МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет радіоелектроніки
Факультет Комп'ютерних наук
Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)

(рівень вищої освіти)

Дослідження та розробка аналітичних методів CRM системи магазину побутової техніки

Виконав:
студент 2 курсу групи ІПЗм-21-3
Нестеров С. О.
(прізвище, ініціали)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми Освітньо-наукова
Керівник доц., к.т.н. Вечур О. В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. Кафедри _____ З.В. Дудар

2023

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва)

Тип програми освітньо-наукова програма

Освітня програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Нестерову Сергію Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження та розробка аналітичних методів CRM системи магазину побутової техніки

затверджена наказом університету від « 29 » березня 2023 р. № 302 Ст

2. Термін подання студентом роботи до екзаменаційної комісії « 19 » травня 2023 р.

3. Вихідні дані до роботи *встановлений календарний план роботи, методичні вказівки до оформлення пояснювальної записки, методи CRM системи магазину побутової техніки.*

4. Перелік питань, що потрібно опрацювати в роботі *аналіз предметної галузі, огляд наявних методів розробки, розробка підходів та формул для обчислень, виділення критеріїв порівняння, створення програмного забезпечення для проведення дослідження, дослідження аналітичних методів та алгоритмів з отриманням та аналізом результатів.*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	26.01.2023	виконано
2	Здійснення огляду існуючих методів розробки	28.01.2023	виконано
3	Постановка задачі	05.02.2023	виконано
4	Виділення критеріїв порівняння, обґрунтування вибору	15.02.2023	виконано
5	Розробка підходів та створення формул для проведення обчислень	22.02.2023	виконано
6	Проведення експериментального дослідження, документування результатів	10.04.2023	виконано
7	Підготовка пояснювальної записки	27.04.2023	виконано
8	Перевірка роботи на антиплагіат	05.05.2023	виконано
9	Нормоконтроль	06.05.2023	виконано
10	Рецензування	07.05.2023	виконано
11	Підготовка презентації та доповіді	08.05.2023	виконано
12	Попередній захист	09.05.2023	виконано
13	Занесення роботи в електронний архів	10.05.2023	виконано
14	Допуск до захисту у зав. кафедри	19.05.2023	виконано

Дата видачі завдання 24 січня 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц., к.т.н. Вечур О. В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Звіт до кваліфікаційної роботи містить: 68 с., 25 рис., 6 табл., 23 джерел, 6 додатків.

АСОЦІАТИВНІ ПРАВИЛА, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, РЕКОМЕНДАЦІЇ, ПОБУТОВА ТЕХНІКА, РЕКОМЕНДАЦІЙНІ СИСТЕМИ, CRM-СИСТЕМА, APRIORI, ELCAT, FP-GROWTH.

Предмет дослідження – інтелектуальний аналіз даних, його методи та підходи з метою заохочення і приваблення клієнтів, для створення спеціалізованих рекомендацій, які будуть запропоновані користувачам системи.

Мета роботи – дослідження, вивчення, розробка та аналіз переліку методів інтелектуального аналізу даних для заохочення та приваблення відвідувачів, потенційний покупців, постійних клієнтів.

У результаті роботи розглянуто існуючі методи та алгоритми створення персональних пропозицій для роботи CRM-системи магазину побутової техніки, включаючи експериментальне дослідження та аналіз його результатів.

CRM-SYSTEM, INTELLIGENT DATA ANALYSIS, RECOMMENDATIONS, HOUSEHOLD APPLIANCES, RECOMMENDATION SYSTEMS, ASSOCIATIVE RULES, APRIORI, ELCAT, FP-GROWTH.

The subject of the study is the intelligent analysis of data, its methods and approaches for the purpose of encouraging and attracting customers, to create specialized recommendations that will be offered to users of the system.

The purpose of the work is to research, study, develop and analyze a list of methods of intelligent data analysis to encourage and attract visitors, potential buyers, regular customers.

As a result of the work, the existing methods and algorithms for creating personal offers for the operation of the CRM system of the household appliances store were considered, including experimental research and analysis of its results.

Я, Нестеров Сергій Олександрович, студент гр. ІПЗм-21-3, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження та розробка аналітичних методів CRM системи магазину побутової техніки», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElAr KhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

З кожним днем все більше віддається перевага людьми здійснення купівлі-продажу продукції через Інтернет через стрімкий ріст та розвиток в галузі інформаційних технологій і бізнесу по частині обробки та доставки замовлень клієнтів в напрямку електронної комерції.

Впровадження електронної комерції на теперішній час являється дуже затребуваним та актуальним напрямком, тому що багатьом компаніям це дозволяє підсилювати та зміцнювати їх позиції в умовах звичайних традиційних ринків і виходити на нові для збільшення продажу, завдяки чому фірми зможуть мати можливість збільшувати прибуток (з мінімальними витратами) та товарообіг в цілому.

Впровадження CRM-системи потребує значно більшої уваги, так як з’являється можливість збору, накопичення та аналізу інформації про уподобання замовників, за допомогою якого є можливість оптимізації асортименту, щоб клієнти змогли знаходити та обирати для себе більше корисних товарів, а іншими словами – більше витрачали грошей. Сюди входить інформування про надходження нових товарів по СМС, електронній скриньці, через що можна безпосередньо збільшити число продажів; розробка спеціальної системи цільових знижок задля найбільш економічно вигідного використання маркетингового бюджету; швидко реагувати та відповідати на звернення і скарги замовників, використовуючи CRM-систему, завдяки чому покращується відношення та збільшується позитивний відгук від клієнтів по відношенню до магазину; створення дисконтних та бонусних карток, так як можна прив’язати покупців до магазину та підвищити їх лояльність.

В більшості схожих систем створення рекомендації представляє складний процес, так як здебільшого не можна покладатися на товари, які додає користувач до кошику або профіль користувача, так як його досить складно сформувати. Тому важливо створити таку систему, яка б дала можливість рекомендацій в обхід традиційним підходам, наприклад, взаємодія (перегляди) користувачів з товарами.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз та дослідження предметної галузі

Глобальна мережа надала унікальну можливість співпраці та взаємодії з потенційними та постійними покупцями за допомогою віртуального ринку, що дозволяє знизити витрати бізнес і розширювати клієнтську базу з великою інтенсивністю.

Бізнес з продажу побутової техніки буде завжди користуватися своєю популярністю серед клієнтів будь-якої категорії. Не всі магазини побутової техніки мали необхідні товари в асортименті та достатню кількість одиниць товарів. Тому раніше придбати техніку за своїми персональними бажаннями було дуже складно та потрібно було витрачати багато часу на пошуки, власної енергії на походи до магазинів, так як заздалегідь неможливо було знати асортимент товарів, і не факт, що хоча б один із локальних магазинів має бажаний товар в наявності – товару може не бути в даний період або ніколи, наприклад, зняття продукції з виробництва та бракування товарів на складі (обидві умови разом).

Завдяки розробкам інформаційних систем для тієї чи іншої предметної галузі, є можливість достатньо заощадити час та енергію клієнтів, а також дозволяють обрати потрібну продукції за оптимальної ціною. Причиною оптимальних цін являється сам процес переходу для до таких систему, завдяки чому можна заощадити багато грошей на оренду приміщення та виплату зарплатні персоналу, отже, ціни в них або такі ж самі, або навіть нижче.

Завдяки автоматизації діяльності продажу побутової техніки є можливість уникнути роботи з папером, зменшити або звузити накопичення інформації, скоротити витрати часу на обробку інформації, скоротити фінансові витрати для обслуговування. Через автоматизації з прийому та обробки замовлень співробітники магазину можуть бути здатні швидко аналізувати та отримувати дані про товари, які придбаються клієнтом. Завдяки збору інформації можна чітко зрозуміти, на яку продукції більше попиту, а на яку – менше, і саме для цього створюються та запроваджуються рекомендаційні алгоритми.

Розглядається автоматизація діяльності (онлайн-збуту) компанії магазину побутової техніки.

Персонал магазину займається торгівельної діяльністю, за керівництво якої відповідає генеральний директор. У складі персоналу є наступні позиції: менеджери по закупці, головний бухгалтер, менеджери з продажу, адміністратори інформаційної системи, прибиральниці та комірники-вантажники.

Задача головного бухгалтеру – відповідати за всю бухгалтерію.

Комірники-вантажники виконують свою роботу на складі, де вони приймають продукції від постачальників та займаються подальшим комплектуванням товарів відповідно до замовлень клієнтів.

Адміністратори інформаційної системи відповідають за цілісність та безперебійність інформаційних ресурсів даного магазину, надають доступи до цих ресурсів окремим співробітника фірми. До переліку зобов'язань головного адміністратора відноситься також проведення процесу резервного копіювання даних та відновлення інформації, тому що втрата інформації неприпустима. Сюди ж входить опрацювання та аналіз даних завдяки CRM-системі – проведення дослідження та аналізу даних про клієнтів та розсилання рекламної продукції до всіх можливих джерел отримання інформації клієнтом.

Обробкою замовлень та консультацією з клієнтами щодо товарів та обговорення процесів про замовлення, доставку та повернення продукції, займаються співробітники на позиції менеджерів з продажу.

За створення, оновлення та ведення каталогу продукції, включаючи заповнення каталогу інформацією про продукції відповідно до прайс-листів, відповідають співробітники за позицією менеджерів по закупці.

Всі замовлення користувачів доставляються до них спеціальними службами доставки, такими як «Нова Пошта». Звідси, користувачам під час оформлення замовлення потрібно вказувати місто, область та номер відділення пошти. Обов'язково клієнт сплачує за доставку його замовлення самостійно будь-яким засобом оплати (термінал, готівка, тощо).

З інформаційною системою буде працювати чотири види користувачів:

- адміністратор;
- менеджер з продажу;
- менеджер по закупці;
- клієнт.

Можливий перелік бізнес-функцій адміністратора інформаційної системи подано наступним чином:

- розсилання рекламної продукції і формування аналітичних даних, завдяки переглядам користувачами продукції магазину та подальше їх рекламування / розсилання);
- управління персоналом, до якого входить створення нових «співробітників» у системі, можливість зміни даних існуючих робітників в системі, або взагалі їх видалення, наприклад, в разі звільнення або зміни посади).

Перелік основних можливостей менеджера з продажу можна подати таким чином:

- перегляд, сортування та фільтрація замовлень;
- безпосередня обробка замовлення, до якої входить вибір конкретного замовлення для оброблення, корегування його даних та підтвердження замовлення або відміна замовлення (в разі відмови від замовлення клієнтом);
- зміна статусу замовлення, яке залежить від того, чи вийшов на зв’язок клієнт, або підтвердив або відмовився клієнт від замовлення за будь-яких обставин.

Список функції, притаманних менеджерам по закупці:

- огляд переліку / каталогу товарів;
- керування каталогом товарів, до якого входить: створення продукції та додання їх до бази даних системи, зміна даних існуючої продукції у системі, приховання товарів – в разі зняття з виробництва або через тимчасові потреби, чи їх видалення із системи зовсім).

До можливостей клієнта відноситься наступне:

- перегляд, фільтрація, сортування (за рейтингом, за ціною, за новизною) каталогу, включаючи повнотекстовий пошук продукції за брендами або найменуваннями конкретних товарів;
- керування кошиком, до якого включено: додання товарів до кошику, зміна кількості товарів у кошику, та звичайно видалення товару з кошику);
- оформлення замовлення – створення замовлення з вказанням всіє обов’язкової та додаткової інформації, а також перегляд статусу та історії замовлення.

1.2 Опис CRM-систем та їх компонентів

З метою підвищення ефективності електронної комерції зазвичай впроваджують концепцію Customer Relationship Management (CRM), завдяки якої є можливість автоматизації керування взаємодією з замовниками, а також формування механізмів з пріоритетними потребами клієнта.

CRM-система представляю собою деяку базу даних з програмним забезпеченням, мета яких полягає в автоматизації стратегій та процесів взаємодії користувачів із системою, а також в оптимізації та покращенні обслуговування клієнтів та маркетингу в цілому.

Головним завданням CRM-систем являється зберігання цінних та дуже значних об’ємів інформації про замовників або мати можливість управління взаємодією користувачів і системи, опираючись на результати обробки аналітичних даних бази даних системи.

До CRM-систем відносять широкий спектр варіантів використання та вони є особливо незамінними, колу потрібно розуміти потреби клієнтів, проведення збору інформації, а також її застосування для залучення набагато більшої кількості потенційних користувачів, які потім можуть стати постійними.

Компонентами системи мають бути: база даних, що зберігає усі дані про взаємодії користувачів із системою; інтерфейс, який надає доступ користувача до бази даних системи; складові системи – операційна – відповідає за авторизацію

операцій та забезпечує звітність; фронтальна – дозволяє застосовувати велику кількість каналів взаємодії з користувачами даної системи, наприклад, СМС, електронною поштою, дзвінками за номером телефону, і т. д.; аналітична – виконується збір, обробка та деяка підготовка даних про замовників для прийняття спеціалізованих організаційних рішень, наприклад, збільшення обсягів продажу, або пошук будь-яких мір задля підняття середнього чеку, або навіть залучення нових потенційних користувачів.

До основних цілей інтеграції CRM-систем є потреба генерації такої система, яка б мала одну єдину базу контрагентів та клієнтів, включаючи повну історію із взаєминами з ними, з метою ефективного виконання та здійснення контролю якості продажу, аналітики роботи та отримування корисної статистики, можливість керування взаємовідношень з клієнтами, також розробка програм лояльності і планування подальшої роботи тощо.

Коли впроваджуються методи та підходи CRM-системи, інформаційна система стає набагато ефективнішою, наприклад, удосконалюються бізнес-процеси та бізнес-функції системи, підвищуються продажі продукції та поліпшуються зв’язки з клієнтами.

1.3 Актуальність роботи

Було виконано та проведено дослідження аналогічних реалізованих інформаційних систем, які також займаються процесами продажу побутової техніки (онлайн-збут). Для того, щоб можна було провести порівняння та опис, було обрано наступні дві аналогічні системи: «f.ua» та «ТехноТочка». На рисунку 1.1 подається головна сторінка інтерфейсу магазину «f.ua».

Дана інформаційна система містить наступні бізнес-функції:

- реєстрація та авторизація користувачів системи різними способами – через Facebook, Google, або локальним обліковим записом;
- перегляд каталогу товарів з можливістю фільтрації та сортування продукції;
- управління кошиком клієнта, додавання продукції до бажаних;

– оформлення і обробка замовлення, а також кредиту;

– складний повнотекстовий пошук товарів в базі.

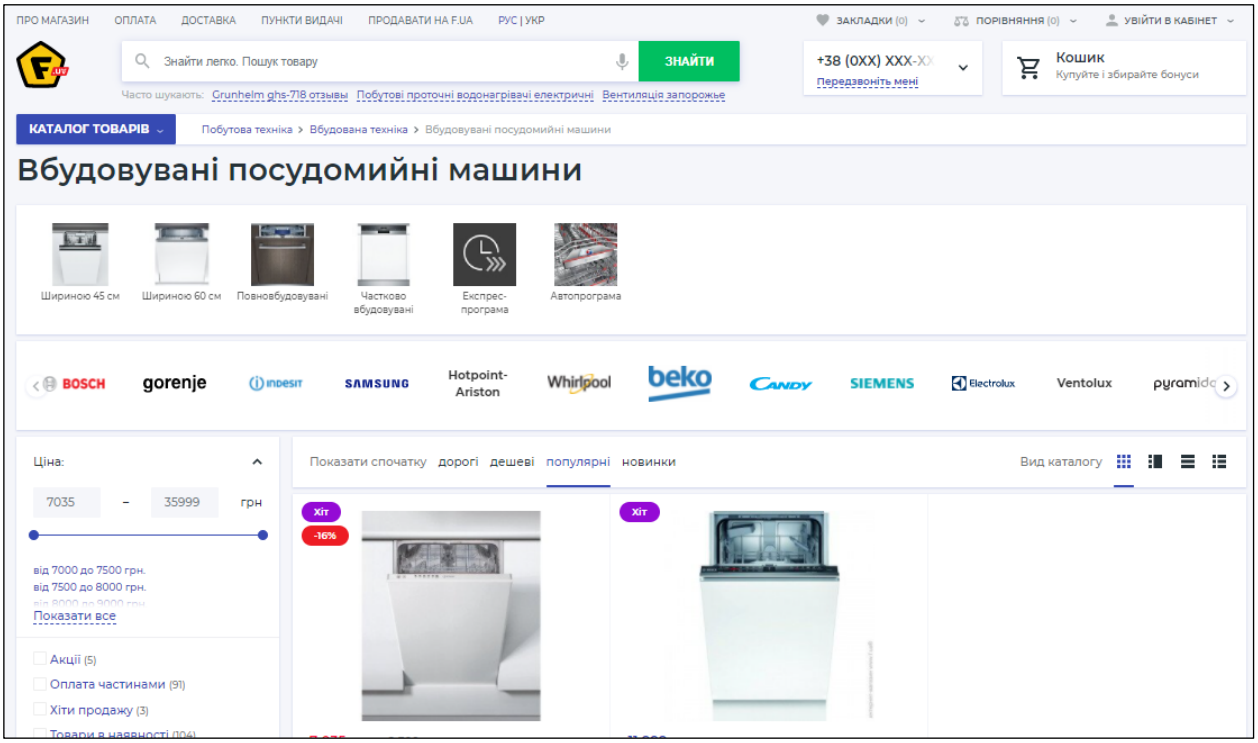

Рисунок 1.1 – Приклад інтерфейсу першого аналогу «f.ua»

На рисунку 1.2 можна побачити приклад інтерфейсу для другого аналогу – також магазину побутової техніки під назвою «ТехноТочка».

Для обох сайтів не вистачає / бракує процесу збору статистичних даних. Системи представляють собою деякий «чорний ящик», тобто, можливо, що дані насправді збираються і зберігаються десь на стороні серверу, але з точки зору клієнта вони не використовуються. Це пояснюється тим, що бракує будь-яких рекомендації та рекламних розсилок схожих товарів, які переглядалися користувачем, відповідно до його побажань та інтересів.

Другою проблемою систем є бракування використання сучасних інформаційних технологій, так як на теперішній час існує багато більш сучасних технології, що дозволяють генерувати чи створювати додатки, наприклад, в моду технологій поступово входять ідеї створення односторінкових додатків, які запускаються прямо в будь-якому браузері користувача. Суть в тому, що усі записи клієнта виконується виключно в рамках однієї сторінки, тобто та чи інша

операція, така як зміна тих чи інших даних не приводить до перезавантаження сторінки цілком, тому що теперішні сучасні технології мають можливість оновити лише

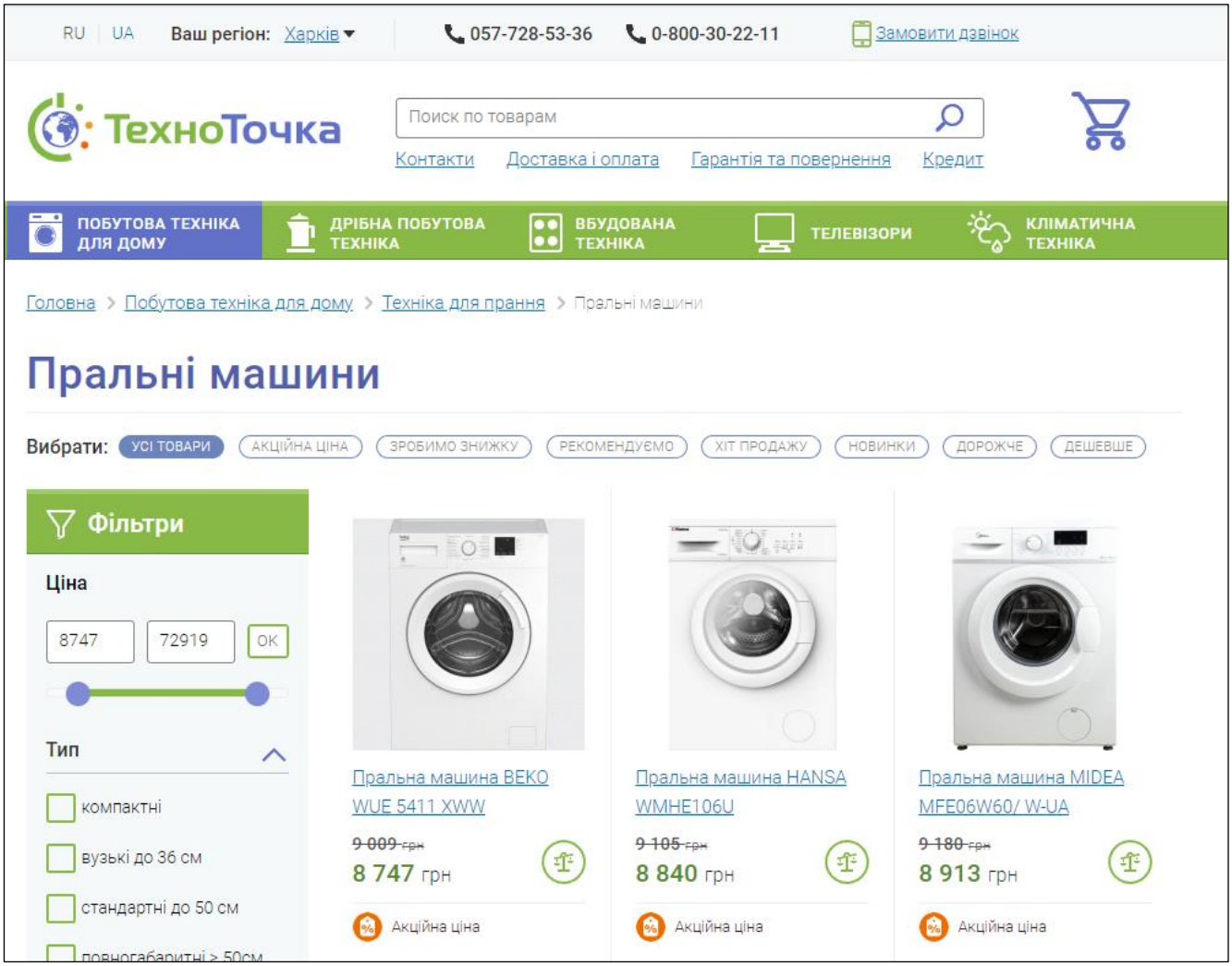

Рисунок 1.2 – Приклад інтерфейсу другого аналогу – магазину побутової техніки під назвою «ТехноТочка»

конкретну частину веб-додатку / веб-сторінки в асинхронному режимі. Найбільш актуальними та кращими інструментами для створення таких типів додатків є фреймворки ReactJS, Angular, AngularJS, Vue.js та інші. Будь-який з цього переліку інструментів може використовуватися залежно від того, з якого задачею необхідно мати справи, при цьому є можливість прискорити написання більш стабільного коду, поліпшити його підтримку, розширюючи та покриваючи код юніт-тестами. Виконується зниження завантаження / навантаження на трафік між

сервером та пристроєм користувача, і суттєво збільшується швидкість роботи веб-додатку на пристрої користувача.

Тому відповідно до попередньо перелічених недоліків описаних існуючих аналогів, було прийнято рішення з розробки CRM-системи, яка б надавала можливість проводити збір даних по переглядам користувачів, а також використовувати більше інформації з профілю користувача, наприклад, стать, вік, і т. д. Базуючись на даних, що були проаналізовані, потрібно виконувати розсилання рекламної продукції до будь-яких джерел користувачів згідно до їх побажань та потреб. Дані можуть збиратися завдяки переглядам веб-сторінок Web-додатку і продажами товарів замовникам. Дані про продажі та перегляди теоретично можуть бути поділені на кілька груп: ті, хто оглядає товари, але не придбає їх за якихось причин; ті, хто купує найдешевші або найдорожчі товари; за групами (кластерами в майбутньому) людей: літні, молоді, чоловіки та жінки, і т. д. і т. п.

1.4 Постановка задачі

Необхідно провести дослідження обраної предметної галузі, визначити основні бізнес-процеси та бізнес-функції. Провести аналіз і дослідження оцінок і методів побудови рекомендацій, включаючи опис найпопулярніших алгоритмів та методів, визначаючи їх переваги та недоліки, кроки виконання та приклади; обрати найбільш відповідні алгоритми чи методи до даної предметної галузі, детально їх описати, провести дослідження на деякому зразковому набору даних, провести їх аналіз, розраховуючи найбільш особливі та впливові фактори чи метрики, за допомогою яких потім визначити, який з алгоритмів краще за інший, у яких випадках один алгоритм буде кращим, а коли інший; побудувати графічні результати для візуального відображення різниці між методами чи алгоритмами; для дослідження розробити власне програмне забезпечення з алгоритмом або використовувати існуючі інструменти / засоби для розрахунків.

2 ДОСЛІДЖЕННЯ МЕТОДІВ ТА ТЕХНОЛОГІЙ, ЯКІ ВИКОРИСТОВУЮТЬСЯ В ПРЕДМЕТНІЙ ГАЛУЗІ

2.1 Методи кластеризації даних

Для класифікації дуже великої кількості або обсягу даних використовується кластерний аналіз, який виконує мету одержання відомостей про вибірку, яка досліджується. Отримані результати інтерпретуються задля детального опису кластерів, що були сформовані, а також їх ознак.

В напрямку маркетингу використання кластеризації дуже широке – виконується дослідження конкурентів, групування замовників за їх власною поведінкою, а також визначається ніша для позиціонування товару тощо. Процес групування замовників в спеціальні однорідні кластери називається сегментацією клієнтів. Вона використовується для одержання максимально можливого повного уявлення стосовно їх факторів впливі та загальної поведінки. Базуючись на цих кластерах для кожного окремого сегменту генерується індивідуальна політика.

Всі методи або підходи кластеризації даних поділяються на дві великі підгрупи, такі як ієрархічні та неієрархічні.

В свою чергу ієрархічні методи поділяються на дівізимні та агломеративні методи. Об’єднання ознак, коли кожна ознака представляє окремий кластер на початку роботи алгоритму, відноситься до агломеративних методів. У разі дівізимних методів виконується інакша операція – кожна ознака належить тільки одному кластеру на початку роботи алгоритму, при цьому кластер повинен дробитися на менші за розміром кластери, і так відбувається на кожній ітерації. Використання агломеративних методів являється найкращим, якщо подивитися з погляду обчислювальної ефективності.

На рисунку 2.1 наведено алгоритм агломеративної ієрархічної кластеризації.

Для знаходження відстаней між різними кластерами використовуються наступні зв’язки – центроїдний (centroid) зв’язок, середній зв’язок (average), повний зв’язок (complete), а також одиночний (single).

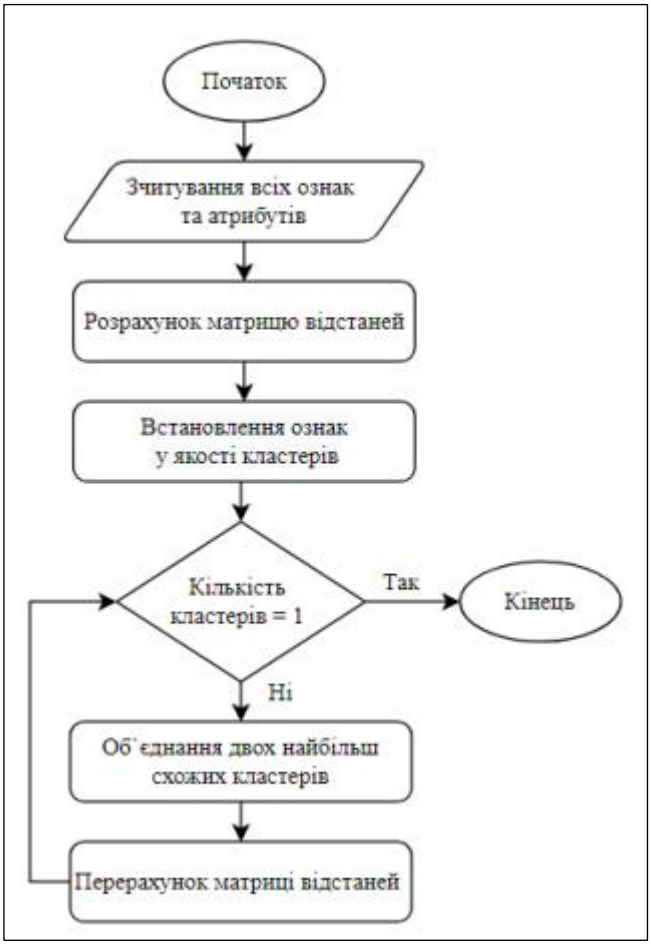

Рисунок 2.1 – Приклад алгоритму агломеративної ієрархічної кластеризації

Суть центроїдного методу полягає в тому, що є розрахунок центроїдів кластерів, тобто це точки, які в межах одного кластера є найближчими сусідами до усіх ознак, а також визначення та розрахування мінімальної відстані між ними.

Основна ідея методу середнього зв’язку полягає в розрахунках відстані між всіма точками двох кластерів. Йому необхідна значна кількість додаткових розрахунків, але він дуже добре справляється з викидами.

Метод повного зв’язку базується на розрахунках різних кластерів як відстань між двома найбільш віддаленими один від іншого ознаками. Цей метод остаточно гарантує, що зв’язані кластери, які будуть сформовані, будуть приблизно однакової форми, при цьому такий метод являється надчутливим до викидів.

Метод одиночного зв’язку полягає в розрахунках відстані між різними кластерами, що лежить між найближчими ознаками. Такий метод дуже зручно

використовувати для виявлення викидів, для цього він є найкращим, тому що вони презентуються у виді поодиноких кластерів, але з однією ознакою. В якості недоліків такого методу, можна виділити форму кластерів, які відповідають скошений вид дендограм та ланцюгів.

На жаль, є один великий недолік використання ієрархічних методів – вони зручні тільки для використання невеликої вибірки даних.

В свою чергу неієрархічні методи кластеризації даних в цілком базуються на задачі оптимізації, що означає, що процес групування вихідної вибірки у кластери являються рішенням, яке відноситься до екстремальних задач. Найпопулярнішим, тривіальним, найпоширенішим методом являється k-середніх [1-2].

На рисунку 2.2 подається алгоритм методу k-середніх.

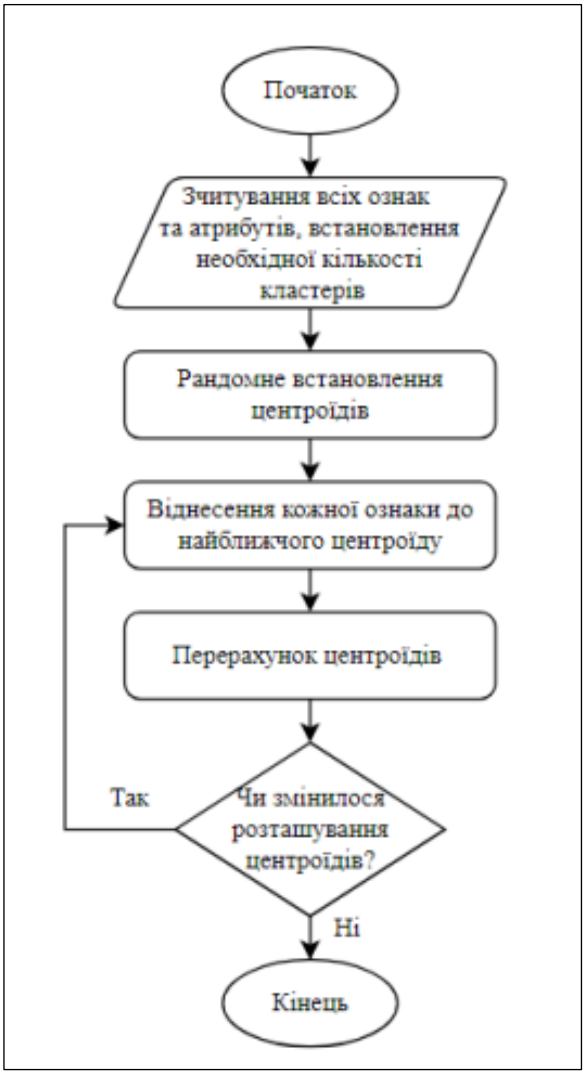

Рисунок 2.2 – Приклад алгоритму методу k-середніх

Якщо порівнювати метод k-середніх з ієрархічними методами, то цей метод можна визначити як швидкий кластерний алгоритм, так як немає потреби окрім вибірки вимагання додаткових відомостей, він також потребує гіпотезу, що визначає найбільш вірогідну кількість кластерів. Суть даного методу полягає в процесі мінімізації відстані між компонентами кластера і процесу максимізації відстані між кластерами.

Даний метод кластеризації використовує одну з наступних метрик для розбиття ознак:

– евклідова відстань, яка являється найбільш популярною серед всіх інших метрик і представляє геометричну відстань в багатомірному просторі (формула 2.1);

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \quad (2.1)$$

– квадрат евклідової відстані – він звичайно використовується для того, щоб надавати віддаленим один від одної ознакам більшої ваги (формула 2.2);

$$d(X, Y) = \sum_{i=1}^{m} (X_i - Y_i)^2 \quad (2.2)$$

– манхеттенська відстань – це відстань, яка розраховується і дорівнює середній різниці по координатам та призводить до результатів, аналогічних, як і евклідова відстань, але є один виняток – відстань не зводиться у ступінь, так як вплив вибросів досить зменшується (формула 2.3);

$$d_1(p, q) = \|p - q\|_1 = \sum_{i=1}^{n} |p_i - q_i| \tag{2.3}$$

– відстань Чебишева – вона використовується для процесу відокремлення обох ознак у різні кластери при умові, якщо ознаки відрізняються за будь-якою однією координатою (формула 2.4).

$$l_\infty(\vec{x}, \vec{y}) = \max_{i=1,\ldots,n} |x_i - y_i| \tag{2.4}$$

Підсумовуючи обидва типи методів кластеризації, метод k-середніх, що відноситься до неієрархічних методів кластеризації, підходить для обробки саме великих вибірок на відміну від інших. В якості недоліків даного методу можна віднести потребу завдання початкової точної кількості кластерів, а також чутливість даного алгоритму до вибору конкретних початкових так званих наближень центроїдів, так як вона змінюється навіть на той самій вибірці в залежності від самого значення центроїду. Але для виправлення останнього недоліку було розроблено модифікований алгоритм – k-середніх++.

Сегментація клієнтів – це просто групування клієнтів зі схожими характеристиками. Ці характеристики включають географічні, демографічні, поведінкові, купівельну спроможність, ситуаційні фактори, особистість, спосіб життя, психографічні характеристики тощо. Цілями сегментації клієнтів є залучення клієнтів, утримання клієнтів, підвищення прибутковості клієнтів, задоволеності клієнтів, розподіл ресурсів шляхом розробки маркетингових заходів або програм та вдосконалення заходів цільового маркетингу.

Кластеризація – ефективний метод, який використовується для сегментації клієнтів. Кластеризація розміщує однорідні точки даних у заданому наборі даних. Кожна з цих груп називається кластером. Хоча об’єкти в кожному кластері схожі між собою, вони не схожі на об’єкти інших груп. Кластеризація – це тип підходу до інтелектуального аналізу даних у машинному навчанні, який класифікується як

неконтрольоване навчання. Це пояснюється тим, що він здатний виявляти шаблони та інформацію з немаркованих даних. Він широко використовується в машинному навчанні, класифікації та розпізнаванні образів.

2.2 Фільтрація на базі контенту (content-based)

Фільтрація на базі контенту виконує аналіз характеристик елементів з метою ідентифікації тих, які мають особливий інтерес для користувачів [3]. Згідно до цього підходу опис / характеристика товару ставиться у відповідність до інтересів користувача, які отримується за його попередніми оцінками. Чим більше відповідність товару до інтересів користувача, тим вище оцінка потенційної зацікавленості користувача [4].

Все, що потрібно зробити для даного способу, – це створити профілі користувачів та профілі елементів. Опираючись на властивості елементів системи, можна зробити висновки про належність конкретного користувачу якогось конкретного елемента. Для створення профілю рекомендаційної системи та опису її елементів необхідно ставити певний набір ключових слів і відповідність кожному елементу.

Об’єктом рекомендацій на базі контенту частіше за все виступає продукція з неструктурованим описом: статті, книги, музика. В якості прикладу таких ознак можна привести рецензії, текстові описи, склад авторів та інше. Але нічого не заважає використовувати категоріальні або звичайні числові ознаки. Даний тип систем рекомендації може бути використаний у різних сферах, наприклад, рекомендації в електронної комерції (інтернет-магазини), контент в соціальних мережах, новинних статей, тощо.

Головна мета контент-орієнтованого методу являється створення профілю для кожного елемента та користувача.

Впродовж виконання даного методу проводиться аналіз набору характеристик предметів, що формують профілі інтересів користувачів, використовуючи характеристики предметів, а також оцінюються цільовими користувачами. В процесі рекомендації лежить процес зіставлення атрибутів

профілів користувачів та набору характеристик вмісту товару. Після чого, можна робити висновки про відповідність кожного елементу користувачу на основі властивостей елементів системи.

Зберігання ключових слів у документах в n-мірних векторах є однією з поширених технік, в межах якої призначається вага для кожного слова, а для визначення самої ваги використовуються такі техніки, як TD-IDF, відстань Кульбака-Лейблера, відстань Окарі, тощо. Після того, як вектори ключових слів були побудовані, потрібно розрахувати переходи між предметами та користувачами. Пропозиції слід рекомендувати користувачам, якщо вони найчастіше взаємодіють з іншими користувачами або з одним із прочитаних ними пунктів. Чим менший кут, тим ближче буде документ до профілю користувача. Формула 2.5 приводить розрахунки для коефіцієнта косинуса.

$$\cos(x, y) = \frac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \times \sqrt{\sum_{i=1}^{n} y_i^2}} \tag{2.5}$$

де x та y – це вектори профілю користувача та документу.

Перевагами даного підходу можна виділити наступні:

- володіння детальною інформацією про продукції не є обов’язковим;
- включає можливість рекомендації навіть тих об’єктів, які не мають жодних оцінок інших користувачів;
- це універсальний метод, який приносить результати дуже швидко.

До недоліків відносяться:

- наявність обмеженості аналізу контенту;
- велика залежність від предметної галузі;
- профіль товарів та користувачів мають містити один й той самий перелік характеристик з метою полегшення їх порівняння;

– проблема «холодного» старту – означає повну відсутність будь-яких вподобань користувачів, коли тільки починається робота з системою (на початку);
– дуже низька точність рекомендацій.

2.3 Колаборативна фільтрація

Колаборативні методи фільтрації – це методи, які дозволяють виконувати прогнозування невідомих вподобань конкретного одного користувача в системі за допомогою порівняння його інтересів з інтересами інших користувачів в системі [5]. Іншими словами, ті користувачі, як однаково оцінили елементи (товари) системи, можуть також однаково оцінювати будь-які інші елементи в системі. Для реалізації даного методу алгоритми створюються на основні машинного навчання.

В основі цього метода лежить поведінка користувача або групи користувачів. Принципом дії даної фільтрації закладається в тому, що, система знаходить будь-яких інших користувачів з однаковими або схожими запитами, якщо конкретний користувач придбав товари, чи просто переходів на сторінки тих чи інших товарів. Система буде рекомендувати йому тільки ті товари, які були зацікавлені іншими користувачами, а даний користувач – поки що ні.

Алгоритми колаборативної фільтрації виконують прогнозування для певного користувача цінність різних елементів системи, використовуючи попередні оцінки, зроблені іншими клієнтами.

«User-based» підхід рекомендує ті товари, які будуть куплені схожими користувачами: буде виконано усереднення рейтингу товарів, що проставляється іншими клієнтами з використанням ваги по ступені схожості клієнтів.

Підходи на основі колаборативної фільтрації, на сьогодні, є більш популярними, чим підходи на основі фільтрації вмісту, оскільки являють собою зображення практичного досвіду. Міра схожості обчислюється за матрицею оцінок R. Найбільш загальновживана метрика схожості – кореляція Пірсона і косинусна відстань рядків (стовпців) матриці.

У формулі 2.6 подається коефіцієнт Пірсона.

$$S_{i,j} = \frac{\sum_{u \in U}(r_{u,j} - \overline{r_i}) \times (r_{u,j} - \overline{r_j})}{\sqrt{\sum_{u \in U}(r_{u,j} - \overline{r_i})^2} \times \sqrt{\sum_{u \in U}(r_{u,j} - \overline{r_j})^2}} \quad (2.6)$$

де $U = Ui \cup Uj$ – це множина користувачів, які оцінили об'єкти i чи j.

Косинус кута між двома векторами i r i j r подано у формулі 2.7.

$$S_{i,j} = \cos(r_i, r_j) = \frac{r_i^o \times r_j}{|r_i| \times |r_j|} \quad (2.7)$$

Проводиться формування кінцевої множини об'єктів S, найбільш близьких до об'єкта o. Обчислення рейтингу об'єкта o здійснюється за формулою 2.8.

$$\hat{r}_{u,o} = \frac{\sum_{j \in S} S_{o,j} \times r_{u,i}}{\sum_{j \in S} |S_{o,j}|} \quad (2.8)$$

Популярний підхід до формування багатьох рекомендацій – це упорядкування всіх об'єктів за критерієм схожості і вибірці деякого фіксованого кількості об'єктів з максимальним рейтингом. В якості запобіжної схожості двох об'єктів виступає косинус кута між n-мірними векторами.

Переваги:

- високий показник точності рекомендацій;
- користувач має змогу ознайомитися з повним спектром товарів ресурсу.

Недоліки:

- нові елементи в системі повинні оцінюватися користувачами до використання в процесі рекомендації. Колаборативні системи керуються тільки вподобаннями користувачів, а нові товари не можуть бути рекомендовані, доки не наберуть необхідну кількість оцінок;

- «білі ворони». Тип користувачів, думка та вподобання котрих не підходять під більшість оцінок;
- кількість оцінок, що треба показати набагато перевищує кількість даних оцінок;
- високий поріг входження – не знаючи нічого про користувача, рекомендації є марними та не точними (проблема «холодного старту»);
- синонімія. Схожі товари мають різне найменування;
- високий поріг входження – не знаючи нічого про користувача, рекомендації є марними та не точними (проблема «холодного старту»).

2.4 Асоціативні правила для виявлення закономірностей

Асоціативні правила – це метод аналізу даних, що використовується для виявлення цікавих зв’язків, що часто зустрічаються, між елементами у великих наборах даних. Вони є виразами виду "Якщо X, то Y", де X і Y являють собою набори елементів. Асоціативні правила визначають, які елементи часто зустрічаються разом і можуть використовуватися для передбачення поведінки чи надання рекомендацій.

Аналіз кошику покупців належить до завдань інтелектуального аналізу даних (data mining).

Data mining – це процес пошуку у великому обсязі даних будь-яких закономірностей і отримання знань, які потрібні для прийняття рішень у багатьох сферах людської діяльності [6-9].

Правило асоціації (associate rule) складається із двох частин: попередньої (якщо) і наступної (тоді). Попереднє завдання – це елемент, що знаходиться в даних. А наступна – це елемент чи множина елементів, що зустрічаються у поєднані із попереднім завданням [10].

В інтелектуальному аналізі даних правила асоціації є корисними та допомагають спрогнозувати поведінку клієнта. Вони відіграють важливу роль у аналізі кошиків покупців.

Загалом навчання на асоціативних правилах можна описати як «Хто купив X, також купив Y». В основі лежить аналіз транзакцій, всередині кожної з яких лежить свій унікальний елемент set з набору елементів. За допомогою алгоритмів навчання на асоціативних правилах знаходяться ті самі «правила» збігу елементів усередині однієї транзакції, які потім сортуються за їх силою.

Для оцінки якості отриманих рекомендацій використовуються такі метрики, як підтримка, достовірність, інтерес та переконливість.

Підтримка (support) – перше базове поняття в ARL – дозволяє дізнатися, в якій частині кошиків покупців містяться всі елементи того чи іншого асоціативного правила (формула 2.9).

$$\text{supp}(X) = \frac{t \in T; X \in t}{|T|} \tag{2.9}$$

де X – itemset, що містить у собі i-items,

T – кількість транзакцій.

Тобто. Загалом, це показник «частотності» даного itemset у всіх аналізованих транзакціях. Але це стосується тільки X. Нам же цікавий скоріше варіант, коли у нас в одному itemset зустрічаються x1 і x2 (наприклад). Але тут теж все просто. Нехай x1 = {Пральна машина}, а x2 = {Мішок для прання}, значить нам потрібно порахувати, у скількох транзакціях зустрічається ця пара (формула 2.10).

$$\text{supp}(X) = \frac{\sigma(x_1 \cup x_2)}{|T|} \tag{2.10}$$

де – σ – кількість транзакцій, які містять x_1 та x_2.

Транзакції (блендер, пральна машина, мішок для прання) подано в таблиці 2.1.

Таблиця 2.1 – Транзакції (блендер, пральна машина, мішок для прання)

Транзакція	Блендер	Пральна машина	Мішок для прання
1	1	1	1
2	0	0	0
3	1	1	0
4	1	1	1
5	0	1	1

Приклад розрахунку підтримки подано у формулі 2.11.

$$\text{supp}(X)=\frac{\text{Транзакції з пр. маш. та міш. для пр.}}{\text{Всі транзакції}}=\frac{3}{5}=60\% \tag{2.11}$$

Достовірність (confidence) – показує, наскільки хорошим є правило для передбачення правої частини, коли умова зліва вірна (формула 2.12). Є показником того, як часто наше правило спрацьовує для всього датасету.

$$\text{conf}(x_1 \cup x_2)= \frac{\text{supp}(x_1 \cup x_2)}{\text{supp}(x_1)} \tag{2.12}$$

Наприклад, є потреба порахувати confidence для правила "хто купує пральну машину, той купує і мішок для прання".

Для цього спочатку порахуємо, який support у правила "купує пральну машину", потім порахуємо support у правила "купую пральну машину та мішок для прання", і просто поділимо одне на інше. Тобто. ми порахуємо в скількох випадках (транзакціях) спрацьовує правило «купив пральну машину» supp(X), «купив пральну машину та мішок для прання» (формули 2.13-2.14).

$$\text{conf}(\text{Пр. машина} \cup \text{Мішок для пр.})= \frac{\text{supp}(\text{Пр. маш. } \cup \text{ Міш. для пр.})}{\text{supp}(\text{Пр. маш})} \tag{2.13}$$

$$\text{conf}=\frac{3}{4}=75\% \tag{2.14}$$

Інтерес (lift) – відіграє важливу роль при аналізі отриманих правил і показує, наскільки добре було передбачено те, що у правій частині. Іншими словами, lift вимірює силу правила, порівнюючи повне правило із припущеною правою частиною і розраховується як ставлення достовірності правила до частоти появи слідства. Тобто, lift – це відношення «залежності» items до їх «незалежності». Lift показує, наскільки items залежать один від іншого, що очевидно з формули 2.15.

$$\text{lift}(x_1 \cup x_2)=\frac{\text{supp}(x_1 \cup x_2)}{\text{supp}(x_1)\times\text{supp}(x_2)} \tag{2.15}$$

Наприклад, ми хочемо зрозуміти залежність покупки пральної машини пива та покупки мішка для прання. Для цього вважаємо support правило «купив пральну машину та мішок для прання» і ділимо його на добуток правил «купив пральну машину» та «купив мішок для прання». Якщо lift = 1, ми говоримо, що items незалежні і правил спільної купівлі тут немає. Якщо ж lift > 1, то величина, за яку lift, власне, більше цієї самої одиниці, і покаже нам «силу» правила. Чим більше одиниці, тим краще. Якщо lift < 1, це покаже, що правило підстави x_2 негативно впливає на правило x_1. Інакше lift можна подати, як ставлення confidence до expected confidence, тобто, відношення достовірності правила, коли обидва (ну або скільки потрібно) елемента купуються разом до достовірності правила, коли один із елементів купувався (неважливо, з другим чи без) (формули 2.16-2.16).

$$\text{lift} = \frac{\text{Confidence}}{\text{Expected confidence}} = \frac{\text{P (Пральна машина | Мішок для прання)}}{\text{P (Пральна машина)}} \tag{2.16}$$

$$\text{lift}= \frac{\frac{3}{4}}{\frac{3}{5}}=1.25 \tag{2.17}$$

Тобто, наше правило, що пральну машину придбають з мішком для прання, на 25% потужніше правила, що пральну машину просто купують.

Переконливість (conviction) – це "частотність помилок" нашого правила. Тобто, наприклад, як часто купували правильну машину з мішком для прання та без (формули 2.18-2.19).

$$\text{conv}(x_1 \cup x_2)= \frac{1\text{-supp}(x_2)}{1\text{-conf}(x_1 \cup x_2)} \tag{2.18}$$

$$\frac{1\text{-supp(Пральна машина)}}{1\text{-conf(Пральна машина}\cup\text{Мішок для прання)}}= \frac{1- 0.6}{1- 0.75}=1.6 \tag{2.19}$$

Чим результат за формулою вище за 1, тим краще. Наприклад, якщо conviction купівлі пральної машини та мішка для прання разом дорівнював би 1.2, то це означає, що правило «купив правильну машину та мішок для прання» було б у 1.2 рази (на 20%) більш вірним, ніж якби збіг цих items в одній транзакції було б чисто випадковим. Трохи не інтуїтивне поняття, але й використовується не так часто, як попередні три.

Існує ряд часто використовуваних класичних алгоритмів, що дозволяють знаходити правила в itemsets згідно з перерахованими вище поняттями – Apriori-алгоритм, ECLAT-алгоритм, FP-Growth алгоритм та інші. Кожен з цих трьох алгоритмів буде детально розглянуто в наступному розділі про дослідження з аналізом результатів для всіх трьох.

Переваги:

– простота: асоціативні правила прості у розумінні та реалізації. Вони вимагають складних математичних обчислень чи глибокого розуміння статистичних моделей;

- висока швидкість навчання: асоціативні правила можуть бути швидко навчені великих обсягах даних. Вони можуть ефективно знаходити приховані зв'язки та шаблони даних без тривалого навчання;
- інтерпретованість: результати асоціативних правил легко інтерпретувати та зрозуміти. Вони є простими "якщо" правилами, які показують зв’язки між елементами даних;
- асоціативні правила можуть бути застосовані в різних галузях, таких як рекомендаційні системи, маркетингові дослідження, аналіз поведінки споживачів і т.д.

Недоліки:

- чутливість до шуму: асоціативні правила можуть бути чутливі до шуму даних. Поодинокі викиди або некоректні значення можуть призвести до появи неправильних асоціацій;
- проблема масштабованості: під час роботи з великими обсягами даних асоціативні правила можуть зіткнутися з проблемою масштабованості. Обробка та аналіз великих наборів даних може бути обчислювально-затратним завданням;
- обмеженість інформації: асоціативні правила працюють тільки на основі наданих даних і не здатні виявляти прихованих причинно-наслідкових зв’язків або складних залежностей між змінними;
- необхідність попередньої обробки даних: Для досягнення кращих результатів асоціативні правила вимагають попередньої обробки даних, такої як видалення дублікатів, нормалізація чи заповнення пропущених значень.

3 ОПИС ПРОВЕДЕНИХ ДОСЛІДЖЕНЬ

3.1 Дослідження алгоритму ECLAT

Eclat розшифровується як Equivalence Class Clustering Bottom-Up Lattice Traversal, і це алгоритм для аналізу правил асоціації (який також перегруповує частий аналіз наборів елементів).

Алгоритм ECLAT не є першим алгоритмом аналізу правил асоціації. Основним алгоритмом у домені є алгоритм Апріорі. Оскільки алгоритм Apriori є першим алгоритмом, який був запропонований у домені, його покращили з точки зору ефективності обчислень (тобто, створено швидші альтернативи).

Існує дві швидші альтернативи алгоритму Apriori, які є найсучаснішими: одна з них – FP Growth, а інша – ECLAT. Між FP Growth і ECLAT немає очевидного переможця з точки зору часу виконання: це залежатиме від різних даних і різних налаштувань в алгоритмі.

У той час, як більшість алгоритмів містять низку ключових показників своїх правил, ECLAT не робить те саме.

Наприклад, ECLAT не надає показників confidence та lift, які є важливими для інтерпретації в альтернативних моделях. З іншого боку, це дозволяє моделі бути швидшою: користувач має вибір між швидкістю та більшою кількістю показників.

Давайте тепер представимо приклад використання, щоб зробити тему трохи більш практичною та прикладною. У цій статті ми візьмемо невеликий набір даних транзакцій нічного магазину. Для кожної операції ми просто маємо список продуктів.

Вихідний набір даних подано у таблиці 3.1

Таблиця 1 – Вихідний набір даних

Номер транзакції	Товари
1	пральна машина, блендер, холодильник
2	пральна машина, мультиварка

Продовження таблиці 3.1

Номер транзакції	Товари
3	праска, фен, чайник, холодильник
4	праска, фен, чайник, пральна машина, мультиварка
5	блендер, холодильник
6	мультиварка
7	праска, фен, чайник, пральна машина, блендер, холодильник
8	праска, фен, чайник, пральна машина, мультиварка
9	блендер, пральна машина
10	пральна машина, мультиварка
11	чайник, праска
12	пральна машина, мультиварка
13	фен, праска
14	пральна машина, мультиварка
15	праска, фен, чайник, блендер, холодильник
16	пральна машина, блендер, мультиварка, холодильник
17	блендер, холодильник
18	пральна машина, мультиварка
19	блендер, холодильник
20	пральна машина, мультиварка

Якщо уважно подивитися на вихідні дані, покупці цього магазину в основному купують пральні машини, мультиварки, блендери тощо, а також деякі інші товари.

Робота алгоритму:

Крок 1 – перелічити набір ідентифікаторів транзакції (TID) кожного продукту.

Першим кроком є створення списку, який містить для кожного товару список ідентифікаторів транзакцій, у яких зустрічається продукт. Цей список представлений у таблиці 3.2.

Таблиця 3.2 – Перелік товарів та їх множини TID

Товар	Множина TID
Блендер	1, 5, 7, 9, 15, 16, 17, 19
Холодильник	1, 3, 5, 7, 15, 16, 17, 19
Пральна машина	1, 2, 4, 8, 9, 10, 12, 14, 16, 18, 20
Мультиварка	2, 4, 6, 8, 10, 12, 14, 16, 18, 20
Праска	3, 4, 7, 8, 11, 13, 15
Фен	3, 4, 7, 8, 13, 15
Чайник	3, 4, 7, 8, 11, 15

Крок 2 – фільтрація з мінімальною підтримкою.

Наступним кроком є визначення значення, яке називається мінімальною підтримкою. Мінімальна підтримка слугуватиме для фільтрації продуктів, які трапляються недостатньо часто, щоб їх розглядати.

У поточному прикладі ми виберемо значення 7 для мінімальної підтримки. Як ви можете бачити в таблиці кроку 1, є два продукти, які мають набір TID, який містить менше 7 транзакцій: фен та чайник. Тому відфільтруємо їх і отримаємо результати, які подані в таблиці 3.3.

Таблиця 3.3 – Результат фільтрації з мінімальною підтримкою

Товар	Множина TID
Блендер	1, 5, 7, 9, 15, 16, 17, 19
Холодильник	1, 3, 5, 7, 15, 16, 17, 19
Пральна машина	1, 2, 4, 8, 9, 10, 12, 14, 16, 18, 20
Мультиварка	2, 4, 6, 8, 10, 12, 14, 16, 18, 20

Продовження таблиці 3.3

Товар	Множина TID
Праска	3, 4, 7, 8, 11, 13, 15

Крок 3 – обчислити набір ідентифікаторів транзакцій для кожної пари товарів.

Тепер переходимо до парних продуктів. В основному треба повторити те саме, що й у кроці 1, але тепер для пар продуктів.

Цікава річ щодо алгоритму ECLAT полягає в тому, що цей крок виконується за допомогою перетину двох вихідних наборів. Цим він відрізняється від алгоритму Apriori.

Алгоритм ECLAT є швидшим, оскільки набагато простіше ідентифікувати перетин набору ідентифікаторів транзакцій, ніж сканувати кожну окрему транзакцію на наявність пар продуктів (як це робить Apriori). У таблиці 3.4 можна побачити, як легко відфільтрувати ідентифікатори транзакцій, які є спільними для пари продуктів «Блендер» і «Холодильник»:

Таблиця 3.4 – Фільтрація ідентифікаторів транзакцій, які є спільними для пари продуктів «Блендер» і «Холодильник»

Блендер	1	-	5	7	9	15	16	17	19
Холодильник	1	3	5	7	-	15	16	17	19

Під час перетину для кожної пари товарів (ігноруючи товари, які не отримали підтримки окремо), виходять результати, подані у таблиці 3.5:

Таблиця 3.5 – Результати перетину кожної пари товарів

Пара товарів	Множина TID
блендер, холодильник	1, 5, 7, 15, 16, 17, 19
блендер, пральна машина	1, 9, 16

Продовження таблиці 3.5

Пара товарів	Множина TID
блендер, мультиварка	16
блендер, праска	7, 15
холодильник, пральна машина	1, 16
холодильник, мультиварка	16
холодильник, праска	3, 7, 15
пральна машина, мультиварка	2, 4, 8, 10, 12, 14, 16, 18, 20
пральна машина, праска	4, 8
мультиварка, праска	4, 8

Крок 4 – відфільтрувати пари, які не досягають мінімальної підтримки.

Як і раніше, нам потрібно відфільтрувати результати, які не досягають мінімальної підтримки. Таким чином, ми залишаємо лише дві пари продуктів: блендер і холодильник, пральна машина і мультиварка. Результаті подані в таблиці 3.6.

Таблиця 3.6 – Відфільтровані пари, які не досягають мінімальної підтримки

Пара товарів	Множина TID
блендер, холодильник	1, 5, 7, 15, 16, 17, 19
пральна машина, мультиварка	2, 4, 8, 10, 12, 14, 16, 18, 20

Крок 5 – продовжувати, поки є змога створювати нові пари з підтримкою.

З цього моменту повторюємо кроки якомога довше. У поточному прикладі, якщо ми створимо пари продуктів із трьох продуктів, мі побачимо, що жодна група з трьох продуктів не досягає мінімального рівня підтримки. Тому правила асоціації будуть такими, якими були отримані на попередньому кроці.

За допомогою розробленого програмного забезпечення мовою програмування Java виконаємо розрахунки за алгоритмом ECLAT. Результати подані на рисунку 3.1.

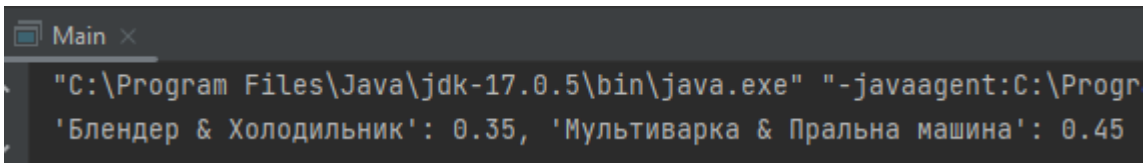

Рисунок 3.1 – Результати роботи ПЗ для алгоритму ECLAT

Інтерпретація цього результату полягає в тому, що в транзакціях нашого магазину є дві комбінації товарів, які є відносно сильними. Люди часто купують блендер та холодильник разом. Люди також часто купують мультиварку та пральну машину разом. Очевидно, було б гарною ідеєю поєднати ці продукти разом, щоб люди могли легко дістатися до обох пар. Або, можливо, власник магазину міг би подумати про упаковку продуктів у привабливу пропозицію, щоб ще більше збільшити продаж цих продуктів.

3.2 Дослідження алгоритму Apriori

Почнемо з початку: у нас є набір даних, у якому клієнти купують кілька продуктів. Мета – з’ясувати, які комбінації продуктів найчастіше купують разом.

Вам потрібно організувати дані таким чином, щоб у кожному рядку був набір продуктів. Кожен із цих наборів містить продукти, які були придбані в одній транзакції.

Найелементарнішим рішенням було б прокрутити всі транзакції, а всередині транзакції перебрати всі комбінації продуктів і підрахувати їх. На жаль, це займе надто багато часу, тому нам потрібно щось краще.

Двоє вчених Агравал і Срікант були першими, хто запропонував вирішення цієї проблеми у своїй статті 1994 року під назвою «Швидкі алгоритми для правил асоціації майнінгу». Їхнє перше рішення – знаменитий алгоритм Апріорі.

Візьмемо той самий датасет, який використовувався для дослідження ECLAT.

Крок 1 – розрахунок підтримки для кожного окремого елемента.

Алгоритм заснований на понятті підтримки. Підтримка – це просто кількість транзакцій, у яких виникає певний товар (або комбінація продуктів).

В межах першого кроку алгоритму є обчислення підтримки кожного окремого елемента. Це в основному зводиться лише до підрахунку кількості транзакцій для кожного продукту. Співвідношення подані у таблиці 3.7.

Таблиця 3.7 – Співвідношення товарів та кількості випадків

Товар	Кількість випадків
Блендер	8
Холодильник	8
Пральна машина	11
Мультиварка	10
Праска	7
Фен	6
Чайник	6

Крок 2 – визначення порогу підтримки.

Тепер, коли ми маємо підтримку для кожного окремого продукту, ми використаємо її, щоб відфільтрувати деякі продукти, які не є частими. Для цього нам потрібно визначитися з порогом підтримки. Для нашого поточного прикладу пропонується використовувати 7 як мінімум.

Крок 3 – вибір частих елементів.

Легко побачити, що є два окремі продукти, які мають підтримку менше 7 (це означає, що вони мають менше 7 випадків). Це фен і чайник.

Тож, на жаль, для фену та чайнику, вони не будуть розглянуті на подальших етапах.

Крок 4 – пошук підтримки частих наборів елементів.

Наступним кроком є проведення такого ж аналізу, але тепер із використанням пар продуктів замість окремих продуктів. Як ви можете собі уявити, кількість комбінацій тут може швидко стати великою, особливо якщо у вас є дуже велика кількість продуктів.

Великий «винахід» за алгоритмом Apriori полягає в тому, що ми будемо безпосередньо ігнорувати всі пари, які містять будь-який з нечастих елементів. Завдяки цьому ми маємо набагато менше пар елементів для сканування.

Перелік всіх пар, які містять чайник, фен або обидва – все це можна відкинути, і це прискорить виконання наших підрахунків. Результати подано у таблиці 3.8.

Таблиця 3.8 – Співвідношення множини товарів та кількості випадків з вилученими товарами – чайник та фен

Множина товарів	Кількість випадків
блендер, холодильник	7
блендер, пральна машина	2
блендер, мультиварка	1
блендер, праска	2
холодильник, пральна машина	2
холодильник, мультиварка	1
холодильник, праска	3
пральна машина, мультиварка	9
пральна машина, праска	2
мультиварка, праска	2

Як можна побачити, на даний момент залишилося лише дві комбінації продуктів, які перевищують підтримку: блендер і холодильник, а також пральна машина і мультиварка.

Крок 5 – повторити для більших наборів.

Тепер повторимо те ж саме для наборів з трьох продуктів. Як і раніше, ми не будемо рахувати сети, які були вилучені на попередньому кроці.

Залишилися пари:

– блендер, холодильник;

– пральна машина, мультиварка.

Отже, які трійки ми можемо зробити, щоб не містити жодної пари, яку вибули?

Блендер, холодильник, пральна машина – виключено через наявність нечастої пари (наприклад, холодильник, пральна машина).

Блендер, холодильник, мультиварка – вилучено через наявність нечастої пари (наприклад, холодильник, мултиварка).

Пральна машина, мультиварка, блендер – вилучено через наявність нечастої пари (наприклад, блендер, мультиварка).

Холодильник, пральна машина, мультиварка – вилучено через наявність нечастої пари (наприклад, холодильник, пральна машина).

Немає трійок, які не містять жодної нечастої пари. Це означає, що часті набори визначено, і це пари блендер, холодильник і пральна машина і мультиварка.

Крок 6 – створити правила асоціації та обчислити надійність.

Тепер, коли у нас є найбільші часті набори елементів, наступним кроком є їх перетворення на правила асоціації. Правила асоціації йдуть далі, ніж просто перелік продуктів, які часто зустрічаються разом.

Правила асоціації записуються у форматі: продукт X => продукт Y. Це означає, що отримується правило, яке говорить вам, що якщо купується продукт X, також, ймовірно, буде придбано продукт Y.

Є додатковий показник, який називається достовірність (confidence). Достовірність говорить вам про відсоток випадків, у яких це правило є дійсним. 100% впевненість означає, що цей зв'язок має місце завжди; Наприклад, 50% означає, що правило діє лише 50% часу.

Крок 7 – обчислити підйом (lift).

Отримавши правила, останнім кроком є обчислення підйому кожного правила. Згідно з визначенням, підйом правила – це показник продуктивності, який вказує на силу зв’язку між продуктами в правилі.

Це означає, що lift в основному порівнює покращення правила асоціації із загальним набором даних. Якщо «будь-який продукт => X» у 10% випадків, тоді як «A => X» у 75% випадків, покращення становитиме 75% / 10%, тобто 7,5.

Якщо підйом правила дорівнює 1, то продукти незалежні один від одного. Будь-яке правило, яке має підйом 1, можна скасувати.

Якщо підйом правила перевищує 1, значення підйому говорить про те, наскільки сильно правий добуток залежить від лівого.

Даний алгоритм дає три показника: support, confidence, lift.

Усі ці три показники мають власну достовірність. Тому вибрати між ними складно. Наприклад, якщо у нас є правило, яке має більший підйом, але нижчу надійність, ніж інше правило, буде важко стверджувати, що одне правило «краще за інше». На цьому етапі ми зможемо просто зберегти обидва правила або спробувати знайти причину віддати перевагу одному показнику над іншим у нашому конкретному випадку використання.

Встановлення параметрів для алгоритму та запуск алгоритму.

Наступним кроком ми застосуємо апріорний алгоритм до цих даних. Нам потрібно буде встановити два параметри:

- min_support: це поріг підтримки, який було пояснено декілька раніше. Невелика відмінність полягає в тому, що тут він виражений у відсотках, а не числом. Встановимо, як 7 / кількість транзакцій;
- min_confidence: це лише фільтр, який відфільтровує правила, які не відповідають мінімальній достовірності. Можна поставити його на нуль, якщо хочемо бачити всі згенеровані правила.

За допомогою розробленого програмного забезпечення виконаємо розрахунки для алгоритму Apriori, результат яких подано на рисунку 3.2.

Як висновок, за цими даними можна стверджувати, що підйом правил Блендер => Холодильник і Холодильник => Блендер є дуже високим. Власник цього магазину, ймовірно, захоче поставити холодильник і блендер поруч один з одним. Асоціації Мультиварка => Пральна машина та Пральна машина =>

Мультиварка дещо менш сильні, але достатньо високі, щоб також поставити пральну машину та мультиварку в одне й те саме місце в магазині.

В цілому, алгоритм Apriori може бути поданий у вигляді блок-схеми, поданій на рисунку 3.3.

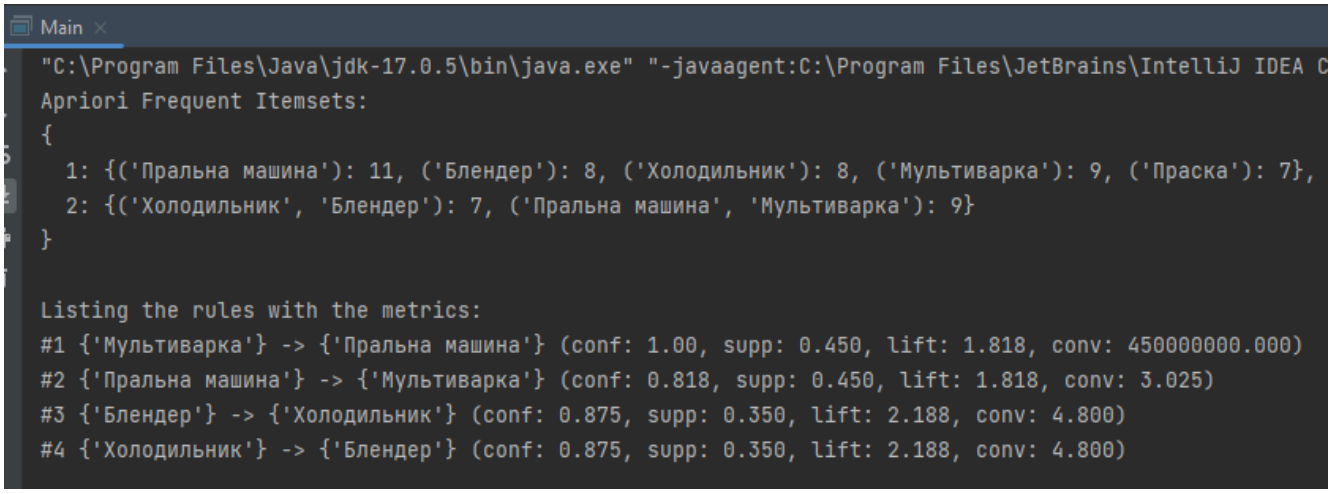

Рисунок 3.2 – Результати роботи ПЗ для алгоритму Apriori

3.3 Дослідження алгоритму FP-Growth

Frequent Itemset Mining – це технічний термін для пошуку комбінацій продуктів, які часто купуються разом. Зазвичай починається зі списку транзакцій, у якому кожна транзакція представлена як список товарів.

Метою Frequent Itemset Mining є виявлення комбінацій продуктів, які часто зустрічаються, за допомогою швидкого та ефективного алгоритму.

Алгоритм FP-Growth можна розглядати як сучасну версію Apriori, оскільки він швидший і ефективніший при досягненні тієї ж мети.

До речі, алгоритми Frequent Itemset Mining не залежать від домену: ми можемо використовувати частий itemset mining для інших полів, крім аналізу кошика.

Ідея алгоритму FP-Growth полягає в тому, щоб знаходити часті набори елементів у наборі даних, але робити це швидше, ніж алгоритм Apriori. Алгоритм Apriori, в основному, повертається до набору даних, щоб перевірити наявність продуктів у наборі даних.

Щоб бути швидшим, алгоритм FP змінив організацію даних у вигляді дерева, а не наборів. Ця деревовидна структура даних забезпечує швидше сканування, і саме тут алгоритм виграє час.

Крок 1 – підрахунок випадків появи окремих елементів.

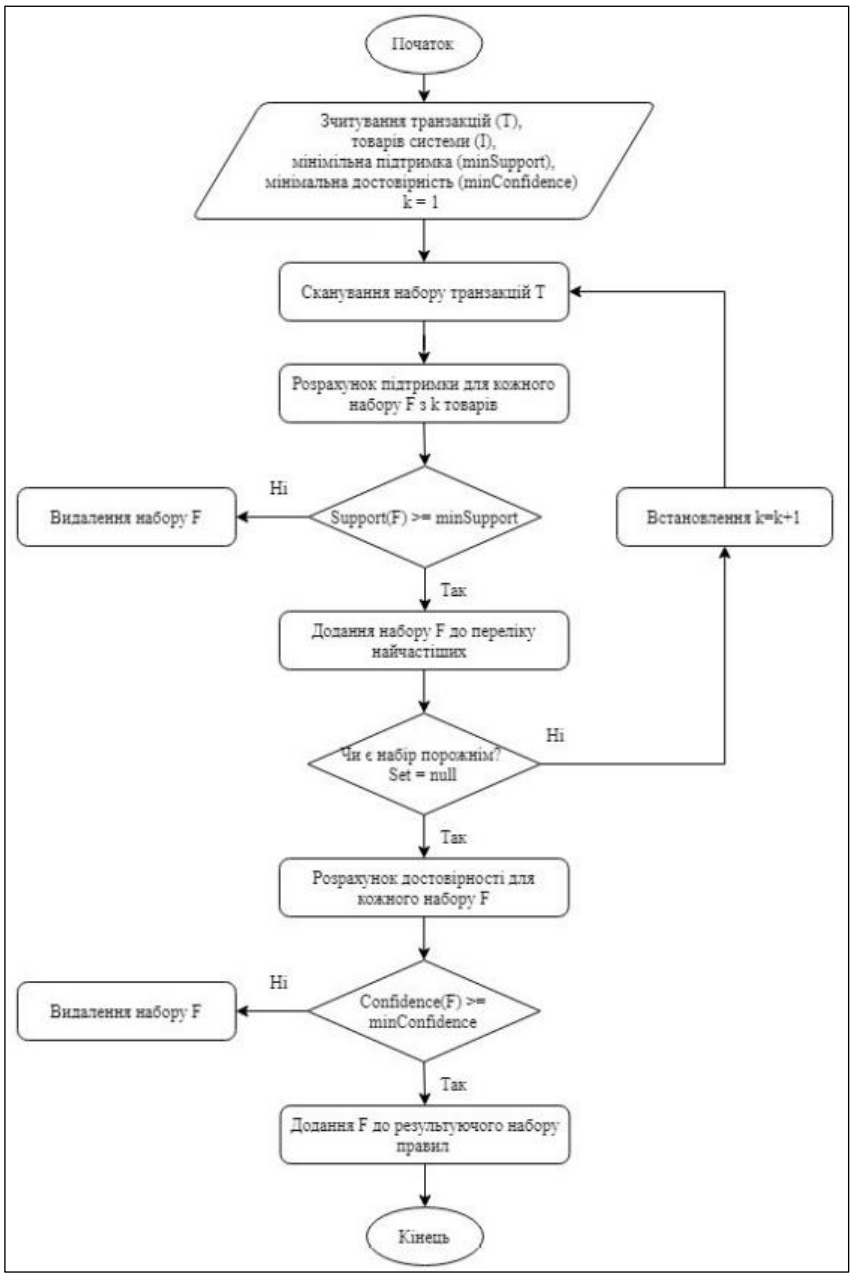

Рисунок 3.3 – Блок-схема алгоритму Apriori

Першим кроком алгоритму FP-Growth є підрахунок поява окремих елементів. У таблиці 3.9 наведено співвідношення товарів та кількості випадків.

Крок 2 – відфільтрувати нечасті елементи за допомогою мінімальної підтримки.

Таблиця 3.9 – Співвідношення товарів та кількості випадків

Товар	Кількість випадків
Блендер	8
Холодильник	8
Пральна машина	11
Мультиварка	10
Праска	7
Фен	6
Чайник	6

Нам потрібно визначитися зі значенням мінімальної підтримки: кожен елемент або набір елементів із меншою кількістю входжень, ніж мінімальна підтримка, буде виключено.

У нашому прикладі виберемо мінімальну підтримку 7. Це означає, що ми збираємося відкинути предмети фен та чайник.

Крок 3 – упорядкування наборів товарів на основі окремих випадків.

Для решти елементів ми створимо впорядковану таблицю. Ця таблиця міститиме елементи, які ще не відхилено, а елементи в транзакції буде впорядковано на основі окремого товару. Упорядкування наборів товарі на основні окремих випадків подано у таблиці 3.10.

Таблиця 3.10 – Упорядкування наборів товарів на основі окремих випадків

Транзакція	Товари	Упорядковані товари
1	пральна машина, блендер, холодильник	пральна машина, блендер, холодильник
2	пральна машина, мультиварка	пральна машина, мультиварка
3	праска, фен, чайник, холодильник	холодильник, праска
4	праска, фен, чайник, пральна машина,	пральна машина,

Кінець таблиці 3.10

Транзакція	Товари	Упорядковані товари
	мультиварка	мультиварка, праска
5	блендер, холодильник	блендер, холодильник
6	мультиварка	мультиварка
7	праска, фен, чайник, пральна машина, блендер, холодильник	блендер, холодильник, праска
8	праска, фен, чайник, пральна машина, мультиварка	пральна машина, мультиварка, праска
9	блендер, пральна машина	пральна машина, блендер
10	пральна машина, мультиварка	пральна машина, мультиварка
11	чайник, праска	праска
12	пральна машина, мультиварка	пральна машина, мультиварка
13	фен, праска	праска
14	пральна машина, мультиварка	пральна машина, мультиварка
15	праска, фен, чайник, блендер, холодильник	блендер, холодильник, праска
16	пральна машина, блендер, мультиварка, холодильник	пральна машина, мультиварка, блендер, праска
17	блендер, холодильник	блендер, холодильник
18	пральна машина, мультиварка	пральна машина, мультиварка
19	блендер, холодильник	блендер, холодильник
20	пральна машина, мультиварка	пральна машина, мультиварка

Крок 4 – створення дерева та додання транзакції одну за одною.

Тепер ми можемо створити дерево, починаючи з першої транзакції. Кожен продукт є вузлом у дереві, а саме – за допомогою програмного забезпечення отримаємо фрейм даних, поданий у таблиці 3.11.

Таблиця 3.11 – Фрейм даних, згенерований ПЗ, для кроку 4

	Пральна машина	Чайник	Холодильник	Праска	Фен	Мультиварка	Блендер
0	True	False	True	False	False	False	True
1	True	False	False	False	False	True	False
2	False	True	True	True	True	False	False
3	True	True	False	True	True	True	False
4	False	False	True	False	False	False	True
5	False	False	False	False	False	True	False
6	False	True	True	True	True	False	True
7	True	True	False	True	True	True	False
8	True	False	False	False	False	False	True
9	True	False	False	False	False	True	False
10	False	True	False	True	False	False	False
11	True	False	False	False	False	True	False
12	False	False	False	True	True	False	False
13	True	False	False	False	False	True	False
14	False	True	True	True	True	False	True
15	True	False	True	False	False	True	True
16	False	False	True	False	False	False	True
17	True	False	False	False	False	True	False
18	False	False	True	False	False	False	True
19	True	False	False	False	False	True	False

Наступним кроком є обчислення частих наборів елементів. За допомогою розробленого програмного забезпечення виконаємо розрахунки для алгоритму FP-Growth. Результати демонструються на рисунку 3.4.

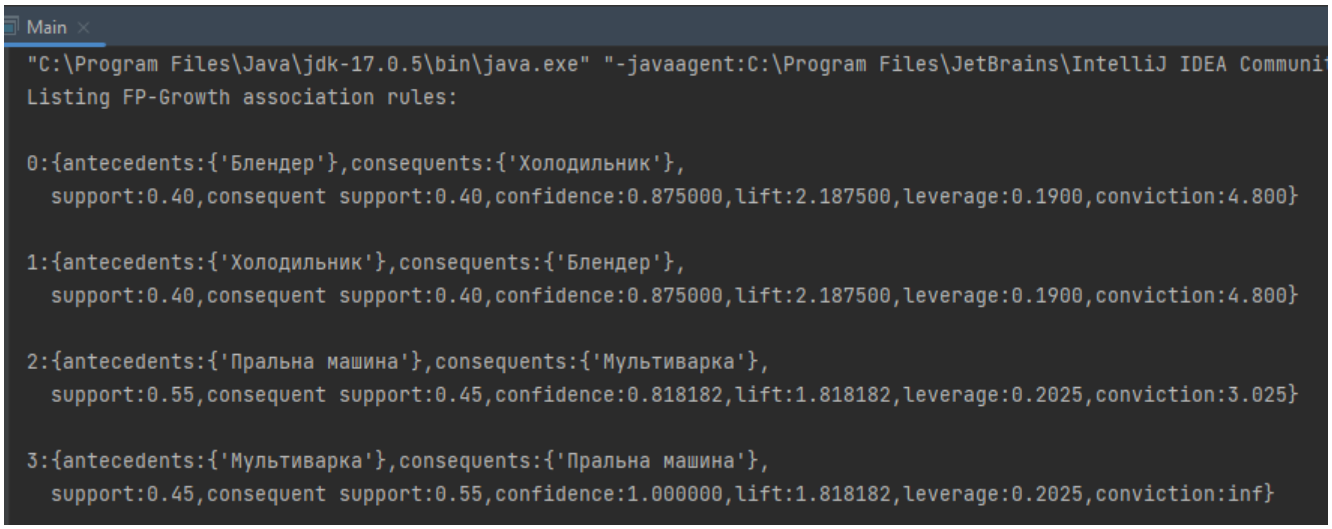

Рисунок 3.4 – Результати роботи ПЗ для алгоритму FP-Growth

Тепер ми переходимо до заключної частини: тлумачення правил і показників, які були згенеровані алгоритмом FP-Growth.

По-перше, ми можемо зробити висновок, що існує дві комбінації продуктів, і обидві асоціації є двонаправленими. Люди, які купують холодильник, також купують блендер, а люди, які купують блендер, також купують холодильник. Окремо ми бачимо, що люди, які купують мультиварку, купують і пральні машини, і навпаки.

Друга річ, на яку цікаво подивитися, це показники правил. Разом вони говорять нам дещо про надійність правил. Важливо звернути увагу на такі три показники:

– support повідомляє нам, скільки разів або у відсотках разом зустрічаються продукти;
– confidence говорить нам, скільки разів правило трапляється. Це можна сформулювати інакше, як умовну ймовірність правої частини, заданої лівою частиною;
– lift дає нам силу асоціації.

В цілому, алгоритм має такі кроки:

- підрахунок частоти появи кожного елемента набору даних;
- видалення елементів, що рідко зустрічаються, шляхом застосування мінімальної підтримки (min_support) для визначення порога;
- побудова FP-дерева з елементів, що залишилися. FP-дерево є структурою даних, яка представляє частотні патерни у наборі даних;
- для кожного елемента у FP-дереві: побудова умовної бази даних, що містить усі шляхи в дереві, що містять цей елемент; рекурсивне застосування алгоритму FP-growth до умовної бази для побудови піддерева. комбінування піддерева з поточним елементом для отримання частих патернів, які називаються FP-патернами;
- повторення кроку 4 для кожного елемента у FP-дереві для отримання всіх можливих FP-патернів;
- виведення частих патернів як результат.

Алгоритм FP-growth дозволяє ефективно знаходити часті патерни у великих обсягах даних, особливо коли набір даних розріджений. Він уникає необхідності явного генерування всіх можливих комбінацій патернів, що робить його швидким та масштабованим для обробки великих наборів даних.

4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ

У таблиці 4.1 дуже коротко подається різниця між трьома алгоритмами за деякими основними / загальними параметрами.

Таблиця 4.1 – Порівняння алгоритмів Apriori, FP-Growth, Eclat

	Apriori	FP-Growth	Eclat
Техніка	Пошук вшир	Розділяй і володарюй	Пошук вшир і перетинання id транзакцій
Сканування бази	База сканується кожен час, коли генерується множина candidate item	Сканується тільки два рази	Сканується декілька разів
Переваги	Легко реалізувати; використовує властивість великого набору елементів	База санується лише два рази	Немає потреби сканування бази кожен раз
Недоліки	Потребує багато пам’яті, забагато набору елементів-кандидатів	FP-дерево дороге для створення – споживає більше пам’яті	Потребує віртуальної пам’яті для виконання транзакції
Формат даних	Горизонтальний	Горизонтальний	Вертикальний
Формат зберігання	Масив	Дерево	Масив
Час	Більший час виконання	Менший, ніж у Apriori	Менший, ніж у Apriori

Далі порівняємо ефективність метрикою Runtime залежно від щільності датасета і довжин транзакцій датасета.

Під щільністю датасета передбачається відношення транзакцій, що містять у собі часті елементи до загальної кількості транзакцій. Щільність набору даних в даному випадку визначає відсоток кошиків, які навмисно містять часті набори елементів. Зі збільшенням частої щільності набору елементів збільшується і час обробки Apriori, Eclat і FP-Growth, як показано на рисунку 4.1.

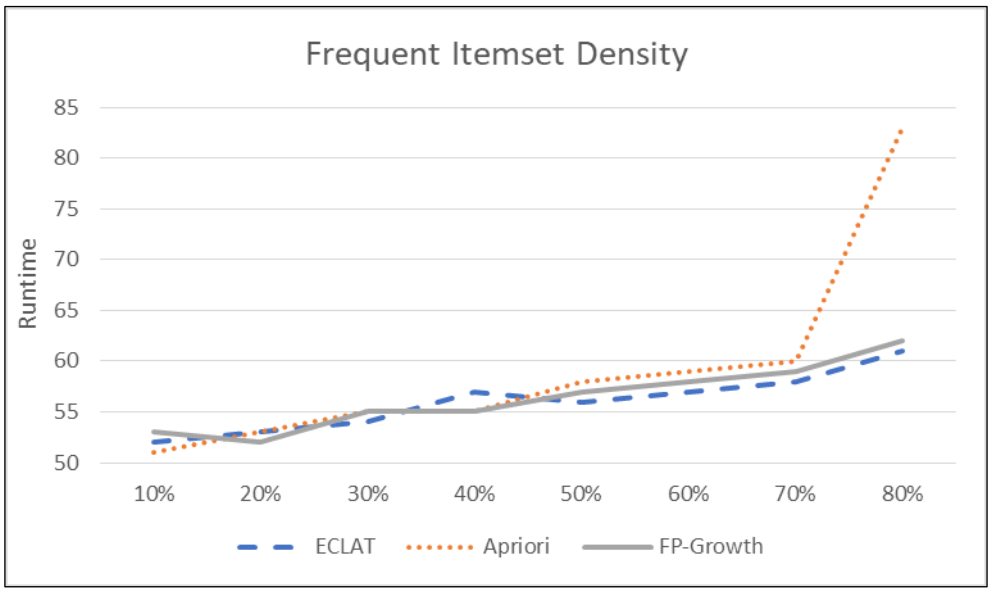

Рисунок 4.1 – Залежність роботи алгоритмів від щільності датасета

Дана діаграма показує результати використання 10 мільйонів кошиків із середнім розміром 50 кошиків різної щільності. Алгоритми Eclat і FP-Growth показують дуже схоже зростання, оскільки часто збільшується щільність набору елементів. Алгоритм Apriori також працює дуже подібно до Eclat і FP-Growth, поки щільність не перевищує 70%. Як згадувалося раніше, Apriori потребує значно більше пам’яті, ніж інші алгоритми. На 70% Apriori виділила всі 8 гігабайт оперативної пам’яті тестової машини. Це призвело до необхідності переходу на фізичну пам’ять і суттєво вплинуло на час роботи алгоритму. Також цікаво відзначити, що Eclat трохи випереджає FP-Growth при низькій щільності. Цей рейтинг змінюється при більшій щільності. Між 10% і 70% всі три алгоритми демонструють приблизно O(NlogN) складність. Понад 70% апріорі наближається

до O(N2) і має гіршу складність, де N – кількість фактичних частих елементів у базі даних.

На рисунку 4.2 подається графік залежності роботи алгоритмів від довжини транзакцій датасета.

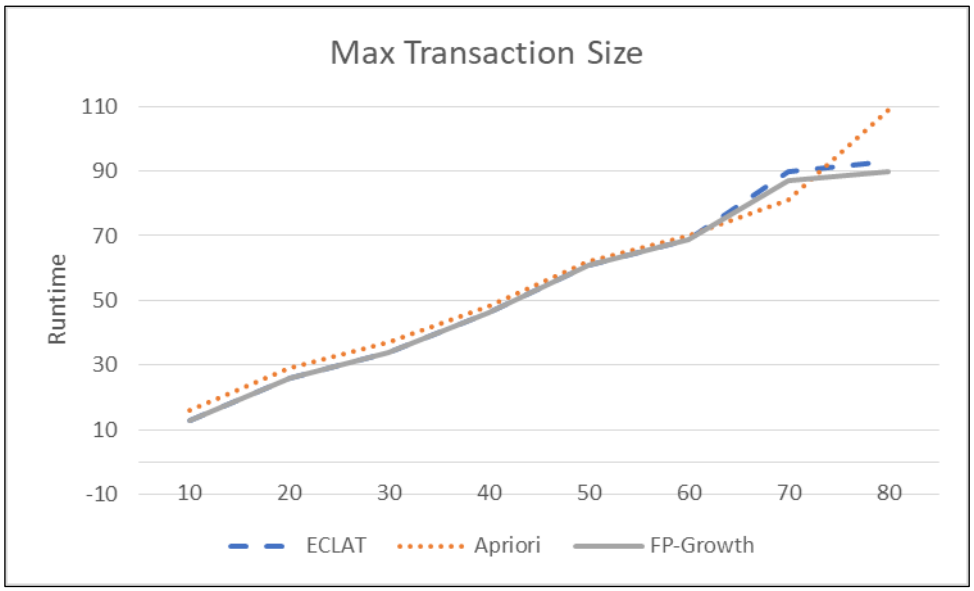

Рисунок 4.2 – Залежність роботи алгоритмів від довжини транзакцій датасета

Звіди, очевидно, що зі збільшенням довжини транзакції Apriori також справляється набагато гірше.

Ця діаграма показує результати запуску 10 мільйонів кошиків, із частою щільністю набору товарів 50 %, при різному максимальному розміру кошика.

Розмір кошика визначає максимальну кількість товарів у рядку кошика. Більший розмір кошика означає, що часті набори товарів також будуть більшими. Це збільшує розміри структур даних, які використовуються для зберігання цих наборів елементів. Ці більші структури даних вимагають більше пам’яті для зберігання та більшого часу обробки для їх проходження.

На цій діаграмі показано результати використання 10 мільйонів кошиків із частою щільністю набору елементів 50% при різних максимальних розмірах кошика. Три алгоритми демонструють майже однакову продуктивність O(N) для розмірів кошика до 60. Після того, як розмір кошика перевищує 60, Apriori, здається, зростає набагато швидше, ніж два інших. Можливо, це пов’язано з збільшенням пам’яті, яку використовує Апріорі. Цікаво, що Apriori показав

найкращі результати між 60-70 максимальними розмірами транзакцій. Потрібні подальші дослідження, щоб визначити, чому Apriori є кращим у цьому невеликому діапазоні.

Далі описується, як підбирати гіперпараметри наших моделей (вони ж ті самі support, confidence і т.д.).

Оскільки алгоритми ARules чутливі до гіперпараметрів, необхідно грамотно підійти до питання про вибір. Різниця в часі виконання при різних комбінаціях гіперпараметрів може суттєво відрізнятись.

Основним гіперпараметром у будь-якому алгоритмі ARules є min support. Він, власне кажучи, відповідає за мінімальну відносну частоту асоціативного правила у вибірці. Вибираючи цей показник необхідно насамперед орієнтуватися на поставлене завдання. Найчастіше замовнику потрібна невелика кількість топових комбінацій, тому можна ставити високе значення min support. Однак, у такому разі до нашої вибірки можуть потрапити якісь викиди (bundle-товари, наприклад) та не потрапити якісь цікаві комбінації. Загалом і в цілому, чим вище ви ставите значення супорт, тим потужніші правила знаходяться, і тим швидше вважається алгоритм. Рекомендується при першому прогоні ставити менше значення, щоб зрозуміти, які пари, трійки і т.д. товарів взагалі зустрічаються у датасеті. Потім поступово збільшувати крок, добираючись до потрібного значення (заданого замовником, наприклад). Насправді хорошим значенням min supp буде і 0.1% (при дуже великому датасеті).

На рисунку 4.3 наводиться зразковий графік залежності часу виконання алгоритму від даного показника.

Частий аналіз шаблонів є найважливішим кроком у правилах асоціації, який, зрештою, допомагає нам у багатьох програмах, таких як аналіз ринкового кошика, кластеризація, ігри, аналіз серій, аналіз об’єктів, прийняття рішень, навігація веб-сайтів тощо. Виявлено, що Apriori використовує метод об’єднання та скорочення, Ecalt працює з вертикальними наборами даних, а FP-Growth створює умовне дерево шаблонів, яке задовольняє мінімальну підтримку.

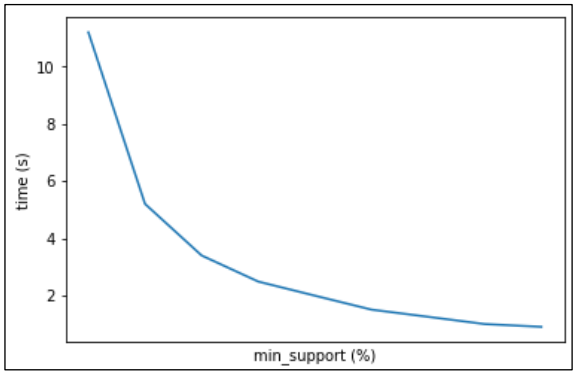

Рисунок 4.3 – Графік залежності часу виконання алгоритму від показника min_support

Основним недоліком алгоритму Apriori є створення великої кількості наборів елементів-кандидатів і великої кількості сканувань бази даних, яка дорівнює максимальній довжині набору частих елементів. Дуже дорого сканувати велику базу даних. Справжньою причиною апріорної невдачі є відсутність ефективного методу обробки бази даних. FP-Growth є найкращим серед трьох алгоритмів і, таким чином, найбільш масштабований. Eclat працює гірше, ніж FP-Growth, а Apriori – гірше.

Apriori – це зрозумілий алгоритм частого видобутку набору елементів. Через це Apriori є популярною відправною точкою для частого вивчення набору елементів. Однак Apriori має серйозні проблеми з масштабованістю та витрачає доступну пам’ять набагато швидше, ніж Eclat і FP-Growth. Через це Apriori не слід використовувати для великих наборів даних. Найбільш частим додаткам itemset слід розглянути можливість використання FP-Growth або Eclat. Ці два алгоритми працювали однаково для дослідження, хоча FP-Growth дійсно показав трохи кращу продуктивність, ніж Eclat. Інші документи також рекомендують FP-Growt для більшості випадків. Частий видобуток наборів предметів є областю активних досліджень. Часто вводяться нові алгоритми, а також модифікації існуючих алгоритмів. Для програми, де продуктивність має вирішальне значення, важливо оцінювати набір даних за допомогою новіших алгоритмів, щойно вони впроваджуються та демонструють кращу продуктивність, ніж FP-Growth або Eclat.

ВИСНОВКИ

У ході виконання даної дослідницької роботи було проведено та описано аналіз предметної галузі магазину побутової техніки із вказанням основних бізнес-процесів та бізнес-функцій. Проведено аналіз існуючих аналогів електронної комерції з їх оцінкою переваг та недоліків, де було розглянуто 2 аналоги з демонстрацією їх зовнішнього вигляду та оцінкою щодо їх функціональності.

Було проведено дослідження та детальний опис різноманітних алгоритмів, які можуть бути використані для CRM-системи, такі як метод кластеризації даних – k-means, колаборативна фільтрація, фільтрація на базі контенту та асоціативні правила. До кожного з цих алгоритмів було подано детальний опис з усіма формулами, які потрібні для розрахунків результатів. Більше уваги для дослідження було віддано асоціативним алгоритмам, такими як Apriori, FP-Growth та Eclat. За кожним алгоритмом було приведено кроки виконання з прикладами на основі даної предметної галузі, з виведенням результатів за допомогою розробленого програмного забезпечення для кожного алгоритму мовою програмування Java. Для кожного з алгоритмів було не тільки представлено недоліки та переваги, але також було проведено збір та аналіз результатів та метрик кожного з них. Побудовано діаграми, які визначають продуктивність та ефективність алгоритмів в залежності від щільності набору даних та довжини транзакції набору даних. Було виявлено, що алгоритм Apriori показав найгірші результати зі збільшенням довжини транзакцій, а також зі збільшенням щільності. Найліпші результати були продемонстровані алгоритмом FP-Growth, який є удосконаленою версією алгоритму Apriori, особливо зі збільшенням показників щільності та довжини транзакції. Алгоритм Eclat також продемонстрував дуже гарні результати, але вони зовсім трохи гірші за результати FP-Growth, але на маленьких показниках транзакцій та щільності алгоритм Eclat показав себе трішечки краще за FP-Growth.

Падіння продуктивності та ефективності алгоритму Apriori виконується за рахунок використання великої кількості пам’яті (один з недоліків). Коли пам’яті не вистачає, використовується пам’ять з диску та продуктивність сильно спадає.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Jigui Sun, Jie Liu and Lianyu Zhao, "Clustering algorithms Research", Journal of Software, vol. 19, no. 1, pp. 48-61, January 2008.

2. Shibao Sun and Keyun Qin, "Research on Modified k-means Data Cluster Algorithm", I. S. Jacobs and C. P. Bean “Fine particles thin films and exchange anisotropy” Computer Engineering, vol. 33, no. 13, pp. 200-201, July 2007.

3. Basu, C., Hirsh, H., Cohen, W.W.: Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In Proceedings of the Fifteenth National Conference on Artificial Intelligence. (1998) Madison, Wisconsin. AAAI Press p. 714-720.

4. Balabanovíc, M., Shoham, Y.: Fab: Content-Based, Collaborative Recommendation. Communications of the ACM, (1997) 40(3): p. 66-72.

5. Canny, J.: Collaborative Filtering with Privacy via Factor Analysis. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in infor-mation retrieval. (2002) Tampere, Finland. ACM Press p. 238-245.

6. J. Han, H. Cheng, D. Xin, and X. Yan, “Frequent pattern mining: Current status and future directions,” Data Mining Knowledge Discovery, vol. 15, no. 1, pp. 55–86, Aug. 2007.

7. Z. Zheng, R. Kohavi, and L. Mason, “Real world performance of association rule algorithms,” in Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2001, pp. 401–406.

8. R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules ´ between sets of items in large databases,” in ACM SIGMOD Record, vol. 22, no. 2. ACM, 1993, pp. 207–216.

9. M. J. Zaki, S. Parthasarathy, M. Ogihara, W. Li et al., “New algorithms for fast discovery of association rules,” in KDD, vol. 97, 1997, pp. 283– 286.

10. R. Agrawal, R. Srikant et al., “Fast algorithms for mining association rules,” in Proceedings of the 20th international conference of very large data bases, VLDB, vol. 1215, 1994, pp. 487–499.