МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ ТА МОДЕЛЮВАННЯ СКЛАДНИХ СИСТЕМ

Допущено до захисту
Завідувач кафедри ПМ та МСС
_____ проф. Лисенко О.В.
(підпис)
«__» _____ 2019 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня «магістр»
спеціальність 113 «Прикладна математика»
освітньо-професійна програма «Прикладна математика»

тема роботи «**Побудова математичної моделі подачі контекстної інформації**»

Виконавець
студент факультету ЕлІТ
Шурига Віктор Валентинович _____
(прізвище, ім'я, по батькові) (підпис)
Науковий керівник
к.ф.-м.н.
(науковий ступінь, вчене звання)
Сушко Тетяна Сергіївна _____
(прізвище, ім'я, по батькові) (підпис)

Суми – 2019

РЕФЕРАТ

Кваліфікаційна робота: 40с., 5 рисунків, 14 джерел, 1 додаток.

Мета і завдання дослідження: Метою кваліфікаційної роботи є дослідження в сфері контекстної інформації, аналіз існуючих рішень та розробка і удосконалення швидкісного алгоритму подання рекомендацій.

Мета кваліфікаційної роботи визначає необхідність розв'язання таких завдань:

- Дослідження сфери подання контекстної інформації та розробки рекомендаційних систем;
- Аналіз існуючих рішень та методів для окреслення проблематики під час розробки рекомендаційних систем;
- Вирішення проблеми холодного старту для алгоритму рекомендаційної системи;
- Мінімізація можливості зовнішнього впливу на систему для зміни її пріоритетів ;
- Оптимізація алгоритму для швидкої роботи з великими даними на пристроях з обмеженою обчислювальною потужністю

Об'єкт дослідження: процес подання контекстної інформації для користувачів.

Предмет дослідження: модель, алгоритми та методи рекомендаційних систем.

Ключові слова: РЕКОМЕНДАЦІЙНА СИСТЕМА, КОНТЕКСТНА ІНФОРМАЦІЯ, РЕКЛАМА, ВЕЛИКІ ДАНІ, АНАЛІЗ.

.

ЗМІСТ

ВСТУП

Рекомендаційні системи - це програмні засоби та методи, що надають пропозиції щодо корисних для користувача товарів. [1,2,3]. Пропозиції можуть стосуватися широкого спектру процесів, таких як, які товари купити, яку музику слухати або які новини читати в Інтернеті.

"Товар" - загальний термін, що використовується для позначення того, що система рекомендує користувачам. Рекомендаційна система зазвичай фокусується на конкретному типі продукту (наприклад, фільмах або новинах), і на основі його характеристик, генерує рекомендації, що мають надавати корисні та ефективні пропозиції щодо конкретного типу товару.

Загалом рекомендаційні системи спрямовані на осіб, яким не вистачає особистого досвіду чи компетенції для оцінки широкого вибору, який, може запропонувати веб-сайт.

Як приклад - система рекомендування книг, яка допомагає користувачам обрати книгу для читання. На популярному веб-сайті Amazon.com використовується рекомендаційна система персоналізації інтернет-магазину для кожного клієнта [4]. Оскільки рекомендації зазвичай персоналізовані, різні користувачі або групи користувачів отримують різноманітні пропозиції. Крім того, існують також неперсоналізовані рекомендації. Їх генерувати набагато простіше і зазвичай вони розміщуються у журналах чи газетах. Типові приклади включають в себе десятку найкращих фільмів, книг, компакт-дисків, тощо. Хоча вони й можуть бути корисними та ефективними в певних ситуаціях, однак ці дослідження неперсоналізованих рекомендацій зазвичай не розглядаються в ході дослідження рекомендаційних систем.

У їх найпростішій формі персоналізовані рекомендації пропонуються як ранжовані списки предметів. Створюючи цей рейтинг, рекомендаційні системи намагаються передбачити, які саме продукти чи послуги є найбільш влучними, виходячи з вподобань та обмежень користувача. Для виконання такого обчислювального завдання рекомендаційні системи аналізують вподобання

користувачів, які або явно виражені, наприклад, як рейтинги для товарів, або є висновком з дій користувача. Наприклад, рекомендаційна система може вважати навігацію до певної сторінки товару за прихований знак переваги серед елементів, показаних на сторінці.

Розвиток рекомендаційних систем розпочався з досить простого спостереження: люди часто покладаються на рекомендації інших під час прийняття звичайних, щоденних рішень [1, 5]. Наприклад, звичною практикою є покладатися на те, що рекомендують однолітки під час вибору книги для читання; роботодавці враховують рекомендаційні листи, приймаючи рішення про прийняття на роботу; а під час вибору фільму для перегляду люди схильні читати та покладатися на огляди фільмів, написані критиком.

Намагаючись наслідувати таку поведінку, перші рекомендаційні системи застосовували алгоритми щоб використовувати рекомендації створені спільнотою користувачів для надання рекомендацій активному користувачеві, тобто тому, хто шукає пропозиції. Рекомендації стосувалися товарів, які подобалися схожим користувачам (тим, хто має схожі смаки). Цей підхід називається спільною фільтрацією, і його обґрунтування полягає в тому, що якщо активний користувач раніше погоджувався з деякими користувачами, то інші рекомендації, що надходять від цих подібних користувачів, повинні бути доречними і цікавими для активного користувача.

По мірі того, як веб-сайти електронної комерції почали розвиватися, виникла нагальна потреба у наданні рекомендацій, отриманих від фільтрації всього спектру доступних альтернатив. Користувачам було дуже важко зробити найбільш слушний вибір серед величезної кількості товарів (речей і послуг), які пропонували веб-сайти.

Значне збільшення кількості наявної в Інтернеті інформації, її різноманітність та швидке впровадження нових послуг електронного бізнесу (купівля товарів, порівняння товарів, аукціони, тощо) часто дезорієнтують користувачів, що призводить до прийняття поганих рішень. Наявність вибору, проте відсутність користі від цього, почали знижувати добробут користувачів.

Стало зрозуміло, що широкий вибір – це не завжди добре [6].

В останні роки рекомендаційні системи стали цінним засобом вирішення проблеми перевантаження інформацією. Загалом, рекомендаційна система вирішує її , вказуючи користувачу на нові, ще не відомі продукти, які можуть відповідати поточним запитам користувачів. На запит користувача, який може бути сформульований, залежно від рекомендаційного підходу, контексту та потреби користувача, рекомендаційні системи генерують рекомендації використовуючи різні типи знань та даних про користувачів, наявні товари та попередні транзакції, що зберігаються у спеціалізованих базах даних. Потім користувач може переглядати рекомендації. Користувач може прийняти їх або ні, одразу ж або на наступному етапі надати прихований чи явий зворотній зв'язок. Усі ці дії та відгуки користувачів можуть зберігатися в базі даних рекомендацій і використовуватися для генерування нових рекомендацій у наступних взаємодіях користувач-система.

Як зазначалося вище, дослідження систем рекомендацій є відносно новим порівняно з дослідженнями інших класичних засобів та методів інформаційної системи (наприклад, баз даних чи пошукових систем). Системи рекомендацій виникли як незалежна дослідницька область в середині 1990-х [7, 1, 5]. В останні роки інтерес до систем рекомендацій різко зріс, про що свідчать наступні факти:

1. Системи рекомендацій відіграють важливу роль на таких рейтингових веб-сайтах, як Amazon.com, YouTube, Netflix, Yahoo, Tripadvisor, Last.fm та IMDb. Більше того, багато медіакомпаній зараз розробляють та впроваджують рекомендаційні системи як частину послуг, які вони надають своїм абонентам. Наприклад, Netflix, онлайн портал фільмів, присудив премію на мільйон доларів команді, яка вперше значно покращила ефективність їх системи рекомендацій [8].

2. Існують спеціальні конференції та семінари, що стосуються цієї галузі. Конференція ACM Recommender Systems (RecSys), створена в 2007 році, дотепер є найважливішою щорічною подією в галузі досліджень та застосувань

технології рекомендацій. Крім того, сесії, присвячені рекомендаційним системам, часто включаються в більш традиційні конференції в галузі баз даних, інформаційних систем та адаптивних систем. Серед цих конференцій варто відзначити ACM SIGIR Special Interest Group on Information Retrieval (SIGIR), User Modeling, Adaptation and Personalization (UMAP), та ACM's Special Interest Group on Management Of Data (SIGMOD).

3. У вищих навчальних закладах у всьому світі магістратура та аспірантура зараз повністю присвячені рекомендаційним системам; навчальні посібники з галузі рекомендаційних систем користуються популярністю на конференціях з інформатики; нещодавно вийшла книга, що описує методики рекомендаційних систем [9].

4. У наукових журналах було декілька спеціальних випусків, що висвітлювали дослідження та розробки в галузі рекомендаційних систем. Серед журналів, які присвятили випуски рекомендаційним системам, є: AI Communications (2008); IEEE Intelligent Systems (2007); International Journal of Electronic Commerce (2006); International Journal of Computer Science and Applications (2006); ACM Transactions on Computer-Human Interaction (2005); and ACM Transactions on Information Systems (2004).

РОЗДІЛ 1 ФУНКЦІЇ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

Визначимо коло можливих функцій, які може виконувати рекомендаційна система. Перш за все, ми повинні відрізняти функції, які виконує система від імені постачальника послуг та від ролі користувача. Наприклад, систему рекомендування подорожей зазвичай запроваджує замовник щоб продати більше номерів у готелях або збільшити кількість туристів до пункту призначення. А основною мотивацією користувача для доступу до системи є пошук відповідного готелю та цікаві події / атракціони під час відвідування пункту призначення. Насправді є різні причини, чому постачальники послуг можуть захотіти використовувати системи рекомендацій:

- **Збільшити кількість проданих товарів**. Це, мабуть, найважливіша функція для комерційної системи рекомендацій. Можливість продати додатковий набір товарів схожих до проданого або того на який закцентовано увагу в даний момент. Ця мета досягається тому що рекомендовані товари, ймовірно, відповідають потребам та побажанням споживача. Некомерційні програми мають подібні цілі, навіть якщо для користувача немає витрат, пов'язаних із покупкою товару або підписки на сервіс. Наприклад, тематична мережа має на меті збільшити кількість новин, прочитаних на її сайті. Загалом, основна мета введення рекомендаційної системи - збільшити коефіцієнт конверсії, тобто кількість споживачів, які приймають рекомендації та споживають товар, порівняно з кількістю простих відвідувачів, які просто переглядають інформацію.
- **Продати більш різноманітні речі**. Ще одна основна функція рекомендаційної системи - дозволяє користувачеві вибирати товари, на які важко звернути увагу без системи рекомендацій. Наприклад, у системі рекомендацій авто постачальник послуг зацікавлений продати або дати в оренду всі автомобілі в каталозі, а не тільки найпопулярніші. Без рекомендаційної системи це може бути складно, оскільки постачальник послуг

або товарів не може дозволити собі рекламувати всі товари в каталозі кожному споживачу. Тому, рекомендаційна система пропонує або рекламує непопулярні моделі авто потрібним користувачам.

- **Підвищення задоволеності користувачів**. Добре розроблена система рекомендацій також може покращити досвід взаємодії користувача із сайтом чи додатком. Якщо користувач знайде рекомендації цікавими, актуальними, тоді при правильно розробленій взаємодії з сайтом чи додатком споживачу сподобається користуватися системою. Поєднання ефективних рекомендацій та зручного інтерфейсу підвищить суб'єктивну оцінку сайту для користувача. Це, в свою чергу, збільшить кількість відвідувань та ймовірність прийняття рекомендацій.
- **Підвищення рівня користувальницьких якостей.** Користувач повинен бути лояльним до веб-сайту, який під час відвідування визнає споживача, та ставиться до нього як до цінного відвідувача. Це звичайна особливість рекомендаційної системи, оскільки вона використовує інформацію отриману від користувача під час попередніх взаємодій. Отже, чим довше користувач взаємодіє із сайтом, тим більш актуальні товари та послуги будуть пропонуватися рекомендаційною системою.
- **Краще зрозуміти, чого хоче користувач.** Ще одна важлива складова рекомендаційної системи, яку можна використовувати для багатьох інших задач - це набір побажань користувача, зібраних явно або передбачуваними рекомендаційною системою. На підставі цих даних постачальник послуг може точніше управляти запасами або виробництвом товару.

Вище описано основні мотивації того, чому постачальники послуг запроваджують рекомендаційні системи. Але користувачі також можуть хотіти використовувати рекомендаційну систему, якщо вона буде ефективно вирішувати їх завдання та цілі. Отже, рекомендаційна система повинна збалансувати потреби цих постачальників послуг та споживачів і запропонувати цінні та актуальні послуги для обох. У статті Herlocker [3], було

визначено одинадцять основних завдань, у виконанні яких може допомогти рекомендаційна система.

- **Визначити зацікавленість у товарі.** Рекомендувати користувачеві товари як список, разом із прогнозом того, наскільки користувач може бути в них зацікавлених (наприклад, від 1-го до 5-ти). Деякі системи не показують прогнозований рейтинг.
- **Знайти всі хороші товари.** Рекомендувати всі товари, які можуть задовольнити потреби користувача. У таких випадках недостатньо лише знайти якісь хороші предмети. Особливо це стосується тих випадків, коли кількість предметів порівняно невелика або коли PC є критично важливим для вибору, наприклад, у медичних та фінансових додатках. У цих ситуаціях, окрім переваг, отриманих від ретельного вивчення всіх можливостей, користувач також може отримати користь від ранжирування цих пунктів в рекомендаційній системі або від додаткових пояснень, які отримує від рекомендаційній системі.
- **Рекомендувати послідовність.** Замість того, щоб зосереджуватись на формуванні єдиної рекомендації, ідея полягає в тому, щоб рекомендувати послідовність пунктів, яка цікавить в цілому. Типові приклади включають в себе рекомендування телесеріалу, серія за серією, книгу про рекомендаційні системи після того як була рекомендована книга з пошуку даних.
- **Рекомендувати пакет.** Запропонувати групу елементів, які добре поєднуються. Наприклад, план подорожей може складатися з різних атракціонів, напрямків та послуг розміщення, які знаходяться в окресленій території. З точки зору користувача, ці різні альтернативи можна розглядати і вибирати як єдине місце подорожі [10].
- **Просто перегляд**. У цьому завданні користувач переглядає каталог, не маючи наміру придбати товар. Завдання рекомендаційної системи - допомогти користувачеві переглядати елементи, які швидше за все, потрапляють у сферу його інтересів, для конкретного сеансу перегляду.

- **Додатковий функціонал**. Деякі користувачі не довіряють системам рекомендацій, тому вони грають з ними, щоб побачити, наскільки вони гарні в наданні рекомендацій. Система може запропонувати спеціальні функції, щоб користувачі могли перевірити її поведінку.
- **Заповнення профілю.** Це стосується можливості користувача надавати (вводити) інформацію до системи рекомендацій про те, що йому подобається та що не подобається. Це фундаментальне завдання, яке вкрай необхідне для надання персоналізованих рекомендацій. Якщо система не має конкретних знань про активного користувача, то вона може надати йому лише ті самі рекомендації, які були б надані «середньому» користувачеві.
- **Допомга у виборі.** Деякі користувачі із задоволенням ставлять оцінку товару, оскільки вони вважають, що громада отримує користь від їх внеску. Це може стати основною мотивацією для введення інформації в систему рекомендацій.

Як свідчать ці пункти, роль рекомендаційної системи може бути дуже різноманітною. Ця різноманітність вимагає використання цілого ряду різних методів.

РОЗДІЛ 2 АНАЛІЗ МЕТОДІВ ТА ОБРОБКА ДАНИХ

Рекомендаційні системи - це системи обробки інформації, які активно збирають різні види даних для формування своїх рекомендацій. Дані насамперед стосуються предметів, які можна запропонувати, та користувачів, які отримають ці рекомендації. Але оскільки джерела даних та знань, доступні для систем рекомендацій, можуть бути дуже різноманітними, в кінцевому рахунку, чи можна їх використовувати чи ні, залежить від методики рекомендацій. Загалом, існують рекомендаційні методики, які мають низький рівень знань, тобто вони використовують дуже прості та основні дані, такі як рейтинги / оцінки користувачів для предметів. Інші методи значно залежать від даних, наприклад, використання описів користувачів, предметів або обмежень, соціальних відносин та діяльності користувачів. У будь-якому випадку, як загальна класифікація, дані, що використовуються рекомендаційною системою, відносяться до трьох видів об'єктів: товарів, користувачів та транзакцій, тобто відносин між користувачами та предметами.

- **Товари** - це об'єкти, які рекомендуються. Товари можуть характеризуватися їх складністю та вартістю чи корисністю. Значення товару може бути позитивним, якщо предмет є корисним для користувача, або негативним, якщо елемент не підходить і користувач прийняв неправильне рішення при його виборі.

Наприклад, рекомендаційна система новин повинна враховувати складність новини, її структуру, текстове подання та свіжість. Але, в той же час, система повинена розуміти що навіть якщо користувач не платить за читання новин, завжди існує пізнавальна вартість, пов’язана з пошуком та читанням новин. Якщо вибраний товар є релевантним для користувача, у цій вартості переважає користь від придбання корисної інформації, тоді як, якщо товар не є відповідним, чиста вартість цього товару для його користувача та його рекомендація негативна. В інших сферах, наприклад, в автомобільній чи фінансовій, справжня грошова вартість предметів стає важливим елементом,

який слід враховувати під час вибору найбільш відповідного рекомендаційного підходу.

Товари з низькою складністю та цінністю: новини, веб-сторінки, книги, фільми. Елементи з більшою складністю та цінністю - це цифрові камери, мобільні телефони, ПК тощо. Найскладнішими предметами, які розглядалися, є страхові поліси, фінансові інвестиції, подорожі, робочі місця [11]. Рекомендаційні системи, можуть використовувати цілий ряд властивостей предметів. Наприклад, у системі рекомендацій фільмів - жанр (наприклад, комедія, трилер та ін.), а також режисер та актори можуть бути використані для опису фільму, та для того щоб дізнатися як корисність товару залежить від його особливостей. Товари можна представити використовуючи різні підходи до інформації та представлення, наприклад, мінімалістично, як єдиний ідентифікаційний код, або в більш насиченій формі, як набір атрибутів.

- **Користувачі**. Як зазначено вище, користувачі рекомендаційних систем можуть мати дуже різноманітні характеристики. З метою персоналізації рекомендацій та взаємодії людина-комп'ютер, системи використовують низку інформації про користувачів. Цю інформацію можна структурувати різними способами, і знову вибір інформації для моделювання залежить від методики рекомендації. Наприклад, у процесі спільної роботи користувачі моделюються як простий список, що містить рейтинги надані користувачем для деяких товарів. У демографічній системі рекомендацій використовуються соціально-демографічні ознаки, такі як вік, стать, професія та освіта. Дані користувачів складають модель користувача . Користувачі можуть також описуватись їхніми моделями поведінки, наприклад, моделями перегляду веб-сайтів (у веб-системі рекомендацій) або моделями пошуку подорожей (у системі рекомендацій щодо подорожей) [13]. Більше того, дані користувачів можуть включати відносини між користувачами. Система може використовувати цю інформацію, щоб рекомендувати товари користувачам, яким віддавали перевагу подібні або довірені користувачі.

- **Транзакції.** Транзакцією називається завершена взаємодія між користувачем та рекомендаційною системою. Транзакції - це схожі на журнал дані, які зберігають важливу інформацію що генерується під час взаємодії людина-комп'ютер і яка корисна для алгоритму генерації рекомендацій, який використовує система. Наприклад, журнал транзакцій може містити посилання на товар, обраний користувачем, та опис контексту (наприклад, ціль / запит користувача) для цієї конкретної рекомендації. Транзакція може також містити явний зворотний зв’язок, який користувач надав, наприклад, рейтинг для вибраного елемента. Насправді, рейтинги - це найпопулярніша форма даних про транзакції, яку збирає рекомендаційна система. Ці оцінки можуть бути зібрані явно або неявно. У явному оцінюванні, користувач просить надати свою думку про предмет за шкалою рейтингу. Відповідно до [12], рейтинги можуть набувати різноманітних форм:

- Числові рейтинги, такі як 1-5 зірок.
- Звичайні рейтинги, такі як "повністю згоден, згоден, нейтрально, не згоден, категорично не згоден", коли користувачеві пропонується вибрати термін, який найкраще вказує на її думку щодо товару (як правило, за допомогою анкети).
- Бінарні рейтинги - це вибір моделі, в якій користувач просто вирішує, є певний товар хорошим чи поганим.
- Одиничні оцінки - можуть свідчити про те, що користувач бачив або придбав товар. У таких випадках відсутність відгуку свідчить про те, що ми не маємо інформації щодо транзакції (можливо, користувач придбав товар десь в іншому місці).

Інша форма оцінки користувачів складається з тегів, асоційованих користувачем з елементами, представленими системою. У системах, що збирають неявні відгуки, система має на меті зробити висновок користувачів на основі їх дій. Наприклад, якщо користувач введе на Amazon.com ключове слово «Йога», йому буде надано довгий список книг з цієї тематики. Натомість користувач може натиснути на певну книгу в списку, щоб отримати додаткову

інформацію. У цей момент система може зробити висновок, що користувач дещо зацікавлений у цій книзі. Користувач може запитати рекомендації, а система може створити список пропозицій. Він також може запросити налаштування свого профілю, щоб забезпечити кращі результати. У моделі транзакцій, система збирає різні запити-відповіді і може навчитися змінювати свою стратегію взаємодії, спостерігаючи за результатами рекомендаційного процесу [1].

2.1 Рекомендаційні методи

Для того, щоб реалізовувати свою основну функцію - визначити корисні товари для користувача, рекомендаційна система повинна визначити, чи товар варто рекомендувати. Для цього система повинна бути в змозі передбачити корисність деяких з них або принаймні порівняти корисність деяких товарів, а потім вирішити, які товари рекомендувати, грунтуючись на цьому порівнянні.

Щоб проілюструвати крок визначення системи рекомендацій, розглянемо, наприклад, простий неперсоніфікований алгоритм рекомендацій, який рекомендує лише найпопулярніші пісні. Обґрунтуванням цього підходу є те, що за відсутності точної інформації про вподобання користувача, популярна пісня, тобто те, що багатьом користувачам подобається (висока корисність), також, можливо, сподобається загальному користувачеві, принаймні більше, ніж інша випадково вибрана пісня. Отже, прогнозується, що корисність цих популярних пісень буде досить високою для цього загального користувача.

Деякі системи рекомендацій не повністю оцінюють корисність перед тим, як зробити рекомендацію, але вони можуть застосувати деякі евристики, щоб висунути гіпотезу, що предмет корисний користувачеві. Це характерно, наприклад, в системах, заснованих на знаннях. Ці прогнози на корисність обчислюються за допомогою конкретних алгоритмів і використовують різного роду знання про користувачів та товари. Важливо також зазначити, що іноді спостерігається, що корисність товару залежить від інших змінних, які

називаються "контекстуальними" [1]. Наприклад, корисність товару для користувача може залежати від глибини знань певної сфери користувачем (наприклад, професійна камера непотрібна початковим користувачам цифрової техніки), або може залежати від часу запиту рекомендації. Або користувач може бути більше зацікавлений предметами (наприклад, рестораном), ближчим до поточного місця розташування.

Отже, рекомендації повинні бути адаптовані до цих конкретних випадків, і в результаті стає все складніше правильно оцінити що таке правильність рекомендації.

Опишемо шість різних класів рекомендаційних підходів:

- **На основі товарів**. Система вчиться рекомендувати предмети, подібні до тих, які сподобалися користувачеві в минулому. Подібність предметів обчислюється виходячи з особливостей, пов’язаних із порівняними предметами. Наприклад, якщо користувач позитивно оцінив фільм, який належить до жанру комедії, система може навчитися рекомендувати інші фільми з цього жанру.
- **Колективна фільтрація.** Найпростіша та оригінальна реалізація цього підходу [12] рекомендує користувачеві предмети, які в минулому сподобались іншим користувачам з подібними смаками. Подібність у смаку двох користувачів розраховується виходячи з подібності рейтингу користувачів. Колаборативне числення вважається найбільш популярною та широко впровадженою технікою в рекомендаційній системі
- **Демографічний**. Цей тип системи рекомендує товари, що базуються на демографічній складовій користувача. Припущення полягає в тому, що для різних демографічних складових слід формувати різні рекомендації. Багато веб-сайтів приймають прості та ефективні рішення щодо персоналізації на основі демографії. Наприклад, користувачі розсилаються на певні веб-сайти залежно від їх мови чи країни. Або пропозиції можна налаштувати відповідно до віку користувача.

- **На основі знань**. Системи, що базуються на знаннях, рекомендують предмети, що базуються на конкретних знаннях предметної сфери про те, як певні функції товару відповідають потребам та вподобанням користувачів, і в кінцевому рахунку, наскільки цей предмет корисний для користувача. У цих системах функція подібності оцінює, наскільки потреби користувача (опис проблеми) відповідають рекомендаціям (рішення проблеми). Тут оцінка схожості може бути безпосередньо інтерпретована як корисність рекомендації для користувача.
- **На базі спільноти.** Цей тип системи рекомендує предмети на основі вподобань друзів користувачів. Докази свідчать, що люди, як правило, більше покладаються на рекомендації своїх друзів, ніж на рекомендації подібних, але анонімних осіб. Це спостереження в поєднанні зі зростаючою популярністю відкритих соціальних мереж породжує інтерес до систем, що базуються на даних про спільноту, або, як їх зазвичай називають, до систем соціальних рекомендацій [14]. Цей тип рекомендаційних систем отримує інформацію про соціальні відносини користувачів та уподобання друзів користувача. Рекомендація базується на рейтингах, наданих друзями користувача. Насправді ці системи слідкують за зростанням популярності соціальних мереж і дозволяють просто та всебічно отримувати дані, пов'язані з соціальними відносинами користувачів.

Дослідження в цій галузі ще на ранній стадії, і результати щодо роботи систем неоднозначні. Наприклад, [14] повідомляють, що загальні рекомендації, що базуються на соціальних мережах, не є більш точними, ніж ті, що випливають із традиційних підходів, за винятком особливих випадків, наприклад, коли оцінка користувачів певного товару сильно відрізняються (суперечливі позиції) або для ситуацій холодного старту, коли користувачі не надали достатньої кількості оцінок для обчислення подібності з іншими користувачами.

- **Гібридні системи рекомендацій**. Ці рекомендаційні системи засновані на поєднанні вищезазначених методик. Гібридна система поєднує методи A і B,

намагається використати переваги методу A до недоліків методу B. Наприклад, деякі методи мають проблеми з новими товарами, вони не можуть рекомендувати предмети, які не мають оцінок. Це не обмежує підхід орієнтований на контент, оскільки прогнозування нових елементів базується на їх описі (особливостях), які зазвичай легко доступні. [3].

Профіль користувача може бути використаний для кращої персоналізації результатів. Наприклад, рекомендації щодо відпусток взимку повинні сильно відрізнятися від тих, що надаються влітку. Або рекомендація ресторану для суботнього вечора з друзями повинна відрізнятися від рекомендації ресторану для обіду у робочий день з колегами.

2.2 Методи збору даних для рекомендаційних систем

Розглянемо основні методи збору даних, які використовуються в контексті рекомендаційних систем. Опишемо загальні методи попередньої обробки даних, такі як вибірки або зменшення розмірності. Опишемо алгоритм кластеризації k-середніх та обговоримо декілька альтернатив, дослідимо їх використання в системах рекомендацій та представимо випадки, коли вони були успішно застосовані.

Процес збору даних, як правило, складається з 3 етапів, які проводяться послідовно:

- попередня обробка даних
- аналіз даних
- інтерпретація результатів (див. Рис. 1)

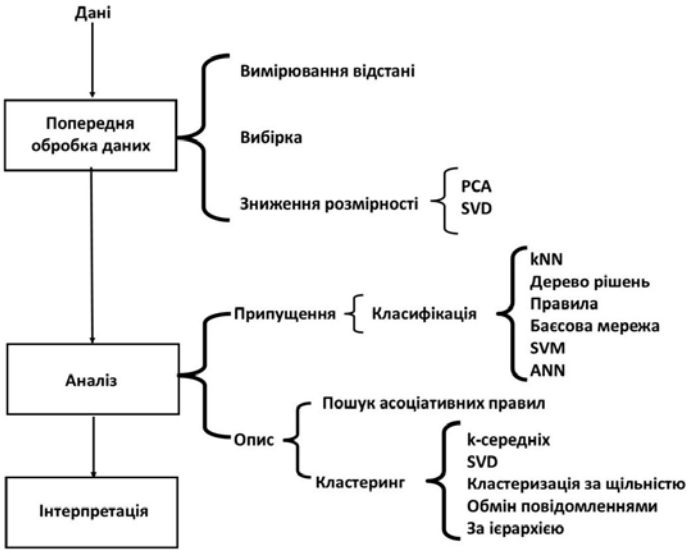


Рис 1 – Основні кроки та методи в задачі збору даних

2.3 Попередня обробка даних

Дані визначаються як сукупність товарів та їх атрибутів, де атрибут визначається як властивість або характеристика товару. Атрибут може називатися змінною, полем, характеристикою або ознакою. Реальні дані, як правило, потрібно попередньо обробити (наприклад, очистити, фільтрувати, трансформувати), щоб їх могли використовувати машинні методи на етапі аналізу.

Одним з кращих підходів до рекомендацій щодо колективної фільтрації є використання класифікатора. Цей метод класифікації - як і більшість класифікаторів та методів кластеризації - сильно залежить від визначення відповідної міри подібності чи відстані.

Найпростішим і найпоширенішим прикладом вимірювання відстані є евклідова відстань:

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2} \qquad (2.1)$$

де n – кількість атрибутів), а x_k і y_k – k-ті атрибути (компоненти) товарів x і y відповідно.

Відстань Міньковського - це узагальнення евклідової відстані:

$$d(x,y) = \left(\sum_{k=1}^{n} |x_k - y_k|^r\right)^{\frac{1}{r}} \qquad (2.2)$$

Відстань махаланобіса визначається як:

$$d(x,y) = \sqrt{(x-y)\sigma^{-1}(x-y)^T} \qquad (2.3)$$

де σ - матриця коваріації даних.

Ще один дуже поширений підхід полягає в тому, щоб розглядати елементи як вектори n-мірного простору та обчислювати їх схожість як косинусом кута, який вони утворюють:

$$cos(x,y) = \frac{(x * y)}{||x||||y||} \qquad (2.4)$$

де * позначає векторний добуток і $\|x\|$ - норма вектора x .

Подібність між товарами може бути також представлена їх співвідношенням, яке вимірює лінійну залежність між об'єктами. Хоча існує декілька коефіцієнтів кореляції, які можна застосувати, найбільш часто застосовується кореляція Пірсона. Враховуючи коваріацію точок даних та їхнє стандартне відхилення σ , ми обчислимо кореляцію Пірсона, використовуючи формулу:

$$Pearson(x,y) = \frac{\sum(x,y)}{\sigma_x \times \sigma_y} \qquad (2.5)$$

Рекомендаційні системи традиційно використовують або косінусну схожість (2.4), або кореляцію Пірсона (2.5) - або одну з їх багатьох варіацій. Було проведень масштабне дослідження, щоб оцінити шість різних методів оцінювання подібності в контексті соціальної мережі. Слід відзначити, що найкращі рекомендації були отримані за допомогою подібності косинусу [9]. Ще одне дослідження дало висновок, що в загальному випадку на вибір точності прогнозування не впливає яким методом буде визначатись схожість товарів. Використання міри випадкової подібності іноді давало кращі результати, ніж використання будь-якого з відомих підходів.

2.4 Вибірка даних

Вибірка - це основна методика, що використовується для вибору підмножини відповідних даних з великого набору даних. Він використовується як на етапах попередньої обробки, так і у фінальній інтерпретації даних. Вибірка може бути корисною, оскільки обробити весь набір даних занадто дорого або довго. Вона також може бути використана для створення наборів навчань та тестування. У цьому випадку навчальний набір даних використовується для вивчення параметрів або конфігурації алгоритмів, що використовуються на етапі аналізу, тоді як набір даних тестування використовується для оцінки моделі або конфігурації, отриманої на етапі тренінгу.

Найпоширеніший підхід до вибірки полягає у використанні вибірки без повторного розміщення: Коли елемент вибирається, він вилучається із сукупності. Однак можливо також здійснити відбір проб із заміною, коли елементи не вилучаються із сукупності після того, як вони були відібрані, що дозволяє відбирати один і той же зразок не один раз.

Загальна практика - застосовувати стандартні випадкові вибірки без заміни пропорцією 80/20. Це означає, що ми використовуємо випадкову

вибірку без заміни, 20% екземплярів для тестового набору, а решту 80% залишаємо для навчання.

2.5 Класифікація

Класифікатор - це відображення між характеристиками товару і класами. Наприклад, система рекомендацій ресторанів може бути реалізованана класифікатором, який класифікує ресторани в одну з двох категорій (хороший, поганий) на основі ряду характеристик, які описують його. Існує багато типів класифікаторів, але загалом ми будемо говорити про класифікацію, що контролюється, або непідконтрольну. Під контрольованою мається на увазі що класи відомі заздалегідь, і ми маємо набір прикладів, які складають навчальний набір. У класифікації, що не контролюється, набір класів заздалегідь невідомий.

2.5.1 Метод найближчих сусідів

Класифікатори, що базуються на екземплярах, працюють, зберігаючи інформацію про тренування та використовують її для прогнозування класу. Тривіальний приклад – цей коли класифікатор запам'ятовує весь навчальний набір та класи і може визначити клас лише у тому випадку, якщо атрибути нового запису точно відповідають одному із навчальних прикладів. Більш досконалим і набагато популярнішим класифікатором на основі екземплярів є класифікатор найближчих сусідів (kNN). Він класифікує k найближчих точок (найближчих сусідів) з навчальних записів. Потім він присвоює товару мітку класу відповідно до міток класів своїх найближчих сусідів. Основна ідея полягає в тому, що якщо запис потрапляє до певного району, де переважають елементи певного класу, ймовірно, сам елемент теж належить до того цього класу.

2.5.2 Дерево рішень

Дерева рішень є класифікаторами цільового класу у вигляді структури дерева. Елементи для класифікації складаються з атрибутів та їх цільового значення. У вузлах дерева перевіряється одна атрибутна величина, щоб визначити, до якої гілки відноситься елемент. Однак з практичних причин більшість реалізацій дерев рішень використовують обмеження, за допомогою якого вузол більше не перевіряється, якщо його кількість спостережень у вузлі нижче певного порогу. Основними перевагами побудови класифікатора за допомогою дерева рішень є те, що побудувати його недорого і це надзвичайно швидко при класифікації невідомих класів. Ще одним важливим аспектом дерева рішень є те, що їх можна використовувати для створення набору даних.

2.5.3 Класифікатори на основі правил

Класифікатори, засновані на правилах, класифікують дані за допомогою набору правил "якщо ... то ...". Правило або умова - це вираз, отриманий із сполук атрибутів. Переваги класифікаторів, заснованих на правилах полягають у тому, що вони є дуже гнучкими, оскільки є строковими та оперують атрибутами даних без будь-яких перетворень. Класифікатори, засновані на правилах і деревах рішень дуже легко інтерпретувати та створювати. Також вони досить ефективно можуть класифікувати нові екземпляри. Подібно до класифікаторів основаних на деревах рішень, дуже складно побудувати повну модель рекомендацій на основі правил. Насправді, цей метод не дуже популярний в контексті рекомендаційних систем, тому що ми повинні мати певні знання процес прийняття рішення. Однак система, заснована на правилах, може бути використана для підвищення ефективності роботи рекомендаційної системи іншого типу, вводячи часткові знання про предметну сферу.

2.6 Кластерний аналіз

Основна проблема масштабування класифікаторів - це кількість операцій, пов'язаних з обчисленням відстаней - наприклад, для визначення k-найближчих сусідів. Навіть якщо ми зменшимо розмірність характеристик, у нас все одно може бути багато об'єктів для обчислення відстані. Тут можуть бути використані алгоритми кластеризації. Те ж саме стосується і систем рекомендацій, де відстані між об'єктами потрібні для отримання подібних елементів. Кластеризація, безумовно, підвищить ефективність, оскільки кількість операцій зменшується. Однак, на відміну від методів зменшення розмірності, навряд чи це може допомогти підвищити точність. Тому кластеризація повинна застосовуватись обережно при розробці рекомендаційної системи, потрібно знайти компроміс між покращеною ефективністю та можливим зниженням точності. Кластеризація складається з присвоєння групам елементів, щоб елементи в одних і тих же групах були більш схожими, ніж предмети в різних групах: мета - виявити природні групи, які існують у даних. Подібність визначається за допомогою вимірювання відстані. Метою алгоритму кластеризації є мінімізація внутрішньокластерних відстаней при максимальному збільшенні міжкластерних відстаней.

Існує дві основні категорії алгоритмів кластеризації: ієрархічний та впорядкування . Алгоритми впорядкування ділять елементи даних на кластери що не перетинаються таким чином, що кожен елемент даних знаходиться в точно одному кластері. Ієрархічні алгоритми кластеризації послідовно кластеризують елементи в знайдених кластерах, створюючи набір вкладених кластерів, організованих як ієрархічне дерево.

Багато алгоритмів кластеризації намагаються мінімізувати функцію, яка вимірює якість кластеризації. Таку функцію якості часто називають цільовою функцією, тому кластеризацію можна розглядати як оптимізаційну проблему: ідеальний алгоритм кластеризації розглядає всі можливі розділи даних та виводить розділ, який мінімізує функцію якості. Багато алгоритмів вдаються до

евристики (наприклад, в алгоритмі k-середніх, використовуючи лише локальні процедури оптимізації, що потенційно закінчуються локальними мінімумами). Головним моментом є те, що кластеризація є складною проблемою, для якої часто неможливо знайти оптимальні рішення. З цієї ж причини вибір конкретного алгоритму кластеризації та його параметрів (наприклад, міра подібності) залежить від багатьох факторів, включаючи характеристику даних.

Одним з найпопулярніших методів кластеризації є метод k-середніх. Метод k-середніх - це метод впорядкування. Функція розділяє набір даних з N елементів на k непересічні підмножини S_j, які містять елементи N_j, щоб вони були максимально наближені один до одного відповідно до заданої міри відстані. Кожен кластер визначається його членами N_j та його центроїдом λ_j. Центроїд для кожного кластеру - це точка, до якої сума відстаней від усіх елементів цього кластеру зводиться до мінімуму. Однак у нього є кілька недоліків:

- він передбачає попереднє знання даних, щоб вибрати відповідний k;
- кінцеві кластери дуже чутливі до виділення початкових центроїді;
- він може створити порожній кластер.

Алгоритм k-середніх також має декілька обмежень щодо даних: він має проблеми, коли кластери різної величини та щільності.

РОЗДІЛ 3 РОЗРОБКА АЛГОРИТМУ

3.1 Постановка задачі

Метою даної кваліфікаційної роботи є розробка програмного продукту, а саме - алгоритму для рекомендаційної системи.

Проаналізувавши існуючі рішення та розробки у даній сфері, була поставлена задача розробити алгоритм, який міг би конкурувати з відомими методами у складних для них ситуаціях, таких як проблема холодного старту, рекомендація нових товарів, робота системи без оцінок користувачів, швидкодія, використання ресурсів, та показувати гарні результати.

Особливістю алгоритму має бути можливість працювати без прив’язки до користувача. Тобто, можливість побудувати набір з великої кількості елементів, описаних характеристиками, у зв’язній послідовності, для кращого сприйняття інформації. Це може бути корисно у таких задачах як перегляд слайд шоу, де використовується інформація тільки про фотографії, або для того щоб побудувати розважально-інформаційний контент у зв’язному порядку для того щоб користувач міг розуміти наскрізний сенс представленої інформації.

За вихідні данні беремо набір характеристик (тегів) з 80 000 фотографій, які були представлені на міжнародному хакатоні від компанії Google у 2019 році спеціально для вирішення задач рекомендаційних систем.

3.2 Попередня обробка даних

Основними структурами даних для роботи нашого алгоритму є матриці, оскільки вони мають певні особливості, які стануть у нагоді під час розробки.

По-перше, розріджені дані у матрицях досить ефективно зберігати, з погляду на використання дискового простору. Оскільки предметна область у якій можуть бути корисні рекомендаційні системи оперує даними про рекламу,

товари, торгівлю, людей,оцінки та відгуки, в основному, це все дуже розріджені дані.

По-друге, існують дуже ефективні методи обробки даних у таблицях, множення таблиць, що може допомогти у вирішенні проблеми швидкодії, що і є метою розробки. Також ми можемо використати всю користь оптимізацій SIMD (Single Instruction, Multiple Data), що дає нам можливість виконувати операції над великою кількістю даних за одну операцію процесору, використовувати кеш процесору та зберігати проміжні значення у швидкій оперативній пам’яті. Виконаємо попередню обробку даних, які мають вигляд:

```
A – { кіт сад яблуко гарбуз кошик квіти }
B – { посмішка сад селфі альтанка телефон }
...
```

Рис. 2 – Неопрацьовані дані

Де A і B елементи які ми хочемо рекомендувати та їх характеристики у фігурних скобках. Створимо матрицю приналежності атрибуту до товарів.

	сад	кіт	авто	телефон	яблуко
A	1	1	0	0	1
B	1	0	0	1	0
C	0	0	1	1	0
D	1	0	0	0	1
E	0	1	1	1	0

Табл. 1 – Матриця приналежності, P

Як ми бачимо, вона розріджена, та має оригінальний розмір 80 000 на 80 000. Оскільки ми маємо на меті розробку універсального алгоритму, розглянемо можливість прив’язатись до користувача, для того щоб отримувати рекомендації які стосуються певної людини (контекстна інформація). Додамо до матриці P користувача, як ще один елемент, та визначимо приналежність

атрибутів до його загальних інтересів, або інтересів за останній час. Дані готові для використання алгоритмом.

3.3 Функція цікавості

Майже кожен алгоритм рекомендаційної системи має функцію яка оцінює наскільки цікавим може бути той чи інший елемент для конкретного користувача. Функцію потрібно підібрати таким чином, щоб рекомендований товар був схожий на поточні інтереси користувача, та міг би бути йому цікавим, але в той же час, повинен в достатній мірі відрізнятись від попереднього елементу або мати певну ступінь різноманітності, наприклад, якщо це рекомендаційна система розважального контенту. У нашому алгоритмі ми будемо використовувати функцію цікавості яка буде мати вигляд:

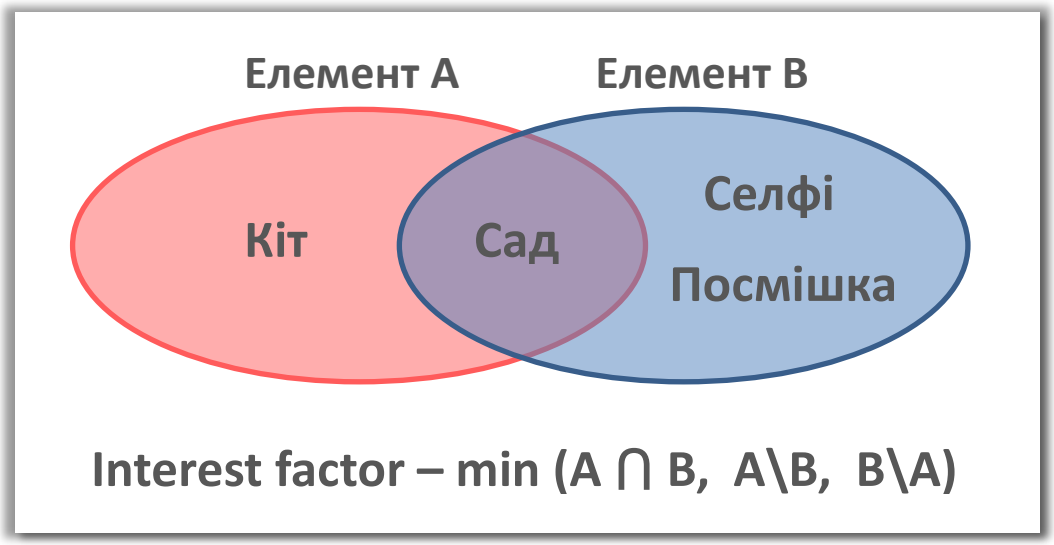


Рис.3 – Функція цікавості

Фактор цікавості буде залежати від кількості однакових характеристик між двома елементами, кількості характеристик які належать елементу А але відсутні у елементі В та навпаки. Фактор цікавості буде найменшим з цих трьох значень. Далі ми будемо максимізувати наш фактор цікавості для знаходження найбільш відповідного елементу для рекомендації.

3.5 Перетин множин

Для подальшого аналізу, знайдемо перетин множин характеристик елементів. Для цього перемножимо матрицю приналежності P на транспоновану матрицю приналежності P^T

$$M = \left(P * P^T\right) \qquad (3.1)$$

У результаті множення та використання оптимізації SIMD ми досить ефективно отримаємо симетричну розріджену матрицю M, яка має вигляд:

	A	B	C	D	E	F	G	H	I
A	n								
B	4	n							
C	1	8	n						
D	0	4	3	n					
E	3	1	0	1	n				
F	0	6	0	7	4	n			
G	9	0	6	4	2	0	n		
H	0	1	9	0	1	2	7	n	
I	7	6	0	8	1	0	5	4	n

Табл. 2 – Матриця перетину, M

На головних осях знаходяться елементи, а на їх перетині ми маємо число спільних характеристик між ними. N – показує загальну кількість характеристик елементу. Наголосимо, що матриця є симетричною, тому 2-гу (верхню) її частину тримати у пам’яті більше немає необхідності, що може бути корисним під час обробки великих об’ємів інформації.

У пункті 3.2 говорилось про прив’язку до користувача. Матриця перетину M яка містить інформацію про перетин інтересів користувача з елементами буде мати вигляд:

	A	B	C	D	E	F	G	H	I	U
A	n									
B	4	n								
C	1	8	n							
D	0	4	3	n						
E	3	1	0	1	n					
F	0	6	0	7	4	n				
G	9	0	6	4	2	0	n			
H	0	1	9	0	1	2	7	n		
I	7	6	0	8	1	0	5	4	n	
U	1	4	3	0	2	5	0	2	4	n

Табл. 3 – Матриця перетину M з даними про користувача

Введемо коефіцієнт k, який буде регулювати схожість/різноманітність між двома характеристиками.

$$\{x | x \geq k\}, \qquad (3.2)$$

де k - коефіцієнт різноманітності

Коефіцієнт виконує 2 головні функції, це можливість обирати наскільки різноманітні товари пропонувати та, обмежувати кількість подальших розрахунків. Оберемо елементи, які мають кількість однакових характеристик з користувачем, більшим або рівним коефіцієнту k. (у табл. 3 коефіцієнт = 4 а елементи позначено жовтим кольором)

Для цієї множини елементів, визначимо фактор подібності згідно функції описаній у пункті 3.3. Для того щоб визначити елемент для рекомендації, оберемо товар з максимальним фактором цікавості.

$$I = \max(\text{interestfactor}(A,B), \text{interestfactor}(A,C), ...) \qquad (3.3)$$

3.6 Орієнтоване дерево

Деякі задачі потребують від рекомендаційних систем вистроїти послідовність товарів один за одним (слайдшоу) або реалізувати швидкий пошук наступного елементу якщо користувач швидко їх переглядає. У цьому завданні стає у нагоді орієнтоване дерево. Добавимо елемент в орієнтоване дерево, та видалимо з таблиці уже переглянутий елемент, та проведемо ті ж дії відносно нового елементу, який порекомендувала система.

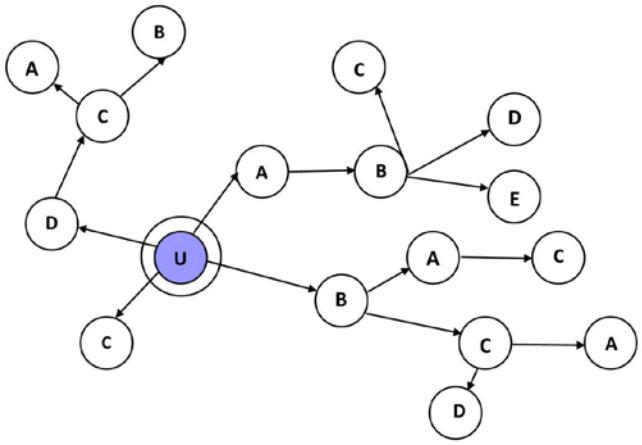
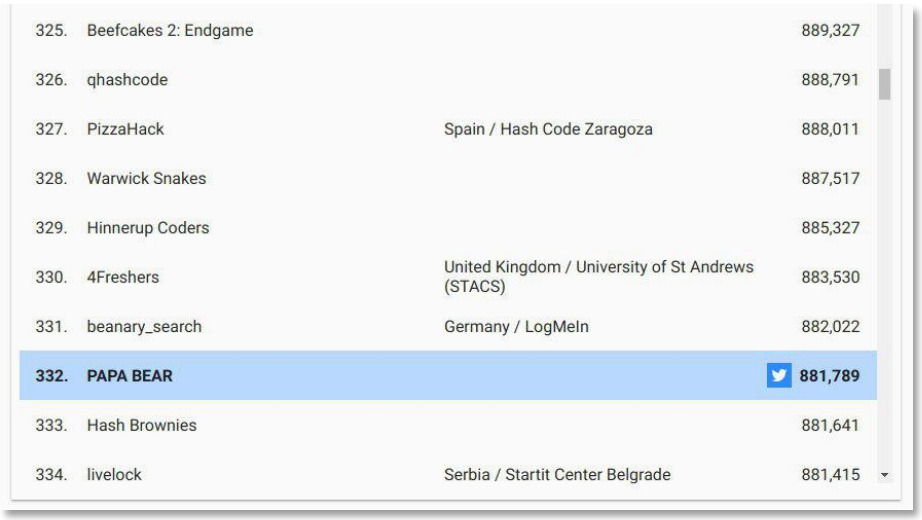


Рис. 4 – Орієнтоване дерево

Таким чином отримаємо дерево можливих рекомендації відносно дій користувача, можемо швидко переходити до наступної рекомендації та переглядати попередні.

3.7 Результат роботи

Розроблений алгоритм був адаптований під конкретну прикладну задачу та протестований на щорічному змаганні з програмування Hash Code від компанії Google. За результатами участі у даному заході, отримано наступні результати:



325.	Beefcakes 2: Endgame		889,327
326.	qhashcode		888,791
327.	PizzaHack	Spain / Hash Code Zaragoza	888,011
328.	Warwick Snakes		887,517
329.	Hinnerup Coders		885,327
330.	4Freshers	United Kingdom / University of St Andrews (STACS)	883,530
331.	beanary_search	Germany / LogMeIn	882,022
332.	**PAPA BEAR**		**881,789**
333.	Hash Brownies		881,641
334.	livelock	Serbia / Startit Center Belgrade	881,415

Рис. 5 – Результати хакатону

- 34 місце серед усіх команд з України (~ 400)
- 332 місце у міжнародному рейтингу (~ 5000)

Алгоритм оброблює дані розміром в 80 000 елементів за 4.2 секунди. на звичайному ПК, що є гарним результатом в плані швидкості роботи. Наприклад, звичайний перебір з використанням функції цікавості зазначеної у пункті 3.3 займає близько 8-ми годин.

ВИСНОВКИ

В цій роботі було проаналізовано базові підходи до створення рекомендаційних систем. В результаті дослідження було зроблено висновок, що не існує універсального алгоритму, який би підійшов до рекомендування будь-яких товарів. Алгоритм потрібно адаптувати та удосконалювати для кожної прикладної задачі для отримання високої точності рекомендації.

Загалом, рекомендаційна система складається з декількох основних частин: попередня обробка даних, аналіз та інтерпретація. І для кожної сфери, будь то наукова, комерційна або розважальна, потрібно підбирати сукупність методів для підвищення ефективності роботи.

Також у роботі наведено головні проблеми з які виникають під час розробки рекомендаційної системи. Одна з найперший проблем з якою зустрічаються розробники - проблема холодного старту, коли користувачі ще не виставили оцінки і немає змоги використовувати рейтингові дані товарів, або смаки користувачів. Друга складність, це можливість зовнішнього впливу на систему, наприклад, якщо система враховує рейтинг товару, можливо штучним шляхом його підняти створивши безліч користувачів, яким би дуже сподобався обраних товар.

За мету роботи ставилось розробити алгоритм для роботи з великою кількістю даних на пристроях з обмеженою. Для цього, головною структурою даних була обрана розріджена таблиця та оптимізації SIMD.

Використання рекомендаційних систем в комерції значно збільшує прибутковість сервісів, адже збільшується залученість користувачів, їх зацікавленість, швидкість взаємодії з продуктом. Вони отримують те що хочуть, а комерційна установа примножує рахунки в банку.

Проаналізувавши результати роботи створеної рекомендаційної системи, можна сказати, що проблема зовнішнього впливу на систему неможлива, оскільки рейтинг товару не впливає на остаточній вибір елементу для рекомендації. Також, з тієї ж причини нові, непопулярні елементи мають

однаковий шанс бути рекомендованими. Також, розроблений алгоритм вдало справляється з великою кількістю даних та має досить високу швидкість роботи та невимогливий до використання ресурсів пристрою.

Отримані результати дослідження пропонується використовувати в якості базового алгоритму розробниками програмного забезпечення під час проектування та розробки рекомендаційних систем. Алгоритм є досить гнучким та може з легкістю поєднуватись з алгоритмами кластеризації, класифікації та іншими для забезпечення кращої точності рекомендації.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mahmood, T., Ricci, F.: Improving recommender systems with adaptive conversational strategies. In: C. Cattuto, G. Ruffo, F. Menczer (eds.) Hypertext, pp. 73–82. ACM (2009)
2. Resnick, P., Varian, H.R.: Recommender systems. Communications of the ACM 40(3), 56–58 (1997)
3. Burke, R.: Hybrid web recommender systems. In: The Adaptive Web, pp. 377–408. Springer Berlin / Heidelberg (2007)
4. Jannach, D.: Finding preferred query relaxations in content-based recommenders. In: 3rd International IEEE Conference on Intelligent Systems, pp. 355–31 (2006)
5. McSherry, F., Mironov, I.: Differentially private recommender systems: building privacy into the net. In: KDD ’09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 627–636. ACM, New York, NY, USA (2009)
6. Schwartz, B.: The Paradox of Choice. ECCO, New York (2004)
7. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Commun. ACM 35(12), 61–70 (1992)
8. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer 42(8), 30–37 (2009)
9. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender Systems An Introduction. Cambridge University Press (2010)
10. Ricci, F., Cavada, D., Mirzadeh, N., Venturini, A.: Case-based travel recommendations. In: D.R. Fesenmaier, K. Woeber, H. Werthner (eds.) Destination Recommendation Systems: Behavioural Foundations and Applications, pp. 67–93. CABI (2006)
11. Montaner, M., L´opez, B., de la Rosa, J.L.: A taxonomy of recommender agents on the inter-net. Artificial Intelligence Review 19(4), 285–330 (2003)

12. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: The Adaptive Web, pp. 291–324. Springer Berlin / Heidelberg (2007)
13. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual in-formation in recommender systems using a multidimensional approach. ACM Trans. Inf. Syst. 23(1), 103–145 (2005)
14. Golbeck, J.: Generating predictive movie recommendations from trust in social networks. In: Trust Management, 4th International Conference, iTrust 2006, Pisa, Italy, May 16-19, 2006, Proceedings, pp. 12–104 (2006)

ДОДАТОК А

Код алгоритму на мові програмування Python 3.6

```
import pandas as pd
import networkx as nx
from scipy.sparse import csr_matrix

input_file = open(r"C:\Users\\Desktop\dataset2.txt", "r")
res_file = open("res.txt", "r")
score_file = open('scores.txt', 'w')
number_of_photos = input_file.readline()

tags = []
tags_set = set()

for line in input_file.readlines():
    tags.append(line.split()[2:])
    for item in line.split()[2:]:
        tags_set.add(item)

tags_dict = dict()

o = 1
for item in tags_set:
    tags_dict[item] = o
    o += 1

ttags = []
for item in tags:
    ttemp_tags = []
    for temp_item in item:
        ttemp_tags.append(tags_dict[temp_item])
    ttags.append(ttemp_tags)

df = pd.DataFrame(ttags)
df = df.fillna(0)

data = [1] * df.shape[0] * df.shape[1]

df_flatten = df.values.flatten()
indices = df_flatten
```

```
indptr = range(0, df_flatten.shape[0] + 51, 51)
matrix = csr_matrix((data, indices, indptr))[:, 1:]
matrix_transpose = matrix.transpose()
mega_matrix = matrix.dot(matrix_transpose)[:, :]
mega_list = list(mega_matrix.data)

unique_values = pd.DataFrame({'A': mega_list}).A.unique()[1:]
mega_str = " ".join([str(l) for l in mega_list])

for i in unique_values:
    mega_str = mega_str.replace(str(i), "1")

mega_distance = [l.count("3") for l in mega_str.split("1")]
mega_distance = mega_distance[:len(mega_distance) - 1]

def get_neighbor(n):
    r = mega_matrix[n, :].nonzero()[1].tolist()
    r.remove(n)
    return r

graph = []
for i in range(0, len(mega_distance)):
    p = []
    n_edge = mega_distance[i]
    p.append(n_edge)
    p.append(get_neighbor(i))
    graph.append(p)


def add_edge_for_node(G, index, neighbors):
    for n in neighbors:
        G.add_edge(index, n)
    return G


def score(res):
    j = 0
    ss = 0
    p_line = ""
    for linee in res[1:]:
        if j == 0:
            j += 1
            p_line = linee
            continue
```

```
        ss += len(set(tags[int(p_line)]).intersection(set(tags[int(linee)])))
        p_line = linee
    return ss


G = nx.Graph()

for i in range(len(graph)):
    node = graph[i]
    G = add_edge_for_node(G, i, node[1])

indices_ones = [i for i, x in enumerate(mega_distance) if x == 17]
print('start of graph ' + str(indices_ones[0]))


def process(index):

    T = nx.dfs_tree(G, index)

    slides = []
    slides.append(len(T.nodes()) - 1)
    for s in T.nodes():
        slides.append(s)
    slides = slides[0: len(slides) - 1]
    return slides


scores = []
g = 0
count_of_iteration = 100
for i in range(len(graph)):

    result = str(score(process(i)))
    print(str(g) + " - ", result)
    scores.append(result)
    score_file.write(result + "\n")
    g += 1

max_index = scores.index(max(scores))
slides = process(max_index)
file = open('res', 'w')

for el in slides:
```

```
    file.write(str(el) + "\n")
file.close()
```