МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ САДІВНИЦТВА

Факультет економіки і підприємництва

Кафедра інформаційних технологій

Допущено до захисту

"_____" _____ 2023 р.

Зав. кафедри, к.т.н., доцент

_____ **Ліщук Р.І.**

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Розробка інформаційно-рекомендаційної системи сайту

Виконав – студент 4 курсу, групи 41-кн
освітньої програми "Комп'ютерні науки"
спеціальності 122 Комп'ютерні науки
Янковий Микола Русланович
Науковий керівник – д.т.н., професор
Кучерук Володимир Юрійович

Умань – 2023

Реферат

Перелік умовних позначень, скорочень і термінів

CBF	Content-based Filtering
CF	Collaborate Filtering
SVD	Singular-Value Decomposition
SOM	Self-Organized Map
SVM	Support Vector Machine
NN	Neural Networks
ANN	Artificial Neural Networks
RS	Recommendation System
FTP	File Transfer Protocol
ID	Identity Document
URL	Uniform Resource Locator
API	Application Programming Interface
СКВ	Середньоквадратичне відхилення

Вступ

Рекомендаційна система — підклас системи фільтрації інформації, яка будує рейтинговий перелік об'єктів (фільми, музика, книги, новини, веб-сайти), яким користувач може надати перевагу. Для цього використовується інформація з профілю користувача.

Існують дві основні стратегії створення рекомендаційних систем: фільтрація вмісту і колаборативна фільтрація.

При фільтрації вмісту створюються профілі користувачів і об'єктів.

Профілі користувачів можуть містити демографічну інформацію або відповіді на певний набір питань.

Профілі об'єктів можуть містити назви жанрів, імена акторів, імена виконавців, тощо. Або якусь іншу інформацію в залежності від типу об'єкта.

При колаборативній фільтрації використовується інформація про поведінку користувачів у минулому — наприклад, інформація про придбання або оцінки. В цьому разі не має значення, з якими типами об'єктів ведеться робота, але при цьому можна брати до уваги неявні характеристики, які складно було б врахувати при створенні профілю. Основна проблема цього типу рекомендаційних систем — «холодний старт»: відсутність даних про користувачів чи об'єкти, які нещодавно з'явились у системі.

У процесі роботи рекомендаційні системи збирають дані про користувачів, використовуючи поєднання явних і неявних методів.

Приклади явного збору даних:

- користувач оцінює запропонований об'єкт за диференційованою шкалою;
- користувач ранжує групу об'єктів від найкращого до найгіршого;
- користувач вибирає кращий з двох запропонованих об'єктів;
- користувачу пропонують створити список його улюблених об'єктів.

Приклади неявного збору даних:

- спостереження за тим, що користувач оглядає в інтернет-магазинах або базах даних іншого типу;
- ведення записів про поведінку користувача онлайн;
- відстеження вмісту комп'ютера користувача.

Рекомендаційні системи порівнюють однотипні дані від різних людей і розраховують список рекомендацій для конкретного користувача. Деякі приклади їх комерційного та некомерційного використання наведені в статті про колаборативну фільтрацію. Для розрахунку рекомендацій використовується граф інтересів. Рекомендаційні системи — зручна альтернатива пошуковим алгоритмам, оскільки дозволяють виявити об'єкти, які не можуть бути знайдені останніми. Цікаво, що рекомендаційні системи часто використовують пошукові машини для індексації незвичайних даних.

Рекомендаційна система - це програма, яка на основі даних про користувача (User) і предмет (Item) дає рекомендації. Така система включає в себе весь процес - від отримання інформації до її подання користувачеві. Важливий кожен етап. Від інформації, яку ви будете збирати залежить, які алгоритми ви зможете застосувати. Хороші алгоритми дають хороші, корисні рекомендації. Критерії оцінки результату дозволяють вибрати найбільш підходящі алгоритми.

Актуальність

Враховуючи, що рекомендаційна система - це програма, яка на основі даних про користувача (User) і предмет (Item) дає рекомендації. Така система включає в себе весь процес - від отримання інформації до її подання користувачеві. Важливий кожен етап. Від інформації, яку ви будете збирати залежить, які алгоритми ви зможете застосувати. Хороші алгоритми дають хороші, корисні рекомендації. Критерії оцінки результату дозволяють вибрати найбільш підходящі алгоритми [1].

Переважно всі IT-компанії в сфері «ecommerce» використовують колаборативну фільтрацію, перша компанія яка реалізувала такі алгоритми –

Amazon. Найпопулярніша в Україні – Rozetka. Проте сучасному світі часто доводиться стикатися з проблемою рекомендації товарів або послуг користувачам будь-якої інформаційної системи.

Вже відомомо, що колаборативна фільтрація — метод, який використовується деякими рекомендаційними системами. В цілому, колаборативна фільтрація — процес фільтрації інформації або зразків за допомогою методів за участю співробітництва між декількома агентами, точками зору, джерелами даних і т. д. Застосування колаборативної фільтрації, як правило, пов'язане з дуже великими наборами даних. Алгоритми рекомендаційних систем використовуються в кожній компанії, яка позиціонує себе як інтернет-магазин, дошка оголошень, сайт-агрегатор і т.д. В нашому випадку колаборативна фільтрація — це один з методів побудови прогнозу в рекомендаційних системах, який використовує відомі переваги (оцінки предметів, дії, поведінку) групи користувачів для прогнозування невідомих переваг іншого користувача. Основне припущення колаборативної фільтрації полягає в наступному: ті, хто однаково оцінювали будь-які предмети в минулому, схильні давати схожі оцінки інших предметів і в майбутньому [2].

Таким чином можна зробити висновок, що продукт як сайт повинен бути максимально корисним для користувача. Та враховуючи стрімкий ріст WEB-ресурсів, спрямованих на покупки в інтернеті, можна зробити висновок, що покращення точності колаборативної фільтрації в рекомендаційних системах - актуальна тематика.

Об’єкт дослідження – процеси, що виникають у інформаційно-рекомендаційних системах сайтів.

Предмет дослідження – метод побудови інформаційно-рекомендаційних систем сайтів.

Мета дослідження

Метою проведених досліджень була розробка рекомендаційних систем, корисних в першу чергу для користувача, допомагаючи зробити правильний вибір в пошуку товару та скоротивши загальний час пошуку, цим самим допомагаючи обрати товар. А також, збільшити точність рекомендацій.

Під час виконання кваліфікаційної роботи бакалавра був проведений аналіз наукової та технічної літератури, технічної документації, міжнародних стандартів у даній предметній галузі, а також проведено розробку математичного та програмного забезпечення системи.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Об’єкт та предмет дослідження

Вибуховий ріст обсягу доступної цифрової інформації та кількості відвідувачів Інтернету створив потенційний виклик перевантаження інформації, що перешкоджає своєчасному доступу до цікавих предметів в Інтернеті. Інформаційно-пошукові системи, такі як Google, DevilFinder та Altavista, частково вирішили цю проблему, однак відсутність пріоритетів та персоналізації (де система відображає доступний вміст до інтересів та уподобань користувачів) інформації. Це посилило попит на системи, що рекомендують, більше, ніж будь-коли раніше. Рекомендаційні системи - це системи фільтрування інформації, що займаються проблемою інформаційної перевантаження шляхом фільтрації фрагмента життєво важливої інформації з великої кількості динамічно створюваної інформації відповідно до переваг, інтересів або поведінки користувача щодо елемента. Система Recommender має можливість прогнозувати, чи буде певний користувач віддавати перевагу елементу, або не базується на профілі користувача.

Системи рекомендацій корисні як постачальникам послуг, так і користувачам. Вони зменшують транзакційні витрати на пошук та відбір елементів в онлайн-магазині. Системи рекомендацій також довели, що вони покращують процес прийняття рішень та якість. У налаштуваннях електронної комерції системи, що рекомендують, збільшують доходи, оскільки вони є ефективним засобом продажу більшої кількості продуктів. У наукових бібліотеках системи, що рекомендують, підтримують користувачів, дозволяючи їм виходити за рамки пошуку в каталозі. Тому необхідність використання ефективних та точних методів рекомендацій в рамках системи, яка забезпечить відповідні та надійні рекомендації для користувачів, не може бути надмірно наголошеною.

Система рекомендацій визначається як стратегія прийняття рішень для користувачів у складних інформаційних середовищах. Крім того, система

рекомендацій була визначена з точки зору електронної комерції як інструмент, який допомагає користувачам здійснювати пошук записів про знання, пов'язані з інтересом та перевагою користувачів. Система рекомендацій була визначена як засіб сприяння та посилення соціального процесу використання рекомендацій інших осіб для вибору, коли немає достатніх особистих знань або досвіду альтернатив. Системи, що рекомендують, обробляють проблему перевантаження інформації, яку користувачі зазвичай стикаються, надаючи їм індивідуальні, виняткові вміст та рекомендації щодо обслуговування. Нещодавно були розроблені різні підходи для побудови систем рекомендацій, які можуть використовувати як спільну фільтрацію, фільтрування на основі вмісту чи гібридну фільтрацію.

Технологія спільної фільтрації є найбільш зрілою і найбільш часто застосовується. Спільна фіксація рекомендує елементи, ідентифікуючи інших користувачів з аналогічним смаком; він використовує свою думку, щоб рекомендувати елементи активному користувачеві. Спільна рекомендаційна система була впроваджена в різних областях застосування. GroupLens - це архітектура на основі новин, яка використовує спільні методи, допомагаючи користувачам знаходити статті з масивної бази новин.

Рінго - це система онлайн-фільтрування соціальної інформації, яка використовує спільну фільтрацію для створення профілів користувачів на основі оцінок їх у музичних альбомах. Amazon використовує алгоритми диверсифікації теми, щоб поліпшити свою рекомендацію. Система використовує метод спільного фільтрування для подолання проблеми масштабування, створивши таблицю подібних елементів поза мережею, використовуючи матрицю "елемент до елемента". Потім система рекомендує інші товари, подібні онлайн, за історією покупки користувачів. З іншого боку, методи, що базуються на вмісті, співпадають з ресурсами вмісту відповідно до характеристик користувача. Технології фільтрування на основі вмісту зазвичай базують свої передбачення на інформації користувача, і вони ігнорують вклади інших користувачів, як у

випадку спільних методів. Фаб багато в чому залежить від рейтингів різних користувачів, щоб створити навчальний комплект, і це є прикладом системи рекомендованих на основі вмісту. Деякі інші системи, які використовують фільтрування на основі вмісту, щоб допомогти користувачам знаходити інформацію в Інтернеті, включають Letizia. Система використовує користувальницький інтерфейс, який допомагає користувачам переглядати Інтернет; він може відстежувати шаблон перегляду користувача, щоб передбачити сторінки, які вони можуть зацікавити. Pazzani et al. розробив інтелектуальний агент, який намагається передбачити, які веб-сторінки зацікавлять користувача за допомогою наївного класифікатора Байєса. Агент дозволяє користувачеві надавати навчальні екземпляри, оцінюючи різні сторінки як гарячі, так і холодні. Дженнінгс і Хігучі описують нейронну мережу, яка моделює інтереси користувача в середовищі новин Usenet.

Незважаючи на успіх цих двох методів фільтрації, було визначено кілька обмежень. Деякі проблеми, пов'язані з технологіями фільтрування на основі вмісту, містять обмежений аналіз вмісту, надспеціалізацію та малу кількість даних. Також спільні підходи демонструють проблеми холодного старту, ранжирування та масштабованості. Ці проблеми зазвичай зменшують якість рекомендацій. Для того, щоб пом'якшити деякі виявлені проблеми, було запропоновано гібридну фільтрацію, яка поєднує два або більше методів фільтрації різними шляхами для підвищення точності та ефективності систем рекомендацій. Ці методики поєднують два або більше фільтруючих підходів, щоб використовувати їх сильні сторони, вирівнюючи їхні відповідні слабкі сторони. Їх можна класифікувати за своїми операціями в гібридний гібрид, змішаний гібрид, гібрид перемикання, гібрид комбінації комбінацій, гібрид каскаду, гібрид гібридного та гібридного рівня, що складається з ознак. Спільна філософія та підходи до фільтрування на основі вмісту широко використовуються сьогодні, реалізуючи різноманітні технології на основі вмісту та спільної роботи, а результати їхнього прогнозування згодом поєднуються або

додають характеристики співвідношення вмісту на основі спільної фільтрації та навпаки. Нарешті, може бути розроблена загальна уніфікована модель, яка включає як властивості фільтрування на основі вмісту, так і спільні. Проблема роздільної здатності даних та "холодного старту" була вирішена шляхом поєднання оцінок, особливостей та демографічної інформації про предмети в гібридній методиці каскадної рекомендації Циглера та ін. було запропоновано гібридний підхід до спільного використання фільтрації для використання основної таксономічної інформації, призначеної для вимогливої класифікації продуктів, для вирішення проблеми обмеженості даних рекомендацій з використанням CF, виходячи з генерації профілів за допомогою висновку над темою та диверсифікацією тем. Гібридна рекомендаційна методика також пропонується в Ghazantar і Pragel-Benett, і це використовує профіль на основі контенту окремих користувачів для пошуку схожих користувачів, які використовуються для прогнозування. У Сарварі та ін., спільна фільтрація була об'єднана з інформаційним агентом фільтрування. Тут автори запропонували рамки для інтеграції фільтрів-агентів на основі вмісту та спільної фільтрації. Гібридний алгоритм рекомендувача застосовується багатьма додатками в результаті нової проблеми користувача з фільтром на основі вмісту та середньою проблемою для спільної фільтрації. Простий та зрозумілий спосіб об'єднання фільтрації на основі вмісту та спільної роботи було запропоновано Каннінгем та ін.. У Konstas і співавторстві була запропонована система рекомендацій щодо музики, в якій поєднувалась інформація про теги, графічні показники та соціальні відносини. Щоб визначити кількість сусідів, які можуть бути автоматично підключені на соціальній платформі, Лі та Брусиловський вбудували соціальну інформацію в спільний алгоритм фільтрації. Кондіфф та ін. запропонували Байєсівську модель змішаних ефектів, яка об'єднує оцінки користувачів, елементи користувача та елементи в єдиній структурі.

1.2 Рекомендаційні методи фільтрації

Використання ефективних та точних методів рекомендацій є дуже важливим для системи, яка забезпечить корисні та корисні рекомендації окремим користувачам. Це пояснює важливість розуміння особливостей та потенціалів різних методів рекомендації. На рис. 1.1 показана анатомія різних методів фільтрування рекомендацій.

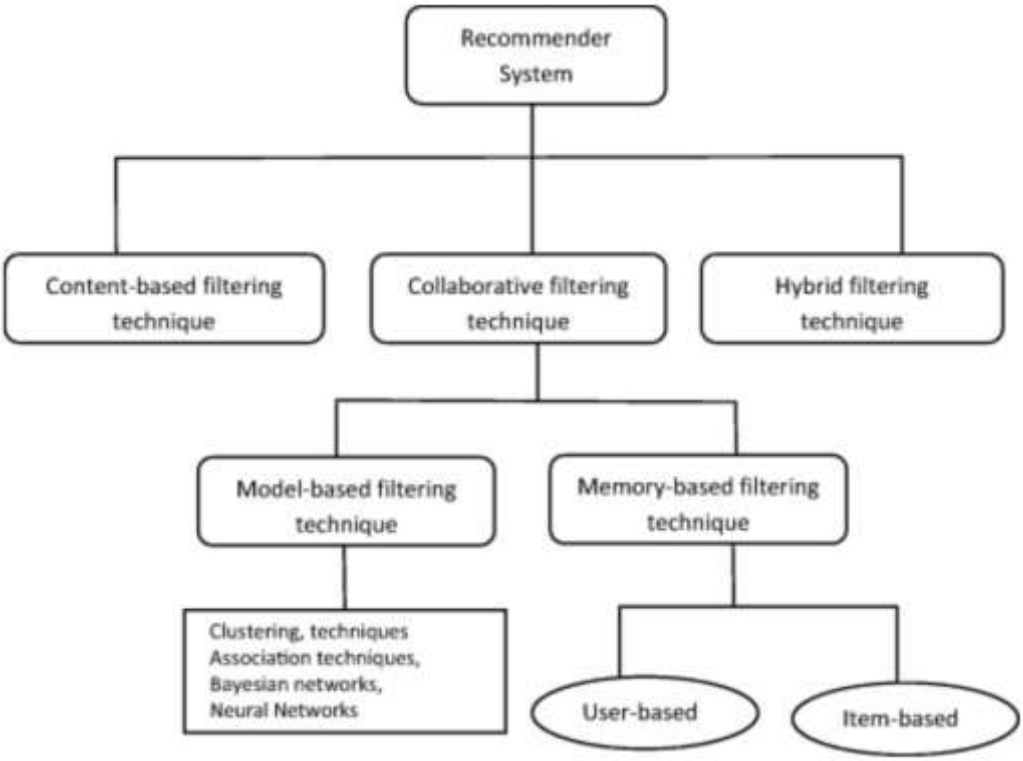


Рисунок 1.1. Рекомендаційні методи

Фільтрування на основі вмісту

Техніка, основана на контенті, - це алгоритм, залежний від домену, і більше підкреслює аналіз атрибутів предметів для створення прогнозів. Якщо потрібно рекомендувати такі документи, як веб-сторінки, публікації та новини, найефективніша методика фільтрування на основі вмісту. У технології фільтрування на основі вмісту, рекомендація робиться на основі профілів користувачів, використовуючи функції, витягнуті з вмісту елементів, які користувач оцінив раніше. Елементи, які в основному пов'язані з позитивно оціненими елементами, рекомендуються користувачеві. CBF використовує різні

типи моделей, щоб знайти схожість між документами, щоб створити корисні рекомендації. Він може використовувати модель Vector Space, таку як Inverse Document Frequency (TF/IDF) або імовірнісні моделі, такі як класифікатор Бейєса, Рішення Дерева або Нейронні мережі [3, 4] моделювати зв'язок між різними документами всередині корпусу. Ці методики дають рекомендації, вивчаючи базову модель за допомогою статистичного аналізу або методів машинного навчання. Технологія фільтрування на основі вмісту не потребує профілю інших користувачів, оскільки вона не впливає на рекомендації. Також, якщо змінюється профіль користувача, то метод CBF все ще може регулювати свої рекомендації протягом дуже короткого періоду часу. Основним недоліком цієї техніки є необхідність мати глибокі знання та опис особливостей елементів у профілі.

Плюси та мінуси технологій фільтрування на основі вмісту полягають у наступному. Вони мають можливість рекомендувати нові елементи, навіть якщо немає оцінок, наданих користувачами. Таким чином, навіть якщо база даних не містить переваг користувача, точність рекомендацій не впливає. Також, якщо змінюються налаштування користувача, він може змінювати свої рекомендації за короткий проміжок часу. Вони можуть управляти ситуаціями, коли різні користувачі не поділяють однакові елементи, а лише ідентичні елементи відповідно до їх властивостей. Користувачі можуть отримувати рекомендації без надання доступу до свого профілю, і це гарантує конфіденційність. Техніка CBF також може дати пояснення щодо того, як користувачі створюють рекомендації. Проте методики страждають від різних проблем, як обговорюється в літературі. Методи фільтрування на основі вмісту залежать від метаданих елементів. Тобто вони потребують насиченого опису елементів і дуже добре організованого профілю користувача, перш ніж рекомендації можуть бути зроблені користувачам. Це називається обмежений аналіз контенту. Отже, ефективність CBF залежить від наявності описових даних. Інша серйозна проблема методу

CBF є над спеціалізацією вмісту. Користувачі можуть отримувати рекомендації, подібні до елементів, вже визначених у їхніх профілях.

Приклади систем фільтрування на основі вмісту. Новини Dude - це персональна новинна система, яка використовує синтезовану мову, щоб читати новини для користувачів.

Модель TF-IDF використовується для опису новин, щоб визначити короткотермінові рекомендації, які потім порівнюються з вимірюванням подібності косинусів і нарешті передаються алгоритму навчання (NN). CiteSeer - це автоматичне індексування цитування, яке використовує різні алгоритми евристики та машинного навчання для обробки документів. Сьогодні CiteSeer є одним із найбільших і широко використовуваних дослідницьких паперових сховищ в Інтернеті.

LIBRA - це система рекомендацій, основана на зміст, яка використовує інформацію про книгу, зібрану з Інтернету. Він реалізує класифікатор Наївного Байєса з інформації, витягнутої з Інтернету, для вивчення профілю користувача для створення рейтингового списку назв на основі навчальних прикладів, наданих окремим користувачем.

Система здатна надавати пояснення щодо будь-яких рекомендацій, зроблених користувачам, шляхом перерахування функцій, що сприяють досягненню найвищих оцінок, а отже, дозволяючи користувачам повністю довіряти рекомендаціям, наданим користувачам системи.

Спільна фільтрація

Спільна фільтрація - це техніка, що не залежить від домену, для вмісту, який неможливо легко та адекватно описати метаданими, такими як фільми та музика. Технологія спільної фільтрації створює базу даних (матрицю користувацьких елементів) для налаштування елементів користувачами.

Потім він відповідає користувачам, які мають відповідні інтереси та переваги, шляхом обчислення подібності між їх профілями для надання рекомендацій. Такі користувачі створюють групу під назвою neighborhood. Користувач отримує рекомендації щодо тих предметів, які він раніше не оцінив,

але які вже були позитивно оцінені користувачами в його околицях. Рекомендації, що виробляються колаборативною фільтрацією, можуть бути як передбаченням, так і рекомендаціями.

Прогноз - це чисельне значення Rij, яке виражає прогнозовану оцінку пункту j для користувача i, тоді як рекомендація - це список найпопулярніших N елементів, які користувачеві найбільше сподобається, як показано на рис. 1.2. Техніку спільної фільтрації можна розділити на дві категорії: на основі пам'яті та на основі моделі.

Процес спільного фільтрування зображений на наступному рисунку в розрізі з користувачами та предметами.

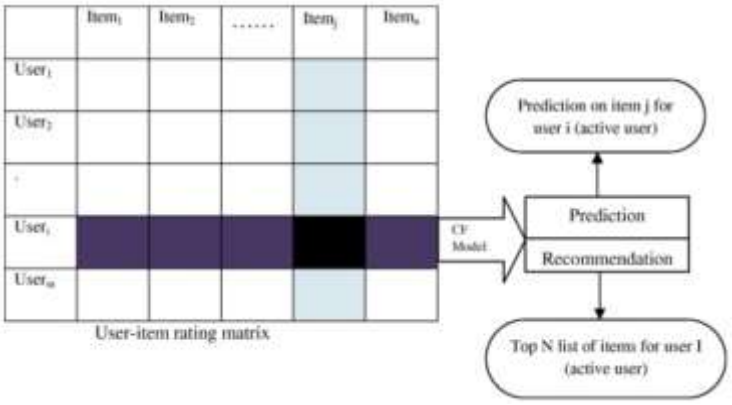


Рисунок 1.2 - Процес спільного фільтрування

Методика на основі пам'яті полягає в тому, що елементи, які користувач уже оцінив, перш ніж відігравати важливу роль у пошуку сусіда, який поділяє оцінку з ним. Після знаходження сусіда користувача різні алгоритми можуть бути використані для поєднання переваг сусідів для створення рекомендацій. Завдяки ефективності цих методів, вони досягли широкого успіху в реальних додатках. На базі пам'яті CF можна досягти двома способами за допомогою користувацьких та елементів-методів. Технологія спільної роботи з фільтрацією, що використовується користувачем, розраховує подібність між користувачами, порівнюючи їхні рейтинги з одним і тим же елементом, а потім обчислює передбачуваний рейтинг для елемента активним користувачем як

середньозважений показник оцінок елемента користувачами, подібними до активного користувача, де ваги - це схожість цих користувачів із цільовим елементом. Методи фільтрації на основі предметів обчислюють прогнози з використанням подібності між елементами, а не подібністю між користувачами. До найбільш схожі елементи і відповідні їм загальні риси також визначаються. Прогноз здійснюється шляхом прийняття середньозваженого середнього показника активних користувачів за подібними пунктами k. Для обчислення подібності між елементом / користувачем використовуються декілька типів заходів подібності. Два найпопулярніших заходи подібності - кореляційні та косинусні. Коефіцієнт кореляції Пірсона використовується для вимірювання ступеня, до якої дві змінні лінійно відносяться один з одним і визначається як

$$s(a,u) = \frac{\sum_{i=1}^{n}(r_{a,i}-\bar{r}_a)(r_{u,i}-\bar{r}_u)}{\sqrt{\sum_{i=1}^{n}(r_{a,i}-\bar{r}_a)^2}\sqrt{\sum_{i=1}^{n}(r_{u,i}-\bar{r}_u)^2}} \quad (1.1)$$

З вищевказаного рівняння $s(a,u)$ позначає схожість між двома користувачами і $r_{a,i}$ - це рейтинг, заданий елементу i користувача a , $\bar{r}_a$ - це середній рейтинг, заданий користувачем в той час, коли n - загальна кількість елементів у просторі користувача-об'єкта. Також передбачення для елемента складається з вагової комбінації оцінок вибраних сусідів, яка обчислюється як ваговане відхилення від середнього значення сусідів. Загальна формула передбачення є

$$p(a,i) = \bar{r}_a + \frac{\sum_{i=1}^{n}(r_{u,i}-\bar{r}_u)\times s(a,u)}{\sum_{i=1}^{n} s(a,u)} \quad (1.2)$$

Косинусна подібність відрізняється від виміру на основі Пірсона, оскільки вона є моделлю вектор-космічної, яка базується на лінійній алгебрі, а не на статистичному підході. Він вимірює подібність двох n- мірних векторів на основі кута між ними. Межа на основі косину широко використовується в областях пошуку інформації та видобування текстів для порівняння двох текстових документів, в даному випадку документи представлені векторами термінів. Подібність між двома пунктами u та v може бути визначена як наступним чином:

$$s(\vec{u}, \vec{u}) = \frac{\vec{u} \cdot \vec{u}}{|\vec{u}| * |\vec{u}|} = \frac{\sqrt{\sum_{i=1}^{n} r_{u,i}^2 r_{u,i}^2}}{\sqrt{\sum_{i=1}^{n} r_{u,i}^2} \times \sqrt{\sum_{i=1}^{n} r_{u,i}^2}} \quad (1.3)$$

Межа схожості також називається показником подібності, і це методи, що використовуються для розрахунку балів, які виражають, як подібні користувачі або елементи відносяться один до одного. Ці бали потім можуть бути використані в якості основи генерації рекомендацій на основі користувачів або елементів. Залежно від контексту використання, показники подібності можна також назвати метрикою кореляції або показниками відстані.

Моделізовані методи - це методика, що використовує попередні рейтинги для вивчення моделі, щоб підвищити ефективність Спільної технології фільтрації. Процес побудови моделі може бути виконаний за допомогою машинного навчання або методів вилучення даних. Ці методики можуть швидко рекомендувати набір предметів для того факту, що вони використовують попередньо обчислену модель, і вони довели, що отримали результати рекомендацій, які схожі на методи рекомендацій на базі сусідніх регіонів. Приклади цих методів включають в себе методики зменшення розміру, такі як розкладання сингулярного значення (SVD), методика завершення матриці, приховані семантичні методи та регресія та кластеризація. Методи, що базуються на моделях, аналізують матрицю користувацького елемента для визначення зв'язків між елементами; вони використовують ці відносини для порівняння переліку рекомендацій "топ-N".

Використання алгоритмів навчання також змінило спосіб рекомендацій із рекомендацій щодо споживання користувачами рекомендацій щодо фактичного споживання продукту. Тому дуже важливо вивчити інші алгоритми навчання, що використовуються в моделях на основі рекомендаційних систем.

Правило асоціації: асоціація правил вишивання алгоритмів вилучає правила, які передбачають виникнення елемента на основі наявності інших елементів у транзакції. Наприклад, якщо вказати набір транзакцій, де кожна транзакція є сукупністю елементів, правило асоціації застосовує форму A → B,

де A та B складають два набори елементів . Правила асоціації можуть скласти дуже компактне представлення даних переваг, які можуть підвищити ефективність зберігання, а також продуктивність. Також, іноді відомо, що ефективність правила асоціації для розкриття шаблонів та керування персоналізованими маркетинговими рішеннями. Однак існує чітка зв'язок між цим методом та ціллю системи рекомендацій, але вони не стали основними.

Кластеризація. Технології кластеризації застосовуються в різних областях, таких як розпізнавання образів, обробка зображень, аналіз статистичних даних та знання. Алгоритм кластеризації намагається розділити набір даних на сукупність підгруп, щоб виявити значущі групи, що існують в них. Після формування кластерів, думки інших користувачів у кластері можуть бути усереднені та використані для надання рекомендацій окремим користувачам. Хороший метод кластеризації буде створювати високоякісні кластери, в яких схожість між кластерами є високою, а подібність між кластерами є низькою. У деяких кластеризованих підходах користувач може частково брати участь у різних кластери, а потім рекомендації базуються на середньому по кластерах участі, який оцінюється ступенем участі. K-засоби та самоорганізуюча карта (SOM) найчастіше використовуються серед різних методів кластеризації. K-means приймає вхідний параметр, а потім розділяє набір n елементів у кластери. Самоорганізаційна карта (SOM) - це метод безконтрольного навчання, який базується на методі кластеризації штучних нейронів. Методи кластеризації можуть бути використані для зменшення кандидата на основі алгоритмів на базі спільної роботи.

Дерево рішень: дерево рішень ґрунтується на методології графіків дерев, яка побудована шляхом аналізу набору навчальних прикладів, для яких є відомості класів. Потім вони застосовуються для класифікації раніше невидимих прикладів. Якщо вони навчаються надто високоякісними даними, вони мають можливість робити дуже точні прогнози. Дерева рішень є більш інтерпретованими, ніж інші класифікатори, такі як Support Vector Machine (SVM)

та Neural Networks, тому що вони об'єднують прості запитання про дані зрозумілим чином. Рішення дерев також є гнучкими в обробці предметів з сумішшю реальних і категоріальних особливостей, а також предмети, які мають деякі конкретні відсутні функції.

Штучна нейронна мережа: ANN є структурою багатьох з'єднаних нейронів (вузлів), які систематично розташовані у шарах. Зв'язки між нейронами мають пов'язані з ними ваги залежно від кількості впливу одного нейрона на інший. Є деякі переваги у використанні нейронних мереж у деяких особливих проблемних ситуаціях. Наприклад, через те, що він містить багато нейронів, а також присвоюється вага для кожного з'єднання, штучна нейронна мережа досить надійна щодо шумних і помилкових наборів даних. ANN має можливість оцінювати нелінійні функції та захоплювати складні відносини в наборах даних, вони також можуть бути ефективними і навіть працювати, якщо частина мережі не працює. Основний недолік полягає в тому, що важко знайти ідеальну топологію мережі для заданої проблеми, і, як тільки буде вирішено топологію, це буде нижчою межами для помилки класифікації.

Аналіз посилань: Аналіз посилань - це процес побудови мереж взаємозв'язаних об'єктів для вивчення моделі та тенденцій. Він представив великі можливості для вдосконалення веб-пошуку. Аналіз посилань складається з алгоритмів PageRank і HITS. Більшість алгоритмів аналізу посилань обробляють веб-сторінку як єдиний вузол у веб-графіку.

Регресія: регресійний аналіз використовується, коли вважається, що дві або більше змінних системно пов'язані між собою за допомогою лінійних взаємозв'язків. Це потужний і різнорідний процес для аналізу асоціативних зв'язків між залежною змінною та однією чи більше незалежними змінними. Використання регресії містить формування кривої, прогнозування та тестування системних гіпотез про співвідношення змінних. Крива може бути корисною для визначення тенденції в межах набору даних, будь то лінійна, параболічна або деяких інших форм.

Байєсівські класифікатори: вони є імовірнісною основою для вирішення проблем класифікації, яка ґрунтується на визначенні умовної ймовірності та теореми Байєса. Байєсівські класифікатори розглядають кожен атрибут і клас етикетки як випадкові величини. З огляду на запис N функцій (1, 2, ..., N), мета класифікатора складається в прогнозі класу C_K, знаходячи значення C до , що максимізує апостеріорна ймовірність класу, враховуючи дані P (Ck | A1, A2,..., H), застосовуючи теорему Байєса. Найбільш часто використовуваний класифікатор Байєса відомий як класифікатор наївних Байєсів. Для оцінки умовної ймовірності наївний класифікатор Байєсів приймає імовірнісну незалежність атрибутів, тобто наявність або відсутність певного атрибуту не має відношення до присутності або відсутності будь-якого іншого.

Основні переваги класифікаторів Naive Bayes полягають у тому, що вони є надійними для ізольованих точок шуму та невідповідних атрибутів, і вони обробляють відсутні значення, ігноруючи цей екземпляр під час обчислення ймовірності. Однак припущення про незалежність може не мати значення для деяких атрибутів, оскільки вони можуть бути співвіднесені. У цьому випадку звичайним підходом є використання Bayesian Networks. Байєсовські класифікатори можуть виявитися доцільними для середовищ, в яких знання переваг користувача повільно змінюється за часом, необхідного для побудови моделі, але не підходить для середовищ, в яких користувачі перевагу модель повинна бути оновлена швидко або часто. Вона також успішна в моделях на основі рекомендаційних систем, оскільки вона часто використовується для отримання моделі для систем рекомендацій на основі вмісту.

Методи завершення матриці: Суть методу завершення матриці - передбачити невідомі значення в матрицях користувацького елемента. Кореляція на основі K-найближчого сусіда є однією з основних методів, використаних у спільних системах рекомендацій щодо фільтрації. Вони значною мірою залежать від історичних рейтингових даних користувачів про предмети. У більшості випадків рейтинг матриці завжди дуже великий і рідкий через те, що користувачі

не оцінюють більшість предметів, представлених у матриці. Ця проблема завжди призводить до неможливості системи надавати користувачам надійні та точні рекомендації. На практиці для виготовлення матриць застосовуються різні варіанти низькосортних моделей, особливо до застосування у спільній фільтрації. Формально завдання матричного методу завершення полягає в оцінці записів матриці, коли підмножина, нових записів спостерігається певний набір низькорівневих матриць. Найбільш широко використовуваний алгоритм на практиці для відновлення M від частково спостережуваної матриці з використанням низькорівневого припущення являє собою мінімізацію змінної меншої площі (ALS), яка включає в себе оптимізацію U і V по черзі, щоб мінімізувати квадратну помилку над записаними записами, зберігаючи інші фіксуючі фактори. Candes and Recht запропонували використання методу завершення матриці в задачі Netflix як практичний приклад використання цієї техніки. В [5] використано метод SVD в алгоритмі OptSpace для вирішення проблеми матричного завершення. Результати їх експерименту показали, що SVD здатна забезпечити надійну первісну оцінку для охоплюючого підпростору, яке можна додатково вдосконалити градієнтним спуском на грассманівському різноманітті. Методи, засновані на техніці, вирішують проблему ранжування Основним недоліком технологій є те, що процес побудови моделі є обчислювальною вартістю, а потужність використання пам'яті є надзвичайно інтенсивною. Також вони не пом'якшують проблему "холодного запуску".

Плюси та мінуси методів спільного фільтрування полягають в наступному: спільна фільтрація має деякі переваги над CBF, оскільки вона може виконуватись у доменах, де нема багато вмісту, пов'язаного з елементами, і для яких комп'ютерний аналіз вмісту складний (наприклад, думки та ідеальні). Крім того, технологія CF має можливість надавати однозначні рекомендації, що означає, що він може рекомендувати елементи, релевантні для користувача, навіть якщо цей вміст не знаходиться в профілі користувача. Незважаючи на успіх методів CF, їх широке використання виявило деякі потенційні проблеми, такі як наступне.

Також існує проблема холодного запуску це стосується ситуації, коли рекомендатор не має адекватної інформації про користувача чи елемент, щоб зробити відповідні прогнози. Це одна з основних проблем, які знижують ефективність системи рекомендацій. Профіль такого нового користувача або елемента буде порожнім, оскільки він не оцінив жодного елемента; отже, його смак не відомий системі.

Крім цього є проблема шорсткості даних, це проблема, яка виникає внаслідок відсутності достатньої інформації, тобто коли користувачі оцінюють лише деякі з загальної кількості доступних у базі даних. Це завжди призводить до рідкісної матриці користувацького елемента, нездатності знайти успішних сусідів і, нарешті, створення слабких рекомендацій. Також, пріоритетність даних завжди призводить до проблем охоплення, тобто відсотком елементів у системі, на які можна внести рекомендації для.

Це ще одна проблема - масштабованість, пов'язана з алгоритмами рекомендацій, оскільки обчислення нормально зростає лінійно з кількістю користувачів і елементів. Метод рекомендацій, який є ефективним, коли кількість даних обмежена, може бути не в змозі створити задовільну кількість рекомендацій при збільшенні обсягу набору даних. Таким чином, важливо застосовувати методи рекомендацій, які можуть успішно розширюватися, оскільки кількість набору даних у базі даних зростає. Методи, що використовуються для вирішення проблеми масштабованості та прискорення генерації рекомендацій, базуються на методах зменшення розмірності, таких як метод декомпозиції сингулярного значення (SVD), який має можливість створювати надійні та ефективні рекомендації.

Синонімія - це тенденція дуже подібних предметів, що мають різні назви або записи. Більшість систем, що рекомендують, ускладнюють розмежування між тісно пов'язаними предметами, такими як різниця між, наприклад, одягом для дітей та дитячою ганчіркою. Спільна система фільтрів зазвичай не знаходить відповідності між двома термінами, щоб мати можливість обчислювати їх

схожість. Різні методи, такі як автоматичне розширення терміну, побудова тезаурусу та декомпозиція сингулярного значення (SVD), особливо латентний семантичний індексування, здатні вирішити проблему синонімії. Недоліком цих методів є те, що деякі додані терміни можуть мати різні значення від того, що призначено, що іноді призводить до швидкої деградації результатів рекомендацій. Рінго - це система, на основі якої працює користувач, що робить рекомендації музичних альбомів та виконавців. У Рінго, коли користувач спочатку входить до системи, користувачеві дається список 125 виконавців, щоб оцінювати їх відповідно до того, наскільки він любить слухати їх. Список складається з двох різних розділів. Перша сесія складається з найбільш часто оцінюваних виконавців, і це дає можливість активного користувача оцінювати виконавців, які інші оцінювали однаково, так що існує рівень подібності між профілями різних користувачів. Друга сесія генерується випадковим вибором елементів з усієї матриці користувацького елемента, завдяки чому всі виконавці та альбоми зрештою оцінюються в певний момент у початковій фазі рейтингування. GroupLens - система CF, яка базується на архітектурі клієнт / сервер; система рекомендує новини Usenet, що є великим обсягом служби обговорень в Інтернеті. Короткий термін служби Netnews, а також основна роздільна здатність матриць для оцінки є двома основними завданнями, що стоять перед цією системою. Користувачі та Netnews об'єднуються на основі існуючих груп новин у системі, а неявні рейтинги обчислюються шляхом вимірювання часу, коли користувачі трактують читання Netnews.

Amazon.com є прикладом механізму електронної комерції, що рекомендує використовувати масштабовані технології спільного фільтрування елементів до елемента, щоб рекомендувати Інтернет-продукти для різних користувачів. Обчислювальний алгоритм масштабує незалежно від кількості користувачів і елементів в межах бази даних. Amazon.com використовує явну методику збору інформації для отримання інформації від користувачів. Інтерфейс складається з наступних розділів, історії веб-перегляду, оцінює ці елементи та покращує ваші

рекомендації та ваш профіль. Система прогнозує інтерес користувачів на основі предметів, які він/вона оцінив. Потім система порівнює шаблон перегляду користувачів у системі та вирішує предмет, що цікавить, для того, щоб рекомендувати користувачеві. Amazon.com популяризувала особливість "людей, які купили цей товар, також купили ці предмети". Приклад інтерфейсу контекстної рекомендації до елемента Amazon.com наведено на рис. 1.3. (Інтерфейс рекомендаційного посібника для Amazon).



Рисунок 1.3 - Інтерфейс рекомендаційного посібника для Amazon

Довіра до RS визначається як кореляція між подібними перевагами до елементів, які зазвичай оцінюються або сподобались двом користувачам. Довіра покращує рекомендаційну систему шляхом поєднання подібності та довіри між користувачами. Тобто спосіб вибору сусідів змінюється завдяки впровадженню довіри, з тим щоб розвивати нові відносини між користувачами, з тим щоб він міг збільшити зв'язність та пом'якшувати проблеми ранжирування даних та холодного запуску, пов'язаних із традиційними методами спільного фільтрування. Деякі емпіричні дослідження, проведені Циглером та співавт. що кореляція існує між довірою та подібністю користувачів, коли мережа довіри до

спільноти пов'язана з певною програмою. Після проведених досліджень можна зробити висновок, що обчислювальні моделі довіри можуть служити належним засобом для доповнення або повністю заміни сучасної технології спільної фільтрації.

Різні показники довіри використовуються в рекомендаційних системах для вимірювання та обчислення вартості між користувачами в мережі. Ці показники мають два типи: локальні та глобальні показники довіри. Місцеві показники довіри використовували суб'єктивну думку активного користувача для прогнозування надійності інших користувачів з точки зору активного користувача. Значення довіри показує обсяг довіри, яку активний користувач надає іншому користувачеві. На підставі цієї методики різні користувачі довіряють активному користувачеві по-різному, і тому їх довірче значення відрізняється один від одного. Глобальні показники довіри представляє думку всієї спільноти щодо поточного користувача; отже, кожен користувач отримує лише одне значення, яке відображає її рівень надійності в спільноті. Коефіцієнти довіри в глобальних показниках довіри розраховуються шляхом сукупності всіх думок користувачів щодо поточного користувача. Репутація користувачів на ebay.com є прикладом використання глобальної довіри на веб-сайті онлайн-магазину. ebay.com обчислює репутацію користувача на основі кількості користувачів, які залишили позитивні, негативні або нейтральні відгуки щодо товарів, проданих поточним користувачем. Коли користувач не має певної думки стосовно іншого користувача, вона зазвичай спирається на ці агреговані показники довіри.

Гібридна фільтрація

Гібридна технологія фільтрації поєднує в собі різні методи рекомендації, щоб отримати кращу оптимізацію системи, щоб уникнути деяких обмежень і проблем чистих систем рекомендацій. Ідея гібридних технологій полягає в тому, що комбінація алгоритмів забезпечить більш точні та ефективні рекомендації, ніж єдиний алгоритм, оскільки недоліки одного алгоритму можуть бути подолані іншим алгоритмом. Використання кількох методів рекомендацій може

придушити слабкі сторони індивідуальної техніки в комбінованій моделі. Поєднання підходів може бути зроблено в будь-який з наступних способів: окрема реалізації алгоритмів і об'єднання результатів, використовуючи деякий контент на основі фільтрації в спільному підході, використовуючи деяку спільну фільтрацію в підході на основі змісту, створюючи єдину систему рекомендацій, яка приносить разом обидва підходи.

Зважена гібридизація поєднує результати різних рекомендацій для створення списків рекомендацій або прогнозування шляхом інтегрування балів за кожною з методів, що використовуються лінійною формулою. Прикладом зваженої гібридизованої системи рекомендацій є P-танго. Система складається з рекомендованого на основі вмісту та спільної роботи. Спочатку вони отримують рівну вагу, але ваги налаштовуються, оскільки прогнози підтверджуються або інакше. Перевага зваженого гібриду полягає в тому, що всі сильні сторони системи рекомендацій використовуються протягом процесу рекомендації просто.

Перемикання гібридизації полягає в тому, що система замінює одну з методів рекомендацій за евристикою, яка відображає спроможність рекомендувача робити хороший рейтинг. Гібрид перемикання має можливість уникати проблем, специфічних для одного методу, наприклад, нової проблеми користувача з рекомендацією на основі змісту, перейшовши на спільну систему рекомендацій. Вигода від цієї стратегії полягає в тому, що система чутлива до сильних і слабких сторін своїх рекомендованих. Основний недолік перемикання гібридів є те, що це зазвичай призводить до збільшення складності в процес рекомендації, оскільки критерій перемикання, який зазвичай збільшує кількість параметрів системи рекомендації має бути певно [6, 7]. Прикладом гібридного перемикача, що рекомендує, є DailyLearner який використовує як контент на основі і спільну роботу гібрида, в якому працює рекомендація на основі контенту, перш ніж спільні рекомендації в ситуації, коли система на основі змісту не може давати рекомендації з достатньою кількістю доказів.

Метод гібридизації каскаду застосовує ітеративний процес уточнення при побудові порядку переваги між різними елементами. Рекомендації однієї методики вдосконалюються іншою технікою рекомендації. Перша методика рекомендації видає грубий перелік рекомендацій, який у свою чергу вдосконалюється наступною методикою рекомендації. Технологія гібридизації дуже ефективна і толерантна до шуму через грубість до більш точного характеру ітерації. EntreeC є прикладом каскадного методу гібридизації, який використовував каскадний рекомендаційний користувач, що базується на знаннях та спільній роботі.

Змішані гібриди поєднують результати рекомендацій різних методів рекомендації одночасно, замість того, щоб мати лише одну рекомендацію для кожного елемента. Кожен елемент має кілька рекомендацій, пов'язаних з ним, з різних методів рекомендації. У змішаній гібридизації індивідуальні вистави не завжди впливають на загальну ефективність місцевого регіону. Прикладом системи рекомендацій в цій категорії, яка використовує змішану гібридизацію, є система PTV, яка рекомендує переглядати графік перегляду телепрограм для користувача, поєднуючи рекомендації з систем, що базуються на вмісті, та спільних систем, щоб створити графік. Profinder і PickAFlick також є прикладами змішаних гібридних систем.

Функції, розроблені спеціальною методикою рекомендації, потрапляють в іншу методику рекомендації. Наприклад, рейтинг подібних користувачів, який є ознакою спільної фільтрації, використовується в техніці рекомендації аргументації на основі випадків як однієї з функцій, що визначають подібність між елементами. Піппер - це приклад методу комбінації функцій, який використовував оцінки спільного фільтру в системі на основі вмісту як функцію рекомендації фільмів. Перевага цієї методики полягає в тому, що вона не завжди виключно залежить від спільних даних. Ця методика використовує рейтинги та іншу інформацію, надану попереднім рекомендаційником, і це вимагає додаткової функціональності від систем, що рекомендують. Наприклад, система

Libra робить рекомендації щодо вмісту на основі даних на Amazon.com за допомогою наївного класифікатора текстів Байєса. Гібриди з функціональністю збільшують переваги методів комбінації функцій, оскільки вони додають невелику кількість функцій до основного рекомендаційного.

Внутрішня модель, створена за допомогою однієї методики рекомендацій, використовується як вхідна для іншої. Зроблена модель завжди багатша в інформації, коли порівнюється з єдиним рейтингом. Гібриди мета-рівня здатні вирішити проблему ранжування методів спільної фільтрації, використовуючи всю модель, отриману за допомогою першої методики як вхід для другої методики. Прикладом методу мета рівня є праця, яка використовує миттєве навчання, щоб створити користувальницький профіль на основі вмісту, який потім порівнюється спільним способом.

РОЗДІЛ 2. ПРАКТИЧНЕ ВПРОВАДЖЕННЯ КОЛАБОРАТИВНОЇ ФІЛЬТРАЦІЇ

2.1 Розробка рекомендаційної системи на основі коефіцієнту кореляції Танімото із аналізом log-файлів сервера

Для прогнозування за допомогою алгоритму *«item to item»* потрібно використати предмет, на відмінну алгоритму «*user to user*», який прогнозує на основі вподобань користувачів. Предмет – це змінна, яка ми позначаємо як «0» - предмет не виконаний, або як «1» - предмет виконаний. Змінна предмету може являти певну дію, зроблену з товаром на сайті: найпопулярніші застосування предмету це оцінка товару, або його покупка. Тобто, якщо користувач купує товар, то його значення предмету = 1. Всі користувачі, які купити товар – об'єднують в когорти - групи людей, схожі між собою за певними ознаками, поведінкою, і вже новому користувачу, який буде відвідувати сторінку певного товару – йому запропонується товар, найбільш схожий на той, який він переглядає, на основі вподобань користувачів. Щоб зрозуміти яку ефективність та користь несе для себе колаборативна фільтрація потрібно з чимось порівняти результати. Якщо не існує жодної рекомендаційної системи, а потім реалізувати, то вийде, що результат порівнюється з нічим.

Тому спочатку вирішено реалізувати рекомендаційний блок, який базується на коефіцієнті кореляції Танімото із аналізом на основі log-файлів сервера, який в реалізації є більш простішим, враховуючи що в рамках цього точність в гіпотезі теж менша. Після цього, рекомендаційна система буде працювати на алгоритмі колаборативної фільтрації *«item to item»*. І тоді ми заміряємо результат.

Маючи необхідний об'єм інформації, реалізували блок на кінцевих сторінках оголошень під назвою «Більш вигідно» опираючись спочатку на log-файли не є проблемою. Маючи необхідний об'єм інформації, реалізували блок

на кінцевих сторінках оголошень під назвою «Більш вигідно» опираючись спочатку на коефіцієнт Танімото, який описує ступінь схожості двох множин:

$$k = \frac{c}{a+b-c}, \quad (2.1)$$

де k - коефіцієнт Танімото (число від 0 до 1), чим він ближче до 1, тим більше схожі множини; a - кількість елементів в першій множині; b - кількість елементів у другій множині; c - кількість спільних елементів в двох множинах.

Код програми на мові програмування php, що реалізує колаборативну фільтрацію на основі коефіцієнту Танімото, наведено в Додатку А. В цьому коді ми використовуємо функцію *calculateTanimotoCoefficient* для обчислення коефіцієнта кореляції Танімото між двома множинами (у нашому випадку, наборами предметів). Функція *collaborativeFiltering* реалізує алгоритм колаборативної фільтрації, використовуючи цей коефіцієнт для обчислення подібності між користувачами і генерації рекомендацій.

У прикладі ми використовуємо масив $*ratings*, який містить рейтинги користувачів для певних предметів. Ми викликаємо функцію *collaborativeFiltering* з користувачем *'user1'* і масивом $*ratings*, щоб отримати рекомендації для цього користувача. Рекомендації виводяться на екран разом із відповідними рейтингами. Пояснимо його більш детально.

1. Функція *calculateTanimotoCoefficient* обчислює коефіцієнт кореляції Танімото між двома множинами (наборами предметів). Вона приймає два аргументи - $*setA* і $*setB*, які є масивами з ключами, що представляють предмети відповідно до кожного набору.

```
// Функція для обчислення коефіцієнта кореляції Танімото
function calculateTanimotoCoefficient($setA, $setB)
{
   $intersection = array_intersect($setA, $setB);
   $union = array_merge($setA, $setB);
   $coefficient = count($intersection) / count($union);
   return $coefficient;
}
```

В першому рядку функції ми використовуємо функцію *array_*intersect для знаходження спільних елементів між $*setA* і $*setB*. Результат зберігається в масиві $*intersection*.

У другому рядку ми об'єднуємо масиви $*setA* і $*setB* за допомогою функції array*_merge*. Результат зберігається в масиві $*union*.

У третьому рядку ми обчислюємо коефіцієнт кореляції Танімото за формулою, поділивши розмір $*intersection* на розмір $*union*. Результат повертається як значення коефіцієнта кореляції.

2. Функція *collaborativeFiltering* реалізує алгоритм колаборативної фільтрації з використанням коефіцієнта кореляції Танімото. Вона приймає два аргументи - $*user* (ідентифікатор користувача) і $*ratings* (масив з рейтингами користувачів для предметів).

```
// Функція для колаборативної фільтрації з використанням коефіцієнта кореляції Танімото
function collaborativeFiltering($user, $ratings)
{
  $similarities = array();

  // Обчислюємо подібність між користувачами
  foreach ($ratings as $otherUser => $otherRatings) {
    if ($otherUser !== $user) {
      $userRatings = $ratings[$user];

      $similarity = calculateTanimotoCoefficient(
        array_keys($userRatings),
        array_keys($otherRatings)
      );

      $similarities[$otherUser] = $similarity;
    }
  }

  arsort($similarities); // Сортуємо подібності в порядку спадання

  $recommendations = array();
```

```
  // Обчислюємо рекомендації для користувача
  foreach ($similarities as $otherUser => $similarity) {
    $otherRatings = $ratings[$otherUser];

    foreach ($otherRatings as $item => $rating) {
      if (!isset($ratings[$user][$item])) {
        if (!isset($recommendations[$item])) {
          $recommendations[$item] = 0;
        }

        $recommendations[$item] += $rating * $similarity;
      }
    }
  }

  arsort($recommendations); // Сортуємо рекомендації в порядку спадання

  return $recommendations;
}
```

У першому рядку функції ми створюємо порожній масив $*similarities*, в якому будемо зберігати подібність між користувачами.

У наступному блоку *foreach* ми обчислюємо подібність між користувачем $*user* і кожним іншим користувачем з $*ratings*. Ми перебираємо кожен інший користувач за допомогою циклу *foreach* і обчислюємо подібність за допомогою функції *calculateTanimotoCoefficient*. Результати зберігаються в масиві $*similarities* з ключами, що відповідають іншим користувачам.

У рядку *arsort*($*similarities*) ми сортуємо масив $*similarities* в порядку спадання подібності. Це потрібно для того, щоб отримати найбільш подібних користувачів спочатку.

У наступному блоку *foreach* ми обчислюємо рекомендації для користувача $*user*. Ми перебираємо кожного іншого користувача в порядку спадання подібності і перебираємо всі їхні рейтинги. Якщо предмет не має рейтингу в $*ratings* користувача $*user*, ми додаємо його до масиву $*recommendations* з вагою, яка обчислюється як добуток рейтингу предмета і подібності між користувачами.

Нарешті, ми сортуємо масив $*recommendations* в порядку спадання, щоб отримати найкращі рекомендації.

3. У прикладі використання ми створюємо масив $*ratings*, який містить рейтинги користувачів для певних предметів. У цьому прикладі ми маємо рейтинги для трьох користувачів - *'user1'*, *'user2'* і *'user3'*. Кожен користувач має рейтинги для чотирьох предметів - *'item1'*, *'item2'*, *'item3'* і *'item4'*.

```
// Приклад використання

$ratings = array(
   'user1' => array(
      'item1' => 5,
      'item2' => 3,
      'item3' => 4,
      'item4' => 2
   ),
   'user2' => array(
      'item1' => 2,
      'item2' => 5,
      'item3' => 1,
      'item4' => 4
   ),
   'user3' => array(
      'item1' => 4,
      'item2' => 2,
      'item3' => 5,
      'item4' => 3
   )
);

$user = 'user1';

$recommendations = collaborativeFiltering($user, $ratings);

echo "Рекомендації для користувача $user:\n";
foreach ($recommendations as $item => $score) {
   echo "$item (Рейтинг: $score)\n";
}
```

4. Ми викликаємо функцію *collaborativeFiltering* з користувачем *'user1'* і масивом $ratings, щоб отримати рекомендації для цього користувача.

5. Нарешті, ми виводимо рекомендації на екран разом із відповідними рейтингами.

Множина приймалась за сторінки, які були відвідані користувачами сайту за певний період, які були взяти із log-файлів маючи доступ до FTP-сервера домену Automoto.ua. Список логів пишеться не в браузер, а в спеціальні файли на сервері (access.log, error.log), і їх розташування залежить від відповідних директив в налаштуваннях веб сервера (nginx/apache і ін.) віртуального хоста-сервера. Якщо не було би FTP доступу до сайту, ми би їх не побачили.

Далі вивантажено програмним шляхом інформацію, де кожному користувачу присвоюємо унікальний ID та фіксувалось сторінки, які користувач переглядає (рис. 2.1). Окрім цього фіксувалась дата та час. Колонка «is visit» відповідає за те, з якої сторінки користувач починає сесію на сайті: якщо 1 – це сторінка входу на сайт, якщо 0 – сторінка відвідана в процесі сесії. Колонка «user id» відповідає за користувача із присвоєним ID. Колонка «url» - сторінка сайту, відвідана користувачем (на основі log-файлів). Примітка: до уваги брались тільки кінцеві сторінки оголошень, типу: https://automoto.ua/uk/Mercedes-Benz-GLE-Class-2017-Khmelnytskyi-18044982.html

	A	B	C	D
1	updated_at	is_visit	user_id	url
2	2017-08-23 08:22:12	1	47149	https://automoto.ua/interesting/yak_oskarzhiti_shtraf_politsiyi_lazivki_pro_yaki_malo_hto_znae-880.html
3	2017-08-23 14:12:34	0	47149	https://automoto.ua/interesting/yak_oskarzhiti_shtraf_politsiyi_lazivki_pro_yaki_malo_hto_znae-880.html
4	2017-08-23 11:31:48	1	51437	https://automoto.ua/auto-review/Nissan/350Z
5	2017-08-24 00:08:55	1	158493	https://automoto.ua/Sea-Ray-260-Sundancer-2004-Kiev-14177337.html
6	2017-08-24 00:26:40	0	158493	https://automoto.ua/Sea-Ray-260-Sundancer-2004-Kiev-14177337.html
7	2017-08-22 10:28:51	1	163895	https://automoto.ua/car/BMW/X5/not-customed
8	2017-08-27 14:55:55	1	196878	https://automoto.ua/spectehnika/Minitraktor
9	2017-08-27 14:56:22	0	196878	https://automoto.ua/Bulat-150-2017-Vinnitsa-16615094.html
10	2017-08-27 14:56:23	0	196878	https://automoto.ua/spectehnika/Minitraktor
11	2017-08-27 15:07:02	0	196878	https://automoto.ua/uk/spectehnika/Minitraktor
12	2017-08-24 17:20:29	0	203140	https://automoto.ua/MTZ-622-Belarus-2016-Mankovka-16456388.html
13	2017-08-24 17:20:47	1	203140	https://automoto.ua/spectehnika/MTZ/622-Belarus
14	2017-08-24 17:22:21	0	203140	https://automoto.ua/MTZ-622-Belarus-2016-Kiev-16363332.html
15	2017-08-24 17:22:22	0	203140	https://automoto.ua/car/MTZ/622-Belarus/customed
16	2017-08-27 08:50:57	1	206567	https://automoto.ua/city/Mariupol
17	2017-08-27 09:42:58	0	208413	https://automoto.ua/car/Tesla/Model-X
18	2017-08-25 23:02:27	1	215574	https://automoto.ua/car/Renault/Master/used
19	2017-08-25 23:03:07	0	215574	https://automoto.ua/Renault-Master-2000-Ternopol-16708169.html
20	2017-08-25 23:03:14	0	215574	https://automoto.ua/car/Renault/Master/used
21	2017-08-25 23:04:29	0	215574	https://automoto.ua/Renault-Master-2004-Kremenchug-16693190.html
22	2017-08-25 23:04:37	0	215574	https://automoto.ua/car/Renault/Master/used
23	2017-08-23 15:39:49	1	218764	https://automoto.ua/uk/q/Mototsikl-Volinska-424128.html
24	2017-08-27 12:17:55	1	221810	https://automoto.ua/news/politseyskie-mogut-ostanovit-avtomobil-na-doroge-tolko-v-9-sluchayah-6411.html
25	2017-08-22 12:48:39	0	231291	https://automoto.ua/uk/moto/IZH/Planeta-4
26	2017-08-22 12:48:43	0	231291	https://automoto.ua/uk/q/Mototsikl-IZH-Planeta-4-Ternopilska-do-4000-7460366.html
27	2017-08-22 12:49:37	0	231291	https://automoto.ua/uk/IZH-Planeta-4-1984-Terebovlya-16380055.html
28	2017-08-22 12:49:41	0	231291	https://automoto.ua/uk/IZH-Planeta-4-1985-Dunaevtsyi-16217579.html
29	2017-08-23 09:56:36	1	233141	https://automoto.ua/car/Europe/Polsha/Mercedes-Benz
30	2017-08-24 14:34:08	1	243822	https://automoto.ua/car/Europe/Germany

Рисунок 2.1 – Інформація про користувачів із присвоєним унікальним ID

Вивантаживши статистику тільки за тиждень ми отримали 776542 рядка даних, яких достатньо для досліджень (рис. 2.2).



776535	2017-08-29 10:04:11	1	40735411	https://automoto.ua/interesting/prava_dlya_es_chi_zobovyazuie_asotsiatsiya_minyati_posvidchennya_vodiya-1593.html
776536	2017-08-29 10:04:12	0	40735413	https://automoto.ua/car/Europe/Polsha/BMW
776537	2017-08-29 10:04:19	1	40735419	https://automoto.ua/city/Galich/car
776538	2017-08-29 10:04:31	1	40735428	https://automoto.ua/LuAZ-967-1978-Ivanofrankovsk-7862763.html
776539	2017-08-29 10:04:31	1	40735429	https://automoto.ua/interesting/edu_takoy_300_kmch_vdrug_otkuda_ne_vozmis-1507.html
776540	2017-08-29 10:04:34	1	40735432	https://automoto.ua/uk/spectehnika/John-Deere/Ternopol
776541	2017-08-29 10:04:42	1	40735438	https://automoto.ua/interesting/yak_oskarzhiti_shtraf_politsiyi_lazivki_pro_yaki_malo_hto_znae-880.html
776542				

Рисунок 2.2 – Загальна кількість інформації за тиждень

Маючи всю необхідну інформацію про відвідування сторінок користувачами, аналізуючи log-файли сервера, успішно реалізували блок «Більш вигідно» на десктопній та мобільній версії сайту.

2.2. Розробка рекомендаційної системи на колаборативної фільтрації методу «item to item» із врахуванням релевантності

Після реалізації блоку «Більш вигідно» на сторінках оголошень ми приступили до реалізації колаборативної фільтрації. Інтуїтивно зрозуміло, що для рекомендації користувачеві №1 будь-якого продукту, вибирати потрібно з продуктів, які подобаються якимось користувачам 2-3-4-n., Які найбільш схожі за своїми оцінками на користувача №1. Отримати чисельне вираження цієї «схожості» користувачів можемо за допомогою векторів в N-вимірному просторі продуктів, а порівнювати вектори ми можемо наступними відомими методами:

- Косинусна міра;
- Коефіцієнт кореляції Пірсона;
- Евклідова відстань;
- Коефіцієнт Танімото;
- Манхеттенська відстань тощо.

Далеко не завжди у користувачів є можливість виставляти оцінки предметів. Тобто для колаборативної фільтрації можуть бути доступні лише двійкові дані (купував користувач предмет чи ні). Проте даний алгоритм не враховує

релевантність (відповідність до пошукового запиту) сторінки для користувача. Алгоритм реагує тільки на оцінку товару або на його покупку. По статистиці в світі – відсоток конверсії на сайтах становить всього 2-3%. Тобто, рекомендаційна система обробляє тільки 2-3% даних для видачі інформації.

Тому в цій роботі розроблений алгоритм, який враховує релевантність сторінки для 100% користувачів, і на основі цього рекомендує. Суть алгоритму полягає в тому, що замість покупки чи оцінки товару, ми, присвоюючи кожному користувачу (User) – ідентифікатор (ID), запускаємо таймер при перегляді URL-сторінки оголошення (карточка товару). Якщо при цьому, користувач перебуває на сторінці більше 15 секунд, то ця сторінки вважається релевантною, якщо менше, то ні. (t>15 сек. = 1, t<15 сек. = 0).

Спочатку даний. алгоритм розраховував подібність предметів як косинус між векторами покупок в матриці користувачів та предметів (2-3%).

Розглянемо роботу алгоритму на прикладі статистики, де за предмет – приймається час перегляду URL-сторінки оголошення в табл. 2.1.

2.1. Приклад матриці користувачів та предметів

Користувач	Предмет 1	Предмет 2	Предмет 3
Віктор	1	0	1
Микола	0	1	1
Марія	0	1	0

В цьому випадку косинус між «Предмет 1» і «Предмет 2» розраховується так:

$$\frac{(1,0,0)\bullet(0,1,1)}{||(1,0,0)||\bullet||(0,1,1)||} = 0; \quad (2.2)$$

між «Предмет 1» і «Предмет 3»:

$$\frac{(1,0,0)\bullet(1,1,0)}{||(1,0,0)||\bullet||(1,1,0)||} = \frac{1}{\sqrt{2}} \approx 0{,}71; \quad (2.3)$$

і між «Предмет 2» і «Предмет 3»:

$$\frac{(0,1,1)\bullet(1,1,0)}{||(0,1,1)||\bullet||(1,1,0)||} = \frac{1}{2} = 0{,}5\ . \qquad (2.4)$$

Таким чином, користувач, що знаходиться на сторінці опису «Предмету 1», отримає «Предмет 3» в якості рекомендації; на сторінці «Предмету 2» - «Предмет 3» і на сторінці «Предмету 3» - «Предмет 1» (і потім «Предмет 2»). В даному алгоритмі використовується один коефіцієнт на кожну пару предметів (косинус), на підставі якого і створюються рекомендації. Тобто для n предметів буде потрібно розрахувати і зберегти n (n-1)/2 косинусів.

Отже, робота алгоритму буде проходити в декілька етапів, оскільки спочатку потрібно визначати значення «предмету», яке буде дорівнювати переглянутій сторінці URL оголошення, на якій користувач перебуває не менше 15 секунд, цим самим оцінюючи релевантність сторінки до пошуку користувача. Тільки після визначення таких відбуваються подальші обрахунки із знаходження косинусної міри, при якій береться до уваги 100% інформації про інтереси користувачів, цим самим покращуючи алгоритм Item to Item методу колаборативної фільтрації для розробки рекомендаційних систем шляхом оцінки релевантності.

2.3 Результати досліджень рекомендаційного блоку «Більш вигідно» на основі коефіцієнту кореляцій Танімото із аналізом log-файлів сервера та роботи алгоритму «item to item» колаборативної фільтрації

За допомогою незалежного визнаного сервісу web-аналітики Google Analytics ми порівняли взаємодію (клік) з блоком «Більш вигідно» під час роботи рекомендацій на log-файлів сервера (жовтий графік) та на основі колаборативної фільтрації (синій графік) (рис. 2.3.).

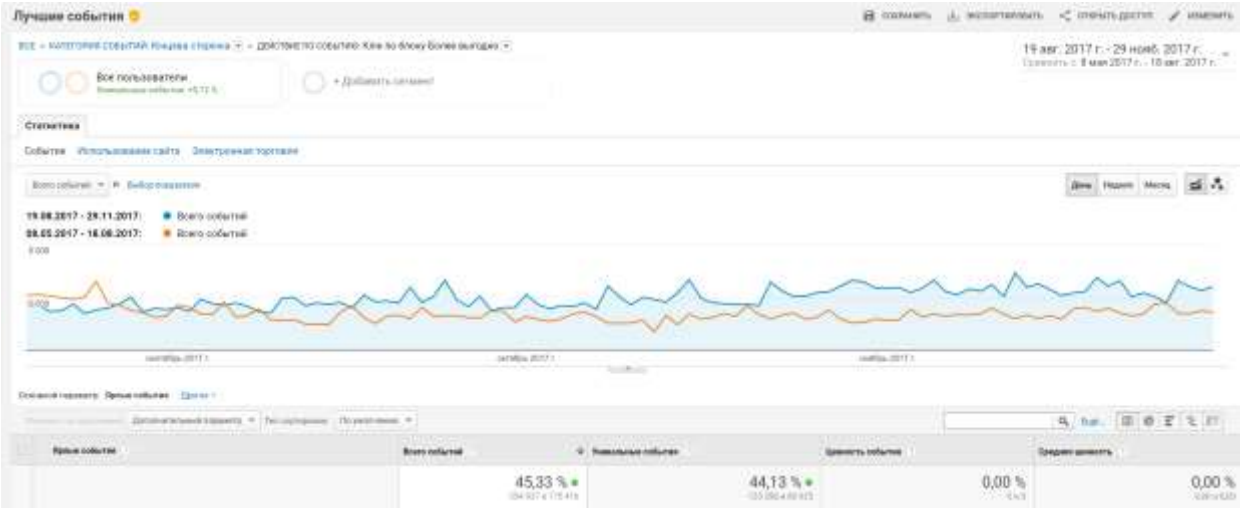


Рисунок 2.3 – Результати взаємодії з блоком «Більш вигідно»

Як можна побачити із графіків, взаємодія із блоком при роботі колаборативної фільтрації вища на 45,33%, що свідчить про її високу точність та користь для користувачів.

РОЗДІЛ 3. АНАЛІЗ ЕФЕКТИВНОСТІ НА ОСНОВІ СТАТИСТИЧНОЇ ОЦІНКИ В РЕАЛІЗОВАНІЙ РЕКОМЕНДАЦІЙНІЙ СИСТЕМІ

3.1. Вибір показників для аналізу ефективності на основі статистичної оцінки

Після реалізації рекомендаційної системи спочатку на основі коефіцієнту кореляції Танімото на основі аналізу log-файлів сервера та методу «item to item» колаборативної фільтрації розпочатий аналіз ефективності статистичної оцінки даних досліджень.

Із попереднього розділу зрозуміло, що для формування рекомендаційної системи потрібно опиратись на дві основні множини даних: предмети (в нашому випадку це переглянуті сторінки) та користувачі.

Оскільки розрахунок статистичних показників для переглянутих сторінок не є ефективним, тому весь подальший розрахунок буде спрямований на оцінку користувачів на сайті.

Проте в незалежному визнаному сервісі Google Analytics ми бачимо загальний показник користувачів за вибраних порівняльних періоди (рис. 3.1).

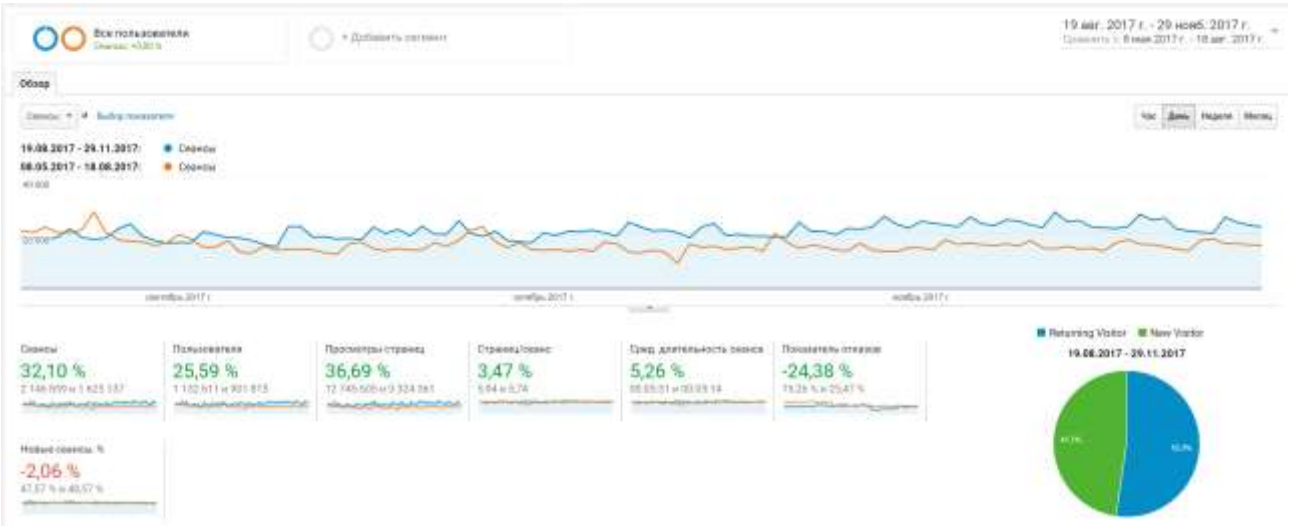


Рисунок 3.1 – Показники аудиторії Automoto.ua

3.2 Парсинг даних про користувачів через API Google Analytics

Оскільки базовий інструмент Google Analytics не дає можливість викачати статистику по кожному дню, прийшли до висновку, що потрібно здійснити парсинг даних із Google Analytics із менеджмент-протоколів через API в Google Таблиці для подальшого аналізу. Для цього налаштували плагін для викачки даних (рис. 3.2).

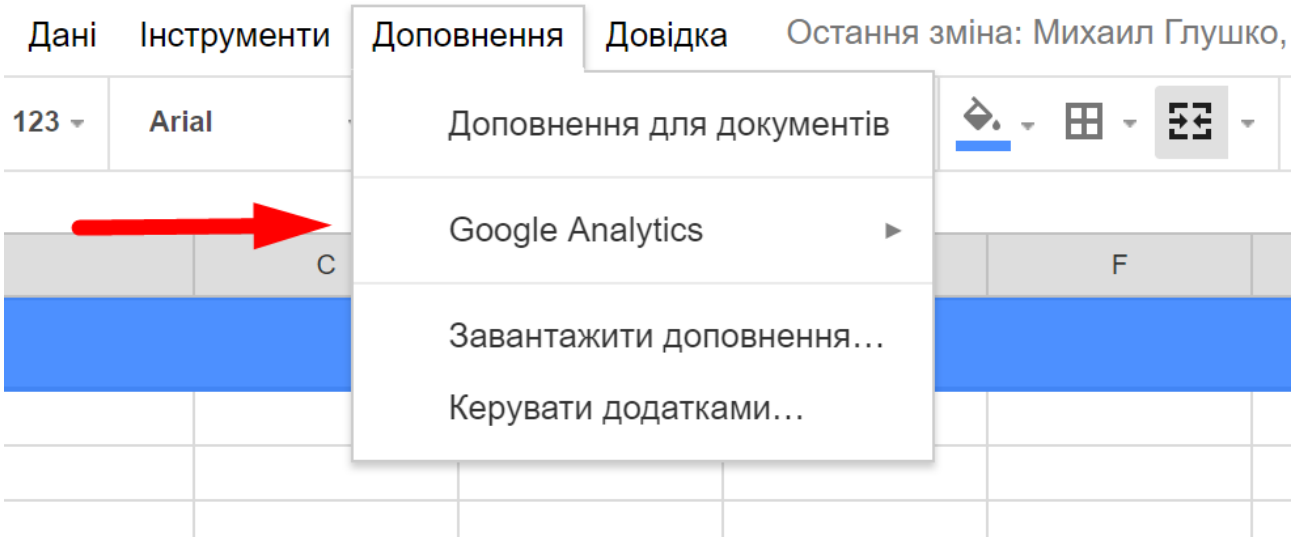


Рисунок 3.2 – Плагін Google Analytics в Google Таблицях

Далі, після запуску плагіна у вікні налаштування вписали назву, вибрали необхідний аккаунт Google Analytics – Automoto.ua та обрали показник користувачів в розрізі із індексом днів (рис. 3.3).

Після налаштування звіту для двох ідентичних по кількості днів періодів (період роботи коефіцієнту Танімото та колаборативної фільтрації) отримали наступний функціонал подальших налаштувань (рис 3.4). В колонки «Start Date» та «End Date» вписали відповідні часові періоди роботи рекомендаційного блоку.

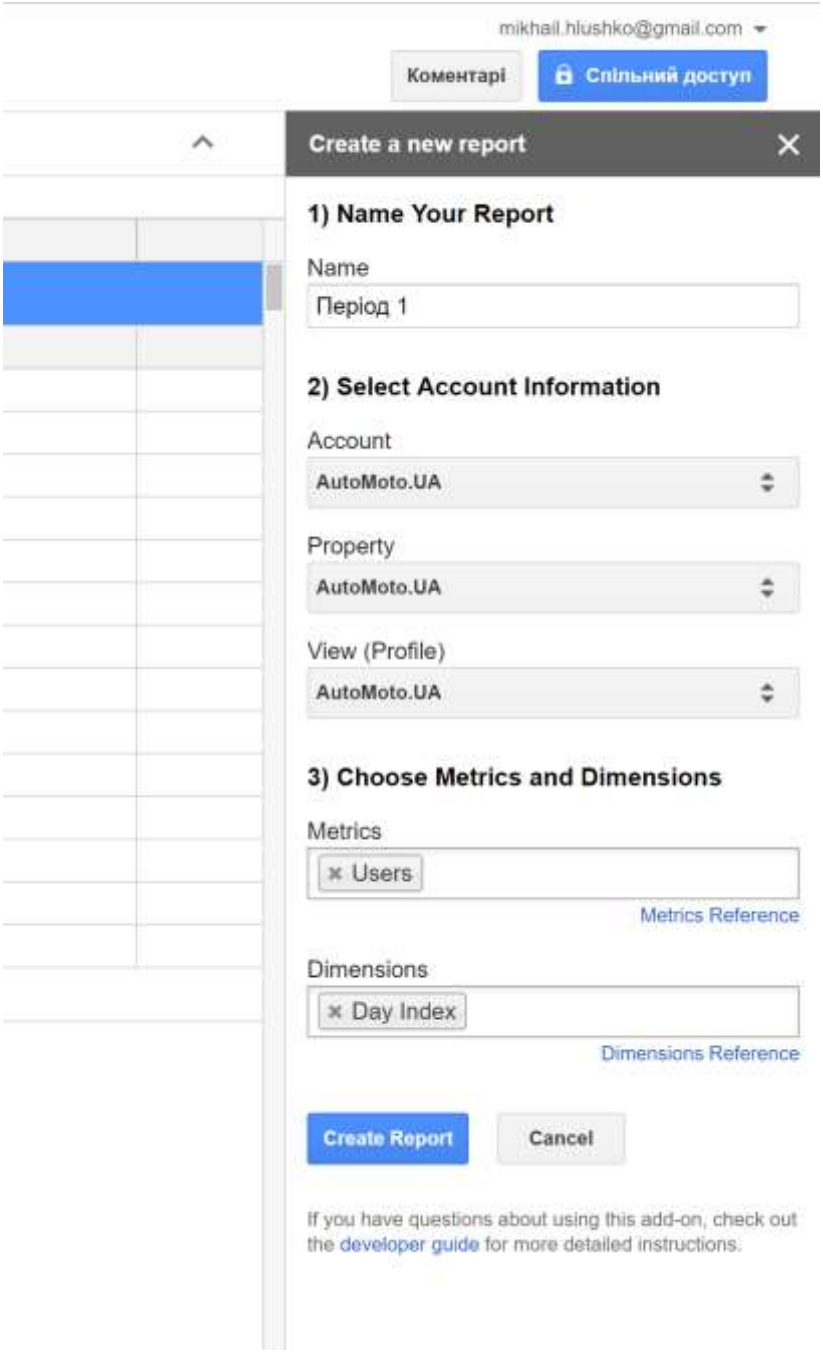
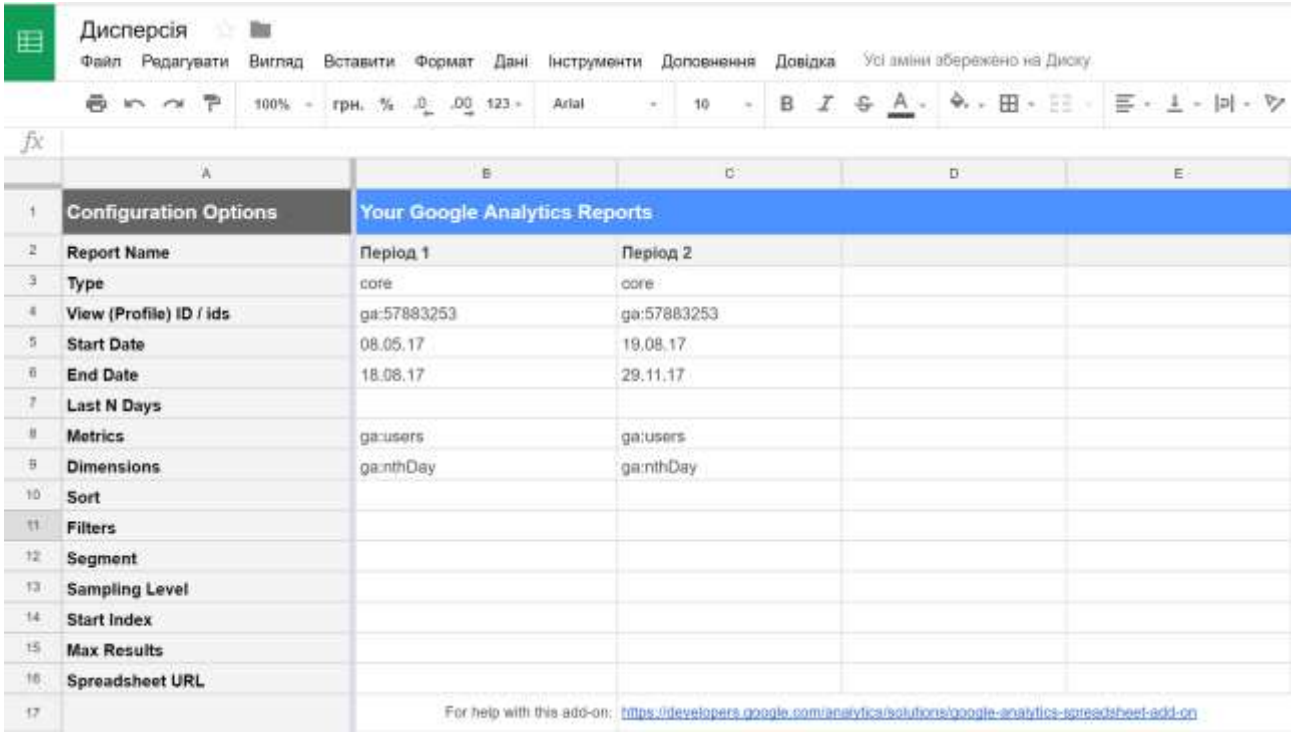


Рисунок 3.3 – Налаштування звіту в плагіні Google Analytics



	A	B	C	D	E
1	Configuration Options	Your Google Analytics Reports			
2	Report Name	Період 1	Період 2		
3	Type	core	core		
4	View (Profile) ID / ids	ga:57883253	ga:57883253		
5	Start Date	08.05.17	19.08.17		
6	End Date	18.08.17	29.11.17		
7	Last N Days				
8	Metrics	ga:users	ga:users		
9	Dimensions	ga:nthDay	ga:nthDay		
10	Sort				
11	Filters				
12	Segment				
13	Sampling Level				
14	Start Index				
15	Max Results				
16	Spreadsheet URL				
17		For help with this add-on: https://developers.google.com/analytics/solutions/google-analytics-spreadsheet-add-on			

Рисунок 3.4 – Функціонал налаштування вигрузки даних

Після запуску всіх налаштувань плагін Google Analytics окремо вивантажив дані по користувачам за перший період та за другий відповідно до встановлених часових періодів для подальших статистичних обрахунків.

В звітах видно, що по кожному дню, починаючи від 0 дня, яким є встановлене 8 травня 2017 року ми отримали статистику по відвідуваності користувачів по дням.

Дана статистика дозволить провести подальші дослідження, проте хочеться зазначити, що таким шляхом можна отримувати будь-які статистичні дані про веб-ресурс та їх відвідувачів.

Кількість різних даних сягає близько 450 видів із можливість фільтрувати та сегментувати необхідну інформацію.

1		
2	**Last Run On**	2017-12-26 23:33:56
3	**View (Profile) Name**	AutoMoto.UA
4	**Total Results Found**	103
5	**Total Results Returned**	103
6	**Contains Sampled Data**	No
7		
8		
9		
10	**Totals For All Results**	
11		**ga:users**
12		1379499
13		
14	**Results Breakdown**	
15	**ga:nthDay**	**ga:users**
16	0	17224
17	1	16997
18	2	18680
19	3	17173
20	4	16893
21	5	17570
22	6	23052
23	7	16999
24	8	14714
25	9	14416
26	10	14206
27	11	13244
28	12	14226
29	13	16402
30	14	15029
31	15	12690
32	16	12656
33	17	14325
34	18	11130
35	19	10802
36	20	12873

Report Configuration | Період 1 | Період 2

Рисунок 3.5 – Дані про кількість користувачів по днях за 1 період

За результати роботи різних алгоритмів, ми вивантажили два періоди по 103 дні: (із врахуванням що 1 день – має число 0).

105	89	11463
106	90	13811
107	91	15030
108	92	13571
109	93	13527
110	94	12722
111	95	12040
112	96	11485
113	97	14678
114	98	14884
115	99	13784
116	100	13624
117	101	13295
118	102	13213

Report Configuration | Період 1 | Період 2

Рисунок 3.6 – Загальна кількість днів збору даних

За другий період, який почався 19 серпня 2017 року вивантажено аналогічні дані із тривалістю в 103 дні (рис. 3.7).

1		
2	**Last Run On**	2017-12-27 00:41:28
3	**View (Profile) Name**	AutoMoto UA
4	**Total Results Found**	103
5	**Total Results Returned**	103
6	**Contains Sampled Data**	No
7		
8		
9		
10	**Totals For All Results**	
11		**ga:users**
12		1827899
13		
14	**Results Breakdown**	
15	**ga:nthDay**	**ga:users**
16	0	15443
17	1	15195
18	2	15095
19	3	15848
20	4	18570
21	5	16042
22	6	15348
23	7	15864
24	8	18443
25	9	19934
26	10	16346
27	11	14814
28	12	14112
29	13	14233
30	14	14241
31	15	17256
32	16	16425
33	17	15750
34	18	15568
35	19	14878
36	20	13589

Report Configuration | Період 1 | Період 2

Рисунок 3.7 - Дані про кількість користувачів по днях за 2 період

3.3 Статистичні обрахунки вивантажених даних із Google Analytics

Розрахунки почали із дисперсії. Це дуже важливий показник, який активно використовується в різних методах статистичного аналізу (перевірка гіпотез, аналіз причинно-наслідкових зв'язків та ін.). Як і середнє лінійне відхилення, дисперсія також відображає міру розкиду даних навколо середньої величини.

Формула дисперсії в теорії ймовірностей має вигляд:

$$D(X) = \sigma^2 = M[X - M(X)]. \quad (3.1)$$

Тобто дисперсія - це математичне очікування відхилень від математичного очікування.

На практиці при аналізі вибірок математичне очікування, як правило, не відомо. Тому замість нього використовують оцінку - середнє арифметичне.

$$s^2 = \frac{\sum_{i=1}^{n}(X-\bar{X})^2}{n}, \quad (3.2)$$

де s^2 - вибіркова дисперсія, розрахована за даними спостережень; X - окремі значення; $\bar{X}$- середнє арифметичне за вибіркою.

Далі, аби повернути дисперсію в реальність, тобто використовувати в більш приземлених цілей, з неї витягають квадратний корінь. Виходить так зване середньоквадратичне відхилення (СКВ). Зустрічаються назви «стандартне відхилення» або «сигма» (від назви грецької букви). Формула стандартного відхилення має вигляд:

$$\sigma = \sqrt{M[X - M(X)]^2} \quad (3.3)$$

Для отримання цього показника по вибірці використовують формулу:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(X-\bar{X})^2}{n}} \quad (3.4)$$

Средньоквадратичне відхилення, також характеризує міру розсіювання даних, але тепер (на відміну від дисперсії) його можна порівнювати з вихідними даними, так як одиниці виміру у них однакові (це випливає з формули розрахунку). Із середньоквадратичним відхиленням вже можна працювати безпосередньо, тому що властивості даного показника добре вивчені і відомі. Середньоквадратичне відхилення, як міра невизначеності, також бере участь у

багатьох статистичних розрахунках. З її допомогою встановлюють ступінь точності різних оцінок і прогнозів. Якщо варіація дуже велика, то стандартне відхилення теж вийде великим, отже, і прогноз буде неточним, що виразиться, приміром, в дуже широких довірчих інтервалах.

Тому, середнє квадратичне відхилення дає абсолютну оцінку міри розсіювання. Тому щоб зрозуміти, наскільки розсіювання велике щодо самих значень (тобто незалежно від їх масштабу), потрібно відносний показник. Такий показник називається коефіцієнтом варіації і розраховується за наступною формулою:

$$V = \frac{S}{\bar{X}} \cdot 100\%. \quad (3.5)$$

Коефіцієнт варіації - це показник, що відображає розкид значень відносно середнього (відношення стандартного відхилення до середнього значення). Коефіцієнт варіації вимірюється у відсотках і відображає однорідність тимчасового ряду.

Коефіцієнт варіації - це відмінний показник, який допоможе вам у підготовці даних для прогнозу. Коефіцієнт варіації - індикатор, який допоможе вам виділити ряди, на які варто звернути увагу перед розрахунком прогнозу і очистити дані від випадкових факторів. Якщо коефіцієнт дорівнює 0%, то ряд абсолютно однорідний, тобто всі значення між собою рівні. Якщо коефіцієнт варіації більше 33%, то за класичною теорією ряд вважається неоднорідним, тобто великий розкид даних щодо середнього значення. Далі, маючи необхідний об’єм даних обчислимо дисперсію за допомогою функцій в Google Таблиці – VARP (рис. 3.8).



fx =VARP(B16:B118)

	A	B	C	D	E
83	67	12049			
84	68	11467			
85	69	13777			
86	70	11916			
87	71	11504			
88	72	11405			
89	73	12539			
90	74	12559			
91	75	11914			
92	76	14880			
93	77	13552			
94	78	14001			
95	79	13507			
96	80	13136			
97	81	13570			
98	82	12981			
99	83	14614			
100	84	12413			
101	85	12197			
102	86	12729			
103	87	12040			
104	88	12363			
105	89	11463			
106	90	13811			
107	91	15030			
108	92	13571			
109	93	13527			
110	94	12722			
111	95	12040			
112	96	11485			
113	97	14678			
114	98	14884			
115	99	13784			
116	100	13624			
117	101	13295			
118	102	13213			
119	Дисперсія	4604567,826			

Рисунок 3.8 – Обчислення дисперсії в Google Таблицях по першому періоду

За формулою дисперсія по першому періоду дорівнює 4604567,826. Далі, аналогічно порахуємо дисперсію для другого періоду (рис. 3.9).



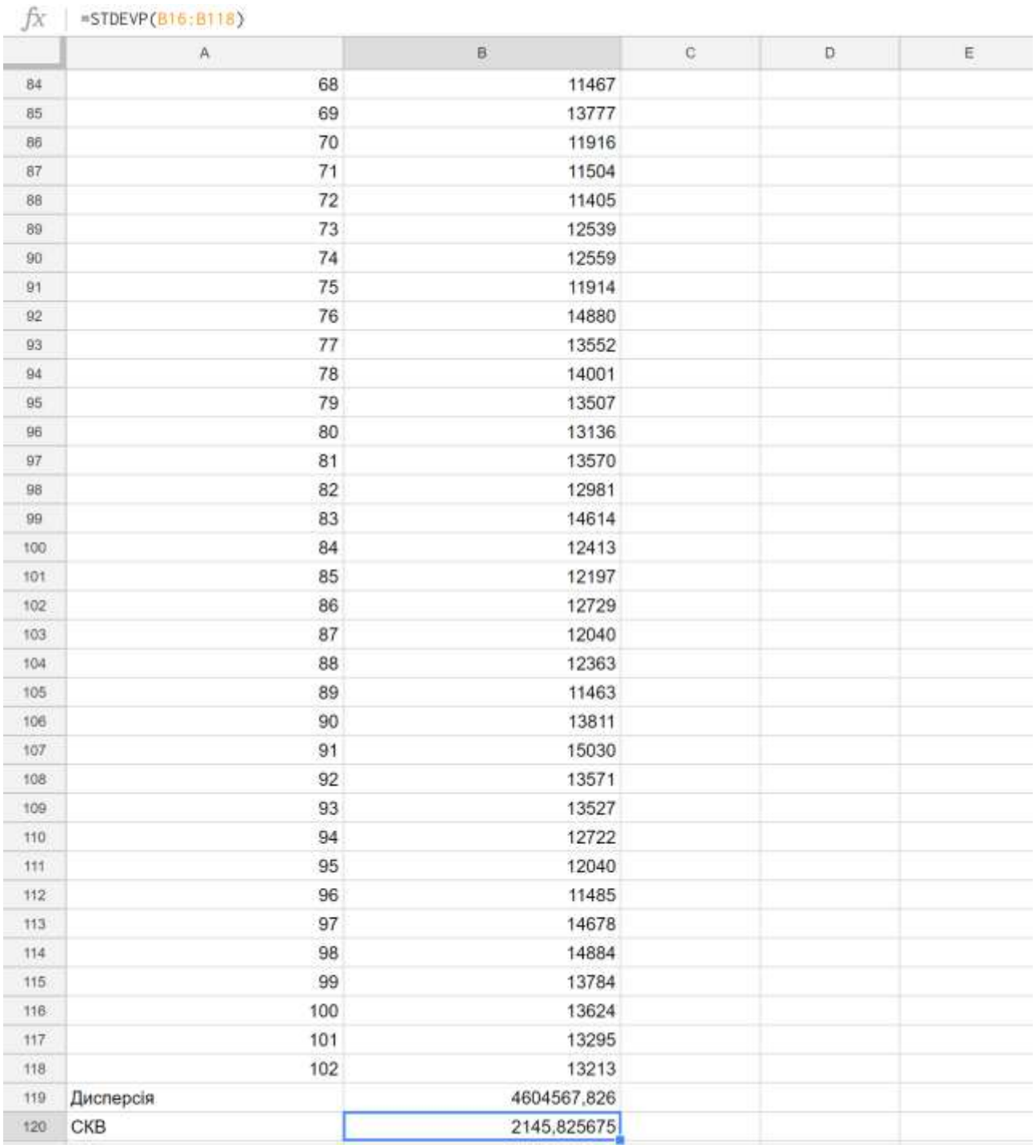
fx =VARP(B16:B118)

	A	B	C	D	E
83	67	16563			
84	68	18426			
85	69	18279			
86	70	18556			
87	71	21721			
88	72	19658			
89	73	18532			
90	74	20572			
91	75	20093			
92	76	19605			
93	77	18752			
94	78	21676			
95	79	19902			
96	80	19416			
97	81	21059			
98	82	20888			
99	83	19368			
100	84	18560			
101	85	23230			
102	86	20463			
103	87	21001			
104	88	19037			
105	89	18808			
106	90	18575			
107	91	19078			
108	92	22736			
109	93	20959			
110	94	21454			
111	95	18293			
112	96	17656			
113	97	17287			
114	98	17094			
115	99	21668			
116	100	20229			
117	101	19394			
118	102	18830			
119	Дисперсія	4725188,688			

Рисунок 3.9 - Обчислення дисперсії в Google Таблицях по другому періоду

За формулою дисперсія по другому періоду дорівнює 4725188,688.

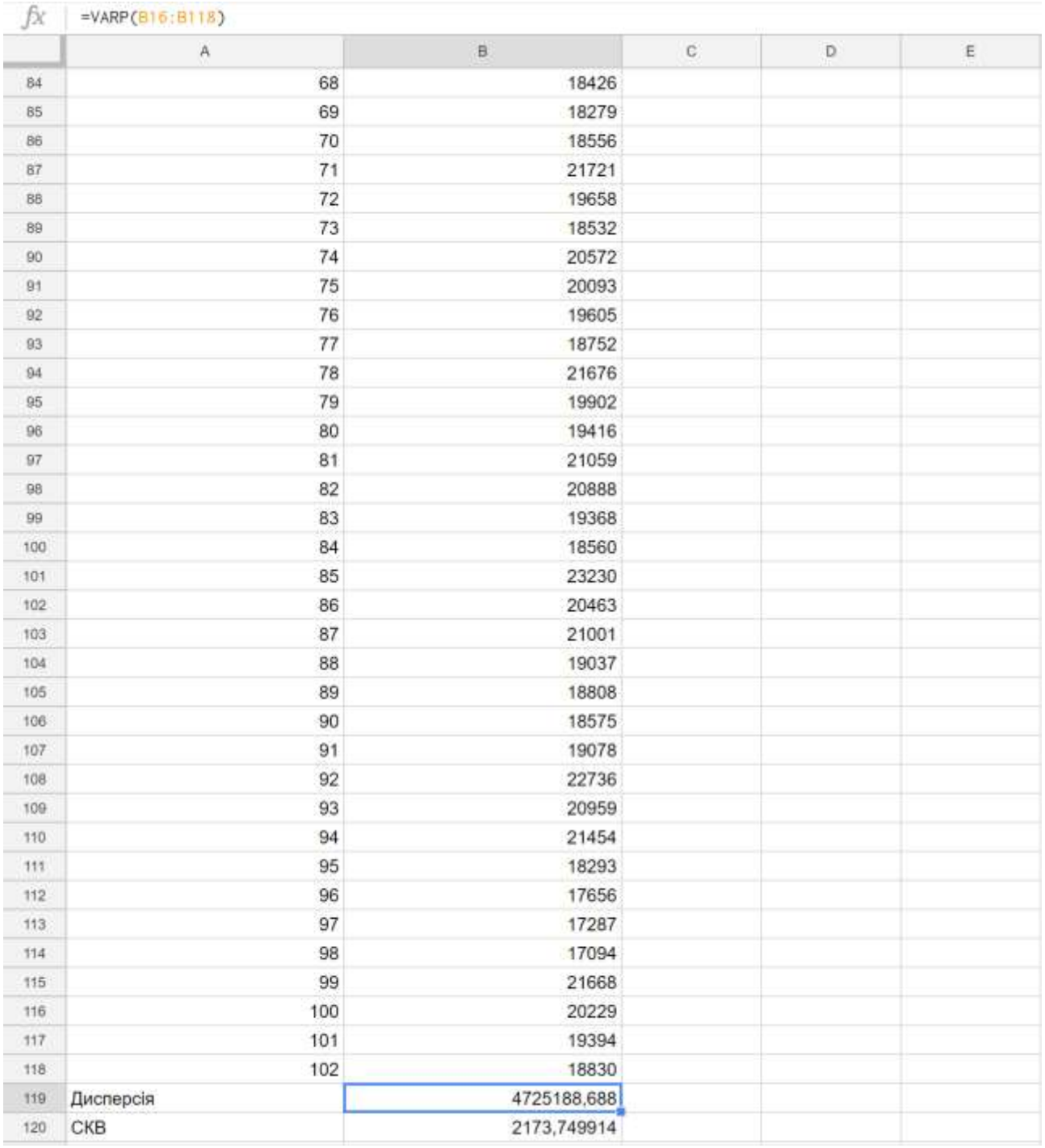
Наступний кроком звісно є обчислення середньоквадратичного відхилення (СКВ). Робимо це за допомогою функції STDEVP (рис. 3.10).



fx =STDEVP(B16:B118)

	A	B	C	D	E
84	68	11467			
85	69	13777			
86	70	11916			
87	71	11504			
88	72	11405			
89	73	12539			
90	74	12559			
91	75	11914			
92	76	14880			
93	77	13552			
94	78	14001			
95	79	13507			
96	80	13136			
97	81	13570			
98	82	12981			
99	83	14614			
100	84	12413			
101	85	12197			
102	86	12729			
103	87	12040			
104	88	12363			
105	89	11463			
106	90	13811			
107	91	15030			
108	92	13571			
109	93	13527			
110	94	12722			
111	95	12040			
112	96	11485			
113	97	14678			
114	98	14884			
115	99	13784			
116	100	13624			
117	101	13295			
118	102	13213			
119	Дисперсія	4604567,826			
120	СКВ	2145,825675			

Рисунок 3.10 - Обчислення СКВ в Google Таблицях по першому періоду



fx =VARP(B16:B118)

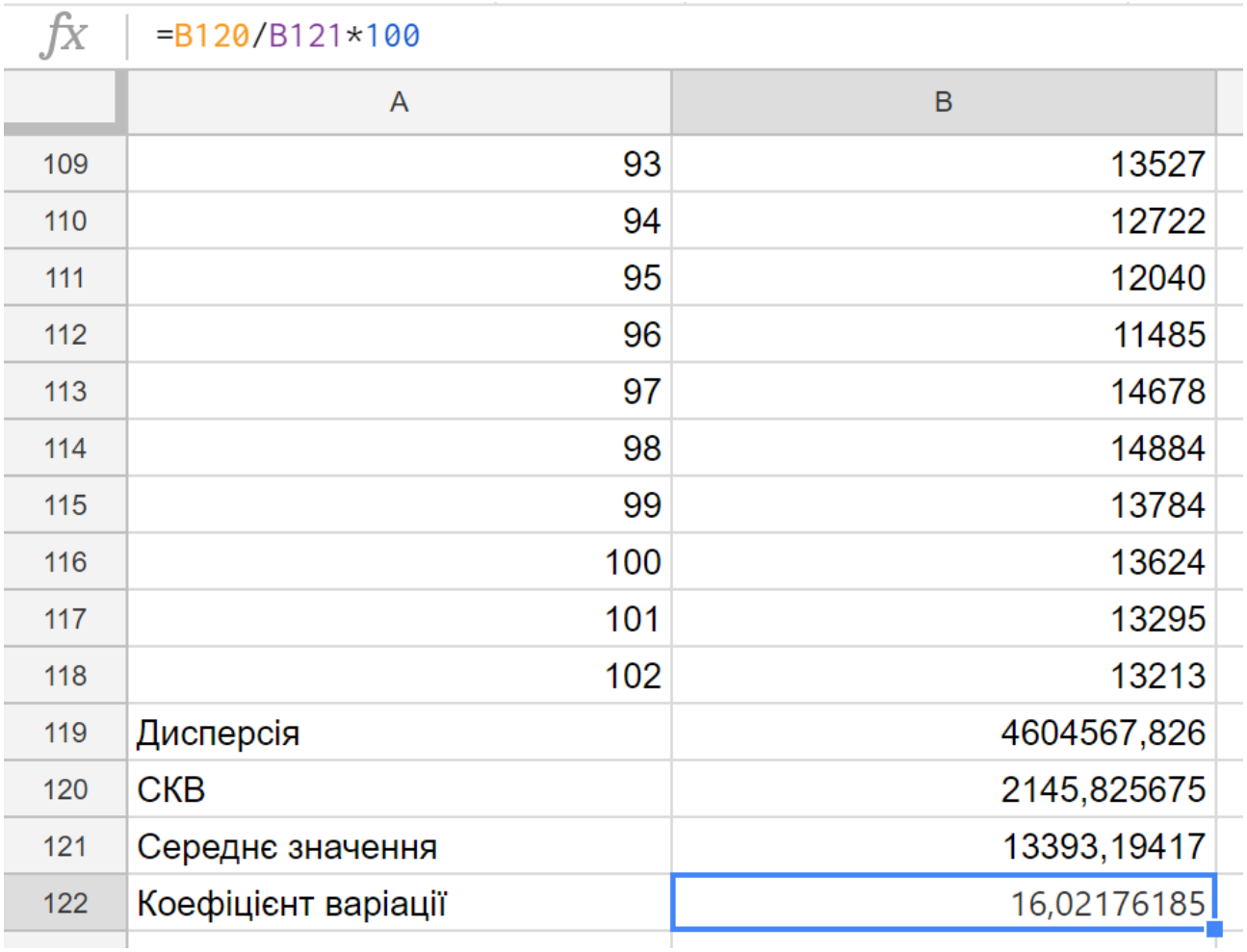
	A	B	C	D	E
84	68	18426			
85	69	18279			
86	70	18556			
87	71	21721			
88	72	19658			
89	73	18532			
90	74	20572			
91	75	20093			
92	76	19605			
93	77	18752			
94	78	21676			
95	79	19902			
96	80	19416			
97	81	21059			
98	82	20888			
99	83	19368			
100	84	18560			
101	85	23230			
102	86	20463			
103	87	21001			
104	88	19037			
105	89	18808			
106	90	18575			
107	91	19078			
108	92	22736			
109	93	20959			
110	94	21454			
111	95	18293			
112	96	17656			
113	97	17287			
114	98	17094			
115	99	21668			
116	100	20229			
117	101	19394			
118	102	18830			
119	Дисперсія	4725188,688			
120	СКВ	2173,749914			

Рисунок 3.11 - Обчислення СКВ в Google Таблицях по другому періоду

Як можемо побачити із рисунків, для першого періоду СКВ становить 2145,825675, для другого періоду - 2173,749914.

Наступним кроком є обчислення коефіцієнту варіації. Враховуючи, що в Google Таблицях немає формули для даного коефіцієнту, ми спочатку знайшли

середнє значення, та відносно формули, розділили СКВ на середнє та помножили на 100%.



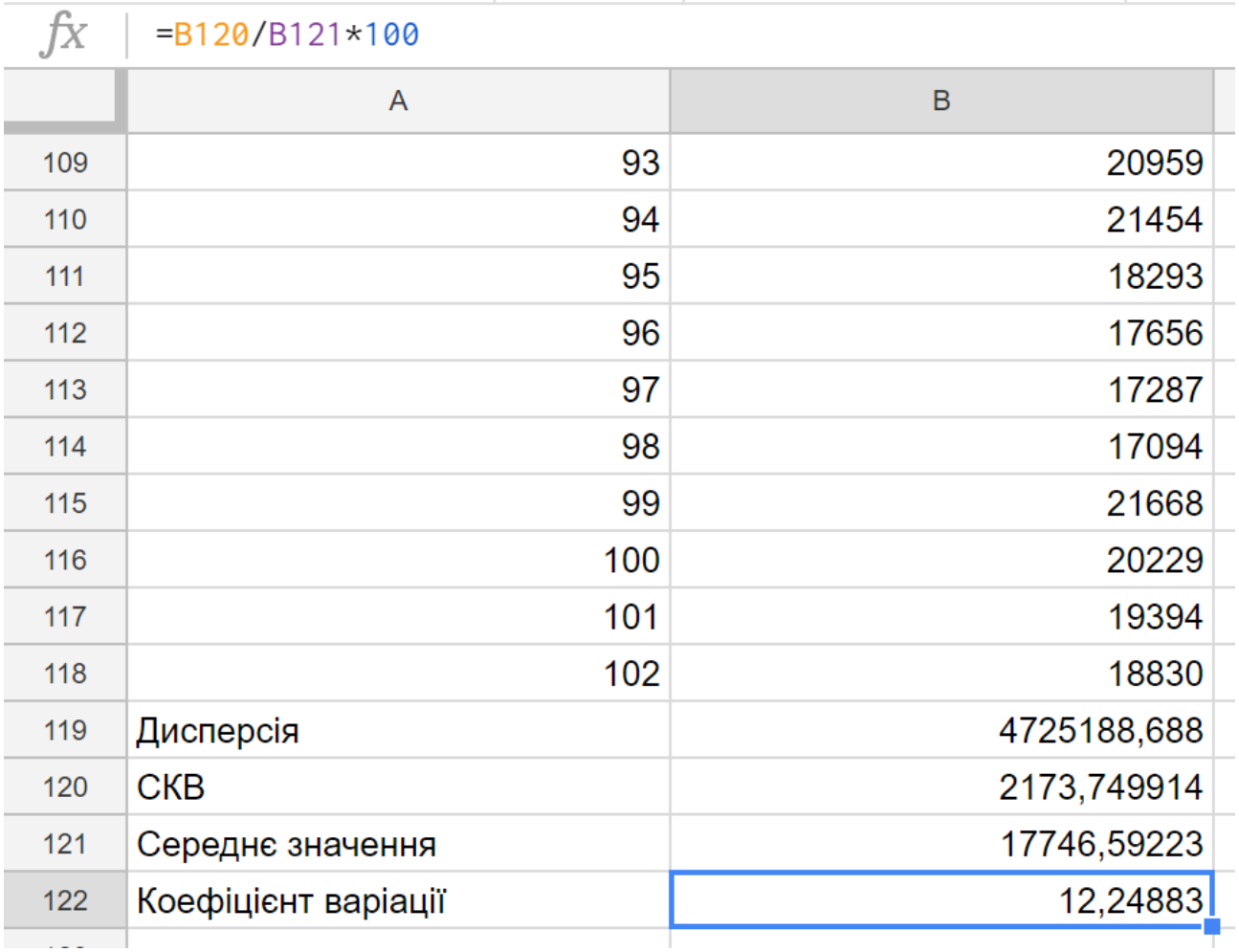
	A	B
109	93	13527
110	94	12722
111	95	12040
112	96	11485
113	97	14678
114	98	14884
115	99	13784
116	100	13624
117	101	13295
118	102	13213
119	Дисперсія	4604567,826
120	СКВ	2145,825675
121	Середнє значення	13393,19417
122	Коефіцієнт варіації	16,02176185

Рисунок 3.12 – Обчислення коефіцієнту варіації для першого періоду

Із обрахованих результатів видно, що для першого періоду коефіцієнт варіації становить - 16,02176185, для другого періоду - 12,24883.

Отже, якщо коефіцієнт варіації в наших результатах не більше 33%, то за класичною теорією ряд не вважається неоднорідним, тобто немає великого відхилення даних щодо середнього значення. Тому, у висновку можна стверджувати, що ефективність статистичної оцінки – хороша.

Також, задля підвищення точності можна використовувати теорію детермінованого хаосу, опираючись що всі показники на сайті мають хаотичний характер, але із певною повторюваністю.



fx =B120/B121*100

	A	B
109	93	20959
110	94	21454
111	95	18293
112	96	17656
113	97	17287
114	98	17094
115	99	21668
116	100	20229
117	101	19394
118	102	18830
119	Дисперсія	4725188,688
120	СКВ	2173,749914
121	Середнє значення	17746,59223
122	Коефіцієнт варіації	12,24883

Рисунок 3.13 – Обчислення коефіцієнту варіації для першого періоду

Висновок

В даній кваліфікаційній роботі бакалавра був здійснений опис рекомендаційних систем. Розглянуто фази процесу рекомендації та види рекомендаційних методів фільтрації. Розкрито суть роботи методів колаборативної фільтрації типу «user to user» та «item to item».

Об’єктом для всіх розрахунків та обчислень став сайт із пошуку автомобілів – Automoto.ua, тому в рамках поставленого практичного завдання:

1. Реалізовано рекомендаційний блок «Більш вигідно» на основі log-файлів сервера опираючись на коефіцієнт кореляції Танімото.

2. Реалізовано рекомендаційний блок «Більш вигідно» на основі алгоритму «item to item» колаборативної фільтрації із врахуванням релевантності.

3. Порівняли результати взаємодії по блоку, де виявилось, що алгоритм «item to item» колаборативної фільтрації на 44,33% ефективніший ніж рекомендації на log-файлів сервера.

4. Провели аналіз ефективності статистичної оцінки в реалізованій рекомендаційній системі, де виявлено, що маємо хороші показники, які не більші за 33%, оскільки за період роботи рекомендаційної системи на основі log-файлів сервера опираючись на коефіцієнт кореляції Танімото має коефіцієнт варіації 16,02176185, а рекомендаційна система на основі алгоритму «item to item» колаборативної фільтрації із врахуванням релевантності має коефіцієнт варіації 12,24883, що свідчить про відсутність великого відхилення даних від середнього значення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Рекомендаційна система [Електронний ресурс] // uk.wikipedia.org. – 2016. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0.

2. Slope One [Електронний ресурс] // en.wikipedia.org. – 2015. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Slope_One.

3. Xiaoyuan Su and Taghi M. Khoshgoftaar "A Survey of Collaborative Filtering Techniques A Survey of Collaborative Filtering Techniques" // Hindawi Publishing Corporation, Advances in Artificial Intelligence archive, USA : 2009. — С. 1-19.

4. Mustansar Ali Ghazanfar ―Building Switching Hybrid Recommender System Using Machine Learning Classifiers and Collaborative Filtering‖ // International Journal of Computer Science, Электронный ресурс: http://www.iaeng.org/IJCS/issues_v37/issue_3/IJCS_37_3_09.pdf

5. Sarwar B. M. Item-based collaborative filtering recommendation algorithms / B. M. Sarwar, G. Karypis, J. A. Konstan // Proceedings of ACM WWW ’01, pp. 285–295, ACM, 2001. 13. Karypis G. Evaluation of item-based top-N recommendation algorithms / G. Karypis // Proceedings of ACM CIKM’01, pp. 247–254, ACM, 2001.

6. Linden G. Amazon.com recommendations: Item-to-item collaborative filltering / G. Linden, B. Smith, J. York // IEEE Internet Computing, vol. 7, no. 1, pp. 76–80, 2003. 15. Hu Y., Koren Y., Volinsky C.: Collaborative filtering for implicit feedback datasets. In ICDM- 08, 8th IEEE Int. Conf. on Data Mining, pages 263–272, Pisa, Italy, 2008.

7. IV-а міжнародна наукова конференція «Вимірювання, контроль та діагностика в технічних системах» у м. Вінниця (31 жовтня – 2 листопада 2017 р. Покращення алгоритму item to item методу колаборативної фільтрації для

розробки рекомендаційних систем шляхом оцінки релевантності / Глушко М. В., Кучерук В. Ю.//
http://mpa.vntu.edu.ua/images/conference/conf2017/VCDTS%202017.pdf с. 215