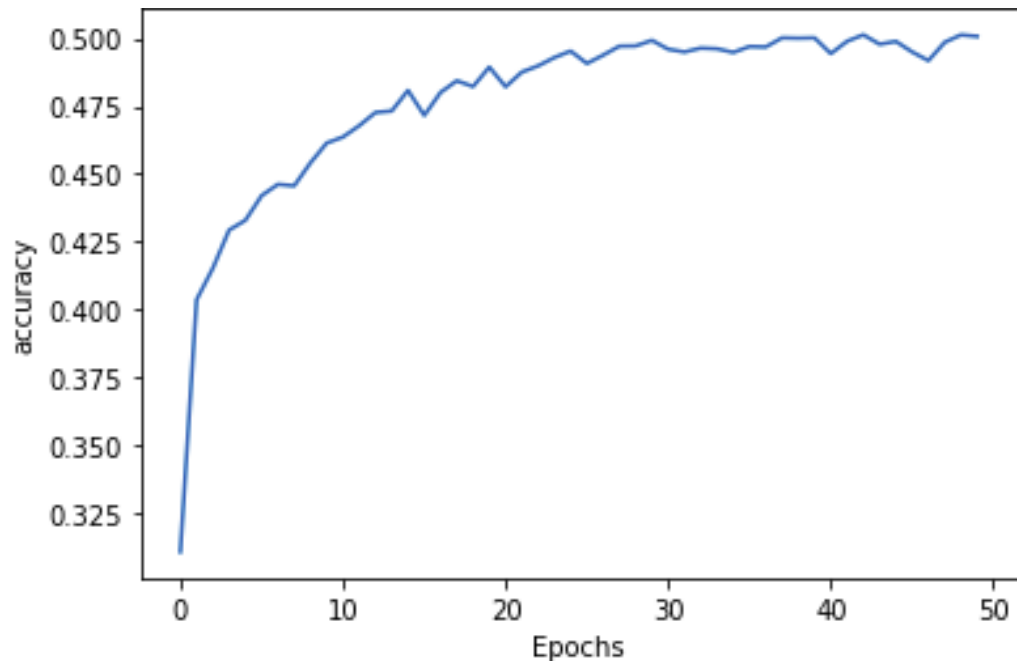


Рис. 3.14. Результати навчання у вигляді графіку

3.4. Створення сайту

Розроблений сайт базується на системі управління контентом (CMS) WordPress і був протестований з майже усіма доступними браузерами. На сьогоднішній день WordPress є найпопулярнішою системою управління контентом у світі. Вона формує основу для понад 30% усіх веб-сайтів у всьому світі, і цей показник постійно зростає.

Загальною характеристикою CMS є веб-додаток, який дозволяє власникам сайтів, редакторам та авторам управляти своїми сайтами та публікувати контент, не потребуючи спеціальних навичок у програмуванні. WordPress використовує PHP і MySQL, і ці технології підтримуються практично всіма постачальниками хостингу. Однак спеціальні тарифні плани

для хостингу WordPress можуть забезпечити кращу швидкість, продуктивність і надійність.

Зазвичай WordPress використовується для створення блогів, але його потужності дозволяють легко перетворити його у інтернет-магазин, портфоліо, періодичний сайт або що завгодно інше, що ви можете уявити. Його інтуїтивно зрозумілий та дружелюбний інтерфейс робить його доступним для широкого кола користувачів, особливо для тих, хто вже знайомий із Microsoft Word.

Найважливішим аспектом є те, що WordPress є відкритим джерелом і абсолютно безкоштовним для всіх. Ця система дозволяє мільйонам людей по всьому світу створювати сучасні та якісні веб-сайти, навіть для невеликих проєктів.

Переваги WordPress:

1. Низька вартість: Основною витратою є лише оплата за домен та веб-хостинг. Крім того, програмне забезпечення WordPress і багато плагінів доступні безкоштовно.

2. Простий процес установки і оновлення: WordPress відрізняється мінімальною настройкою, і його можна легко оновити одним кліканням миші, на відміну від деяких інших систем управління контентом.

3. Легкість в управлінні: Для щоденних завдань, таких як написання та редагування публікацій, завантаження та редагування зображень, управління користувачами, додавання меню, установка плагінів і тем, вам не потрібні знання в програмуванні.

4. Індивідуальний дизайн: З великою кількістю готових тем для WordPress ви можете легко створювати унікальний дизайн, що відповідає вашій сфері діяльності.

5. Призначені для користувача функції: Використовуйте плагіни для розширення стандартних можливостей WordPress. Зазвичай для будь-якого завдання від оптимізації для пошукових систем до бронювання подій можна знайти відповідний плагін.

6. Спільнота завжди готова допомогти: У світовому співтоваристві WordPress і на корисному форумі підтримки завжди можна знайти допомогу та поради від інших користувачів.

Недоліки WordPress:

1. Проблеми безпеки: Оскільки WordPress є основою для понад 30% сайтів в мережі, він часто стає об'єктом кібератак. Застосування плагінів для забезпечення безпеки може значно знизити цей ризик.

2. Сторонній контент: Багато плагінів і тем для WordPress створюються сторонніми розробниками і можуть містити помилки. Перед встановленням нового плагіна чи теми завжди рекомендується уважно читати опис та відгуки, а в разі сумнівів звертатися до спільноти для отримання порад.

3. Час завантаження сторінки: З великою кількістю плагінів ваш сайт може почати повільно завантажуватися. Встановлення плагіна кешування зазвичай допомагає вирішити цю проблему.

Для розробки сторінки з інфопродуктами на WordPress і виконання оформлення її дизайну я використовувала декілька спеціальних плагінів: Smart Slider 3, Elementor, WooCommerce та WPForms.

WPForms визначається як найбільш дружній WordPress плагін для створення контактних форм. Ми використовуємо його на численних наших веб-сайтах. Створення контактних форм за допомогою WPForms легке і зручне завдяки функції drag & drop. Однією з особливостей WPForms є його доброзичливий інтерфейс, що робить його найбільш користувачепривітним серед WordPress плагінів для контактних форм.

Плагін також надає вичерпну документацію, і у випадку виникнення проблем можна звертатися за підтримкою через електронну пошту. Незважаючи на міць WPForms, його основна мета полягає в тому, щоб бути максимально дружнеливним до користувача. Саме тому в цьому плагіні відсутні безліч складних функцій, які можуть бути присутні в інших плагінах.

Зовнішній вигляд головної сторінки показано на рис. 3.15.

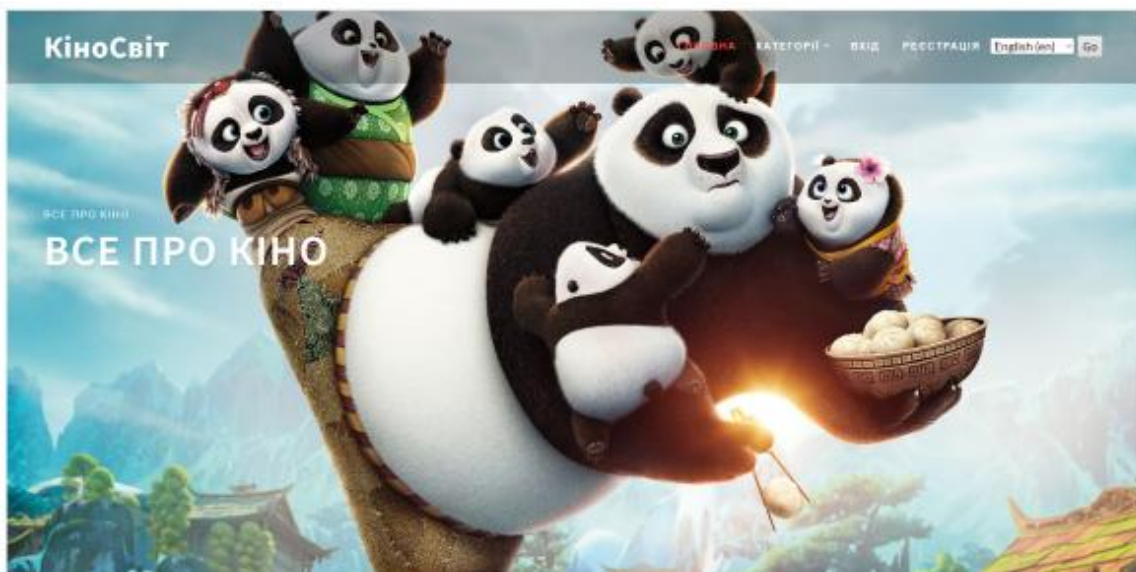


Рис. 3.15. Головна сторінка сайту

Адаптація сторінок під різні пристрої є невід'ємним етапом в розробці веб-сайту. Згідно з офіційною статистикою, кожен третій відвідувач використовує мобільні пристрої для доступу до Інтернету. Це підтверджується як офіційними даними найбільших пошукових систем, наприклад Google, які уважно перевіряють ресурси на наявність адаптованих версій для кишенькових пристроїв.

На платформі WordPress одним з ефективних інструментів адаптації сторінок під планшети і мобільні пристрої є функціонал плагіна Elementor. Процес адаптації зображений на рис. 3.16 та рис. 3.17 відображає етапи оптимізації сторінок сайту для відображення на планшетах і мобільних пристроях.

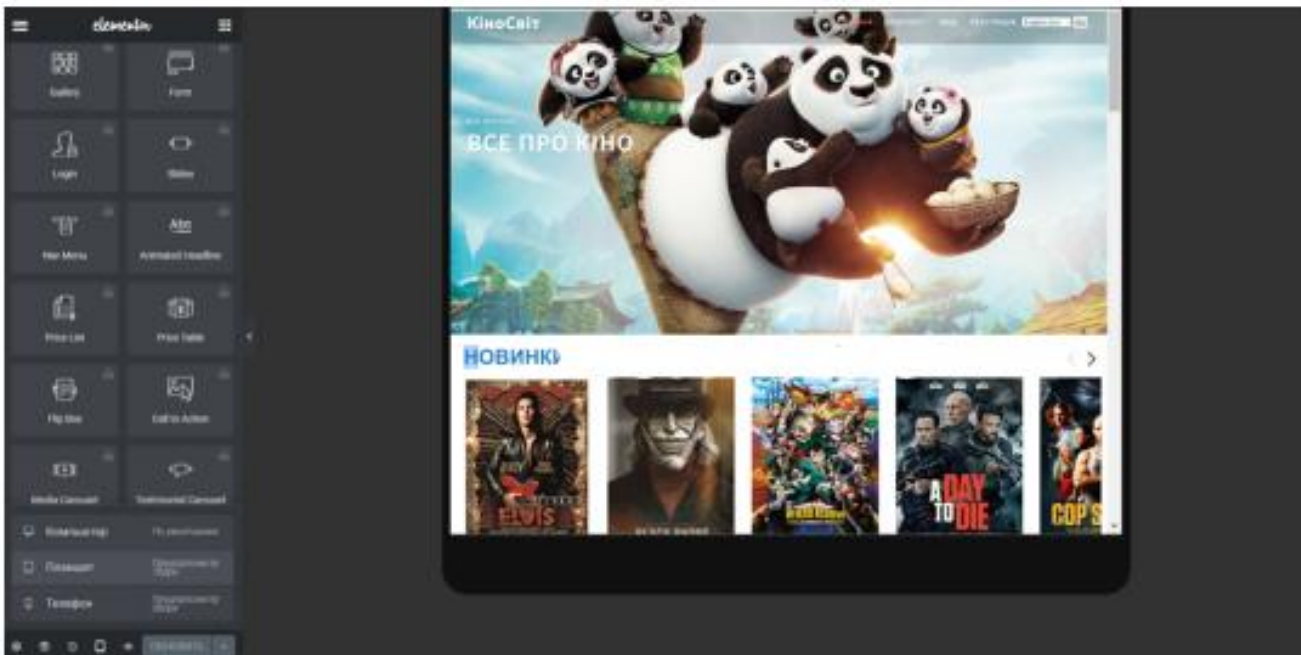


Рис. 3.16. Процес оптимізації сторінки під планшети

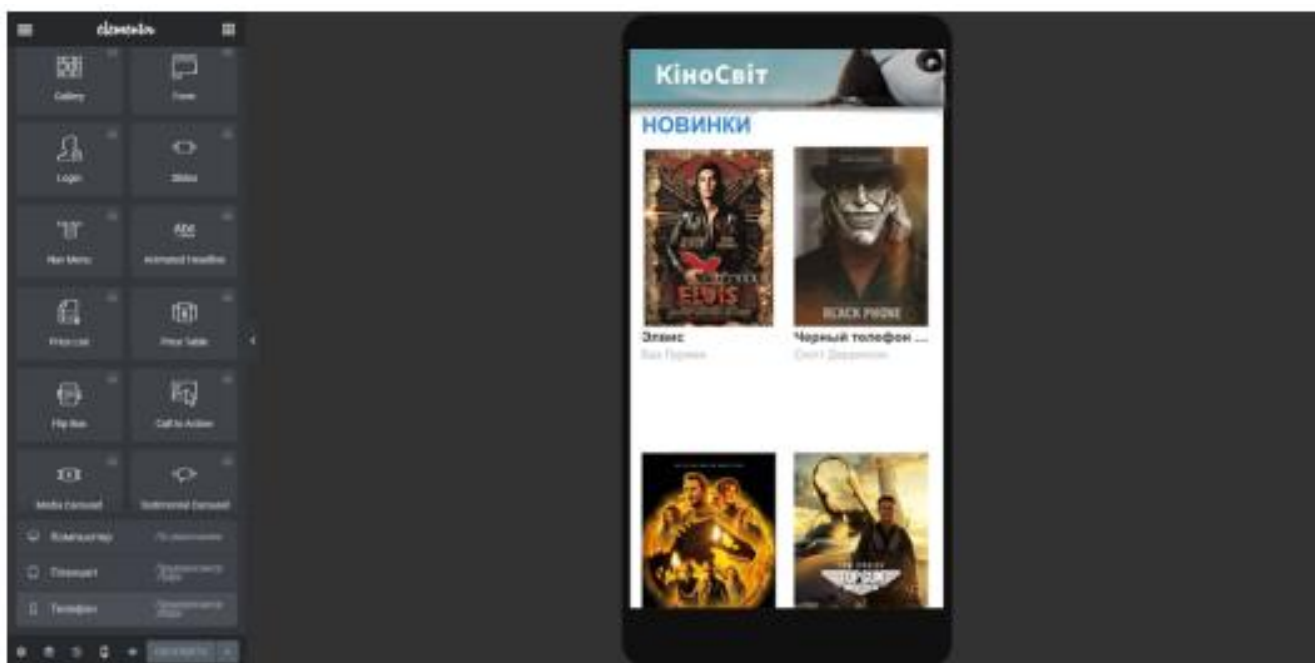


Рис. 3.17. Процес оптимізації сторінки під мобільні девайси

ВИСНОВКИ

Під час виконання роботи було виконано такі завдання:

- проведено аналіз основних методів розробки рекомендаційних систем;
- створено рекомендаційну систему на основі нейронної мережі, використовуючи метод колаборативної фільтрації;
- зібрано та оброблено дані для тренування моделі;
- проведено експерименти для тренування моделі;
- створено та протестовано інтерфейс взаємодії з рекомендаційною системою.

Розроблений продукт може використовуватися як швидка та уніфікована платформа для передбачення рекомендацій щодо фільмів. Завдяки використанню веб-технологій для побудови рекомендаційної системи, досягнута універсальність та зручність використання на будь-якому пристрої, необхідності в адаптації під конкретну операційну систему відсутнє (браузер та його віртуальна машина виступають як цільова універсальна операційна система та комп'ютер).

Системи рекомендацій відкривають нові можливості для отримання персоніфікованої інформації в Інтернеті і сприяють полегшенню проблеми перевантаження інформації, що часто виникає в системах пошуку. У цьому дослідженні розглянуті дві традиційні методики рекомендацій, висвітлені їх сильні та слабкі сторони, а також проаналізовано різні стратегії гібридизації для покращення їхньої ефективності. Розглянуті алгоритми навчання та метрики оцінки, які використовуються при формуванні рекомендаційних моделей.

На основі цього аналізу розроблена рекомендаційна система, використовуючи нейронні мережі. Також розроблена веб-система для взаємодії з рекомендаційною системою на основі REST API. Ці кроки сприятимуть розширенню можливостей досліджень у цій області та служитимуть дорожньою картою для подальших вдосконалень технічних рекомендацій

Під час виконання дипломного проекту був створений готовий до застосування веб-сайт для перегляду трейлерів кінофільмів. Проект включав в себе визначення актуальності теми, постановку завдання та концептуальне та логічне проектування сайту.

Основна мета сайту — надавати глядачеві можливість з легкістю обирати фільми для перегляду, а також створювати приватні чи публічні колекції для подальшого перегляду. Сайт спрямований на поліпшення розуміння користувача актуальності фільму для нього, уникання спойлерів, та допомогу у вирішенні питання щодо необхідності купівлі фільму перед переглядом.

Сервіс надає коротку інформацію про фільм, акторів, які беруть участь у його створенні, а також відгуки інших глядачів, на основі яких формується рейтинг сайту. Використовується SQL база даних для зберігання інформації про фільми, а трейлери завантажуються через посилання з відеохостингу YouTube.

Під час розробки веб-сайту були враховані сучасні веб-технології, і в результаті аналізу виявлено, що для досягнення поставленої мети найбільш підходить система керування вмістом WordPress та мова програмування Python.

Сайт відповідає всім вимогам, визначеним на етапі створення завдання. Зокрема, як можливі напрямки подальшого вдосконалення сайту, розглядається можливість додавання функції покупки та перегляду фільмів на самому сайті. Також передбачається доопрацювання інтерфейсу, його оптимізація та просування в браузерях для підвищення інформативності, ефективності, швидкості, привабливості та зручності для користувачів.

Основні переваги веб-сайту включають зручність у користуванні, надійність у роботі, зрозумілий і приємний інтерфейс, швидкість та високий рейтинг серед різних веб-браузерів, а також можливість легкої модифікації та оновлення інформації за допомогою системи керування вмістом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Konstan JA, Riedl J. Рекомендаційні системи: від алгоритмів до користувацького досвіду. Модель користувача. Взаємодія з адаптацією користувача 2019;22:101–23.
2. Pan C, Li W. Рекомендація дослідницької роботи з аналізом тем. У галузі комп'ютерного дизайну та додатків IEEE 2019; pp. V4-264.
3. Pu P, Chen L, Hu R. Орієнтована на користувача система оцінки рекомендаційних систем. В: Матеріали п'ятої конференції ACM щодо рекомендаційних систем (RecSys'11), ACM, New York, NY, USA; 2019. p. 57–164.
4. Hu R, Pu P. Потенційні проблеми прийняття с рекомендаційних систем. Матеріали конференції ACM щодо рекомендаційних систем (RecSys'09), New York City, NY, USA; October 2019. p. 22–5.
5. Pathak B, Garfinkel R, Gopal R, Venkatesan R, Yin F. Емпіричний аналіз впливу систем рекомендацій на продажі. J Manage Inform Syst 2019;27(2):159–88.
6. Rashid AM, Albert I, Cosley D, Lam SK, McNee SM, Konstan JA et al. Знайомство з вами: вивчення нових уподобань користувачів у рекомендаційних системах. Матеріали міжнародної конференції з інтелектуальних користувацьких інтерфейсів; 2022. p. 127–34.
7. Schafer JB, Konstan J, Riedl J. Рекомендаційні системи в e-commerce. In: Праці 1-ї ACM конференції з питань електронної комерції; 2019. p. 158–66.
8. Resnick P, Varian HR. Рекомендаційні системи. Commun ACM 2017;40(3):56
9. Acilar AM, Arslan A. Метод колаборативної фільтрації на основі штучної нейронної мережі. Exp Syst Appl 2019;36(4):8324–32.
10. Chen LS, Hsu FH, Chen MC, Hsu YC. Розробка систем рекомендацій з урахуванням прибутковості продукції для продавців. Int J Inform Sci 2018;178(4):1032–48.

11. Jalali M, Mustapha N, Sulaiman M, Mamat A. WEBPUM: веб-система рекомендацій для прогнозування подальшого руху користувачів. *Exp Syst Applicat* 2019;37(9):6201–12.
12. Глібовець, Н.М. Створення рекомендаційної системи навчального типу з використанням фреймворку / Н.М. Глібовець, М.О.Сидоренко // Проблеми інтелектуалізації комп'ютера: зб. ст. / Інститут кібернетики ім. В.М. Глушкова НАН України. - Київ, 2019. С. 176 - 181.
13. Джонс, М. Рекомендаційні системи: Частина 1. Введення в підходи і алгоритми.: <http://www.ibm.com/developerworks/ru/library/os-recommender1.html> .
14. Джонс, М. Рекомендаційні системи: Частина 2. Механізми з відкритим вихідним кодом. URL: <http://www.ibm.com/developerworks/ru/library/os-recommender2.html>.
15. Жернакова, О. Системи рекомендацій і пошуку відеоконтенту. URL : <http://www.telemultimedia.ua/art.php?id=464.html>
16. Netflix prize rules. URL: <https://www.snellman.net/blog/archive/2006-10-15-netflix-prize.html> (дата звернення: 25.10.2023)
17. Kuhlman D. А Книга Python: Початкові вправи Python, Advanced Python Dave Kuhlman., 2019.
18. TensorFlow: A System for Large-Scale Machine Learning / [М. Abadi, Р. Barham, С. Jianmin та ін.]. // Симпозіум з проектування та впровадження операційних систем.– 2016. – С. 265–283.
19. Machine Learning Mastery.[веб-ресурс]. Режим доступу: <https://machinelearningmastery.com/types-of-learning-in-machine-learning/>
20. Pattern Recognition and Machine Learning.[веб-ресурс]. Режим доступу: <http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20-%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%20%20200.pdf>
21. Deep Learning. An MIT Press book.[веб-ресурс]. Режим доступу: <https://www.deeplearningbook.org/>

22. Reinforcement Learning.[веб-ресурс]. Режим доступа:
<https://www.andrew.cmu.edu/course/10-703/textbook/BartoSutton.pdf>
23. Long Short-term Memory.[веб-ресурс]. Режим доступа:
https://www.researchgate.net/publication/13853244_Long_Short-term_Memory
24. TensorFlow Core v2.7.0. Input [веб-ресурс]. Режим доступа:
https://www.tensorflow.org/api_docs/python/tf/keras/Input
https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM
25. TensorFlow Core v2.7.0. BatchNormalization [веб-ресурс]. Режим доступа:
https://www.tensorflow.org/api_docs/python/tf/keras/layers/BatchNormalization
26. TensorFlow Core v2.7.0. Attention [веб-ресурс]. Режим доступа:
https://www.tensorflow.org/api_docs/python/tf/keras/layers/Attention
27. TensorFlow Core v2.7.0. Dense[веб-ресурс]. Режим доступа: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense
28. TensorFlow Core v2.7.0. Dropout [веб-ресурс]. Режим доступа:
https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout
29. TensorFlow Core v2.7.0. Activation [веб-ресурс]. Режим доступа:
https://www.tensorflow.org/api_docs/python/tf/keras/layers/Activation
30. Keras [веб-ресурс]. Режим доступа: <https://keras.io/>
31. Numpy [веб-ресурс]. Режим доступа:
<https://numpy.org/doc/stable/user/whatisnumpy.html>
32. Tensorflow [веб-ресурс]. Режим доступа:
<https://www.tensorflow.org/overview>