

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ПАВЛЕНКО ОЛЕНА ОЛЕКСІЇВНА

Допускається до захисту:
завідувач кафедри інформаційних
технологій

Тетяна НЕСКОРОДЕВА

« ___ » _____ 2022 р.

**ДОСЛІДЖЕННЯ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ ОБРОБКИ ДАНИХ
АВТОМАТИЗОВАНОЇ МЕРЕЖЕВОЇ СИСТЕМИ ЗБОРУ І
РЕКОМЕНДАЦІЙ ВІДСОКОНТЕНТУ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (магістерська) робота

Науковий керівник:
Т.В. Нескородева,
завідувач кафедри
інформаційних технологій,
д.т.н, доцент

(підпис)

Оцінка: _____ / / _____

(бали/за шкалою ЕКТС/за
національною шкалою)

Голова ЕК: _____

(підпис)

Вінниця 2022

АНОТАЦІЯ

Павленко О.О. Дослідження інтелектуальних методів обробки даних автоматизованої мережевої системи збору і рекомендацій відеоконтенту. Спеціальність 122 «Комп'ютерні науки», Освітня програма «Комп'ютерні технології обробки даних (Data Science)». Донецький національний університет імені Василя Стуса, Вінниця, 2022.

В роботі здійснено дослідження інтелектуальних методів і моделей обробки даних автоматизованої мережевої системи збору і рекомендацій відеоконтенту в залежності від їх призначення, набору показників, що накопичуються в даних системах, обсягів даних. Виконана програмна реалізація експериментальної побудови оцінок рейтингу відеоконтенту мережевої системи. Зроблено висновок, що існуючі методи можна ефективно використовувати для формування рекомендацій відеоконтенту. Подальший розвиток інтелектуальних методів і моделей передбачає підвищення їх точності, зменшення обчислювальної складності і розширення функціональних можливостей.

Ключові слова: автоматизована мережева система, збір і рекомендації відеоконтенту, інтелектуальні методи.

Р. 77 с., 2 табл., 14 рис., 52 джерела.

ABSTRACT

Pavlenko O.O. Research of intelligent methods of data processing of the automated network system of collection and recommendations of video content. Specialty 122 "Computer science", educational program "Computer technologies of data processing (Data Science)". Vasyl' Stus Donetsk National University, Vinnytsia, 2022.

In the work, a study of intelligent methods and models of data processing of the automated network system of video content collection and recommendations is carried out depending on their purpose, a set of indicators accumulated in these systems, data volumes. The software implementation of the experimental construction of ratings of the video content of the network system has been completed. It was concluded that the existing methods can be effectively used to form video content recommendations. Further development of intelligent methods and models involves increasing their accuracy, reducing computational complexity, and expanding functional capabilities.

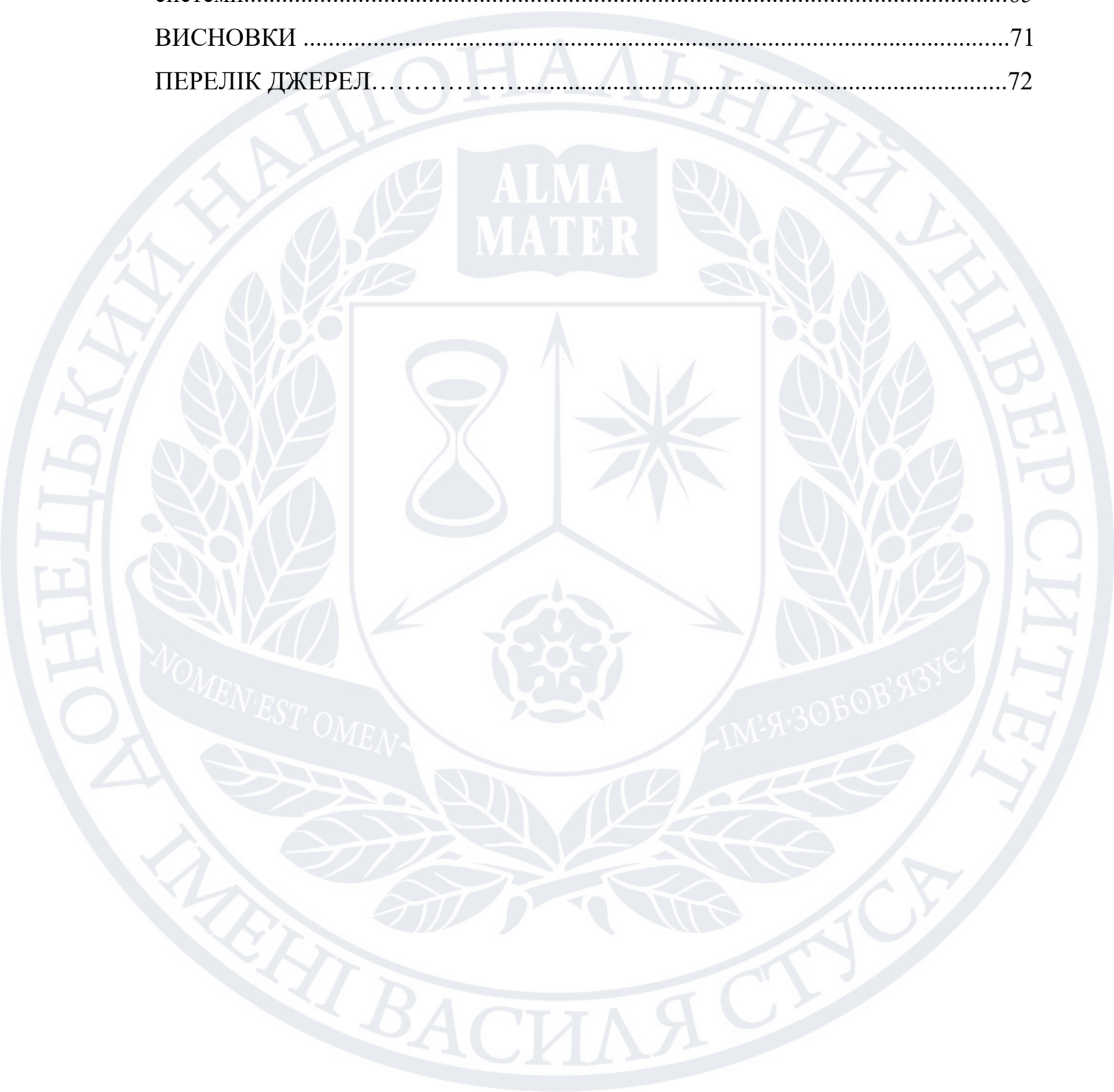
Keywords: automated network system, collection and recommendations of video content, intelligent methods.

P. 77. Tabl. 2. Fig. 14. Add. 2. Sources 52.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	10
1.1 Дослідження існуючих мережевих систем збору і рекомендацій відеоконтенту.....	10
1.2. Дослідження реалізації зворотного зв'язку щодо ефективності рекомендацій відеоконтенту.....	13
1.3. Дослідження технологій обробки даних в мережевих системах збору і рекомендацій відеоконтенту.....	15
1.4 Постановка задачі.....	24
РОЗДІЛ 2 ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ОСНОВИ ОБРОБКИ ДАНИХ АВТОМАТИЗОВАНОЇ МЕРЕЖЕВОЇ СИСТЕМИ ЗБОРУ І РЕКОМЕНДАЦІЙ ВІДЕОКОНТЕНТУ.....	25
2.1 Дослідження наборів даних для рекомендацій відеоконтенту мережевої системи.....	25
2.2 Дослідження вимог до моделей рекомендацій відеоконтенту мережевої системи.....	40
2.3 Дослідження моделей формування рекомендацій, отримання оцінок рейтингу відеоконтенту мережевої системи.....	46
2.4 Висновки за розділом 2.....	47
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДЕЛЕЙ ОЦІНКИ ЕФЕКТИВНОСТІ СИСТЕМ ПОШУКУ РОБОТИ ЗБОРУ І РЕКОМЕНДАЦІЙ ВІДЕОКОНТЕНТУ.....	58
3.1 Програмні засоби оцінки рейтингу відеоконтенту мережевої системи	58

3.2 Експериментальна побудова оцінок рейтингу відеоконтенту мережевої системи.....	65
ВИСНОВКИ	71
ПЕРЕЛІК ДЖЕРЕЛ.....	72



ВСТУП

В сучасному інтернет-просторі значний сегмент ринку займають рекомендаційні системи. Вони супроводжують широкий набір інтернет-послуг. Від покупок до потокової передачі даних та пошукових систем. Основне призначення рекомендаційних систем полягає в допомозі користувачам отримати більш персоналізований досвід від програмних продуктів. Як наслідок це підвищує їх рівень задоволеності, робить лояльними до продукту, що дозволяє розробникам, ІТ-компаніям утримувати та збільшувати аудиторію продукту, зокрема постійних підписників.

Сервіси на зразок Netflix, YouTube або TikTok рекомендують зареєстрованим відвідувачам той чи інший контент в залежності від їх попередніх переглядів або прослуховувань і/або рейтингів того чи іншого відео, фільму або виконавця.

YouTube [1,2] – популярний відеохостинг, що надає послуги розміщення відеоматеріалів. Заснований у 2005 році. Зараз є підрозділом компанії Google. Функціонал YouTube: користувачі можуть додавати, переглядати і коментувати ті чи інші відеозаписи. Завдяки простоті та зручності використання, YouTube став одним із найпопулярніших місць для розміщення відеофайлів. Служба містить як професійні, так і аматорські відеозаписи, у тому числі відеоблоги.

Характеристику обсягу ринку YouTube наступні. До січня 2012 кількість щоденних переглядів відео на сайті сягала 4 млрд, а в травні того ж року на сайт щохвилини завантажувалося близько 72 годин відеоматеріалів. Станом на 2019 рік на YouTube завантажують близько 300 годин відео щохвилини, а кількість щоденних переглядів відео сягнула 5 млрд.

Головна конку рента перевага за думкою засновника YouTube Чад Герлі полягає не тільки в тому, що за допомогою сайту користувачі можуть показати

свої відео всьому світу; на YouTube можна легко знайти потрібне відео і порекомендувати його іншим.

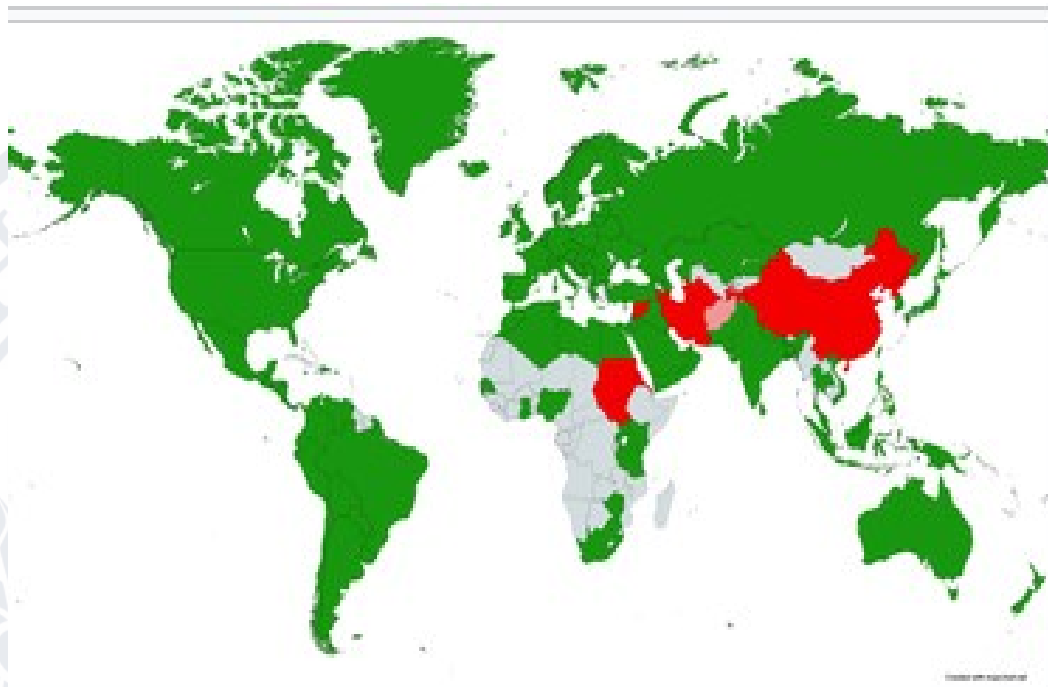


Рис.1. Мапа розповсюдження YouTube на травень 2021 року. Зелене — місцева версія YouTube, сіре — доступний, червоне — в даний час заблоковано, рожеве — раніше блокувалося

Netflix, Inc. — американський провайдер медійних послуг та продюсерська компанія. Основною сферою бізнес-діяльності компанії є надання передплатуваних послуг із мережевої трансляції бібліотеки кінофільмів та телепередач (включно зі створеними самою Netflix). Станом на квітень 2020 року, Netflix налічує 182 млн передплатників по всьому світу, з них 69 млн у США. Netflix доступний у всіх країнах і регіонах, окрім континентального Китаю (через місцеві обмеження), Ірану, Сирії, Північної Кореї та окупованих Росією Криму та Донбасу (через санкції США). Офіси компанії також розташовані в Бразилії, Нідерландах, Індії, Японії та Південній Кореї. Netflix є членом Американської асоціації кінокомпаній (ААК). Виробництво та поширення вмісту компанією теж відбувається по всьому світу.

Netflix займає активнішу позицію як продюсер і дистриб'ютор фільмів та телесеріалів, а зокрема пропонує різноманітний оригінальний контент (Netflix Original) у своїй мережевій бібліотеці. Станом на січень 2016 року, Netflix провадив діяльність у понад 190 країнах. 2016 року компанія випустила близько 126 оригінальних серіалів та фільмів — більше за будь-яку іншу компанію чи канал. У процесі виробництва нового контенту, оплати прав на додатковий вміст і диверсифікації в 190 країнах, компанія залучила мільярди доларів позик: \$21,9 млрд станом на вересень 2017 року (на тлі \$16,8 млрд попереднього року). З цієї суми \$6,5 млрд є довгостроковими позиками, решта — довгостроковими облігаціями. У жовтні 2018 року Netflix анонсував залучення ще \$2 млрд позики для фінансування нових проєктів.

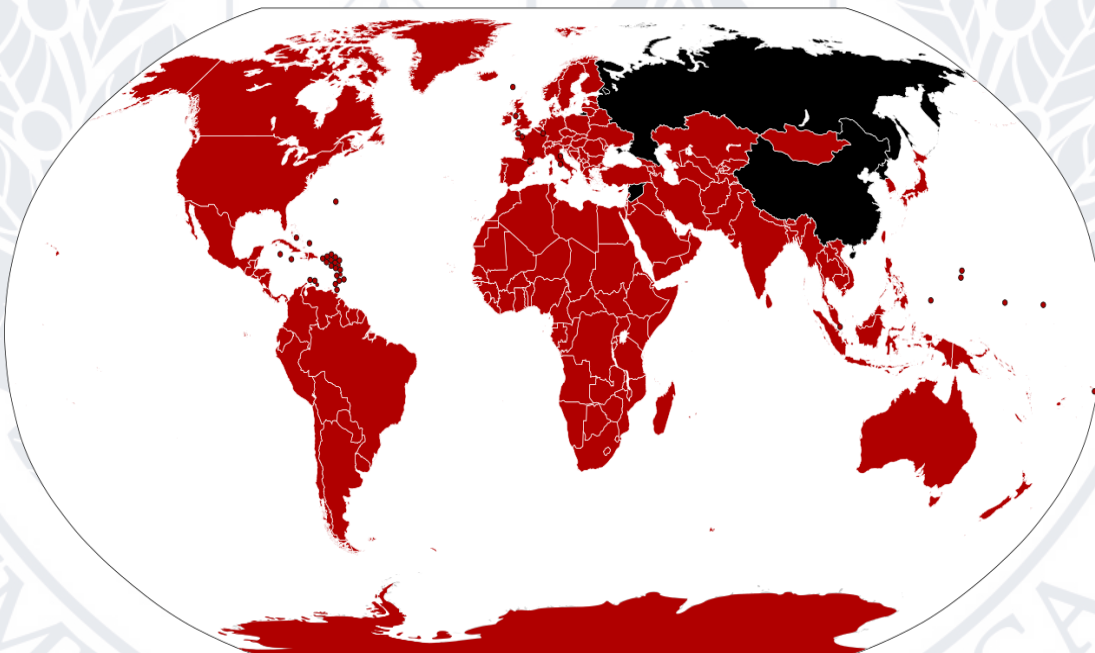


Рис. 2. Доступність Netflix по країнам на березень 2022 року
червоне – доступний; чорне - недоступний

Предмет дослідження системи обробки даних автоматизованої мережевої системи збору і рекомендацій відеоконтенту,

Об'єкт - інтелектуальні методи обробки даних автоматизованої мережевої системи збору і рекомендацій відеоконтенту,

Мета роботи полягає в дослідженні інтелектуальних методів обробки даних автоматизованої мережевої системи збору і рекомендацій відеоконтенту та розробка моделей для формування рекомендацій відеоконтенту.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати сучасні інформаційні та рекомендаційні системи відеоконтенту;
- дослідити набори даних систем автоматизованої мережевої системи збору і рекомендацій відеоконтенту;
- дослідити методи обробки та аналізу даних автоматизованих мережевих систем збору і рекомендацій відеоконтенту;
- дослідження моделей отримання оцінок рейтингу для формування рекомендацій відеоконтенту;
- обґрунтувати вибір методів оцінок рейтингу для формування рекомендацій відеоконтенту;
- побудувати моделі оцінок рейтингу для формування рекомендацій відеоконтенту.

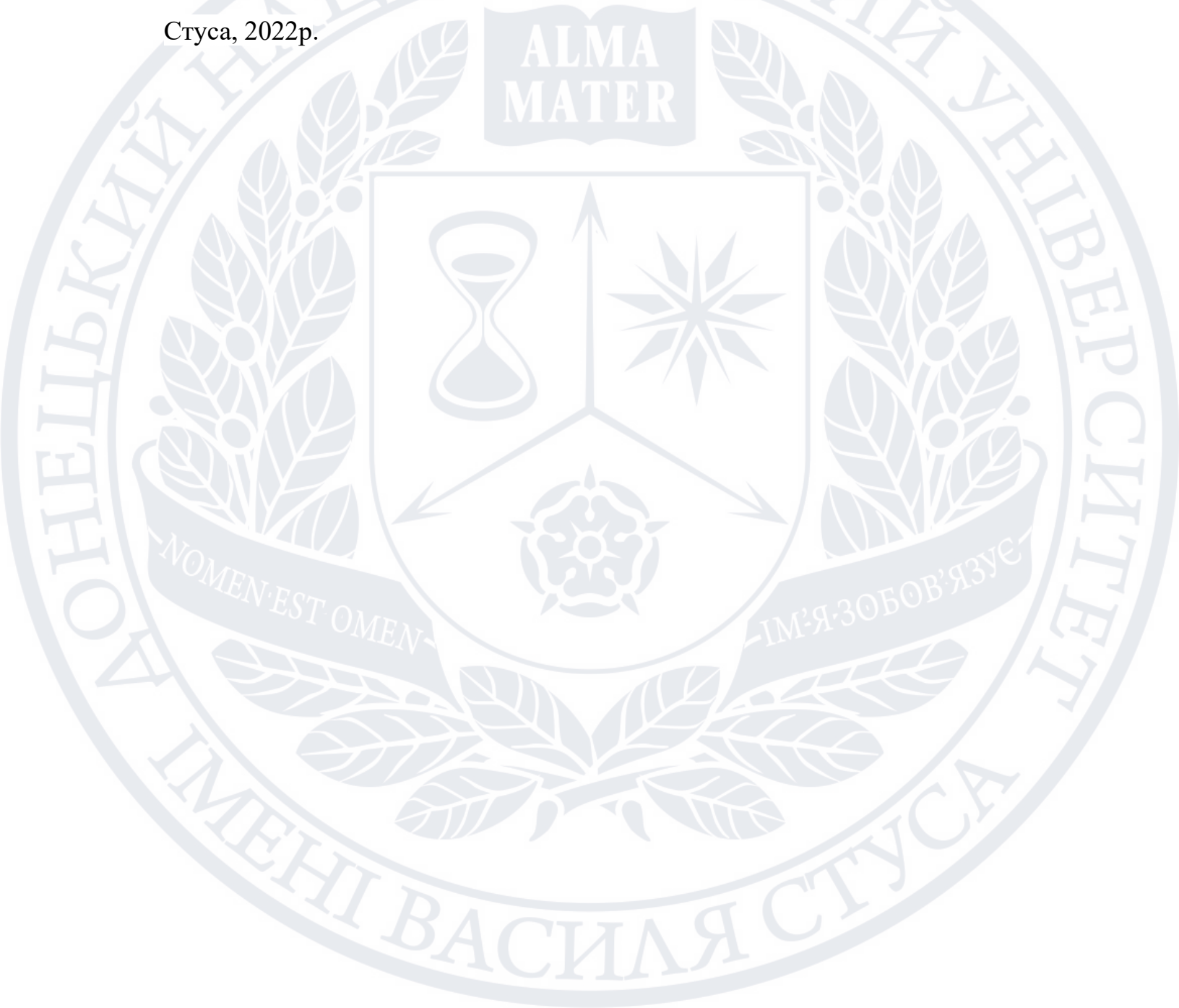
Методи дослідження: системний підхід для при формалізації та класифікації факторів системи пошуку роботи; методи машинного навчання, колаборативної фільтрації великих обсягів даних за набором показників і кількістю спостережень з використанням сучасних програмних засобів (R, Python) та технологій (Google Collab).

Практичне значення результатів роботи полягає в тому, що проведений аналіз дозволяє обрати методи і моделі з урахуванням їх обчислювальної

складності, наявності вхідних даних і даних, що дозволяють оцінки ефективності рекомендацій відеоконтенту.

Основні результати за темою магістерської роботи пройшли апробацію:

– на всеукраїнській науково-практичній конференції «Комп’ютерні технології обробки даних» Донецький національний університет імені Василя Стуса, 2022р.



РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Огляд існуючих мережевих систем збору і рекомендацій відсоконтенту.

В області застосування систем рекомендацій сучасному інтернет- просторі можна виділити наступні сегменти:

- Рекомендації фільмів / музики / відео. Сервіси на зразок Netflix, YouTube або TikTok рекомендують зареєстрованим відвідувачам той чи інший контент в залежності від їх попередніх переглядів або прослуховувань і / або рейтингів того чи іншого відео, фільму або виконавця;
- Рекомендації новин. Система рекомендує «схожі» на раніше прочитані матеріали новини, ґрунтуючись або на схожості ключових слів, або, знову ж таки, на виборі користувачів зі схожими інтересами.
- Рекомендації товарів в інтернет-магазинах. Цей варіант застосування на сьогодні найбільш актуальним; системи рекомендацій інтернет-магазинів пропонують користувачеві придбати певний товар. Рекомендації формуються на основі оцінок рейтингу товару розрахованих за певним алгоритмом, який в якості вхідних даних використовує дані про користувача і його попередню взаємодію з програмним продуктом;

Рекомендаційні системи пропонують вміст після врахування уподобань користувача або групи користувачів, виражених у взаємодії з додатком, як-от публікація коментаря або підписки. Ці сигнали забезпечують отримання зворотного зв'язку від користувача і є основою для класифікацію змісту на той, який вам подобається, і той, який ви б хотіли пропустити. Такі рекомендаційні механізми обробляють великі обсяги інформації для позначення потенційних переваг користувачів. Рекомендаційні системи - це програми, які намагаються визначити що хочуть знайти користувачі, що може

їх зацікавити, і рекомендують їм це. Ці механізми вдосконалюють взаємодію між користувачем і сайтом. Замість статичної інформації вони надають інформацію, що динамічно змінюється: рекомендації генеруються окремо для кожного користувача, ґрунтуючись на його попередньої активності на даному веб-ресурсі, до того ж може враховуватися інформація, що виходить від інших відвідувачів.

Таким чином основні функції рекомендаційних систем, полягають у формуванні персоналізованих та/або групових рекомендацій щодо:

- товару, групи товарів;
- послуги, групи послуг;
- вмісту контенту (відео, аудіо);
- інших програмних додатків;
- інших функцій додатку.

YouTube – це найбільша у світі платформа для створення, обміну та пошуку відеовмісту. Рекомендації YouTube допомагають понад мільярд користувачів відкривають персоналізований вміст із корпусу, що постійно зростає відео. У цій статті ми зосередимося на величезному впливі глибокого навчання на відео YouTube система рекомендацій. На малюнку 1 показано рекомендації на головній сторінці програми YouTube для мобільних пристроїв. Рекомендувати відео YouTube надзвичайно складно з трьох основних точок зору:

- Масштаб: перевірено багато існуючих алгоритмів рекомендацій добре працювати над невеликими проблемами, не працювати на наших масштаб. Вузкоспеціалізовані розподілені алгоритми навчання і ефективні системи обслуговування є важливими для обробки Величезна база та група користувачів YouTube.

- Свіжість: YouTube має дуже динамічний корпус із за секунду завантажується багато годин відео. Система рекомендацій має бути достатньо

швидкою моделювати нещодавно завантажений вміст, а також останні дії, виконані користувачем. Збалансування нового контенту з добре встановленими відео можна зрозуміти перспектива розвідки/експлуатації.

- Шум: історичну поведінку користувачів на YouTube за своєю суттю важко передбачити через рідкість і низку неспостережуваних зовнішніх факторів. Складно отримати основну правду про задоволеність користувачів і натомість моделювати зашумлені сигнали неявного зворотного зв'язку. Крім того, метадані, пов'язані з вмістом, погано структуровані без чітко визначеної онтології. Наші алгоритми потребують бути стійкими до цих наших особливостей навчальні дані.

У поєднанні з іншими продуктами Google, У YouTube відбулася фундаментальна зміна парадигми в напрямку використання глибокого навчання як рішення загального призначення для майже всі проблеми з навчанням. Дана система побудована на Google Brain, який нещодавно був відкритий як TensorFlow.

TensorFlow надає гнучку структуру для експериментів з різними глибокими архітектурами нейронної мережі з використанням широкомасштабного розподіленого навчання. Моделі навчаються приблизно один мільярд параметрів і навчаються на сотнях мільярдів прикладів.

1.2 Дослідження даних для формування рекомендацій відеоконтенту

Розвиток сучасних рекомендаційних систем полягає в підвищенні ефективності алгоритмів формування рекомендацій. Мета цього прогресу - давати відвідувачам сайтів найбільш точні рекомендації, що задовольняють їх запити в певний момент часу. Для досягнення цього алгоритми, що лежать в основі рекомендаційних сервісів, повинні постійно вдосконалюватися. На сьогодні це зокрема алгоритми інтелектуального аналізу даних (Data mining). Схематично роботу цих алгоритмів можна представити наступним чином:

сервіс веб-сайту надає користувачеві набір рекомендацій, далі розробник отримує від нього зворотний зв'язок, аналізує її на предмет відповідності наданих раніше рекомендацій інтересам відвідувача, виконується перенавчання математичної моделі, потім знову пропонує рекомендації і так далі за циклом.

Методи збору інформації, що надаються Інтернетом, значно спростили використання громадської думки за допомогою методів колоборативної фільтрації. Але, з іншого боку, великий обсяг інформації ускладнює втілення цієї можливості. Наприклад, поведінка одних людей досить ясно піддається моделюванню, в той час як інші поведуться абсолютно непередбачувано. І саме другі впливають на зміщення результатів рекомендаційної системи і зниження її ефективності. Інший приклад: користувачі можуть використовувати рекомендаційну систему для навмисного зростання популярності одного продукту щодо іншого. Вони можуть залишати не просто позитивні відгуки про вподобаний контент, але і писати негативні про конкурентів.

Інша проблема, притаманна великим рекомендаційним системам - масштабованість. Традиційні алгоритми непогано справляються з невеликою кількістю даних. Але чим більше інформації має оброблятися, тим важче отримати точні результати. Рекомендаційні системи, що працюють на основі неявних призначених для користувача оцінок, тягнуть за собою проблему дотримання конфіденційності користувачів, захисту персональних даних.

Ще однією проблемою є проблема холодного старту. Вона ділиться на два види: холодний старт по відношенню до користувача (що показувати тільки що зареєструвався людині?) і холодний старт по відношенню до сайтів (кому можна порекомендувати тільки створені сайти?).

Для користувачів існує не так багато прозорості щодо алгоритмів, але все ж існує кілька теорій, заснованих на досвіді користувачів та аналізі експертів. Відповідно до теорії партій, коли відео завантажується, воно демонструється групі партій з різними ідеологіями або простими словами на основі кількох

відмінних факторів, таких як історія перегляду, місце розташування, особисті уподобання. Зміст контенту базується на цих різних партіях. Повторний перегляд, вподобання, спільний доступ та коментарі – це вхідні дані, які використовує алгоритм для оцінки балів вашого вмісту.

Якщо контент користувача подобається групам, тоді його розширюють (рекомендують) для більшої аудиторії того самого процесу мислення, що робить його «вірусним». Як допоміжний фактор, чим більше користувач бере участь у тенденціях, тим більше це його просуває.

У світлі принципу рейтингу влади, доля викласти «вірусне» відео залежить від перших кількох записів. Якщо ті набирають певних поглядів і вподобань, то це створює основу для творця, і його єдиною роботою буде якомога більше взаємодіяти з аудиторією. Отримавши рейтинг авторитетів, користувач опиниться в центрі уваги і буде на кілька рівнів вище початківців, хоча все ще новачком.

Дотримуючись принципу імпульсу затримки, деякі користувачі радили не видаляти будь-яку частину вмісту, незалежно від того, наскільки вона погана. Якщо вміст не приносить ніякої користі, і блогери починають бути менш активними на платформі, програма намагатиметься вдосконалювати їхній вміст самостійно через деякий час, стимулюючи їх створювати більше для своєї аудиторії. Вдосконаленням може бути як збільшення випадків попадання до рекомендованих.

1.3 Дослідження технологій обробки даних в мережевих системах збору і рекомендацій відеоконтенту

При розробці алгоритмів обробки відео контенту можна виокремити наступні технології:

- комп'ютерний зір для аналізу відео,
- технології обробки природною мовою;

- штучний інтелект, який обробляє результати сегментації відео, аудіо, субтитри та метадані (опис, хештеги).

Математичні моделі та алгоритми рекомендаційних систем на їх основі при розробці мають наступні аспекти:

- 1) кількість рекомендованого контенту - його повинно бути достатня кількість, щоб було що і з чого радити і був сенс розробки рекомендаційної системи;
- 2) кількість інформації про рекомендований контент - чим її більше, тим точніше формуються рекомендації;
- 3) обсяг інформації про користувачів (стать, ім'я, вік, країна проживання) - також, чим її більше, тим точніше будуть підібрані рекомендації;
- 4) зручність для користувача частини програми, через який здійснюється зворотній зв'язок, тобто інтерфейсу - чим комфортніше для відвідувача, тим більше інформації можна отримати про його реакції на рекомендації.

Моделі та методи штучного інтелекту аналізують кожную частину викладеного відео, включаючи аудіо, субтитри та метадані (опис, хештеги), щоб створити розуміння змісту та контексту відео тобто класифікувати відео контент.

Це дозволяє програмі обслуговувати користувачів вмістом, який оцінюється як найбільш цікавий і надавати можливість творцям ділитися важливими моментами у повсякденному житті з глобальною аудиторією. Набір характеристик відео: опис відео, хештеги та навіть назва музичного супроводження аналізуються у сукупності.

Після перевірки відео за правилами та положеннями, вміст буде передано невеликій частині аудиторії, і відбудеться оцінка на основі того, як вибіркова група користувачів відреагувала на цей вміст. Кожен відстежуваний показник має можливий пов'язаний бал. Але повторна публікація відео не зробить його «вірусним», оскільки цей алгоритм може розрізнити зайві записи. Як тільки

загальний бал для вмісту буде достатнім, він підштовхне відео до ширшої аудиторії (рис. 1.1).

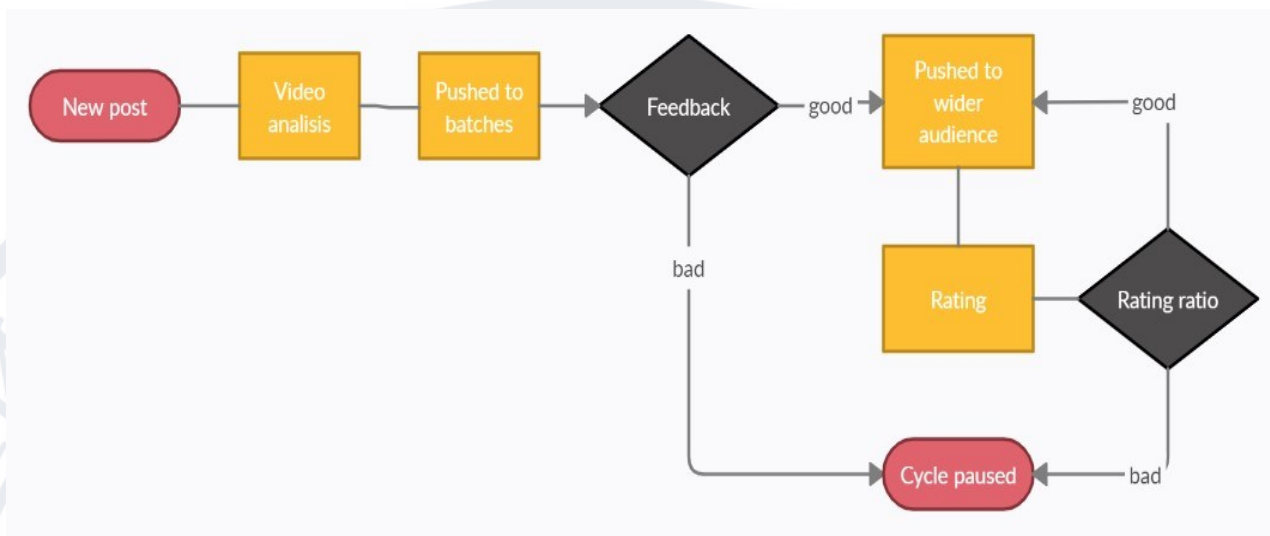


Рисунок 1.1 – Схема обробки інформації нового творця

Якщо вміст не стає «вірусним», цикл не закінчується, а просто призупиняється, і відповідно до теорії затримки імпульсу, він може повернутися назад, коли користувач почне залишати коротший слід у програмі. Це ланцюгова реакція, яка буде тривати і надалі, залучаючи більше творців контенту і, отже, більше аудиторії. В цьому суть цієї платформи, яка може затягувати своїх користувачів протягом тривалого часу.

Додаток починає дізнаватися про відвідувача якомога більше. Перша послідовність відео, які він споживає, допомагає алгоритмам визначити смак людини. Перш за все, щоб втримати нового відвідувача у програмі якомога довше, додаток покаже лише відео, які подобаються ширшому колу аудиторії, таким чином намагаючись зберегти низький рівень виходу.

Після того, як користувач потрапляє в цю екосистему, алгоритми відстежуватимуть метрики користувача та створюватимуть оцінку профілю для кожного жанру. Кожного разу, коли він переглядає відео або споживає всю

тривалість відео, алгоритм бере до уваги ще кілька відео подібного типу для вашого профілю (рис. 1.2).

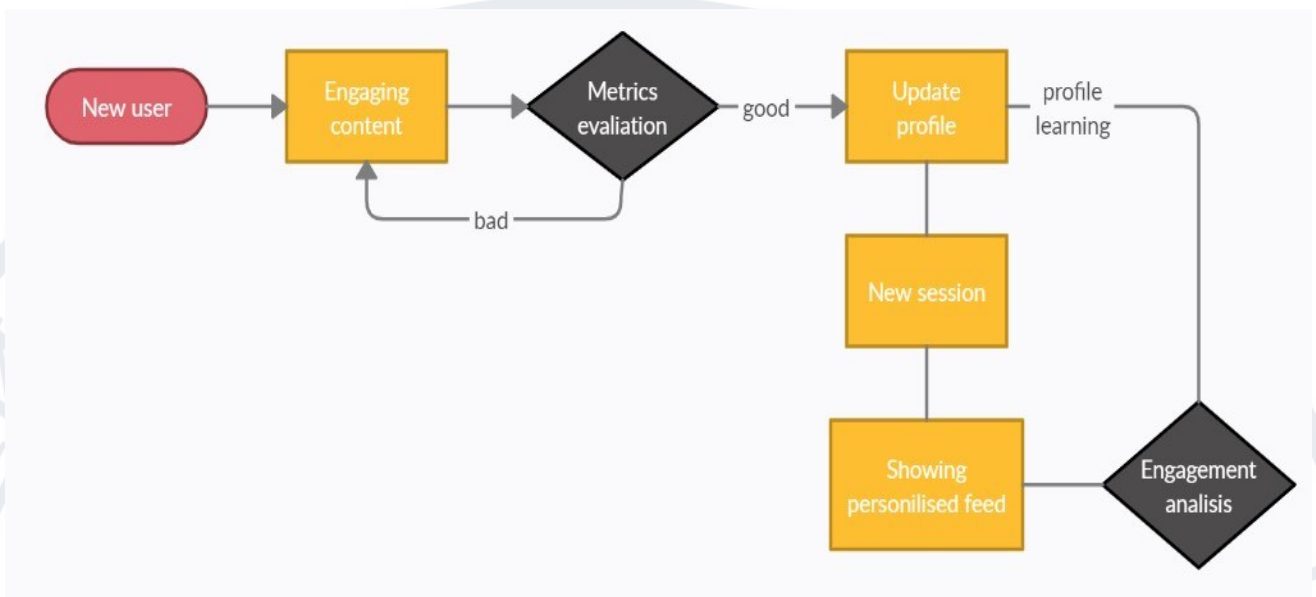


Рисунок 1.2 – Схема обробки інформації нового користувача

Разом з даними метрик, алгоритм також збиратиме особисту інформацію, таку як місцезнаходження, історія пошуку Інтернету, вік, щоб створити більш точний персоналізований канал, тим самим утримуючи користувача в екосистемі якомога довше. Унікальність такого алгоритму полягає в тому, що він може залучити користувачів у всьому світі з різних вікових груп завдяки своїй винятковій системі рекомендацій щодо вмісту.

Мільйони вмісту, що завантажуються користувачами щодня. Практично неможливо відстежувати кожен вміст вручну, тому перевірка проводиться машиною, і якщо вміст позначений нечутливим до протоколів компанії, він передається для перевірки людиною.

Після проходження людського огляду вміст перевіряється на дублювання, далі передається невеликій партії користувачів для початкового зворотного зв'язку. На основі системи метрик надається оцінка вмісту, що визначає долю контенту. Якщо оцінка перевищує деякий поріг, тоді відео розповсюджується

для ширшої аудиторії, але заспокоюється приблизно через тиждень, щоб користувачі постійно відчували новий контент.

Але якщо оцінка зворотного зв'язку низька, цикл тут не зупиняється, а призупиняється, і додаток спостерігає за поведінкою користувача та його активним часом, якщо програма відчуває високу швидкість виходу, тоді цикл знову стане активним і збільшить вміст тим самим заохочуючи творця створювати більше і більше залучати на платформі.

Цей алгоритм діє таким чином, що забезпечує новим та цікавим персоналізованим контентом кожного користувача. Він спостерігає за відвідувачем та покращуватимете рекомендаційну стрічку відповідно до зафіксованих інтересів людини, що не дає нудьгувати та заохочує більшу аудиторію.

Інтелектуальний аналіз даних — це процес вилучення раніше невідомих знань і виявлення цікавих закономірностей із масивного набору даних. Завдяки широкому використанню інформаційних технологій і останнім розробкам мультимедійних систем кількість мультимедійних даних, доступних користувачам, зростає експоненціально. Відео є прикладом мультимедійних даних, оскільки воно містить кілька типів даних, таких як текст, зображення, метадані, зображення та аудіо. Він широко використовується в багатьох основних потенційних застосуваннях, таких як безпека та спостереження, розваги, медицина, освітні програми та спорт. Мета інтелектуального аналізу відеоданих полягає в тому, щоб виявити й описати цікаві закономірності у величезній кількості відеоданих, оскільки це одна з основних проблем дослідницького співтовариства інтелектуального аналізу даних. Порівняно з видобутком інших типів даних, видобуток відеоданих все ще знаходиться в зародковому стані. З відеомайнінгом існує багато складних дослідницьких проблем.

Саме прогрес у технології отримання та зберігання мультимедійних даних призвів до величезного зростання мультимедійних баз даних. Мультимедійний аналіз має справу з вилученням неявних знань, зв'язків мультимедійних даних або інших шаблонів, явно не збережених у мультимедійних даних [4]. Управління мультимедійними даними є одним із найважливіших завдань інтелектуального аналізу даних через неструктуровану природу мультимедійних даних. Основним завданням є обробка мультимедійних даних зі складною структурою, таких як зображення, мультимедійний текст, відео та аудіодані [5-7]. Сьогодні люди мають доступ до величезної кількості відео як на телебаченні, так і в Інтернеті. Отже, існує великий потенціал для додатків на основі відео у багатьох сферах, включаючи безпеку та спостереження, особисті розваги, медицину, спорт, відео новини, освітні програми та фільми тощо. Відеодані містять кілька видів даних, таких як відео, аудіо та текст [8]. Відео складається з послідовності зображень з певною тимчасовою інформацією. Аудіо складається з мови, музики та різних спеціальних звуків, тоді як текстова інформація представляє її лінгвістичну форму.

Відеовміст можна розділити на три категорії, а саме [9].

- 1) Інформація про функції низького рівня, яка включає такі характеристики, як колір, текстура, форма тощо,
- 2) Синтаксична інформація, яка описує вміст відео, включаючи помітні об'єкти, їхнє просторово-часове положення та просторово-часові відносини між ними
- 3) семантична інформація, яка описує те, що відбувається у відео, а також те, що сприймають користувачі. Семантична інформація, яка використовується для ідентифікації відеоподій, має два важливі аспекти [10].

Це:

- (a) Просторовий аспект, представлений відеокадром, наприклад розташування, персонажі та об'єкти, що відображаються у відеокадрі.

(b) Часовий аспект, представлений послідовністю відеокадрів у часі, таких як дії персонажа та рухи об'єкта, представлені в послідовності.

Семантична інформація вищого рівня з відео витягується шляхом вивчення характеристик аудіо, відео та тексту відео. Численні підказки з різних модальностей, включаючи аудіо- та візуальні функції, повністю використовуються та використовуються для захоплення семантичної структури відео, подолаючи розрив між семантичними концепціями високого рівня та функціями низького рівня. У відео визначено три модальності [11]. Це:

- 1) Візуальна модальність, що містить сцену, яку можна побачити на відео;
- 2) Слухова модальність із мовою, музикою та звуками навколишнього середовища, які можна почути разом із відео;
- 3) Текстова модальність, що містить текстові ресурси, які описують вміст відеодокумента.

Відеобазы даних широко поширені, а набори відеоданих надзвичайно великі. Існують інструменти для керування та пошуку в таких колекціях, але потреба в інструментах для вилучення прихованих і корисних знань, вбудованих у відеодані, стає критичною для багатьох програм прийняття рішень.

Технологія обробки відео. Хоча отримання та зберігання відеоданих є легким завданням, отримання інформації з відеоданих є складним завданням. Одним із найважливіших етапів є перетворення відеоданих із неструктурованих даних у структурований набір даних, оскільки обробка відеоданих за допомогою обробки зображень або методів комп'ютерного бачення вимагає функцій структурованого формату [12]. Перед застосуванням методів інтелектуального аналізу даних до ключового кадру відео відео, аудіо та текст виділяються за допомогою методів обробки зображень, усуваючи шум оцифрування та зміни освітлення, щоб уникнути помилкового виявлення.

Відеодані можна структурувати двома способами відповідно до структури вмісту. По-перше, і головне, відеобазы даних зі сценарієм [13] ретельно створюються відповідно до сценарію або плану, який пізніше редагується, компілюється та розповсюджується для використання. Ці бази даних відео мають деякі структури вмісту, такі як фільми, драми та новини. По-друге, бази даних відео без сценаріїв [14] не мають структур вмісту, оскільки «необроблені» відео, такі як відео спостереження та спортивні відео, не мають змін сцени.

Найфундаментальнішим завданням у обробці відео є розділення довгих відеопослідовностей на кілька кадрів і пошук ключового кадру кожного кадру для подальших завдань пошуку відеоінформації. В [15] представив кілька стратегій індексування та пошуку відео на основі візуального вмісту, щоб зосередитися на аналізі структури відео, включаючи виявлення меж кадру, виділення ключових кадрів і сегментацію сцени, виділення функцій і аналіз відеоданих. В [16] обговорювали методи виділення функцій відео, трансформації та представлення відео, а також методи інтелектуального аналізу відеоданих.

Модель відеоданих. Оскільки реляційна або об'єктно-орієнтована модель даних не надає достатніх засобів для ефективного управління та отримання відеоконтенту, для цього потрібна відповідна модель даних. Для цього можна виділити три основні причини [17]:

- (1) відсутність засобів для управління просторово-часовими зв'язками,
- (2) відсутність заснованих на знаннях методів для інтерпретації необроблених даних у семантичний вміст;
- (3) відсутність представлень запитів для складні конструкції.

Модель відеоданих — це представлення відеоданих на основі їх характеристик і вмісту, а також програм, для яких вони призначені [18]. Він заснований на ідеї сегментації відео або відеоанотації. Інтелектуальний аналіз

відеоданих вимагає хорошої моделі даних для представлення відео. Різними авторами були запропоновані різні моделі. Петкович і Йонкер [19] запропонував модель пошуку даних на основі вмісту з чотирма шарами. Це: (i) Рівень необроблених відеоданих із послідовністю кадрів, а також деякими атрибутами відео. (ii) Рівень ознак, що складається з незалежних від домену функцій, які можуть бути автоматично витягнуті з необроблених даних, що характеризує кольори, текстури, форми та рух. (iii) Рівень об'єктів, що має сутності, що характеризуються помітним просторовим виміром і призначені регіонам у кадрах. (iv) Рівень подій із сутностями, які мають помітну часову протяжність, що описує рухи та взаємодії різних об'єктів у просторово-часовій манері.

Чжу та ін. [111] описав ієрархічну структуру керування базою даних відео з використанням семантичних одиниць відео для побудови індексів бази даних. Вони представили ієрархічну модель бази даних відео [20], яка фіксує структури та семантику відеоконтенту в базах даних. Він забезпечує основу для автоматичного відображення від концепцій високого рівня до репрезентативних функцій низького рівня. Він використовується шляхом поділу відеовмісту на набір ієрархічно керованих одиниць, таких як кластери, підкластери, підобласті, кадри або об'єкти, кадри або площини та області відеооб'єктів. Він підтримує більш ефективне представлення відео, методи індексування та доступу до відеоданих.

Технологія сегментації відео.

Першим кроком у будь-якій системі керування відеоданими незмінно є сегментація відеодоріжки на менші одиниці [21,22], що дозволяє виконувати наступні операції обробки відеокадрів, такі як індексація відео, семантичне представлення або відстеження вибраної відеоінформації та визначення кадрів, на яких відбувається перехід від одного кадру до іншого. Візуальна сегментація визначає межі кадру, а сегментація на основі руху визначає панорамування та масштабування [23]. Загалом, більшість відео з повсякденного життя можна

представити за допомогою ієрархії рівнів [24], показаної на рис. 1. Наступні терміни визначаються в [25]. В [26] запропоновано контекстно-залежний метод для виявлення аномалій. Відстежуючи всі рухомі об'єкти у відео, розглядаються три різні рівні просторово-часових контекстів, тобто точкова аномалія відео об'єкта, послідовна аномалія траєкторії об'єкта та аномалія спільного розташування кількох відео об'єктів. Було запропоновано ієрархічний підхід аналізу даних (рис. 1.3). На кожному рівні проводився частотний аналіз, щоб автоматично виявити регулярні правила нормальних подій, і таким чином події, що відхиляються від цих правил, ідентифікуються як аномалії.

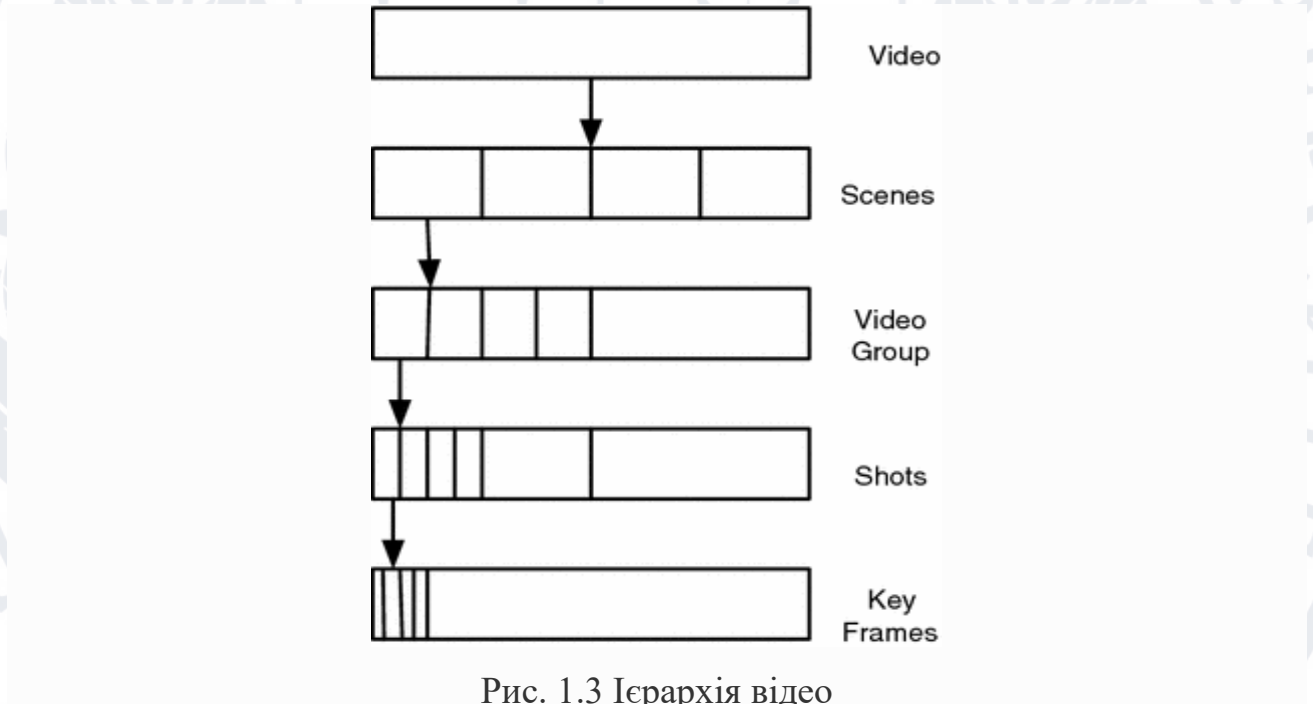


Рис. 1.3 Ієрархія відео

Gowsikhaa та ін. [30] запропонував метод виявлення підозрілих дій, таких як обмін предметами, вхід нової людини, підглядання в бланк відповідей іншого та обмін особами з відео, знятих камерою спостереження, під час обстеження на основі розпізнавання обличчя та рук. Рисунок 1.4 ілюструє короткий дизайн системи розпізнавання діяльності людини.

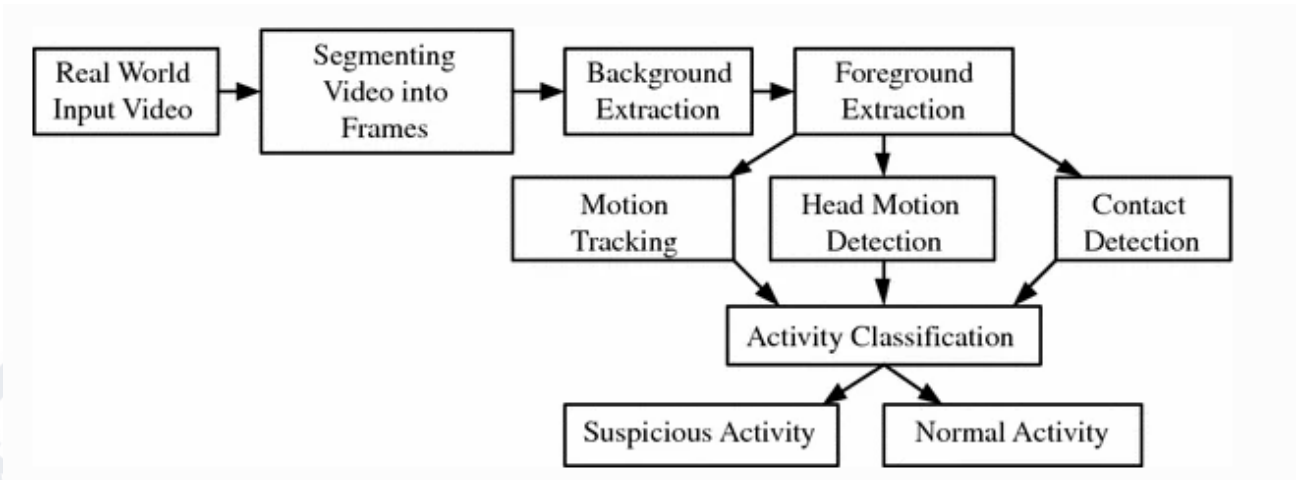


Рисунок 1.4 Узагальнена функціональна схема системи розпізнавання діяльності людини

1.4 Постановка задачі

На основі огляду у попередніх пунктах, можна зробити висновок, що існуючі технології залежать від етапу обробки відеоконтенту, задач вирішуються в системі рекомендацій, наборів даних, що накопичуються в мережевої системи збору і рекомендацій відеоконтенту.

Таким чином завданням даної роботи є дослідження інтелектуальних методів і моделей обробки даних автоматизованої мережевої системи збору і рекомендацій відеоконтенту в залежності від їх призначення, набору показників, що накопичуються в даних системах, обсягів даних.

РАЗДЕЛ 2 ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ОСНОВИ ОБРОБКИ ДАНИХ АВТОМАТИЗОВАНОЇ МЕРЕЖЕВОЇ СИСТЕМИ ЗБОРУ І РЕКОМЕНДАЦІЙ ВІДЕОКОНТЕНТУ

2.1 Формування наборів даних для рекомендацій відеоконтенту мережевої системи. Групові рекомендації відіграють важливу роль у сучасних системах соціальних медіа, де користувачі формують соціальні групи, щоб разом отримувати мультимедійний вміст і взаємодіяти один з одним, замість того, щоб споживати онлайн-контент окремо. Обмеження традиційних підходів групових рекомендацій полягають у наступному. По-перше, вони зазвичай роблять висновок про вподобання членів групи на підставі їх історичної поведінки, не враховуючи вподобання неактивних користувачів з розріджених історичних даних. По-друге, стосунки між членами групи не вивчаються цими підходами, які не в змозі охопити невід’ємну особистість членів групи. Щоб вирішити ці проблеми, ми пропонуємо структуру групових рекомендацій із соціальною підтримкою, яка спільно використовує як соціальні стосунки, так і соціальну поведінку, щоб не лише визначити уподобання групи, але також моделюють характеристики толерантності та альтруїзму членів групи. Базуючись на спостереженні, що наступні відносини в соціальній онлайн-мережі відображають спільні інтереси користувачів, ми пропонуємо модель групових переваг, засновану на зовнішніх експертах членів групи. Крім того, ми моделюємо толерантність користувачів (готовність отримувати небажаний контент) і альтруїзм (готовність отримувати контент, який віддають перевагу друзям). Нарешті, на основі моделі групових переваг ми розробляємо алгоритми рекомендацій для користувачів у різних соціальних контекстах. Експериментальні результати демонструють ефективність нашого підходу, який значно підвищує точність рекомендацій порівняно з традиційними підходами, особливо у випадках неактивних членів групи. Базуючись на спостереженні, що наступні відносини в соціальній онлайн-мережі відображають спільні інтереси

користувачів, ми пропонуємо модель групових переваг, засновану на зовнішніх експертах членів групи. Крім того, ми моделюємо толерантність користувачів (готовність отримувати небажаний контент) і альтруїзм (готовність отримувати контент, який віддають перевагу друзям). Нарешті, на основі моделі групових переваг ми розробляємо алгоритми рекомендацій для користувачів у різних соціальних контекстах. Експериментальні результати демонструють ефективність нашого підходу, який значно підвищує точність рекомендацій порівняно з традиційними підходами, особливо у випадках неактивних членів групи. Базуючись на спостереженні, що наступні відносини в соціальній онлайн-мережі відображають спільні інтереси користувачів, ми пропонуємо модель групових переваг, засновану на зовнішніх експертах членів групи. Крім того, ми моделюємо толерантність користувачів (готовність отримувати небажаний контент) і альтруїзм (готовність отримувати контент, який віддають перевагу друзям). Нарешті, на основі моделі групових переваг ми розробляємо алгоритми рекомендацій для користувачів у різних соціальних контекстах. Експериментальні результати демонструють ефективність нашого підходу, який значно підвищує точність рекомендацій порівняно з традиційними підходами, особливо у випадках неактивних членів групи. Крім того, ми моделюємо толерантність користувачів (готовність отримувати небажаний контент) і альтруїзм (готовність отримувати контент, який віддають перевагу друзям). Нарешті, на основі моделі групових переваг ми розробляємо алгоритми рекомендацій для користувачів у різних соціальних контекстах. Експериментальні результати демонструють ефективність нашого підходу, який значно підвищує точність рекомендацій порівняно з традиційними підходами, особливо у випадках неактивних членів групи. Крім того, ми моделюємо толерантність користувачів (готовність отримувати небажаний контент) і альтруїзм (готовність отримувати контент, який віддають перевагу друзям). Нарешті, на основі моделі групових переваг ми розробляємо алгоритми

рекомендацій для користувачів у різних соціальних контекстах. Експериментальні результати демонструють ефективність нашого підходу, який значно підвищує точність рекомендацій порівняно з традиційними підходами, особливо у випадках неактивних членів групи.

На відміну від величезної кількості досліджень матричної факторизації методів [27], є відносно мало робіт з використанням глибоких нейронних мереж для рекомендаційних систем. Нейронні мережі використовуються для рекомендацій новин у [28], цитат у [29] і оцінки огляду в [30]. Спільна фільтрація сформульована як глибока нейронна мережа в [31] і автокодерах в [32]. Elkahky та ін. використовував глибоке навчання для міждоменого моделювання користувача [33]. У контексті контенту Burges et al. використовували глибокі нейронні мережі для музичних рекомендацій [34].

Загальна структура системи рекомендацій проілюстрована на малюнку 2. Система складається з двох нейронних мереж: одна для створення кандидатів і одна для рейтингу. Мережа створення кандидатів приймає події з історію дій користувача на YouTube як вхідні дані та отримує а невелика підмножина (сотні) відео з великого корпусу. Ці кандидати мають бути загальнодоречними для користувача з високою точністю. Лише мережа генерації кандидатів забезпечує широку персоналізацію за допомогою спільної фільтрації.

Подібність між користувачами виражається в термінах coarse такі функції, як ідентифікатори переглядів відео, маркери пошукових запитів і демографія.

Представлення кількох «найкращих» рекомендацій у списку вимагає представлення тонкого рівня для визначення відносної важливості серед кандидатів з високою запам'ятовуваністю. Мережа рейтингу виконує це завдання, призначаючи бали кожному відео відповідно до бажаної цільової функції, використовуючи широкий набір характеристики, що описують відео та користувача. Найвищий бал відео представлені користувачеві, упорядковані за їх балом.

Двоетапний підхід до рекомендації дозволяє давати рекомендації з дуже великого корпусу (мільйони) відео, але все ще впевнений, що невелика кількість відео, які з'являються на пристрої, персоналізовані та цікаві для користувача. Крім того, ця конструкція дозволяє змішувати кандидатів, створених іншими джерелами, такими як описані в попередній роботі [35].

Під час розробки ми широко використовуємо офлайн-метрики (точність, запам'ятовуваність, втрата рейтингу тощо), щоб керувати ітераційними покращеннями нашої системи. Однак для остаточного визначення ефективності алгоритму чи моделі використовується A/B-тестування за допомогою живих експериментів. У прямому експерименті ми можемо виміряти незначні зміни рейтингу кліків, час перегляду та багато інших показників, які вимірюють залучення користувачів. Це важливо, оскільки живі результати A/B не завжди корелюються з офлайн-експериментами

Під час формування кандидатів величезний корпус YouTube розбирається до сотень відео, які можуть бути актуальні для користувача.

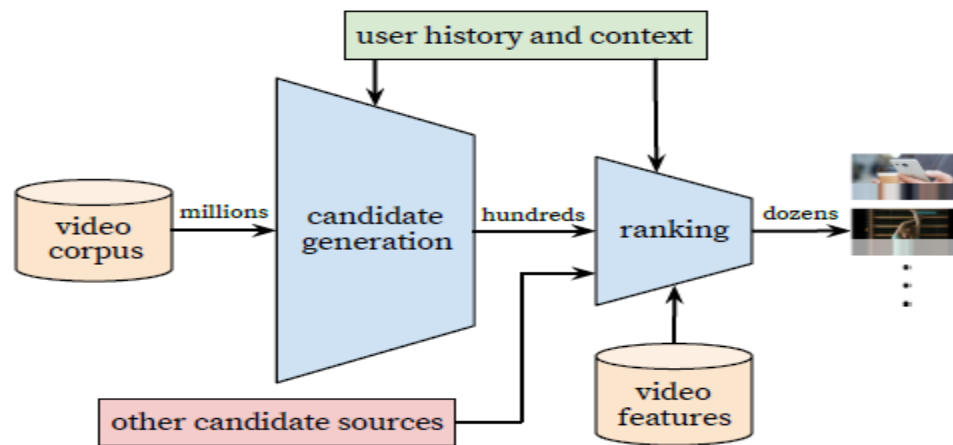


Рисунок 2.1: Архітектура системи рекомендацій демонстрація «воронки», де відео кандидатів витягуються та ранжуються перед представленням.

Ранні ітерації нейронної мережі модель імітувала цю поведінку факторизації з неглибоким мережі, які лише вбудовані попередні годинники

користувача. З цієї точки зору підхід можна розглядати як нелінійне узагальнення методів факторизації.

Моделі рекомендацій як класифікацій

Рекомендація представляється як надзвичайна багатокласова класифікація, де проблема передбачення стає точною класифікацією певного перегляду відео в час t серед мільйонів відео i (класів) із корпусу V на основі користувача U і контекст C ,

$$P(w_{\tau} = i | U, C) = \frac{e^{v_i u}}{\sum_{j \in V} e^{v_j u}},$$

де

$u \in R^N$ являє собою багатовимірне «вбудовування» користувача, контекстна пара та $v_i \in R^N$ представляють вбудовування кожного відео кандидата. У цьому налаштуванні вбудовування є просто зіставлення розріджених об'єктів (окремих відео, користувачів і т. д.) в щільний вектор в R^N .

Завдання глибинної навчання мережи полягає в тому, щоб вивчати вбудовування користувача u як функцію історію користувача та контекст, які корисні для розрізнення серед відео з класифікатором softmax.

Хоча на YouTube існують чіткі механізми зворотного зв'язку (великі пальці вгору/вниз, опитування в продукті тощо) використовується неявний зворотний зв'язок [36] годинників для навчання моделі, де позитивним прикладом є те, що користувач завершує відео. Цей вибір є на основі на порядки більш неявної історії користувача доступні, що дозволяє нам виробляти рекомендації глибоко в хвіст, де явний зворотний зв'язок надзвичайно рідкісний.

Моель Extreme Multiclass. Щоб ефективно навчати таку модель із мільйонами класів, використовується техніка вибірки негативних класів із фонового розподілу («вибірка кандидатів»), а потім виправлення цієї вибірки через зважування важливості [37]. Для кожного прикладу втрата крос-ентропії мінімізована для справжньої мітки і вибірковій негативній класу. На практиці відбирається кілька тисяч негативів, що відповідає понад 100 прискорення в рази порівняно з традиційним softmax. Популярним альтернативним підходом є ієрархічний softmax [38]. В ієрархічному softmax обхід кожного вузла в дереві передбачає розрізнення між наборами класів, які часто не пов'язані, роблячи проблему класифікації набагато складнішою і знижуючою продуктивність.

Під час навчання потрібно обчислити найімовірніше N класів (відео), щоб вибрати N найкращих для презентації до користувача. Оцінка мільйонів елементів із строгою затримкою обслуговування в десятки мілісекунд вимагає приблизно лінійна схема нарахування балів за кількістю класів. Попередні системи на YouTube поклалися на хешування [39], а даний класифікатор використовує подібний підхід. Так як відкалібровані ймовірності з вихідного рівня softmax не потрібні під час подачі, проблема підрахунку очок зменшується до найближчої пошуку сусідів у просторі скалярного добутку для якого загального можна використовувати цільові бібліотеки [40]. В даному випадку А/В тести не особливо чутливі до вибору алгоритм пошуку найближчих сусідів.

Архітектура моделі.

На основі мовних моделей безперервної сумки слів [41], досліджуються багатовимірні вбудовування для кожного відео в фіксований словниковий запас і ці вбудовування передаються у нейронну мережу. Представлена історія переглядів користувача за допомогою послідовності розріджених ідентифікаторів відео змінної довжини відображається в щільне векторне представлення через вкладення. Мережа потребує щільних входів фіксованого

розміру та просто усереднюючи вбудовування, які показали найкращі результати серед кількох стратегій (сума, покомпонентний максимум тощо). важливо, вкладення вивчаються разом з усіма іншими параметри моделі через нормальний градієнтний спад у зворотному напрямку оновлення. Функції об'єднані в широкий перший шар, а потім кілька шарів повністю з'єднаних Rectified Linear Units (ReLU) [42]. На малюнку 3 показано мережева архітектура з додатковими функціями, не пов'язаними з переглядом відео описані нижче.

Неоднорідні сигнали

Ключова перевага використання глибоких нейронних мереж як узагальнення матричної факторизації є довільна неперервна і категоріальні ознаки можна легко додати до моделі.

Історія пошуку обробляється так само, як і історія переглядів запит розбивається на уніграми та біграми, і кожен токен є вбудованим. Після усереднення користувач токенізується, вбудовується запити представляють узагальнену щільну історію пошуку.

Демографічні особливості є важливими для надання попередніх так що рекомендації поведуться прийнятно для нових користувачів. Географічний регіон і пристрій користувача вбудовані та зчеплені. Прості двійкові та безперервні функції, такі оскільки стать користувача, стан входу та вік вводяться безпосередньо у мережу як реальні значення, нормовані на $[0; 1]$.

Функція «Приклад віку». Щосекунди завантажується багатогодинне відео YouTube. Рекомендую нещодавно завантажене (\свіже) контент надзвичайно важливий для YouTube як продукту. Однак ми постійно спостерігаємо, що користувачі віддають перевагу свіжому вмісту не за рахунок актуальності. На додаток до першого порядку Це просто рекомендація нових відео, які потрібні користувачам спостерігати, існує критичне вторинне явище початкового завантаження і поширення вірусного контенту [43].

Системи машинного навчання часто демонструють неявну упередженість до минулого, тому що вони навчені передбачати майбутнє поведінки з історичних прикладів. Розповсюдження відео популярність є дуже нестационарною, але мультиноміальним розподілом над корпусом, створеним нашим рекомендувачем повторно середня ймовірність перегляду у вікні навчання кілька тижнів. Щоб виправити це, ми годуємо вік навчальний приклад як особливість під час навчання. При подачі часу ця функція встановлюється на нуль (або злегка негативне значення), щоб повторно

Тобто модель робить прогнози в самому кінці вікна навчання.

Рисунок 2.2 демонструє ефективність цього підходу на довільно вибране відео [44].

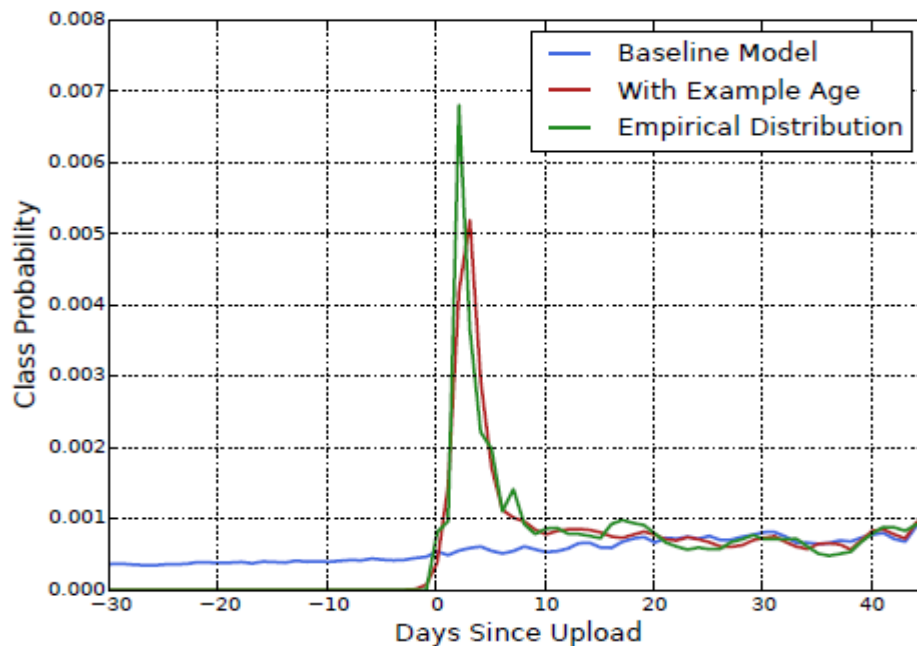


Рисунок 2.2. Для заданого відео модель навчена з прикладом віку як ознаки.

Для заданого відео [45] модель навчена з прикладом віку як ознаки здатний точно представляють час завантаження та залежні від часу спостерігаються в даних. Без функції, модель передбачила б приблизно середнє значення ймовірність за навчальне вікно.

Вибір мітки та контексту

Важливо часто наголошувати на цій рекомендації передбачає вирішення сурогатної проблеми та передачу привести до певного контексту. Класичним прикладом є припущення що точне прогнозування рейтингів веде до ефективності рекомендації фільмів [2]. Ми виявили, що вибір ця сурогатна проблема навчання має величезне значення на продуктивність у тестуванні A/B, але це дуже важко виміряти за допомогою експериментів.

Приклади навчання генеруються з усіх переглядів YouTube (навіть ті, що вбудовані на інших сайтах), а не просто годинники за нашими рекомендаціями. Інакше було бути дуже складним для появи нового вмісту та рекомендувача буде надмірно упередженим до експлуатації. Якщо користувачі знаходять відео не за допомогою наших рекомендацій, ми хочемо мати можливість швидко поширювати це відкриття для інших за допомогою спільного фільтрування. Інший ключ розуміння того, що покращені показники в реальному часі полягають у створенні фіксованої кількості навчальних прикладів на користувача, ефективно зважування наші користувачі однаково у функції втрати. Цьому завдав мал. когорта високоактивних користувачів від домінування втрати.

Дещо суперечить інтуїції, але потрібно бути дуже обережним приховувати інформацію від класифікатора з метою запобігання модель від використання структури сайту та над проблемою сурогату. Розглянемо як приклад.

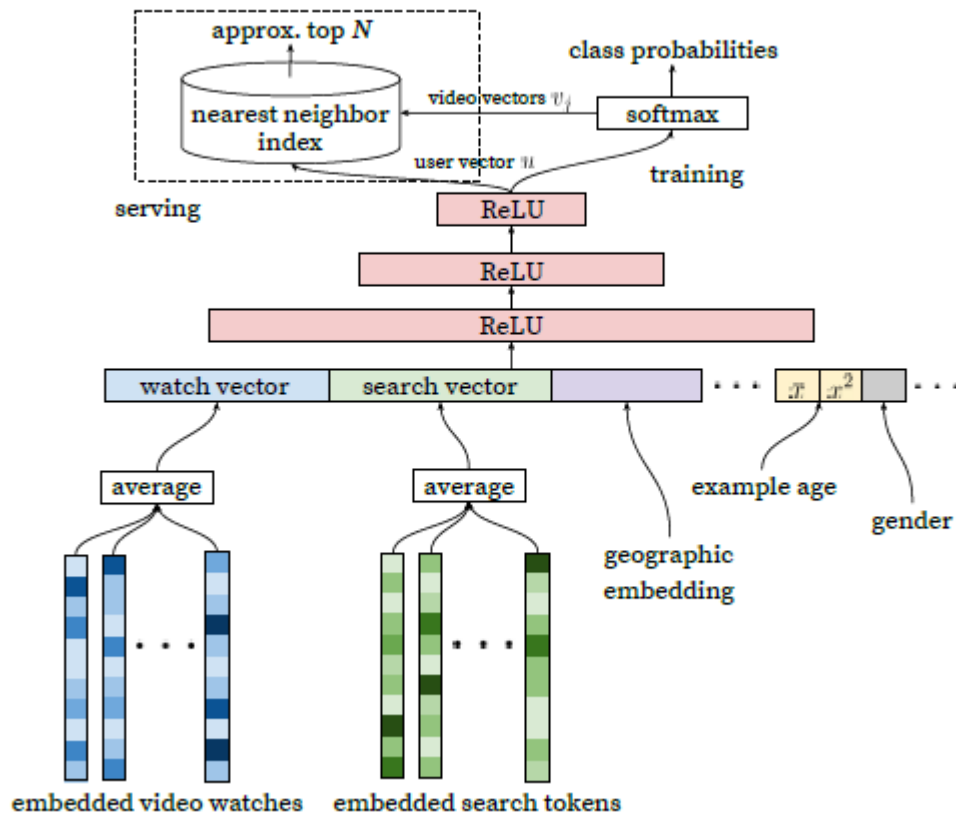


Рис. 2.3. Глибока архітектура моделі генерації кандидатів, яка показує вбудовані розріджені функції, об'єднані з щільні особливості.

Вбудовування усереднюється перед конкатенацією для перетворення мішків змінного розміру розріджених ідентифікаторів у вектори фіксованої ширини, придатні для введення в приховані шари. Усі приховані шари повністю з'єднані. В процесі навчання, втрата перехресної ентропії мінімізована за допомогою градієнтного спуску на виході вибірки softmax.

Під час обслуговування виконується приблизний пошук найближчого сусіда для створення сотень відео-кандидатів рекомендації. У випадку, коли користувач щойно ввів пошуковий запит для "taylor swift". Оскільки наша проблема поставлена як передбачення наступного переглянутого відео, класифікатор, враховуючи цю інформацію, спрогнозує що найвірогідніше переглядати ті відео, які відображаються на відповідній сторінці результатів

пошуку для "taylor swift". Як не дивно, відтворюється останній пошук користувача сторінка як рекомендації домашньої сторінки працює дуже погано.

Відкидаючи інформацію про послідовність і представляючи пошук запити з неупорядкованим мішком токенів, класифікатор – ні довше безпосередньо обізнаний про походження мітки.

Природні моделі споживання відео зазвичай призводять до дуже асиметричних ймовірностей спільного перегляду. Епізодичні серіали є зазвичай переглядаються послідовно, і користувачі часто знаходять виконавців у жанрі, що починається з найпопулярнішого раніше зосередження на менших нішах. Тому ми знайшли набагато краще прогнозування продуктивності наступного перегляду користувача, а не прогнозування випадково триманого годинника (рис. 2.4). Багато спільних Системи фільтрації неявно вибирають мітки та контексти, тримаючи випадковий предмет і прогнозуючи його інші елементи в історії користувача (5a). Це призводить до витoku майбутньої інформації і ігнорує будь-які асиметричні моделі споживання.

На відміну від цього, ми «відкочуємо» історію користувача, вибираючи випадковий спостерігати та вводити лише ті дії, які користувач виконав перед годинник з висунутою етикеткою (5b)

Експерименти з характеристиками та глибиною.

Додавання функцій і глибини значно підвищує точність на дані про утримання, як показано на рис. 2.6. У цих експериментах словниковий запас із 1 мільйона відео та 1 мільйона пошукових токенів були вбудовані з 256 вівса кожен у мішок максимального розміру з 50 останніх переглядів і 50 останніх пошуків. Softmax шар виводить мультиноміальний розподіл на той самий 1M відеокласи з розмірністю 256 (що можна подумати як окреме вбудовування вихідного відео). Ці моделі були навчені до конвергенції для всіх користувачів YouTube, відповідно на кілька епох над даними. Структура мережі дотримувалася загального шаблону "вежа", в якому дно мережа найширша, і

кожен наступний прихований шар ділиться навпіл кількість одиниць (аналогічно рис. 2.3). Нульова глибина мережа є ефективною схемою лінійної факторизації.

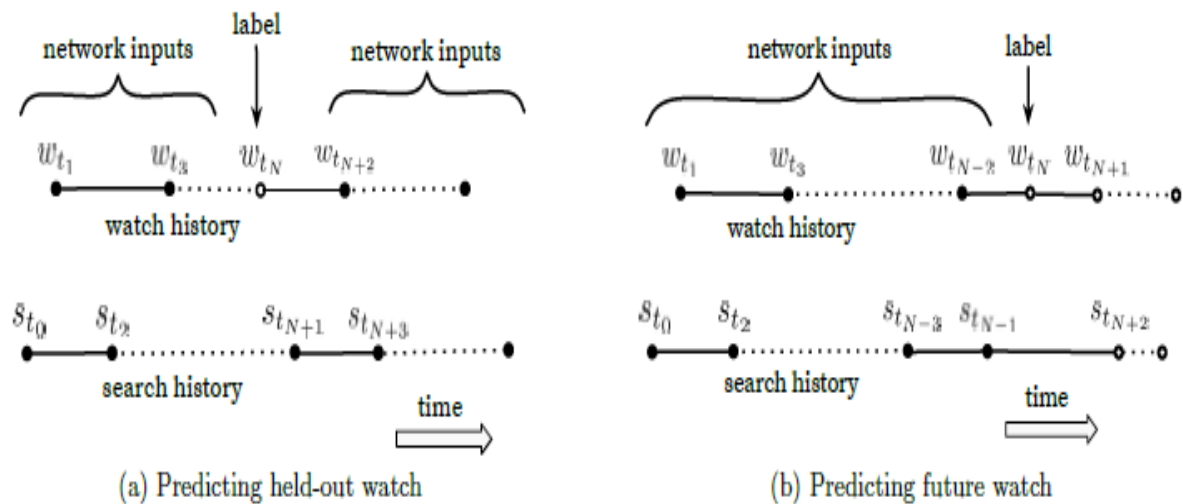


Рис 2.4. Вибір міток і вхідного контексту для моделі складно оцінити, але він має великий вплив на живе виконання.

Тут суцільні події є вхідними функціями в мережу, тоді як порожні події виключено. Ми виявили, що прогнозування майбутнього годинника (5b) показало кращі результати в А/В-тестуванні. У (5b) приклад вік виражається як $t_{max} - t_N$, де t_{max} — це максимальний час, який спостерігається в даних тренування. Працює дуже схоже на попередню систему. Ширина і глибину додавали, поки додаткова користь не зменшувалась і конвергенція стала важкою: Глибина 0: лінійний шар просто перетворює конкатенацію шар, щоб відповідати розміру softmax 256

Глибина 1: 256 ReLU

Глибина 2: 512 ReLU ! 256 ReLU

Глибина 3: 1024 ReLU ! 512 ReLU ! 256 ReLU

Глибина 4: 2048 ReLU ! 1024 ReLU ! 512 ReLU !
256 ReLU

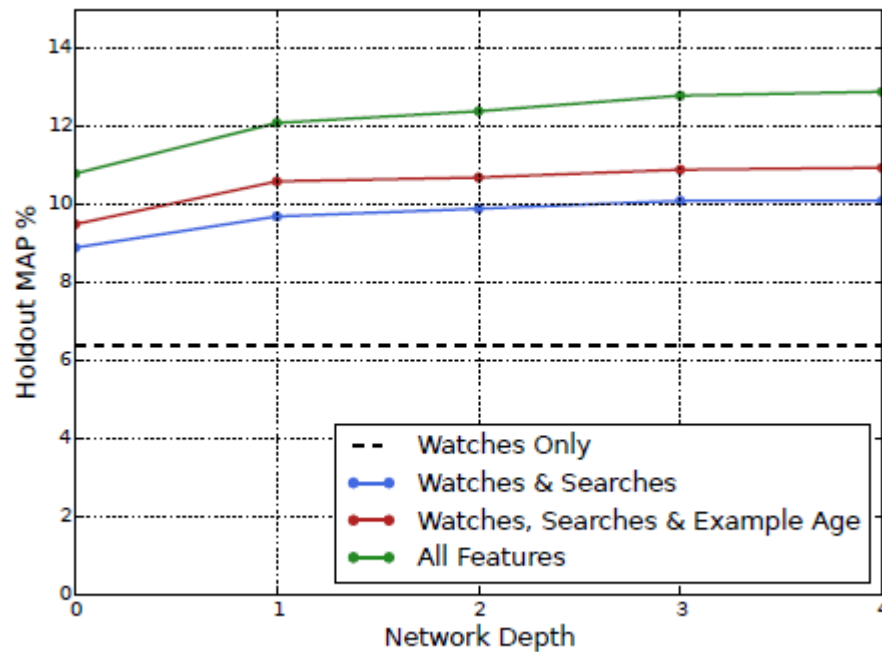


Рисунок 2.6. Функції, окрім вбудованих відео підтвердити середню точність (MAP) і шари глибини додають виразності, щоб модель може ефективно використовувати ці додаткові функції за допомогою покращуючи їх взаємодію.

2.2 Визначення вимог до моделей рекомендацій відеоконтенту мережевої системи

Рейтинг.

Основна роль рейтингу полягає у використанні даних про покази спеціалізуватися та калібрувати прогнози кандидатів для конкретного інтерфейс користувача. Наприклад, користувач може дивитися задане відео з високою ймовірністю, але навряд чи клацне на конкретній домашній сторінці

завдяки вибору ескіз зображення. Під час рейтингу є доступ до багатьох більше функцій, що описують відео та стосунки користувача до відео, оскільки лише кілька сотень відео набрані, а не мільйони, набрані кандидатом покоління. Рейтинг також має вирішальне значення для створення різних ансамблів потенційні джерела, оцінки яких не можна порівнювати безпосередньо.

Ми використовуємо глибоку нейронну мережу зі схожою архітектурою покоління кандидатів, щоб призначити незалежний бал кожному відеовраження за допомогою логістичної регресії (рис. 2.7). Список відео потім сортується за цією оцінкою та повертається до користувача. Наша кінцева мета рейтингу постійно вдосконалюється базується на результатах живого тестування A/B, але загалом є простим функція очікуваного часу перегляду на показ. Рейтинг за рейтингом кліків часто рекламує оманливі відео, які користувач не завершує (\clickbait), тоді як час перегляду краще фіксує залучення [13, 25].

Представлення функції.

Функції відокремлені традиційною таксономією категоріальних і безперервних/порядкових ознак. Категоричний функції, які використовуються, дуже відрізняються за своєю потужністю – деякі є двійковими (наприклад, чи ввійшов користувач), а інші мати мільйони можливих значень (наприклад, останній пошук користувача запит). Функції далі поділяються залежно від того, чи вони внести лише одне значення (\univalent) або набір значень (\multivalent"). Приклад однозначної категоріальної ознаки – це ідентифікатор відео для оцінюваного показу, а відповідна багатовалентна ознака може бути мішком останнього N ідентифікаторів відео, які переглянув користувач. Ми також класифікуємо ознаки відповідно до того, чи описують вони властивості предмета (\impression") або властивості користувача/контексту (\query"). Функції запиту обчислюються один раз на запит під час показу характеристики обчислюються для кожного елемента, який оцінюється.

Розробка функцій

Зазвичай використовуються сотні функцій у моделях рейтингу, приблизно рівномірно розподілені між категоричними та безперервними.

Незважаючи на обіцянку глибокого навчання для полегшення тягар інженерних особливостей вручну, характер необроблених даних не піддаються безпосередньому введенню нейронній мережі прямого зв'язку. За рахунок цього витрачається чимало даних.

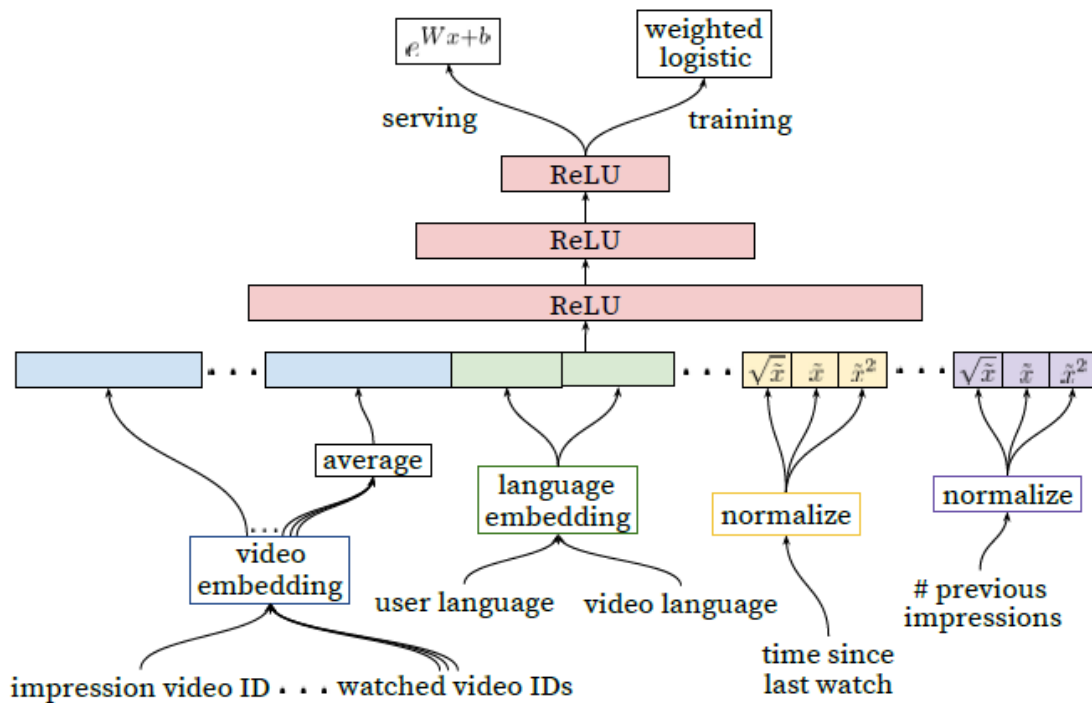


Рисунок 2.7. Мережева архітектура глибокого ранжирування, що зображує вбудовані категоріальні функції (як однозначні, так і multivalent) із спільними вкладеннями та потужностями нормалізованих неперервних ознак.

Всі верстви повністю підключені. На практиці в мережу подаються сотні функцій. Інженерні ресурси, що перетворюють дані користувачів і відео на корисні властивості. Основна проблема полягає в представленні часової послідовності дій користувача і те, як ці дії пов'язані між собою до підрахунку показу відео.

Найважливішими сигналами є ті, що описати попередню взаємодію користувача з самим елементом і інші подібні предмети, що відповідають досвіду інших у рейтингу оголошення [7]. Як приклад розглянемо минулу історію користувача з каналу, який завантажив відео, оцінюється – скільки відео, які користувач дивився з цього каналу? Коли користувач востаннє дивився відео на цю тему? Ці безперервні функції, що описують минулі дії користувача на пов'язаних елементах є особливо потужними, тому що вони добре узагальнюють між різнорідними предметами. Також виявлено, що важливо поширювати інформацію від покоління кандидатів до рейтингу у формі ознак, наприклад, які джерела висунули це відео кандидат? Які бали вони поставили?

Характеристики, що описують частоту минулих показів відео також критичні для введення "churn" у рекомендаціях (послідовні запити не повертають ідентичні списки). Якщо користувачеві нещодавно було рекомендовано відео, але він не дивився тоді модель природним чином знизить це враження завантаження наступної сторінки. Послуга до другого показу а історія переглядів — це інженерний подвиг назовні рамки цього документу, але життєво важливі для створення чуйного рекомендації.

Вбудовування категоріальних ознак

Подібно до створення кандидатів, ми використовуємо вбудовування для відображення розріджені категоріальні ознаки підходять для щільних уявлень для нейронних мереж. Кожен унікальний ідентифікатор ("словник") має окреме вивчене вбудовування зі збільшенням розмірності приблизно пропорційна логарифму кількості унікальних значень. Ці словники прості таблиці пошуку, створені шляхом попередньої передачі даних навчання. Дуже великі простори ідентифікаторів потужності (наприклад, ідентифікатори відео або терміни пошукового запиту) скорочуються шляхом включення лише перші N після сортування на основі їх частоти в кліках.

Значення поза словниковим запасом просто зіставляються з нульове вбудовування. Як у кандидатській генерації, багатовалентний категоричні вбудовування ознак усереднюються перед подачею в мережу.

Важливо, що категориальні ознаки в тому самому ідентифікаційному просторі також поділитися базовими `embeddings`. Наприклад, існує єдиний глобальне вбудовування ідентифікаторів відео з багатьма різними функціями використовувати (ідентифікатор відео показу, ідентифікатор останнього переглянутого відео користувачем, ідентифікатор відео, яке `\seeded`" рекомендація, тощо). Незважаючи на спільне вбудовування, кожна функція подається окремо у мережу, щоб вищезгадані рівні могли навчатися спеціалізовані представлення для кожної функції. Спільний доступ до вставок має велике значення для вдосконалення узагальнення, прискорення навчання і зменшення вимог до пам'яті. Переважна більшість параметрів моделі мають високу потужність простори вбудовування - наприклад, один мільйон вбудованих ідентифікаторів у 32-вимірному просторі мають у 7 разів більше параметрів ніж повністю з'єднані шари шириною 2048 одиниць.

Вбудовування категориальних ознак.

Подібно до створення кандидатів, використовується вбудовування для відображення розріджені категориальні ознаки підходять для щільних уявлень для нейронних мереж. Кожен унікальний ідентифікатор (`\словник`") має окреме вивчене вбудовування зі збільшенням розмірності приблизно пропорційна логарифму кількість унікальних значень. Ці словники прості таблиці пошуку, створені шляхом попередньої передачі даних навчання. Дуже великі простори ідентифікаторів потужності (наприклад, ідентифікатори відео або терміни пошукового запиту) скорочуються шляхом включення лише перші N після сортування на основі їх частоти в кліках.

Значення поза словниковим запасом просто зіставляються з нульове вбудовування. Як у кандидатській генерації, багатовалентні категоричні вбудовування ознак усереднюються перед подачею в мережу.

Важливо, що категориальні ознаки в тому самому ідентифікаційному просторі також поділитися базовими `embeddings`. Наприклад, існує єдиний глобальне вбудовування ідентифікаторів відео з багатьма різними функціями використовувати (ідентифікатор відео показу, ідентифікатор останнього переглянутого відео користувачем, ідентифікатор відео, яке `\seeded`" рекомендація, тощо). Незважаючи на спільне вбудовування, кожна функція подається окремо у мережу, щоб вищезгадані рівні могли навчатися спеціалізовані представлення для кожної функції. Спільний доступ до вставок має велике значення для вдосконалення узагальнення, прискорення навчання і зменшення вимог до пам'яті. Переважна більшість параметрів моделі мають високу потужність простори вбудовування - наприклад, один мільйон вбудованих ідентифікаторів у 32-вимірному просторі мають у 7 разів більше параметрів ніж повністю з'єднані шари шириною 2048 одиниць.

Моделювання очікуваного часу перегляду

В даному випадку мета полягає в тому, щоб передбачити очікуваний час перегляду з огляду на навчання приклади, які є або позитивними (відео враження було натиснув) або негативний (показ не натиснув). Позитивний приклади анотовано кількістю часу користувач витратив на перегляд відео. Передбачити очікуваний годинник коли ми використовуємо техніку зваженої логістичної регресії, який був розроблений для цієї мети.

Модель навчена за допомогою логістичної регресії під крос-ентропією втрати (рис.2.6). Однак позитив (клацнув) покази зважуються за спостережуваним часом перегляду на відео. Усі негативні (ненатиснуті) покази отримують одиницю вага. Таким чином, коефіцієнти, отримані за допомогою логістичної регресії є

$$\frac{\sum T_i}{N - k}$$

де

N – кількість навчальних прикладів,

k – кількість позитивних вражень,

T_i – годинник часу i -го враження.

Вважаючи частку позитивною вражень малою (що вірно в нашому випадку), шанси приблизно дорівнюють $E[T](1+P)$, де P є імовірність клацання, а $E[T]$ – очікуваний час перегляду враження. Оскільки P мале, цей добуток близький до $E[T]$.

Для висновку використовуємо експоненціальну функцію e^x як - кінцеву функцію активації, щоб отримати ці шанси, що близько оцінити очікуваний час перегляду.

Експерименти з прихованими шарами

Таблиця 1 показує результати, які ми отримали під час очікування наступного дня дані з різними конфігураціями прихованого шару. Значення показано для кожної конфігурації («зважені, втрати на користувача») отримано шляхом розгляду як позитивного (клацання), так і негативного (ненатиснуті) покази, показані користувачеві на одній сторінці.

Спершу ми оцінюємо ці два враження за допомогою нашої моделі. Якщо негативне враження отримує вищу оцінку, ніж позитивне враження, то ми вважаємо позитивне враження час перегляду бути неправильним прогнозом часу перегляду. Тоді втрата – це загальна кількість неправильно передбаченого часу перегляду як частка загального часу перегляду над утриманим показом пари.

Ці результати показують, що збільшення ширини приховано шарів покращує результати, як і збільшення їх глибини. The trade-off, однак, це час процесора сервера, необхідний для висновку. Конфігурація 1024-широкого

ReLU з подальшим 512-широкий ReLU, за яким слідує 256-широкий ReLU, дали нам найкраще результати, дозволяючи нам залишатися в межах нашого обслуговуючого ЦП бюджет.

Для моделі 1024!512!256 ми спробували лише подати нормалізовані безперервні функції без їх повноважень, які збільшив збиток на 0,2%. З тим самим прихованим шаром гурації, ми також навчили модель, де позитивні та негативні приклади мають однакову вагу. Як це не дивно збільшив втрату годинника за часом на драматичні 4,1%.

Таблиця 2.1

Ефекти ширшого та глибшого прихованого ReLU шари на годиннику, зважена за часом попарна втрата поміщено на дані очікування наступного дня

Hidden layers	weighted, per-user loss
None	41.6%
256 ReLU	36.9%
512 ReLU	36.7%
1024 ReLU	35.8%
512 ReLU → 256 ReLU	35.2%
1024 ReLU → 512 ReLU	34.7%
1024 ReLU → 512 ReLU → 256 ReLU	34.6%

Таким чином, представлена глибока архітектура нейронної мережі для рекомендації відео YouTube, виокремили дві проблеми: формування та рейтинг кандидатів. Представлена глибока модель спільної фільтрації здатна ефективно працювати асимілювати багато сигналів і моделювати їх взаємодію з шарами глибини, перевершуючи попередню матричну факторізацію підходи, що використовуються на YouTube [23]. Існує більше мистецтва, ніж науки у виборі сурогатної проблеми для рекомендацій і виявилось, що класифікація майбутнього годинника показала хороші результати на живих показниках, фіксуючи асиметричну поведінку спільного перегляду і запобігання витоку

інформації в майбутньому. Утримання дискримінаційні сигнали від класифікатора також були важливими досягнення хороших результатів - інакше модель перевищить т сурогатну проблему і не перенести добре на головну сторінку.

Ми продемонстрували це на прикладі віку навчання як вхідна функція усуває властиве упередження до минулого і дозволяє моделі представляти залежне від часу поведінка популярних відео. Це покращило стійкість точні результати та значно збільшили час перегляду на нещодавно завантажених відео в тестуванні A/B.

Ранжування поки що є більш класичною проблемою машинного навчання і підхід глибокого навчання перевершив попередній лінійний і деревоподібні методи прогнозування часу перегляду. Рекомендації системи особливо виграють від спеціалізованих функцій опису минулої поведінки користувача з елементами. Глибока нервова мережі вимагають спеціальних уявлень категоріальних і безперервні функції, які ми перетворюємо за допомогою вбудовування і квантильна нормалізація відповідно. Шари глибини було показано, що вони ефективно моделюють нелінійні взаємодії між сотні функцій.

Логістичну регресію було модифіковано за допомогою зважених прикладів навчання з часом спостереження за позитивними прикладами та єдністю для негативні приклади, що дозволяє нам детально вивчати шанси очікуваний час перегляду моделі. Цей підхід показав багато краще за показниками оцінки зваженого рейтингу за часом перегляду порівняно з прямим прогнозуванням рейтингу кліків.

2.3 Дослідження моделей формування рекомендацій, отримання оцінок рейтингу відеоконтенту мережевої системи

Кілька онлайн-компаній, зокрема Yahoo!, Amazon.com і Кінокритик, рекомендують документи та продукти споживачам. Як правило, рекомендації

базуються на методах фільтрації вмісту та/або спільної роботи. Автори досліджують переваги цих методів, припускають, що моделі переваг, які використовуються в маркетингу, пропонують хороші альтернативи, і описують байєсівську модель переваг, яка дозволяє статистичну інтеграцію п'яти типів інформації, корисної для надання рекомендацій: виражені переваги людини, переваги інших споживачів, експертні оцінки, характеристики предметів та індивідуальні характеристики. Запропонований метод враховує не тільки неоднорідність уподобань між користувачами, але й неспостережувану неоднорідність продукту, вводячи взаємодію неспостережуваних атрибутів продукту з характеристиками споживача.

Системи рекомендацій забезпечують тип масового налаштування, який стає все більш популярним в Інтернеті. Пошукові системи, такі як Yahoo! і Alta Vista використовують їх, щоб рекомендувати відповідні документи на основі ключових слів, наданих користувачем. Los Angeles Times дозволяє налаштовувати онлайн-новини. Amazon.com і barnesandnoble.com рекомендують книги та фільми на основі вподобань інших клієнтів. Така настройка нібито зменшує пошукові зусилля для користувачів. Це також обіцяє фірмі більшу лояльність споживачів, більші продажі, більші доходи від реклами та переваги цільових рекламних акцій.

Сучасні системи налаштування діляться на два класи, які використовують різні джерела інформації для надання рекомендацій. Перший клас включає спільну фільтрацію, яка імітує рекомендації з уст в уста. Оперативно ці методи передбачають уподобання людини як лінійну, зважену комбінацію уподобань інших людей. Відомі комерційні реалізації спільної фільтрації пропонують Net Perceptions, Likeminds і Firefly, які вже не існують. Другий клас, відомий як фільтрація вмісту, дає рекомендації на основі споживчих переваг щодо атрибутів продукту. Доступні комерційні системи, запропоновані PersonalLogic, Frictionless Commerce і Active Research, використовують оцінки важливості, що

визначаються самостійно, і/або компроміси атрибутів для надання своїх рекомендацій.

Обидва типи методів фільтрації мають обмеження. Спільна фільтрація потребує щільних наборів даних; можна використовувати лише тоді, коли принаймні кілька людей оцінили продукт; не відображає невизначеність у прогнозах; і надає кілька, якщо такі є, причин для рекомендації. Системи, засновані на атрибутах, можуть рекомендувати абсолютно нові елементи, але не обов'язково включають інформацію про схожість уподобань між індивідами. Подібно до спільної фільтрації, ці методи також не можуть давати рекомендації для людей, які не надають інформації про переваги. І хоча в онлайн-ситуаціях бажано мінімізувати кількість даних, зібраних від людини, мало відомо про компроміс між точністю прогнозування та кількістю даних, зібраних від людини для цього класу системи рекомендацій. Нарешті, наскільки нам відомо.

Як зрозуміло з цього опису, методи спільної роботи та фільтрації вмісту використовують один або два типи інформації. Однак існує щонайменше п'ять джерел інформації, які можна використовувати для надання рекомендацій: (1) висловлені людиною переваги або вибір серед альтернативних продуктів, (2) переваги щодо атрибутів продукту, (3) переваги або вибір інших людей, (4) експертні оцінки та (5) індивідуальні характеристики, які можуть передбачити переваги. Хороша система рекомендацій повинна мати можливість використовувати будь-який або всі з цих п'яти типів інформації, потенційно створюючи кращі рекомендації, коли стає доступною більше інформації. Іншими словами, метод повинен мати можливість інтегрувати альтернативні джерела інформації за допомогою вкладених специфікацій, які дозволяють робити прогнози на основі підмножин інформації.

Ми використовуємо ієрархічний байєсівський підхід для розробки системи рекомендацій. Подібно до моделей, описаних Алленбі та Гінтером (1995) і Россі та Алленбі (1996), наша робота допускає неспостережувану

неоднорідність споживчих уподобань. Крім того, ми представляємо вплив неспостережуваної неоднорідності продукту на переваги, щоб дозволити введення неспостережуваних атрибутів продукту, таких як цілісні судження клієнтів і структури привабливості продукту. В онлайн-контексті врахування неоднорідності продукту має вирішальне значення, оскільки варіанти продукту/продавця, доступні споживачеві, часто змінюються на постійній основі. Тому має сенс розглядати альтернативи в певний час як випадкові вибірки з відповідного розподілу. Це контрастує з моделями покупок у супермаркетах групою домогосподарств, для яких фіксована (часто невелика) кількість брендів, що продаються однією роздрібною мережею, найкраще представлена моделлю фіксованих ефектів.

У наступному розділі ми коротко обговоримо системи рекомендацій; потім ми описуємо набір запропонованих моделей та їх оцінку за допомогою методів ланцюга Маркова Монте-Карло (МСМС). У останньому розділі ми представляємо результати тестування моделей з використанням великого загальнодоступного набору даних, який використовували інші дослідники, щоб рекомендувати споживачам онлайн-фільми.

Огляд систем рекомендацій

Хоча ця ідея походить від Негропonte (1970) і Кея (1984), практичне впровадження «розумних агентів» відносно нещодавно і підживлюється успіхами онлайн-компаній, таких як Firefly і Amazon.com. Ранні випадки використання, яких не було в Інтернеті, включали короткочасні кіоски в магазинах Blockbuster Video, які рекомендували фільми на основі минулої історії прокату учасника (West et al. 1999). Завдяки цьому стали можливими такі цікаві рекомендації, як порнографічні фільми для дітей і програми «Телепузики» для бабусь і дідусів, які проживали разом. Його вплив на сімейне благополуччя невідомий. Потім був Магніт (Леві 1993), який претендував на статус першого інтелектуального агента для Macintosh. По суті це файловий

менеджер, він відправляв файли у кошик, якщо користувач неправильно вводив папку призначення (Foner 1993).

Системи рекомендацій — це агенти типу Blockbuster. Використовуючи інформацію про поведінку або переваги, вони фільтрують альтернативи та роблять пропозиції користувачеві. Прикладом таких систем, що базуються на вмісті, є пошукові системи Інтернету, оскільки вони отримують документи за допомогою ключових слів. В одній загальноживаній системі частота цільового слова використовується для оцінки релевантності документа, а відносна частота слів використовується для оцінки схожості документа (Салтон і Баклі 1988).

Подібно до спільного аналізу, системи рекомендацій відбирають привабливі альтернативи. Але тоді як об'єднаний аналіз зазвичай використовується для перевірки багатьох продуктів, щоб знайти кілька привабливих ринкових варіантів, системи рекомендацій є моделями для прогнозування на індивідуальному рівні, які можуть бути корисними, навіть якщо є кілька альтернатив. Наприклад, людина, яка вибирає між новими випусками музики, виставами чи фільмами, може мати лише кілька варіантів. Системи рекомендацій також чудово підходять для пропонування книг, ресторанів, хімчисток, сантехніків, лікарів, юристів, фінансових установ і брокерів з нерухомості. І люди можуть шукати рекомендації не лише за досвідом чи послугами репутації такого роду. Як будь-хто, хто користувався Consumer Reports і так звані онлайн-конфігуратори продуктів знають, що рекомендації можуть бути корисними для таких продуктів, як автомобілі та комп'ютери, які вважаються пошуковими товарами, але які люди не завжди мають можливість або засоби оцінити. Мабуть, найважливішим є те, що системи рекомендацій повинні добре працювати з набагато меншою кількістю інформації, ніж зазвичай збирають у дослідженнях маркетингових досліджень, у яких або респондент отримує винагороду за участь у одноразовому дослідженні, або дані про вибір доступні з панелей клієнтів протягом відносно тривалого

часу. проміжок часу. Навпаки, фольклор для онлайн-систем рекомендацій полягає в тому, що більшість людей не бажають відповідати на занадто багато запитань, перш ніж отримати рекомендації.

Алгоритми спільної фільтрації були вперше представлені Голдбергом та його колегами (1992). Вони використовуються Los Angeles Times, London Times, CRAYON і Tango для персоналізації онлайн-газет; від Bostondine, щоб рекомендувати ресторани в Бостоні та його околицях; Sepia Video Guide, щоб створювати персоналізовані відео рекомендації; від Movie Critic, Moviefinder і Morse для рекомендації фільмів; і від barnesandnoble.com рекомендувати книги. У найпростішому випадку спільне фільтрування передбачає вподобання людини як зважену суму уподобань інших людей, у якій ваги пропорційні кореляції над загальним набором елементів, оцінених двома людьми. Зовсім недавно була представлена спільна фільтрація на основі моделі. Ми відсилаємо читача до Breese, Heckerman і Kadie (1998) для опису цих альтернативних реалізацій, які включають байєсівські мережі (Heckerman 1996) і моделі кінцевої суміші (Chien and George 1999). Рання оцінка цих методів Breese, Heckerman і Kadie (1998) не є обнадійливою щодо повідомлених прогностичних критеріїв.

Як уже зазначалося, алгоритми спільної фільтрації мають кілька обмежень. По-перше, коли дані розріджені, кореляції (ваги) базуються на кількох загальних елементах і тому є ненадійними. Бріз, Хекерман і Кеді (1998) показують, що в такій ситуації ефективність прогнозування різко погіршується. По-друге, алгоритми спільної фільтрації можна використовувати лише тоді, коли дані про переваги для елемента вже існують у базі даних. Іншими словами, ці системи не можуть обробляти запити, які стосуються нових елементів. Наприклад, більшість алгоритмів спільної фільтрації не можуть допомогти користувачеві, якому потрібно знати, чи хороший новий фільм. У таких ситуаціях база даних не містить інформації про фільм, і тому система не

може обробити такі запити. По-третє, ці методи використовують спеціальні алгоритми прогнозування, які не базуються на статистичній моделі. Отже, вони не враховують невизначеність, яка може бути менш важливою для таких покупок із низьким рівнем ризику, як фільми та компакт-диски, але може бути дуже важливою, коли ставки вищі для споживача чи компанії. По-четверте, системи спільної фільтрації явно не включають інформацію про атрибути, хоча вони завантажуються шляхом створення «віртуальних користувачів», які представляють певні смаки (наприклад, фанат віртуальних бойовиків, який має високі оцінки для всіх бойовиків). Наслідки такого непрямого обліку характеристик продукту не зрозумілі. Нарешті, оскільки методи спільної фільтрації кореляційні, вони мало пояснюють рекомендацію, яка може бути важливою для побудови довіри та підвищення лояльності клієнтів.

Системи на основі атрибутів дозволяють давати рекомендації щодо абсолютно нових предметів, але не обов'язково включають інформацію про подібність уподобань між індивідами. Подібно до спільної фільтрації, ці методи не можуть давати рекомендації людям, які не надають інформації про переваги. Комерційно доступні системи на основі атрибутів збирають велику кількість інформації від респондентів, і мало відомо про їх точність прогнозування. Крім того, системи, які використовують нейронні мережі, часто відчувають труднощі з наданням пояснень до рекомендацій. Щоб подолати недоліки існуючих систем, ми розробляємо гнучкі, але прості статистичні моделі, які описані в наступному розділі.

Ієрархічна байєсовська система рекомендацій. Розробляється сукупність статистичних методів, які ми оцінили, використовуючи рейтинги клієнтів щодо невеликих, ідіосинкратичних підмножин продуктів. Потім ми використовуємо ці моделі, щоб надавати персоналізовані рекомендації щодо затриманих елементів — у нашому випадку, нових випусків у кінотеатрах і відеопрокату (або повторного випуску) старих фільмів. Ми використовуємо регресійний

підхід і моделюємо оцінки клієнтів як функцію атрибутів продукту, характеристик споживачів і експертних оцінок. Моделі, які ми розробляємо, відрізняються тим, як вони враховують неспостережувані джерела неоднорідності в уподобаннях клієнтів і структурі привабливості продукту. У тій мірі, в якій оцінки параметрів відображають причинно-наслідкові структури переваг, вони дозволяють нам сказати не тільки про те, що людям може подобатися, але й про те, чому вони можуть реагувати передбаченим чином.

Неоднорідність клієнтів. База даних складається з рейтингів, наданих клієнтами для багатьох різних фільмів. Клієнти різняться за кількістю фільмів, які вони оцінюють, що призводить до незбалансованого набору даних. Нехай $i = \overline{1, I}$ представляють клієнтів, а $j = \overline{1, J}$ представляють фільми. Клієнт i надає оцінки для n_i фільмів у базі даних; нехай $M_i = \{j_1, j_2, \dots, j_{n_i}\}$ позначає набір індексів n_i фільмів, оцінених клієнтом i . Нехай r_{ij} представляє рейтинг, наданий клієнтом i для фільму j , де $j \in M_i$. Загальна кількість оцінок для всіх клієнтів визначається як $N = \sum_{i=1}^I n_i$. Спостереження для кожного клієнта використовуються для визначення моделі регресії на рівні клієнта:

$$r_{ij} = w_{ij}\beta_j + e_{ij}, \quad e_{ij} \sim N(0, \sigma^2), \quad (1)$$

де $j \in M_i$, \mathbf{w}_j — вектор атрибутів фільму (жанр і експертні оцінки) для фільму j , а β_j — вектор параметрів, які представляють структуру переваг для клієнта i .

Якщо база даних містить багато спостережень для кожного клієнта, ми можемо в принципі оцінити попередню модель регресії для кожного клієнта. Однак у багатьох ситуаціях база даних розріджена, і для деяких клієнтів доступні лише кілька спостережень. Тому ми не можемо виконувати окремі регресії для кожного клієнта. Однак ми можемо використовувати ієрархічний

байєсівський підхід, який адекватно об'єднує інформацію про клієнтів, щоб робити висновки, які стосуються конкретного клієнта. У цьому підході безперервний розподіл суміші використовується для опису того, як параметри індивідуального рівня в рівнянні 2 змінюються між клієнтами в сукупності. Популяційна модель, яка враховує як спостережувані, так і неспостережувані джерела неоднорідності

$$\beta_i = z_i \mu + \lambda_i, \beta_i = z_i \mu + \lambda_i, \quad (2)$$

для $i = 1$ до I . У рівнянні 2 z_i містить характеристики клієнта i , а λ_i представляє неспостережуваний ефект клієнта для i -го клієнта.

Повну модель можна альтернативно записати як

$$r_{ij} = x'_{ij} \mu + w'_j \lambda_i + e_{ij}, e_{ij} \sim N(0, \sigma^2), \lambda_i \sim N(0, \Lambda), \quad r_{ij} = x'_{ij} \mu + w'_j \lambda_i + e_{ij}, e_{ij} \sim N(0, \sigma^2), \lambda_i \sim N(0, \Lambda), \quad (3)$$

для $i = 1$ до I та $j \in M_i$. У рівнянні 3 x_{ij} — це вектор, що містить усі спостережувані атрибути фільму (тобто жанрові змінні та оцінки експертів), а також характеристики людей і їх взаємодію, а w_j — вектор, який містить спостережувані атрибути фільму. Вектор μ представляє фіксовані ефекти та документує вплив спостережуваних змінних клієнта та фільму та їх взаємодії. Вектор λ_i містить усі випадкові ефекти, що відносяться до i -го клієнта. Коваріаційна матриця Λ надає інформацію про ступінь неспостережуваної неоднорідності в структурах переваг клієнтів.

Неоднорідність продукту

У попередніх підходах до моделювання неоднорідності в маркетингу (переважно в контекстах об'єданого та дискретного вибору) використовувалися дані, які стосуються кількох продуктів, які добре описуються спостережуваними атрибутами. У такому контексті відмінності в структурах уподобань клієнтів головним чином сприяють неоднорідності даних. Навпаки, системи рекомендацій працюють на базах даних, які містять рейтинги багатьох продуктів. Крім того, як і у випадку з фільмами та музикою, продукти

не можна адекватно описати за допомогою кількох спостережуваних атрибутів. Споживчі переваги в таких категоріях формуються безліччю атрибутів, які взаємодіють складним чином, що призводить до тематичних відмінностей, які вимагають обліку цих складних, але не спостережуваних атрибутів продукту (див. Gershoff and West 1998). Ці непомічені атрибути фільму призводять до відмінностей у структурах привабливості продукту. Врахування цих відмінностей між фільмами стає вирішальним у моделюванні рейтингів клієнтів. У цьому розділі ми розробляємо модель, яка враховує неспостережувані атрибути фільму в параметрах моделювання.

Нехай $C_j = \{i_1, i_2, \dots, i_{n_j}\}$ представляє набір індексів n_j клієнтів, які оцінили фільм j . Нехай r_{ji} представляє рейтинг, наданий клієнтом i для фільму j , де $i \in C_j$. Кількість клієнтів, які надають оцінки для фільму, різна, що призводить до незбалансованого набору даних. Спостереження для фільму j можна використовувати для визначення моделі регресії на рівні фільму таким чином:

$$r_{ji} = z_i' \beta_j + e_{ij}, e_{ij} \sim N(0, \sigma^2), r_{ji} = z_i' \beta_j + e_{ij}, e_{ij} \sim N(0, \sigma^2), \quad (4)$$

для всіх $i \in C_j$. Вектор z_i містить характеристики клієнта для клієнта i , а β_j є вектором параметрів для фільму j , який представляє структуру привабливості фільмів для клієнтів. Популяційну модель, яка визначає, як фільми відрізняються своєю привабливою структурою, можна записати як

$$\beta_j = w_j' \mu + \gamma_j, \gamma_j \sim N(0, \Gamma), \beta_j = w_j' \mu + \gamma_j, \gamma_j \sim N(0, \Gamma), \quad (5)$$

для $j = 1$ до J . Вектор w_j містить спостережувані характеристики фільму, а γ_j представляє неспостережувані ефекти фільму. Повну модель можна альтернативно записати як

$$r_{ji} = x'_{ij} \mu + z'_{ij} \gamma_j + e_{ji}, e_{ji} \sim N(0, \sigma^2), \gamma_j \sim N(0, \Gamma), r_{ji} = x_{ij}' \mu + z_{ij}' \gamma_j + e_{ji}, e_{ji} \sim N(0, \sigma^2), \gamma_j \sim N(0, \Gamma), \quad (6)$$

для $j = 1$ до J і для $i \in C_j$. Вектор rx_{ji} містить усі спостережувані атрибути фільму, характеристики клієнтів і їх взаємодію. Матриця дисперсії Γ надає інформацію про ступінь неспостережуваної неоднорідності в структурах привабливості продукту.

Неоднорідність клієнтів і продуктів

Як стало очевидним з нашого попереднього обговорення, системи рекомендацій працюють у контекстах, які включають багато продуктів і клієнтів. Тому необхідно враховувати неоднорідність як клієнта, так і продукту в моделюванні переваг. Ми поєднуємо обидві форми неоднорідності. У комбінованій моделі рейтинг r_{ij} для клієнта і можна записати як

$$r_{ij} = x'_{ij}\mu + z'_i\gamma_i + w'_j\lambda_j + e_{ij}, \quad e_{ij} \sim N(0, \sigma^2), \lambda_j \sim N(0, \Lambda), \gamma_j \sim N(0, \Gamma), \quad r_{ij} = x_{ij}'\mu + z_i'\gamma_i + w_j'\lambda_j + e_{ij}, \quad e_{ij} \sim N(0, \sigma^2), \lambda_j \sim N(0, \Lambda), \gamma_j \sim N(0, \Gamma), \quad (7)$$

для $i=1$ до I та $j \in M_i$. Вектор x_{ij} містить змінні фільму та клієнта, z_i — вектор характеристик клієнта, а w_j — вектор атрибутів фільму. Випадкові ефекти λ_j враховують неспостережувані джерела неоднорідності клієнтів і з'являються в моделі в інтерактивному режимі разом із спостережуваними атрибутами фільму. Випадкові ефекти γ_j пояснюють неспостережуване джерело неоднорідності в структурах привабливості фільмів і взаємодіють із спостережуваними характеристиками споживачів. Така модель забезпечує гнучку основу для фіксації відмінностей у структурах уподобань клієнтів і структурах привабливості фільмів.

Жанрові змінні визначаються дев'ятьма бінарними індикаторами, які описують, чи належить фільм до однієї чи кількох із таких жанрових категорій: бойовик, мистецтво/іноземний, класика, комедія, драма, сімейний фільм, жахи, романтика та трилер. Фільм можна одночасно віднести до кількох із цих жанрів. Жанрові ефекти включені в модель таким чином:

$$\text{Genre}_j = \sum_k \mu_k \text{Genre}_{jk}, \quad \text{Genre}_j = \sum_k \mu_k \text{Genre}_{jk}, \quad (8)$$

де $Genre_{jk}$ відноситься до жанрової змінної k , а μ_s представляють фіксовані ефекти.

Демографічні змінні включають вік і стать клієнтів у базі даних і представлені в моделі таким чином:

$$Demographics_i = \sum_k \mu_k Demographics_{ik}, Demographics_{ik} = \sum_j \mu_{jk} Demographics_{ijk}, \quad (9)$$

де $Demographics_{jk}$ відноситься до демографічної змінної k , а μ_s представляють фіксовані ефекти.

У модель можна включити два типи взаємодії. По-перше, можна включити взаємодію демографічних показників з різними жанровими та експертними змінними, щоб охопити вплив спостережуваних джерел неоднорідності. По-друге, можна включити взаємодію між різними характеристиками фільму, щоб зафіксувати спільний вплив різних жанрових і експертних змінних.

Розділ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДЕЛЕЙ ОЦІНКИ ЕФЕКТИВНОСТІ СИСТЕМ ПОШУКУ РОБОТИ

3.1 Програмна реалізація оцінки рейтингу відеоконтенту мережевої системи

Програмна реалізація методів обчислення рейтингу виконується за допомогою наступних функцій модуля Surprise.

```
class surprise.prediction_algorithms.matrix_factorization.SVD (n_factors=100,
n_epochs=20, biased=True, init_mean=0, init_std_dev=0.1, lr_all=0.005,
reg_all=0.02, lr_bu=None, lr_bi=None, lr_pu=None, lr_qi=None, reg_bu=None,
reg_bi=None, reg_pu=None, reg_qi=None, random_state=None, verbose=False).
```

Створює модель SVD

де `n_factors` – кількість чинників. за замовчуванням 100,
`n_epochs` – кількість ітерацій процедури SGD. За замовчуванням 20,
`biased` – Якщо True, то використовується поріг. За замовчуванням True,
`init_mean` – середнє нормального розподілу для ініціалізації векторів факторів. За замовчуванням 0,

`init_std_dev` – середньоквадратичне відхилення нормального розподілу для ініціалізації векторів факторів. За замовчуванням 0.1.

`lr_all` – параметр, що визначає швидкість навчання всім параметрів . За замовчуванням 0.005.

`reg_all` – параметр регуляризації всім параметрів. За замовчуванням 0.02,
`lr_bu` – параметр, що визначає швидкість навчання для bu. Має пріоритет над аргументом `lr_all`. За замовчуванням None,

`lr_bi` – параметр, що визначає швидкість навчання для bi. Має пріоритет над аргументом `lr_all`. За замовчуванням None,

`lr_pu` – параметр, що визначає швидкість навчання для pu. Має пріоритет над аргументом `lr_all`. За замовчуванням None,

`lr_qi` – параметр, що визначає швидкість навчання для `qi`. Має пріоритет над аргументом `lr_all`. За замовчуванням `None`

`reg_bu` – параметр регуляризації для `bu`. Має пріоритет над аргументом `reg_all`. За замовчуванням `None`,

`reg_bi` – параметр регуляризації для `bi`. Має пріоритет над аргументом `reg_all`. За замовчуванням `None`,

`reg_pu` – параметр регуляризації для `pu`. Має пріоритет над аргументом `reg_all`. За замовчуванням `None`,

`reg_qi` – параметр регуляризації для `qi`. Має пріоритет над аргументом `reg_all`. За замовчуванням `None`,

`random_state` – генератор випадкових чисел. Може бути числом типу `int` (початкове число для генератора випадкових чисел) чи об'єктом класу `numpy.random.RandomState`. Може бути корисним для методу `surprise.prediction_algorithms.matrix_factorization.SVD.fit()`. За замовчуванням `None`,

`verbose` – Якщо `True`, то друкує поточну епоху. За замовчуванням `False`.

Атрибути

`surprise.prediction_algorithms.matrix_factorization.SVD.pu` – об'єкт класу `numpy.ndarray` з формою `(n_users, n_factors)`, що містить фактори користувачів, де `n_users` – кількість користувачів.

`surprise.prediction_algorithms.matrix_factorization.SVD.qi` – об'єкт класу `numpy.ndarray` з формою `(n_items, n_factors)`, що містить фактори предметів, де `n_items` – кількість предметів.

`surprise.prediction_algorithms.matrix_factorization.SVD.bu` – об'єкт класу `numpy.ndarray` з формою `(n_users,)`, що містить пороги користувача, де `n_users` – кількість користувачів.

`surprise.prediction_algorithms.matrix_factorization.SVD.bi` – об'єкт класу `numpy.ndarray` з формою `(n_items,)`, що містить пороги предмета, де `n_items` – кількість предметів.

Успадковані методи від класу `surprise.prediction_algorithms.algo_base.AlgoBase`

`fit(trainset)`

Налаштовує модель (модифікує даний об'єкт), використовуючи вказані навчальні дані. Повертає модифікований цей об'єкт

де `trainset` - навчальні дані.

`predict(uid, iid, r_ui = None, clip = True, verbose = False)`

Обчислює рейтинг для вказаного користувача та предмета. Якщо обчислення рейтингу неможливе (наприклад, відсутній користувач та/або предмет), то повертається глобальне середнє рейтингів. Повертає об'єкт класу `surprise.prediction_algorithms.predictions.Prediction` (містить ідентифікатор користувача, ідентифікатор предмета, правильний рейтинг, оцінку рейтингу)

де `uid` – ідентифікатор користувача,

`iid` – ідентифікатор предмета,

`r_ui` – правильний рейтинг типу `float`. За замовчуванням `None`,

`clip` – Якщо `True`, то кліпує рейтинг за рейтинговою шкалою. Наприклад, якщо обчислений рейтинг дорівнює `0.5` або `5.5`, а шкала рейтингів дорівнює `[1,5]`, то обчислений рейтинг встановлюється `1` або `5` відповідно. За замовчуванням `True`,

`verbose` – Якщо `True`, то друкує деталі обчислення рейтингу. За замовчуванням `False`.

`test(testset, verbose = False)`

Тестує модель, використовуючи вказані тестові дані. Повертає список об'єктів класу `surprise.prediction_algorithms.predictions.Prediction`, які містять усі обчислені рейтинги

де `testset` – тестові дані, повернуті ітератором перехресної перевірки, функцією `surprise.model_selection.split.train_test_split()` або методом `surprise.Trainset.build_testset()`,

`verbose` – Якщо `True`, то друкує деталі обчислення рейтингу. За замовчуванням `False`.

```
class surprise.prediction_algorithms.matrix_factorization.NMF (n_factors=15,
n_epochs=50, biased=False, reg_pu=0.06, reg_qi=0.06, reg_bu=0.02, reg_bi=0.02,
lr_bu=1, random_state=None, verbose=False).
```

```
predict(uid, iid, r_ui = None, clip = True, verbose = False)
```

Обчислює рейтинг для вказаного користувача та предмета. Якщо обчислення рейтингу неможливе (наприклад, відсутній користувач та/або предмет), то повертається глобальне середнє рейтингів. Повертає об'єкт класу `surprise.prediction_algorithms.predictions.Prediction` (містить ідентифікатор користувача, ідентифікатор предмета, правильний рейтинг, оцінку рейтингу)

де `uid` – ідентифікатор користувача,

`iid` – ідентифікатор предмета,

`r_ui` - правильний рейтинг типу `float`. За замовчуванням `None`

`clip` – Якщо `True`, то кліпує рейтинг за рейтинговою шкалою. Наприклад, якщо обчислений рейтинг дорівнює 0.5 або 5.5, а шкала рейтингів дорівнює [1,5], то обчислений рейтинг встановлюється 1 або 5 відповідно. За замовчуванням `True`,

`verbose` – Якщо `True`, то друкує деталі обчислення рейтингу. За замовчуванням `False`.

```
test(testset, verbose = False)
```

Тестує модель, використовуючи вказані тестові дані. Повертає список об'єктів класу `surprise.prediction_algorithms.predictions.Prediction`, які містять усі обчислені рейтинги

де `testset` – тестові дані, повернуті ітератором перехресної перевірки, функцією `surprise.model_selection.split.train_test_split()` або методом `surprise.Trainset.build_testset()`,

`verbose` – Якщо `True`, то друкує деталі обчислення рейтингу. За замовчуванням `False`.

Зауваження. Існує також клас `surprise.prediction_algorithms.matrix_factorization.SVDpp()`, який реалізує метод SVD++

```
class surprise.prediction_algorithms.knns.KNNBasic(k=40, min_k=1,
sim_options={}, verbose=True, **kwargs)
```

Створює модель k найближчих сусідів (KNN)

де k – максимальна кількість сусідів. За замовчуванням 40,

`min_k` - мінімальна кількість сусідів. Якщо сусідів недостатньо, то обчислений рейтинг встановлюється у глобальне середнє рейтингів. За замовчуванням 1,

`sim_options` – словник міри подоби. За замовчуванням {},

`verbose` – Якщо `True`, то друкує трасуючи повідомлення оцінки порогів, подібності та інших. За замовчуванням `True`.

Наслідувані методи від класу `surprise.prediction_algorithms.algo_base.AlgoBase`

```
fit(trainset)
```

Налаштовує модель (модифікує даний об'єкт), використовуючи вказані навчальні дані. Повертає модифікований цей об'єкт

де `trainset` – навчальні дані.

```
predict(uid, iid, r_ui = None, clip = True, verbose = False)
```

Обчислює рейтинг для вказаного користувача та предмета. Якщо обчислення рейтингу неможливе (наприклад, відсутній користувач та/або предмет), то повертається глобальне середнє рейтингів. Повертає об'єкт класу

`surprise.prediction_algorithms.predictions.Prediction` (містить ідентифікатор користувача, ідентифікатор предмета, правильний рейтинг, оцінку рейтингу)

де `uid` – ідентифікатор користувача,

`iid` – ідентифікатор предмета,

`r_ui` - правильний рейтинг типу `float`. За замовчуванням `None`,

`clip` – Якщо `True`, то кліпує рейтинг за рейтинговою шкалою. Наприклад, якщо обчислений рейтинг дорівнює 0.5 або 5.5, а шкала рейтингів дорівнює [1,5], то обчислений рейтинг встановлюється 1 або 5 відповідно. За замовчуванням `True`,

`verbose` – Якщо `True`, то друкує деталі обчислення рейтингу. За замовчуванням `False`.

`test(testset, verbose = False)`

`class surprise.prediction_algorithms.knns.KNNWithMeans(k=40, min_k=1, sim_options={}, verbose=True, **kwargs)`

Створює модель `k` найближчих сусідів (KNN) із додаванням середніх рейтингів

де `k` – максимальна кількість сусідів. За замовчуванням 40,

`min_k` - мінімальна кількість сусідів. Якщо сусідів недостатньо, то обчислений рейтинг встановлюється у глобальне середнє рейтингів. За замовчуванням 1,

`sim_options` – словник міри подібності. За замовчуванням {},

`verbose` – якщо `True`, то друкує трасу повідомлення оцінки порогів, подібності та інших. За умовчанням `True`.

Успадковані методи від класу

`surprise.prediction_algorithms.algo_base.AlgoBase`

`fit(trainset)`

Налаштовує модель (модифікує даний об'єкт), використовуючи вказані навчальні дані. Повертає модифікований цей об'єкт

де `trainset` – навчальні дані.

```
predict(uid, iid, r_ui = None, clip = True, verbose = False)
```

Обчислює рейтинг для вказаного користувача та предмета. Якщо обчислення рейтингу неможливе (наприклад, відсутній користувач та/або предмет), то повертається глобальне середнє рейтингів. Повертає об'єкт класу `surprise.prediction_algorithms.predictions.Prediction` (містить ідентифікатор користувача, ідентифікатор предмета, правильний рейтинг, оцінку рейтингу)

де `uid` – ідентифікатор користувача,

`iid` – ідентифікатор предмета,

`r_ui` – правильний рейтинг типу `float`. За замовчуванням `None`,

`clip` – Якщо `True`, то кліпує рейтинг за рейтинговою шкалою. Наприклад, якщо обчислений рейтинг дорівнює 0.5 або 5.5, а шкала рейтингів дорівнює [1,5], то обчислений рейтинг встановлюється 1 або 5 відповідно. За замовчуванням `True`,

`verbose` – Якщо `True`, то друкує деталі обчислення рейтингу. За замовчуванням `False`.

```
test(testset, verbose = False)
```

Тестує модель, використовуючи вказані тестові дані. Повертає список об'єктів класу `surprise.prediction_algorithms.predictions.Prediction`, які містять усі обчислені рейтинги

де `testset` – тестові дані, повернуті ітератором перехресної перевірки, функцією `surprise.model_selection.split.train_test_split()` або методом `surprise.Trainset.build_testset()`,

`verbose` – Якщо `True`, то друкує деталі обчислення рейтингу. За замовчуванням `False`.

```
compute_similarities()
```

Створює матрицю подібності

```
get_neighbors(iid, k)
```


Повертає список k найближчих сусідів користувача чи предмета.

де iid – ідентифікатор користувача чи предмета,

k – кількість найближчих сусідів.

3.2 Експериментальна побудова оцінок рейтингу відеоконтенту мережевої системи

Розглянемо реалізацію оцінки рейтингу на основі двох методів: колаборативної фільтрації на базі сусідства та на основі моделі латентних факторів з урахуванням SVD та NMF. Оцінка рейтингу фільму на базі колаборативної фільтрації на основі сусідства. Обчислимо відсутню оцінку рейтингу на основі коефіцієнта кореляції Пірсона та скоригованої косинусної подоби на прикладі 5-бальної рейтингової матриці розмірності $m \times n$, де $m=n=5$.

	відео 1	відео 2	відео 3	відео 4	відео 5
користувач 1	2	4	2	2	r_{15}
користувач 2	1	3	3	5	1
користувач 3	4	5	2	3	3
користувач 4	1	1	5	2	1
користувач 5	1	3	3	2	1

Програмна реалізація

```
!pip install scikit-surprise
```

```
# !conda install -y -c conda-forge scikit-surprise # If you use conda on a non-Colab environment
```

```
from surprise import SVD
```

```
from surprise import Dataset
```

```
from surprise.model_selection import cross_validate
```

```
from surprise import Reader
```

```
from surprise import model_selection
```

```
from surprise import prediction_algorithms
```

```

from surprise import accuracy

import numpy, pandas

number_variant = 5 # номер варіанту 5

# Створення таблиці

df = pandas.DataFrame(data=[["user1", "item1", 1], ["user1", "item2", 5], ["user1", "item3", 3], ["user1", "item4", 1], ["user1", "item5", 5],
                            ["user2", "item1", 2], ["user2", "item2", 4], ["user2", "item3", 2], ["user2", "item4", 4], ["user2", "item5", 2],
                            ["user3", "item1", 3], ["user3", "item2", 6], ["user3", "item3", 3], ["user3", "item4", 2], ["user3", "item5", 4],
                            ["user4", "item1", 5], ["user4", "item2", 2], ["user4", "item3", 5], ["user4", "item4", 2], ["user4", "item5", 5],
                            ["user5", "item1", 2], ["user5", "item2", 5], ["user5", "item3", 3], ["user5", "item4", 2], ["user5", "item5", 2]],
                        columns=["user_id", "item_id", "rating"])

# Створення читача таблиці

reader = Reader(name=None, line_format="user item rating", sep=None, rating_scale=(1, 5), skip_lines=0)

# Створення навчального та тестового датасетів.

Dataset = Dataset.load_from_df(df[["user_id", "item_id", "rating"]], reader)

train_Dataset, valid_Dataset = model_selection.split_train_test_split(data=Dataset, test_size=0.2,
                                                                    train_size=0.8, random_state=None,
                                                                    shuffle=True)

print("-----")

# Обчислення оцінки рейтингу на основі скоригованої косинусної подібності

model = prediction_algorithms.knns.KNNBasic(k=6, min_k=6, sim_options={"name": "cosine", "user_based": False,
                                                                    "min_support": 1}, verbose=False)

model.fit(trainset=train_Dataset)

test_pred = model.test(testset=valid_Dataset, verbose=False)

```

```

print(test_pred[number_variant - 1])
# Обчислення RMSE
accuracy.rmse(predictions=test_pred, verbose=True)
print("-----")
# Обчислення оцінки рейтингу на основі коефіцієнта кореляції Пірсона
model = prediction_algorithms.knns.KNNWithMeans(k=6, min_k=6,
                                                sim_options={"name": "pearson", "user_based": T
rue,
                                                "min_support": 6}, verbose=False)
model.fit(trainset=train_Dataset)
test_pred = model.test(testset=valid_Dataset, verbose=False)
print(test_pred[number_variant - 1])
# Обчислення RMSE
accuracy.rmse(predictions=test_pred, verbose=True)
print("-----")

```

Був отриманий наступний результат

```

user: user1    item: item5    r_ui = 4.00    est = 3.36    {'actual_k': 3, 'was_impossible':
False}

```

RMSE: 0.8774

```

user: user1    item: item5    r_ui = 4.00    est = 3.48    {'actual_k': 2, 'was_impossible':
False}

```

RMSE: 0.4958

Метод колаборативної фільтрації на основі моделі латентних факторів. Обчислимо відсутню оцінку рейтингу з урахуванням SVD та NMF. Задано 5-бальну рейтингову матрицю] [uirR=розмірності $m \times n$, де $m=n=5$, у вигляді наступної таблиці.

Результат:

```

-----
user: user5      item: item1      r_ui = 2.00    est = 1.00    {'was_impossible': False}
RMSE: 2.0976
-----
user: user5      item: item1      r_ui = 2.00    est = 1.96    {'was_impossible': False}
RMSE: 1.7274
-----

```

Програмний код

```

!pip install scikit-surprise
# !conda install -y -c conda-forge scikit-surprise # If you use conda on a non-
Colab environment
from surprise import SVD
from surprise import Dataset
from surprise.model_selection import cross_validate
from surprise import Reader
from surprise import model_selection
from surprise import prediction_algorithms
from surprise import accuracy
import numpy, pandas
number_variant=3
df = pandas.DataFrame(data=[["user1", "item1", 1], ["user1", "item2", 5], ["user1", "it
em3", 3], ["user1", "item4", 1], ["user1", "item5", 5],
                        ["user2", "item1", 2], ["user2", "item2", 4], ["user2", "item3", 2], ["u
ser2", "item4", 4], ["user2", "item5", 2],
                        ["user3", "item1", 3], ["user3", "item2", 6], ["user3", "item3", 3], ["u
ser3", "item4", 2], ["user3", "item5", 4],
                        ["user4", "item1", 5], ["user4", "item2", 2], ["user4", "item3", 5], ["u
ser4", "item4", 2], ["user4", "item5", 5],
                        ["user5", "item1", 2], ["user5", "item2", 5], ["user5", "item3", 3], ["u
ser5", "item4", 2], ["user5", "item5", 2]],
                    columns=["user_id", "item_id", "rating"])
# ["user1", "item5", 4] => ["user1", "item5", 0] для number_variant=5df.iloc[number
_variant-1][2]=0

```

```

reader = Reader(name=None, line_format="user item rating", sep=None, rating_
scale=(1, 5), skip_lines=0)

#Створення навчального та тестового датасетів.

Dataset = Dataset.load_from_df(df[["user_id", "item_id", "rating"]], reader)

train_Dataset, valid_Dataset = model_selection.split.train_test_split(data=Dataset, test
_size=0.2, train_size=0.8, random_state=None, shuffle=True)

print("-----")

# Обчислення оцінки рейтингу на основі SVD

model = prediction_algorithms.matrix_factorization.SVD( n_factors=3, n_epochs=2
0, biased=False, init_mean=0, init_std_dev=0.1, lr_all=0.005, random_state=0,verbo
se=False)

model.fit(trainset=train_Dataset)

test_pred = model.test(testset=valid_Dataset, verbose=False)

print(test_pred[number_variant-1])

# Обчислення RMSE

accuracy.rmse(predictions=test_pred, verbose=True)

print("-----")

# Обчислення оцінки рейтингу на основі NMF

model = prediction_algorithms.matrix_factorization.NMF(n_factors=3, n_epochs=20,
biased=False, random_state=0, init_low=0, init_high=1, verbose=False)

model.fit(trainset=train_Dataset)

test_pred = model.test(testset=valid_Dataset, verbose=False)

print(test_pred[number_variant-1])

# Обчислення RMSE

accuracy.rmse(predictions=test_pred, verbose=True)

print("-----")

```

Був отриманий наступний результат

```

user: user1 item: item5 r_ui = 4.00 est = 1.02 {'was_impossible': False}
RMSE: 1.8891

```

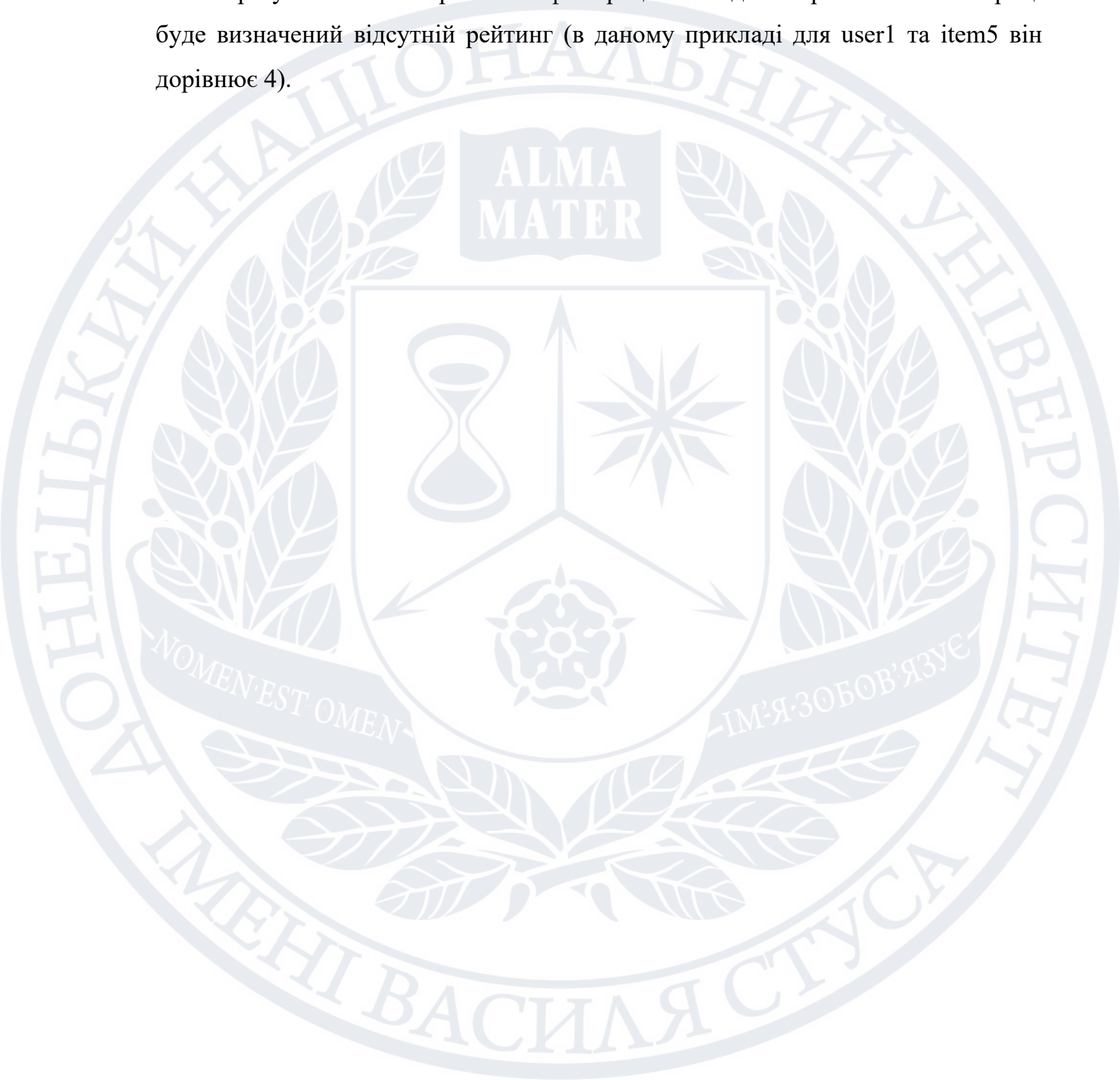
```

-----
user: user1 item: item5 r_ui = 4.00 est = 5.00 {'was_impossible': False}

```

RMSE: 1.1468

В результаті колаборативної фільтрації на підставі рейтингової матриці буде визначений відсутній рейтинг (в даному прикладі для user1 та item5 він дорівнює 4).



ВИСНОВКИ

В сучасному інтернет-просторі значний сегмент ринку займають рекомендаційні системи. Вони супроводжують широкий набір інтернет-послуг. Від покупок до потокової передачі даних та пошукових систем. Основне призначення рекомендаційних систем полягає в допомозі користувачам отримати більш персоналізований досвід від програмних продуктів. Як наслідок це підвищує їх рівень задоволеності, робить лояльними до продукту, що дозволяє розробникам, ІТ-компаніям утримувати та збільшувати аудиторію продукту, зокрема постійних підписників.

В роботі вирішені такі завдання:

- проаналізовані сучасні інформаційні та рекомендаційні системи відеоконтенту;
- визначені набори даних систем автоматизованої мережевої системи збору і рекомендацій відеоконтенту;
- визначені методи обробки та аналізу даних автоматизованих мережевих систем збору і рекомендацій відеоконтенту;
- дослідженні моделі оцінок рейтингу для формування рекомендацій відеоконтенту;
- побудовані оцінки рейтингу для формування рекомендацій відеоконтенту.

Таким, чином, в роботі здійснено дослідження інтелектуальних методів і моделей обробки даних автоматизованої мережевої системи збору і рекомендацій відеоконтенту в залежності від їх призначення, набору показників, що накопичуються в даних системах, обсягів даних. Виконана програмна реалізація експериментальної побудови оцінок рейтингу відеоконтенту мережевої системи. Зроблено висновок, що існуючі методи можна ефективно використовувати для формування рекомендацій відеоконтенту. Подальший розвиток інтелектуальних методів і моделей передбачає підвищення їх точності, зменшення обчислювальної складності і розширення функціональних можливостей.

Джерела

1. Linden G., Smith B., York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering, IEEE Internet Computing, Jan/Feb 2003, Volume: 7, Issue: 1, pp. 76-80 (Рекомендації Amazon.com: спільна фільтрація від елемента до елемента)
2. Brusilovsky P., Nejd W. (2004). Adaptive Hypermedia and Adaptive Web, 2004 CSC Press LLC.
3. Brusilovsky P. From Adaptive Hypermedia to the Adaptive Web, G.Szwillus, J.Ziegler (Hrsg): Mensch & Computer 2003: Interaktion in Bewegung, Stuttgart: B. G. Teubner, 2003, S. 21-24.
4. Jin and Zaiane. Using Distinctive Information Channels for a Mission-based Web Recommender System, In WebKDD 2004. (Використання відмінних інформаційних каналів для системи веб-рекомендації на основі місії)
5. Nasraoui O. and Pavuluri M. (2004). Complete this Puzzle: A Connectionist Approach to Accurate Web Recommendations based on a Committee of Predictors, In Proceedings of WebKDD-2004. workshop on Web Mining and Web Usage Analysis, Seattle, WA, 2004. (Підхід до точних веб-рекомендацій на основі комітету прогнозів,)
6. Nasraoui O., Petenes C. Combining Web Usage Mining and Fuzzy Inference for Website Personalization, In Proceedings of WebKDD 2003–KDD Workshop on Web mining as a Premise to Effective and Intelligent Web Applications, Washington DC, August 2003, p. 37. (Поєднання аналізу використання веб-сайту та нечіткого висновку для персоналізації веб-сайту,)
7. Anjulan A, Canagarajah N. A novel video mining system. In: Proceedings of 14th IEEE 2007international conference on image processing, San Antonio, Texas, 2007. pp 185–189. (Нова система відеомайнінгу)

8. Aradhye H, Toderici G, Yagnik J () Video2Text: learning to annotate video content. In: Proceedings of IEEE international conference on data mining workshops, 2009, pp 144–152. (Video2Text: навчитися коментувати відеовміст.)
9. Bhatt CA, Kankanhalli MS () Multimedia data mining: state of the art and challenges. *Multimedia Tools*. 2011, *Appl* 51:35–76. (Інтелектуальний аналіз мультимедійних даних: сучасний стан і виклики.)
10. Brezeale D, Cook DJ. Automatic video classification: a survey of the literature. *IEEE Trans Syst Man Cybern Part C*: 2008. *Appl Rev* 38(3):416–430 (Автоматична класифікація відео: огляд літератури.)
11. Burl MC. Mining Patterns of activity from video data. In: Proceedings of the SIAM international conference on discrete mathematics, 2004, pp 532–536 (Видобуток шаблонів діяльності з відеоданих.)
12. Cui P, Liu Z-Q, Sun L-F, Yang S-Q. Hierarchical visual event pattern mining and its applications. *J Data Mining Knowl Disc*. 2011 22(3):467–492. (Інтелектуальний аналіз ієрархічного шаблону візуальних подій та його застосування.)
13. Chen B-W, Wang J-C, Wang F. A novel video summarization based on mining the story-structure and semantic relations among concept entities. *IEEE Trans Multimedia* 2009, 11(2):295–313. (Нове відеореzumвання, засноване на дослідженні структури сюжету та семантичних зв'язків між сутностями концепту.)
14. Chen F, Cooper M, Addock. Video summarization preserving synamic content. In: Proceedings of the international workshop on TRECVID video summarization, 2007, pp 40–44. (Реzumвання відео зі збереженням синамічного вмісту)
15. Dai K, Zhang J, Li G () Video mining: concepts, approaches and applications. *Proc IEEE* 2006:477–481

16. Dai KX, Li GH, Gan YL. A probabilistic model for surveillance video mining. In: Proceedings of the fifth international conference on machine learning and, cybernetics, 2006. pp 1144– 1148
17. Fan J, Luo H, Elmagarmid AK () Concept-oriented indexing of video databases: towards semantic sensitive retrieval and browsing. IEEE Trans Image Process 2004, 13(7):974–992 (Концепційно-орієнтоване індексування відеобаз даних: до семантично чутливого пошуку та перегляду.)
18. Fan J, Zhu X, Hacid M-S, Elmagarmid AK (2002) Multimedia tools and applications. Model-based video classification toward hierarchical representation indexing and access. Kluwer, Dordrecht, pp 97–120 (Мультимедійні засоби та програми. Класифікація відео на основі моделі для індексації та доступу до ієрархічного представлення.)
19. Kiran Sree P. Video data mining framework for information retrieval. In: Proceedings of NCKM-2008, Annamalai University, Tamilnadu, India (Фреймворк інтелектуального аналізу відеоданих для пошуку інформації)
20. Vassiliadis B, Stefani A, Drossos L, Ioannou K. Knowledge discovery in multimedia repositories: the role of metadata. In: Proceedings of 7th WSEAS international conference on mathematical methods and computational techniques in electrical engineering, 2005. pp 330–335
21. Wang F, Lu W, Liu J, Shah M, Xu D. Automatic video annotation with adaptive number of key words. In: Proceeding of 19th international conference on pattern recognition (ICPR 2008), pp. 1–4.
22. Wang Y, Xing C, Zhou L. Video semantic models: survey and evaluation. Int J Comput Sci Network Sec 2006, 6(2A):10–21
26. X. Amatriain. Building industrial-scale real-world recommender systems. In Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12, pages 7–8, New York, NY, USA, 2012. ACM.

27. J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath. The Youtube video recommendation system. In Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10, pages 293–296, New York, NY, USA, 2010. ACM.
29. A. M. Elkahky, Y. Song, and X. He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In Proceedings of the 24th International Conference on World Wide Web, WWW'15, pages 278–288, New York, NY, USA, 2015. ACM.
30. X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In G. J. Gordon and D. B. Dunson, editors, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11), volume 15, pages 315–323. Journal of Machine Learning Research – Workshop and Conference Proceedings, 2011.
31. X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, and J. Q. n. Candela. Practical lessons from predicting clicks on ads at facebook. In Proceedings of the Eighth International Workshop on Data Mining for Online Advertising, ADKDD'14, pages 5:1–5:9, New York, NY, USA, 2014. ACM.
32. W. Huang, Z. Wu, L. Chen, P. Mitra, and C. L. Giles. A neural probabilistic model for context based citation recommendation. In AAAI, pages 2404–2410, 2015.
- 33 S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR, abs/1502.03167, 2015.
34. S. Jean, K. Cho, R. Memisevic, and Y. Bengio. On using very large target vocabulary for neural machine translation. CoRR, abs/1412.2007, 2014.
35. L. Jiang, Y. Miao, Y. Yang, Z. Lan, and A. G. Hauptmann. Viral video style: A closer look at viral videos on youtube. In Proceedings of International Conference on Multimedia Retrieval, ICMR '14, pages 193:193–193:200, New York, NY, USA, 2014. ACM.

36. E. Meyerson. Youtube now: Why we focus on watch time. <http://youtubecreator.blogspot.com/2012/08/youtube-now-why-we-focus-on-watch-time.html>. Accessed: 2016-04-20.
37. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. CoRR, abs/1310.4546, 2013.
38. F. Morin and Y. Bengio. Hierarchical probabilistic neural network language model. In AISTATS aA Z05, pages 246–252, 2005.
39. D. Oard and J. Kim. Implicit feedback for recommender systems. In Proceedings of the AAAI Workshop on Recommender Systems, pages 81–83, 1998.
40. K.J.Oh, W. J. Lee, C. G. Lim, and H. J. Choi. Personalized news recommendation using classified keywords to capture user preference. In 16th International Conference on Advanced Communication Technology, pages 1283–1287, Feb 2014.
41. S. Sedhain, A. K. Menon, S. Sanner, and L. Xie. Autorec: Autoencoders meet collaborative filtering. In Proceedings of the 24th International Conference on World Wide Web, WWW’15 Companion, pages 111–112, New York, NY, USA, 2015. ACM.
42. X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.
43. D. Tang, B. Qin, T. Liu, and Y. Yang. User modeling with neural network for review rating prediction. In Proc. IJCAI, pages 1340–1346, 2015.
44. A. van den Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2643–2651. Curran Associates, Inc., 2013.
45. H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In Proceedings of the 21th ACM SIGKDD International

Conference on Knowledge Discovery and Data Mining, KDD '15, pages 1235–1244, New York, NY, USA, 2015. ACM.

46. J. Weston, S. Bengio, and N. Usunier. Wsabie: Scaling up to large vocabulary image annotation. In Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI, 2011.

47. J. Weston, A. Makadia, and H. Yee. Label partitioning for sublinear ranking. In S. Dasgupta and D. Mcallester, editors, Proceedings of the 30th International Conference on Machine Learning (ICML-13), volume 28, pages 181–189. JMLR Workshop and Conference Proceedings, May 2013.

48. X. Yi, L. Hong, E. Zhong, N. N. Liu, and S. Rajan. Beyond clicks: Dwell time for personalization. In Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14, pages 113–120, New York, NY, USA, 2014. ACM.

49. Zayn. Pillowtalk. <https://www.youtube.com/watch?v=C3d6GntKbk>. Mo Y. et al. Cloud-based mobile multimedia recommendation system with user behavior information //IEEE Systems Journal. – 2014. – Т. 8. – №. 1. – С. 184-193.

50. Gao J., Zhang T., Xu C. A unified personalized video recommendation via dynamic recurrent neural networks //Proceedings of the 25th ACM international conference on Multimedia. – 2017. – С. 127-135.

51. Davidson J. et al. The YouTube video recommendation system //Proceedings of the fourth ACM conference on Recommender systems. – 2010. – С. 293-296.

52. Cui L. et al. A video recommendation algorithm based on the combination of video content and social network //Concurrency and Computation: Practice and Experience. – 2017. – Т. 29. – №. 14. – С. 390-410.