

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «магістр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Розробка веб-додатку з реалізацією адаптивного
алгоритму генерування релевантних пропозицій
користувачам інтернет магазину»**

Виконав студент групи ПЗМ-22-2
ГОРНИЦЬКИЙ Олексій Валентинович

Керівник роботи:
ЛОБАНЧИКОВА Надія Миколаївна

Рецензент:
ВАКАЛЮК Тетяна Анатоліївна

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
 ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
 ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

зав. кафедри інженерії
 програмного забезпечення
 Тетяна ВАКАЛЮК
 «18» вересня 2023 р.

ЗАВДАННЯ на кваліфікаційну роботу

Здобувач вищої освіти: ГОРНИЦЬКИЙ Олексій Валентинович

Керівник роботи: ЛОБАНЧИКОВА Надія Миколаївна

Тема роботи: **«Розробка веб-додатку з реалізацією адаптивного алгоритму генерування релевантних пропозицій користувачам інтернет магазину»**,
 затверджена Наказом закладу вищої освіти від **«18» вересня 2023 р. № 479/с**

Вихідні дані для роботи: об'єктом дослідження є технології розробки веб-додатку з реалізацією адаптивного алгоритму генерування релевантних пропозицій користувачам інтернет магазину. Предметом дослідження є моделі, методи та алгоритми розробки веб-додатку з реалізацією адаптивного алгоритму генерування релевантних пропозицій користувачам інтернет магазину.

Консультанти з магістерської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Лобанчикова Н.М.	22.09.2023р.	22.09.2023р.
2	Лобанчикова Н.М.	22.09.2023р.	22.09.2023р.
3	Лобанчикова Н.М.	22.09.2023р.	22.09.2023р.

РЕФЕРАТ

Метою кваліфікаційної роботи є реалізація програмного додатку, що реалізує алгоритм рекомендацій клієнтів інтернет магазину.

Об'єктом дослідження є технології розробки веб-додатку з реалізацією адаптивного алгоритму генерування релевантних пропозицій користувачам інтернет магазину.

Предметом дослідження є моделі, методи та алгоритми розробки веб-додатку з реалізацією адаптивного алгоритму генерування релевантних пропозицій користувачам інтернет магазину.

Теоретична значущість роботи полягає у розкритті теоретичних основ реалізації алгоритмів рекомендацій в інформаційних системах.

Практична цінність полягає в дослідженні шляхів впровадження новітніх технологій у виробництві та покращення фінансово-економічної стабільності підприємства шляхом забезпечення стабільної роботи бізнес-клієнтів підприємства за допомогою поширених алгоритмів рекомендацій та створення ефективнішого підбору контенту для клієнта щоб максимізувати економічний ефект від діяльності інтернет магазину за рахунок підбору оптимальних пропозицій для користувача.

Пояснювальна записка до кваліфікаційної роботи магістра на тему «Розробка веб-додатку з реалізацією адаптивного алгоритму генерування релевантних пропозицій користувачам інтернет магазину» складається з вступу, трьох розділів, висновків, списку використаної літератури та додатків.

Текстова частина викладена на 56 сторінках друкованого тексту.

Список використаних джерел містить 21 найменування і займає 3 сторінки. В роботі наведено 23 рисунка. Загальний обсяг роботи – 56 сторінок.

У першому розділі було проведено порівняння алгоритмів рекомендацій на основі вмісту та колабораційного алгоритму. Також було виділено переваги та недоліки кожного з них. Також було розглянуто можливість комбінування алгоритмів для досягнення різноманіття релевантності.

У другому розділі було проведено пошук механізмів генерування рекомендацій для створення веб-додатку з персоналізованими рекомендаціями.

У третьому розділі було розроблено модель мікросервісного додатку з взаємодією фреймворка Shopware та системи рекомендації на основі бібліотеки PredictionIO.

Висновки містять в собі результати виконаної роботи проектування, розробки та візуалізації рекомендацій на основі колаборативного алгоритму за допомогою Universal recommender та бібліотеки predictionIO.

КЛЮЧОВІ СЛОВА: аналіз даних, статистичний набір, модель класифікації, атрибут, рекомендація, кореляція, ймовірність.

ABSTRACT

The goal of the qualification work is the implementation of a software application that implements the algorithm of recommendations for online store customers.

The object of the research is the technology of developing a web application with the implementation of an adaptive algorithm for generating relevant offers to users of an online store.

The subject of the study is models, methods and algorithms for developing a web application with the implementation of an adaptive algorithm for generating relevant offers to online store users.

The theoretical significance of the work lies in the disclosure of the theoretical foundations of implementation of recommendation algorithms in information systems.

The practical value is in the study of ways to introduce the latest technologies in production and improve the financial and economic stability of the enterprise by ensuring the stable operation of business clients of the enterprise with the help of common recommendation algorithms and creating a more effective selection of content for the client in order to maximize the economic effect of the online store by selecting the optimal suggestions for the user.

The explanatory note to the master's thesis on the topic "Development of a web application with the implementation of an adaptive algorithm for generating relevant offers to users of an online store" consists of an introduction, three sections, conclusions, a list of used literature and appendices.

The text part is laid out on 56 pages of printed text. The list of used sources contains 21 names and occupies 3 pages. There are 23 drawings in the work. The total volume of work is 56 pages.

In the first section, a comparison of content-based recommendation algorithms and a collaborative algorithm was made. The advantages and disadvantages of each

of them were also highlighted. The possibility of combining algorithms to achieve diversity of relevance was also considered.

In the second section, recommendation generation mechanisms were searched for creating a web application with personalized recommendations.

In the third chapter, a model of a microservice application with the interaction of the Shopware framework and a recommendation system based on the PredictionIO library was developed.

The conclusions contain the results of the design, development and visualization of recommendations based on a collaborative algorithm using the Universal recommender and the predictionIO library.

KEY WORDS: data analysis, statistical set, classification model, attribute, recommendation, correlation, probability.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	8
ВСТУП	9
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПОКАЗНИКІВ КОНВЕРСІЇ ТА ЗАЛУЧЕНОСТІ ПРИ ВИКОРИСТАННІ РЕЛЕВАНТНИХ РЕКОМЕНДАЦІЙ.....	11
1.1 Дослідження показників конверсії та залученості при використанні релевантних рекомендацій	11
1.2 Формування стратегії покращення фінансово-економічної безпеки підприємства.....	14
1.3 Огляд алгоритмів генерування персоналізованих рекомендацій	16
РОЗДІЛ 2. АРГУМЕНТАЦІЯ ВИБОРУ ІНСТРУМЕНТУ РЕАЛІЗАЦІЇ СИСТЕМИ РЕКОМЕНДАЦІЙ	25
2.1 Вибір інструменту реалізації системи рекомендації	25
2.2. Постановка завдання.....	30
2.3 Проектування моделі взаємодії додатку з користувачами	32
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ	36
3.1 Створення програмного додатку	36
3.2 Створення середовища роботи додатків та їх взаємодії	41
3.3 Тестування додатку.....	46
ВИСНОВКИ.....	51
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	54

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

HTTP	протокол передачі даних, що використовується в комп'ютерних мережах. Назва скорочена від HyperText Transfer Protocol, протокол передачі гіпертекстових документів
SDK	набір із засобів розробки, утиліт і документації, який дає програмістам змогу створювати прикладні програми за визначеною технологією або для певної платформи
REST API	підхід до архітектури мережових протоколів, які надають доступ до інформаційних ресурсів
Content-based filtering	це тип системи рекомендацій, яка намагається вгадати, що може сподобатися користувачеві на основі активності цього користувача
Collaborative filtering	це техніка, яка може відфільтрувати елементи, які можуть сподобатися користувачеві, на основі реакції схожих користувачів
EventClient	клас програмного коду, який реалізує збір та передачу інформації про події в додатку
EngineClient	клас програмного коду, який реалізує створення запитів для отримання списку рекомендацій

ВСТУП

Найбільшими викликами для українського ритейлу після початку повномасштабного вторгнення стали: проблеми з постачанням товарів, брак коштів, порушення логістичних маршрутів, втрата персоналу, дефіцит пального та енергетична криза.

Ритейл – одна із ключових галузей економіки, яка гарантує фінансові надходження до бюджетів усіх рівнів і надає роботу сотням тисяч людей. Саме стабільна економіка є запорукою безпеки країни, тому ритейл, попри втрати та складну ситуацію, продовжує працювати, відкривати нові торговельні точкию.

Релевантність – це параметр, який показує, наскільки створений інтернет-магазин і кожна його сторінка пов'язана з пошуковими запитами потенційного покупця. Залежно від того, наскільки релевантними будуть сторінки, категорії та окремі картки товарів – буде залежати не тільки відвідуваність, а й підсумковий товарообіг.

Сучасні інформаційні технології значною мірою залучені для вирішення проблеми генерації релевантного контенту. Створюються розумні алгоритми індексації, класифікації даних, створення пошукових моделей [1]. Саме за допомогою таких алгоритмів вдається систематизувати дані, які надходять від користувачів, передбачати тренди та рекомендувати користувачу найбільш відповідну, потрібну та актуальну інформацію, що в свою чергу збільшує показники залучення користувачів та збільшує економічний ефект.

Метою роботи є розробка веб-додатку з реалізацією адаптивного алгоритму генерування релевантних пропозицій користувачам інтернет магазину.

Встановлена мета обумовлює наступні завдання:

- провести аналіз методів та засобів реалізації алгоритмів рекомендацій;
- провести аналіз атрибутів, необхідних для реалізації алгоритму;
- розробити додаток для перевірки роботи рекомендацій на основі колаборативного алгоритму;

- зробити висновки щодо роботи алгоритмів рекомендацій в середовищі з динамічними користувачами.

Об'єктом дослідження є технології розробки веб-додатку з реалізацією адаптивного алгоритму генерування релевантних пропозицій користувачам інтернет магазину.

Предметом дослідження є моделі, методи та алгоритми розробки веб-додатку з реалізацією адаптивного алгоритму генерування релевантних пропозицій користувачам інтернет магазину.

Методи дослідження в кваліфікаційній роботі було використано такі: класифікації, візуальне програмування, математичне моделювання, веб-програмування.

Наукова новизна роботи полягає у тому, що було використано інструменти машинного навчання в популярній веб платформі для підвищення економічної ефективності інтернет-магазину, побудована модель взаємодії інтернет магазину та системи рекомендатора, сформовано алгоритм роботи додатку.

Практична цінність полягає в дослідженні шляхів впровадження новітніх технологій у виробництві та покращення фінансово-економічної стабільності підприємства шляхом забезпечення стабільної роботи бізнес-клієнтів підприємства за допомогою поширених алгоритмів рекомендацій та створення ефективнішого підбору контенту для клієнта, щоб максимізувати економічний ефект від діяльності інтернет магазину за рахунок підбору оптимальних пропозицій для користувача.

Особистим внеском здобувача є інтеграція інструменту машинного навчання в популярну веб платформу та аналіз атрибутів товарів, необхідних для генерації персональних рекомендацій.

Апробація результатів роботи. Матеріали кваліфікаційної роботи доповідалися на II Всеукраїнська науково-технічна конференція «Комп'ютерні технології: інновації, проблеми, рішення» 23-24 листопада 2023 року (подано до друку) [2].

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПОКАЗНИКІВ КОНВЕРСІЇ ТА ЗАЛУЧЕНОСТІ ПРИ ВИКОРИСТАННІ РЕЛЕВАНТНИХ РЕКОМЕНДАЦІЙ

1.1 Дослідження показників конверсії та залученості при використанні релевантних рекомендацій

Відвідувачі часто заходять до інтернет-магазину без певної мети, тому важливо розповісти їм про бестселери та акційні пропозиції. Цікавий підхід до вирішення цього завдання можна побачити в інтернет-магазині електроніки "Цитрус" [3]. Фактично все, що пропонується нижче головного банера, – це добірки з рекомендаціями, а саме:

- турбота про здоров'я – ситуативна добірка, до якої входять не тільки пульсометри й термометри, але також дезінфектори та інші засоби індивідуального захисту, зокрема маски, антисептики, рукавички. Ця добірка може привернути увагу навіть тих, хто прийшов за іншим товаром. Вона є актуальною зараз, а через деякий час її можна замінити іншою ситуативною добіркою, приуроченою до якогось свята (наприклад, до 1 вересня – все для школи);
- лідери продажів – тут представники різних сегментів покупців знайдуть цікаві пропозиції. У продажах, як і в інших сферах, діє принцип Парето: 80% виручки приносять 20% товару. Тому показувати добірку бестселерів – гарна ідея;
- новинки – для тих, хто бажає бути в тренді;
- незабаром у продажі – позиції, доступні для попереднього замовлення;
- переглянуті товари;
- рекомендовані товари – персональна добірка.

У цьому випадку такий підхід спрощує відвідувачу навігацію на сайті й одразу залучає його до взаємодії з пропозиціями магазину.

Відповідно до звіту організації [go.euromonitor.com](https://www.go.euromonitor.com) [4], у 2023 році створювати тенденції, які впливатимуть на роботу ритейлерів, будуть саме споживачі.

Основними запитами покупців будуть:

- автоматизація процесів;
- наявність дешевшого товару;
- гарна репутація бренду у політичних та соціальних переконаннях.

Майже 40% опитаних гравців ринку планують у 2023 році інвестувати у робототехніку, яка допомагатиме споживачам на етапі пошуку товару. Але сума інвестицій буде меншою, якщо порівнювати з піковим значенням 2020 року. Також потрібно впроваджувати рішення, які допоможуть впоратися з економічною невизначеністю, щоб завоювати довгострокову лояльність клієнта.

Зазначається, що фактично 30% споживачів купуватимуть товари, враховуючи рівень соціальної відповідальності компанії та її політичних переконань.

У мережі Comfy відзначають декілька тенденцій на ринку, які з'явилися після початку повномасштабного вторгнення.

- зріс попит на цифрову техніку та електроніку, у той час як інші категорії впали приблизно на 30-35%. Зараз підвищений попит підтримується високою потребою у ноутбуках, планшетах і смартфонах серед військових та волонтерів;
- зменшився попит на побутову техніку, оскільки люди переносять такі покупки до "кращих часів". Але у Comfy впевнені, що після перемоги України попит на велику побутову техніку злетить через масове повернення переселенців;
- зменшився загальний попит на дорогі товари, але зростають продажі сегмента "середній мінус". Це пов'язано з інфляцією та падінням доходів українців.

Також у компанії помітили зміни в географії продажів, через внутрішню міграцію. Переселенці, які залишили дім з мінімумом речей, генерували попит на товари дрібного побуту – фени, тримери, чайники, тощо. Але при цьому серед ритейлерів відбувається посилення конкуренції за клієнта, тож завдання

усіх українських рітейлерів на найближчі кілька років - утримання прибутковості бізнесу, збереження команди, сплата податків.

На основі даних досліджень, приведених в [5] вказано, що найбільш прогресивні електронні ресурси, такі як соціальні мережі захоплюють аудиторію інтернет користувачів та стимулюють активніше користуватися інтернетом шляхом пропозиції користувачу контенту, який відповідає інтересам аудиторії. В результаті цього значно збільшується комунікація з брендом, що було і є основою комерціалізації. Відсоток користувачів, залучених до отримання інформації про товари чи послуги та отримання комерційних послуг наближається до 35 %.

Це вказує на те, що вплив інтернет ресурсів на споживчу поведінку користувачів значний, отже компанії можуть застосовувати релевантні засоби взаємодії зі своєю аудиторією, що дозволяє збільшувати лояльність або ж генерувати трафік для онлайн магазинів. Поряд з цим, можна досліджувати потреби користувачів, підбирати найбільш ефективний асортимент продукції, просувати окремі акційні пропозиції та стимулювання до здійснення покупки шляхом задоволення потреб користувачів.

Також, за даними дослідження, правильно підібрані пропозиції в соціальних мережах можуть генерувати конверсію до 20%, а також, 38% людей рекомендують бренд, послідовниками якого, вони є в соціальних мережах. 43% користувачів отримували інформацію перед покупкою, 51% шукали відгуки чи поради, та 62% шукали відгуки про варіанти товарів.

В дослідженнях порталу Briansoli [6], 81% покупців запитують поради у соціальних мережах про доцільність придбання того чи іншого товару, що вказує на необхідність використання досвіду попередників для формування торгівельної пропозиції для користувачів. Це означає, що автоматизована пропозиція «поради» користувачів значною мірою впливає на успіх у пошуку необхідного товару та прийняття рішення про покупку.

Разом з цим, в дослідженні The Trusted Conversion Optimization Agency [7] вказується, що покупці вважають ефективнішим рішенням користуватися персоналізованими пропозиціями, а саме:

- 59% покупців вважають, що за допомогою персоналізованих пропозицій простіше знайти відповідний товар;
- 56% користувачів скоріше за все повернуться до магазину де їм пропонували персоналізовані пропозиції;
- 53% користувачам здається, що ритейлери, які пропонують персоналізовані пропозиції, мають професійніший підхід до обслуговування;
- 57% подобаються пропозиції, що генеруються магазином та вважають їх корисними;
- персоналізована пропозиції генерують до 10 разів більше переходів у порівнянні зі звичайними списками продукції;

В свою чергу, 61% маркетологів вважає, що персоналізовані рекомендації важливі, але 51% не розуміють як їх впроваджувати, а 59% маркетологів які використали механізми персоналізації відчують вигоду від їхнього використання.

1.2 Формування стратегії покращення фінансово-економічної безпеки підприємства

Перш за все треба зазначити, що під політикою системи фінансово-економічної безпеки пропонуємо розуміти загальні орієнтири для дій і прийняття рішень, які полегшують досягнення економічних цілей [8]. Таким чином, для встановлення цих загальних орієнтирів необхідно спочатку сформулювати цілі політики забезпечення безпеки підприємства.

То ж, пропонуємо в якості цілей політики системи фінансово-економічної безпеки ФОП обрати наступні:

- зміцнення дисципліни праці і підвищення її продуктивності;
- захист законних прав та інтересів підприємства;

- зміцнення інтелектуального потенціалу підприємства;
- збереження і примноження власності;
- підвищення конкурентоспроможності продукції;
- максимально повне інформаційне забезпечення діяльності підприємства та підвищення його ефективності;
- орієнтація на світові стандарти і лідерство в розробці та освоєнні нової технології і продукції, що випускається;
- виконання виробничих програм;
- орієнтація на світові стандарти і лідерство в розробці та освоєнні нової технології і продукції, що випускається;
- надання сприяння управлінським структурам в досягненні цілей;
- недопущення залежності від випадкових і несумлінних ділових партнерів.

З урахуванням вищевикладеного, можна визначити наступні основні загальні орієнтири для дій і прийняття рішень, які полегшують досягнення цих цілей організації системи фінансово-економічної безпеки (рис. 1.1).



Рис. 1.1. Орієнтири політики фінансово-економічної безпеки

Працюючи в сфері e-commerce та ставлячи за мету успішність роботи та розвитку ритейл клієнтів, можна вважати, що стратегія організації фінансово-економічної безпеки повинна складатися з трьох ключових розділів:

- 1) зміцнення інтелектуального потенціалу;

- 2) підвищення конкурентоспроможності продукції;
- 3) виконання виробничих програм.

Дані розділи є взаємопов'язаними та необхідними для виконання для розвитку у сфері ІТ. Загальна мета роботи обраних розділів є орієнтація на світові стандарти і лідерство в розробці та освоєнні нової технології і продукції, що випускається. Це в свою чергу надасть якісну перевагу клієнтам підприємства та буде спонукати виконання планів продажів ритейл платформ, оскільки основна сфера діяльності клієнтів підприємства – це e-commerce.

Таким чином, створення інноваційного додатку створить конкурентну перевагу клієнта, що в свою чергу надасть можливість інвестувати в підприємство, що розробило додаток та зміцнить економічні стосунки.

1.3 Огляд алгоритмів генерування персоналізованих рекомендацій

Більшість систем рекомендацій базуються на двох основних стратегіях:

1. Content-based filtering System;
2. Collaborative filtering System.

Фільтрування на основі вмісту – це тип системи рекомендацій, яка намагається вгадати, що може сподобатися користувачеві на основі його активності. Він дає рекомендації за допомогою ключових слів і атрибутів, призначених об'єктам у базі даних (класифікація змісту), наприклад, у випадку товарів, кожен продукт можна описати за допомогою назви, виробника, складу, типу застосування, року випуску, оцінки та відгуку користувачів. Ці дані зіставляються із профілем користувача. Профіль користувача створюється на основі даних, отриманих від дій користувача, таких як покупки, оцінки (уподобання), завантаження, пошук товарів на веб-сайті та/або розміщення в кошику та натискання посилань на продукти [9]. Крім ключових слів, таких як виробник, марка та модель, профіль користувача вказує на те, що попередні покупки включають аксесуари для продукту. На основі цієї інформації система рекомендацій може запропонувати подібні аксесуари для наступної аналогічної покупки.

Рекомендатори на основі вмісту будують на основі матриці елементів. Матриця містить дані про користувачів, їх атрибути та рекомендації з їхніми атрибутами. Задачею матриці є створити взаємодію між користувачем і бажаними елементами [10]. Дані, зібрані під час повсякденної діяльності користувача, зберігаються в структурованому форматі, щоб визначити, що подобається та що не подобається для різних елементів, з якими користувач взаємодіяв. Кожній взаємодії призначається значення, відоме як «ступінь переваги». Зразок матриці елементів наведено в таблиці. 1.1.

У наведеному прикладі матриці корисності бракує кількох значень. Це тому, що деякі користувачі не взаємодіють з усіма елементами, доступними на платформі. Зауважимо, що мета рекомендуючої моделі полягає в тому, щоб запропонувати нові елементи на основі цієї матриці елементів.

Таблиця 1.1

Зразок матриці елементів

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9
User 1	?	?	0	1	4	1	?	0	3
User 2	1	1	2	2	?	5	0	1	4
User 3	2	3	1	0	?	0	0	0	0
User 4	1	1	?	0	4	4	4	1	?

Профіль користувача – це набір векторів, які визначають уподобання користувача. Профіль базується на діяльності та смаках користувача; наприклад, оцінки користувачів, кількість кліків на різних елементах, оцінки вмісту вгору або вниз тощо. Ця інформація допомагає системі рекомендацій найкраще оцінити нові пропозиції.

Профіль предмета для фільтрації на основі вмісту вимагає, щоб різні характеристики кожного окремого предмета відображали його основні якості. Звертаючись до прикладу фільму, деякі необхідні атрибути фільмів, які допоможуть системі рекомендацій розрізняти їх, це актори, режисер, рік випуску, жанр, рейтинги IMDb тощо. У фільтрації на основі вмісту зазвичай

використовуються два популярні методи: косинусна відстань і підхід класифікації.

Косинусна відстань розраховується між векторами користувача та елемента. Наприклад, цільовий користувач любить дивитися бойовики і дещо не любить жахи та трилери. Вектор для бойовиків має додатні значення, а вектор для фільмів жахів має від'ємне значення для цього конкретного користувача. А тепер уявимо, що вийшов новий фільм у жанрі науково-фантастичного бойовика. Оскільки наш користувач віддає перевагу бойовикам, кут косинуса між вектором фільму та вектором користувача буде великою додатною часткою, що призведе до меншого кута. Це означає, що цей фільм – гарна рекомендація для користувача. Якщо ж косинусна відстань велика, то можна ігнорувати елемент, оскільки це погана рекомендація. Схема розрахунку косинусної відстані приведена на (рис. 1.4).

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}, \quad (1.1)$$

де A і B – вектори елементів на основі їхніх атрибутів.

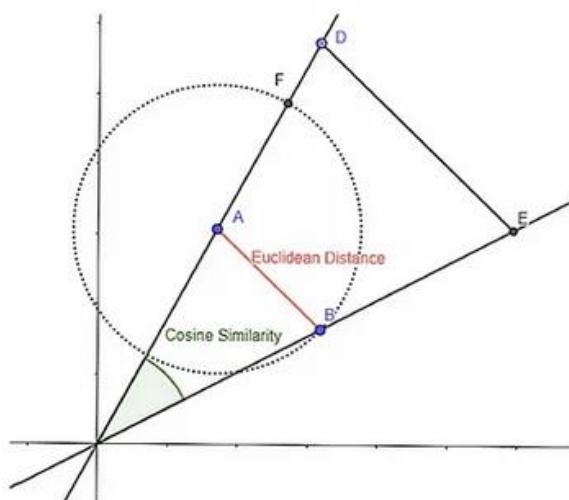


Рис. 1.4. Схема розрахунку косинусної відстані

Алгоритми класифікації, такі як байєсівські класифікатори або моделі дерева рішень, можна використовувати для надання рекомендацій. Наприклад, кожен рівень дерева рішень можна використовувати для фільтрації різноманітних уподобань користувача, щоб зробити більш точний вибір.

Обчислюючи евклідові відстані, можна знайти значення відстані між двома елементами, яке виражає подібність між ними. Помічено, що зі зменшенням відстані схожість зростає. Таким чином можна здійснити процес рекомендацій [11]. Евклідова відстань обчислюється за формулою приведеною у формулі (1.2).

$$d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad , \quad (1.2)$$

де q і p – це значення вподобання користувача за елементами

Хоча в Евкліда існує поняття відстані, поняття подібності з'являється в косинусній подібності. Відстань-близькість, схожість-несхожість тут відповідають одним і тим же поняттям [11].

Альтернатива Content-based типу, який покладається лише на попередній поведінці користувача - Collaborative filtering (колаборативна фільтрація) - аналізує стосунки між користувачами та взаємозалежності між продуктами та визначає нові асоціації користувачів.

Collaborative filtering використовує рекомендації на основі сусідства [12]. Тут вибираються підмножини користувачів на основі їх схожості з активним користувачем, а для пропозиції використовується зважена комбінація їхніх рейтингів передбачення для активного користувача.

Алгоритм наведено в наступні кроки.

Крок 1. Усі активні користувачі мають однакову подібність.

Крок 2. Вимірюємо схожість між користувачами як коефіцієнт кореляції Пірсона між їхніми векторами оцінок. Коефіцієнт кореляції Пірсона можна обчислити за формулою (1.3).

$$P_{a,u} = \frac{\sum_{i=1}^n (r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^n (r_{a,i} - \bar{r}_a)^3 \times \sum_{i=1}^n (r_{u,i} - \bar{r}_u)^3}} , \quad (1.3)$$

де $r_{a,i}$ - це оцінка елемента i користувачем a ;

$\bar{r}_a$ - це середня оцінка користувача a .

Крок 3. Вибрати n користувачів із найбільшим рівнем схожості з активним користувачем.

Крок 4. Сформувані сусідів активного користувача за рекомендаціями, які обчислюються як середньозважене відхилення від сусіднього середнього значення за допомогою формули (1.4).

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^n (r_{u,i} - \bar{r}_u) \times P_{a,u}}{\sum_{u=1}^n P_{a,u}} , \quad (1.4)$$

де $P_{a,i}$ - рекомендація для активного користувача a для елемента i ;

$P_{a,u}$ - це подібність між користувачами a та u ;

n - це кількість користувачів у сусідстві.

Крок 5. Обчислити прогноз зваженої комбінації рейтингів сусідів.

Коефіцієнт ваги значущості (SWF) використовується для відкидання кореляцій, створених на основі кількох подібних елементів, щоб запобігти поганим прогнозам[13]. Якщо кількість спільно оцінених предметів (n) менше 50, тоді SWF є добутком їх кореляцій. Якщо $n \geq 50$, то коефіцієнт дорівнює 1.

Процес реалізації починається з наївного байєсівського текстового класифікатора, щоб дізнатися профіль користувача з набору оцінених елементів. Завданням передбачення можна вважати проблемою текстової категоризації, а інформацію про вміст елемента можна вважати текстовими документами, а оцінки користувача (1-5) можуть бути одним із шести міток класу. Приймається мультиноміальна текстова модель, у якій документ моделюється як упорядкована послідовність подій слів, взятих з одного словника, V . Наївне припущення Байєса стверджує, що ймовірність кожної події залежна від класу елемента[14]. Для кожного класу C_i , ймовірності класу

та виникнення слова словника від класу можна оцінити з навчальних даних. Подальша ймовірність кожного класу заданому документу D , обчислюється за правилом Байєса за формулою (1.5).

$$P(D) = \frac{P(C_i)}{P(D)} \prod_{i=1}^{|D|} P(a_i|C_i) \quad , \quad (1.5)$$

де a_i - це i -те слово в документі;

$|D|$ - це кількість слів у документі.

У процесі реалізації елементи представлені як вектор документів d_m , по одному для кожної позиції, ймовірність кожного слова з урахуванням атрибута та позиції має бути оцінена. Ймовірності наступного атрибута для елемента залежно від позиції розраховується за формулою (1.6).

$$P(F) = \frac{P(C_i)}{P(F)} \prod_{m=1}^S \prod_{i=1}^{|d_m|} P(a_{mi}|C_i, S_m) \quad , \quad (1.6)$$

де S – кількість позицій;

a_{mi} – це i -те слово в елементі m -ї позиції.

Таким чином, можна розрахувати ймовірність виникнення певного слова – атрибута елемента від будь-якого запропонованого наданого елемента.

Також використовуються гібридні підходи контентної та колаборативної фільтрації. Їм властиві збільшення ефективності та точності[15]. Гібридні методи дозволяють враховувати результати як контентного методу так і колаборативного та об'єднувати результати. В цьому підході можуть бути використані різноманітні алгоритми, запозичені з області машинного навчання [15]. Відтак, можна точно вирахувати схожість між двома користувачами і їх атрибутами за допомогою кореляції Пірсона. Цей алгоритм не обчислює міру схожості за всією сукупністю користувачів. Спочатку необхідно власноруч відфільтрувати подібних користувачів елементів на основі високорівневих показників подібності, а далі алгоритм вимірює лінійну залежність між двома

користувачами за їхніми атрибутами як функцію їх змінних і підбере найбільш схожих. Порівняння алгоритмів наведено у (табл. 1.2).

Таблиця 1.2

Порівняння алгоритмів рекомендацій

Порівняльна характеристика	Content-based filtering	Collaborative filtering
1	2	3
Необхідність наявності даних користувача	-	+
Простота генерації рекомендацій	Проста	Складніша, вимагає взаємодії користувачів
Програмна простота реалізації	Проста	Складна
Кількість користувачів у вибірці	Можуть бути відсутні або мала	Необхідна кількість користувачів пропорційна до пропозицій
Сумісність з великою кількістю користувачів	+	+
Пристосування до інтересів користувача	Повне	Часткове, впливає поведінка споріднених користувачів
Чутливість до атрибутів товару та профілю користувача	Висока	Середня
Можливий розмір каталогу товарів	Великий	Середній
Можливе різноманіття типів товарів	Мале	Велике
Доступність для розуміння користувача	Висока	Середня
Проблема холодного пуску	-	+
Динамічне оновлення рекомендацій	-	+
Необхідність реалізації механізму ревалідації	+	-
Масштабованість	Складна	Середня
Необхідність реалізації механізму ревалідації	+	-
Масштабованість	Складна	Середня

Відповідно до таблиці порівняння Content-based filtering краще підходить для простих інтернет магазинів, які мають великий асортимент різнотипних товарів зі статичними атрибутами. Його простота дозволяє швидко додавати прості товари без залучення значних обчислювальних ресурсів. В свою чергу, Collaborative filtering дозволяє значно різноманітнити наповнення товарів атрибутами, він забезпечує більшу варіативність списку пропозицій, але й

вимагає більше обчислювальних потужностей, має складність до адаптації до нових атрибутів.

Висновки до першого розділу

1. В результаті проведеного аналізу методів та засобів реалізації алгоритмів рекомендацій були визначені рекомендаційні блоки, що допомагають вирішити цілу низку завдань, зокрема:

- збільшення конверсії – відвідувач сайту може придбати рекомендований товар замість того, який він переглядав, або на додачу до нього, і з меншою вірогідністю піде із сайту без покупки;
- збільшення тривалості сесії та глибини перегляду – користувач переглядатиме більшу кількість сторінок протягом сеансу;
- внутрішнє перелінкування сторінок;
- спрощення навігації – клієнтові легше знайти супутні або подібні товари в асортименті магазину;
- формування довіри – покупець бачить турботу про свої інтереси;
- ненав'язливе просування різних асортиментних груп;
- підвищення середнього чеку за рахунок cross-sell та upsell (завдяки рекомендації дорожчих товарів і продажу супутніх, а також альтернативних пропозицій щодо товарів, яких немає в наявності);
- збільшення частоти купівель.

2. Проведений аналіз засобів реалізації алгоритмів рекомендацій показав, що рекомендовані товари можуть відображатися на інформативних сторінках сайту, та зокрема у кошику.

3. Виявлено, що, якщо додати рекомендації в набір товарів для продажу, продавці можуть у середньому збільшити кількість конверсій до 40% за тієї самої відвідуваності. Визначено, що поєднання каталогів з привабливими пропозиціями і унікальними рішеннями для аудиторій, ритейлери можуть збільшити обсяги продажів потенційним клієнтам на 94%, а рентабельність витрат на рекламу зменшити на 28%.

4. В результаті проведеного аналізу показано, що для клієнто-орієнтованих підприємств надзвичайно важливо забезпечити стабільність та успішність роботи ритейлерів для покращення економічних зв'язків та партнерської співпраці. Визначено, що існує два базових алгоритми генерації релевантних рекомендацій та їхні похідні. Виявлено, що використання будь-якого алгоритму передбачає збір та аналіз даних користувачів, розрахунку подібності уподобань користувачів та порівняння метрик товарів. Обґрунтовано, що найбільш подібним варіантом товару і є релевантною пропозицію, яка має зацікавити користувача та як результат призвести до транзакції.

5. Було проведено аналіз роботи успішних ритейл платформ, запропоновано стратегію фінансово-економічної безпеки підприємства та розглянуто моделі створення персоналізованих рекомендацій користувачу.

РОЗДІЛ 2. АРГУМЕНТАЦІЯ ВИБОРУ ІНСТРУМЕНТУ РЕАЛІЗАЦІЇ СИСТЕМИ РЕКОМЕНДАЦІЙ

2.1 Вибір інструменту реалізації системи рекомендацій

Серед рішень які можуть реалізовувати алгоритми рекомендацій можна знайти систему рекомендацій від Google [16]. Це клаудне рішення, яке опрацьовує статичні товарні фіди та формують рекомендації за такими показниками, як покази й кліки та використовують моделі атрибуції на основі даних [17]. Така модель має обмеження. Для деяких типів дій-конверсій потрібно зібрати щонайменше 300 конверсій і 3000 взаємодій з оголошеннями в підтримуваних мережах протягом 30 днів. Ви не зможете й надалі використовувати атрибуцію на основі даних для цих типів дій-конверсій, якщо буде зареєстровано менше ніж 2000 взаємодій з оголошеннями в підтримуваних мережах або дія-конверсія отримає менше ніж 200 конверсій протягом 30 днів. Саме тому було обрано спеціалізовану систему PredictionIO, яка не має обмежень по кількості взаємодій та має змогу працювати з колаборативною моделлю даних.

Apache PredictionIO – це платформа машинного навчання з відкритим вихідним кодом для розробників, спеціалістів із обробки даних і кінцевих користувачів. Він підтримує збір подій, розгортання алгоритмів, оцінку, запит передбачуваних результатів через REST API. Він заснований на масштабованих службах з відкритим кодом, таких як Hadoop, HBase (та інші БД), Elasticsearch, Spark, і реалізує так звану лямбда-архітектуру [18].

PredictionIO складається з таких компонентів (рис 2.1):

Сервер подій – аналітичний рівень машинного навчання з відкритим вихідним кодом для об'єднання подій із кількох платформ.

Галерея шаблонів – місце для завантаження шаблонів механізмів для різних типів програм машинного навчання.

Сервер подій PredictionIO постійно збирає дані з програми. Потім механізм PredictionIO будує прогностичну модель (моделі) з одним або декількома алгоритмами, використовуючи дані. Після розгортання як веб-служби він прослуховує запити від вашої програми та відповідає прогнозованими результатами в режимі реального часу. Він також може уніфікувати дані, пов'язані з конкретною програмою з кількох платформ.

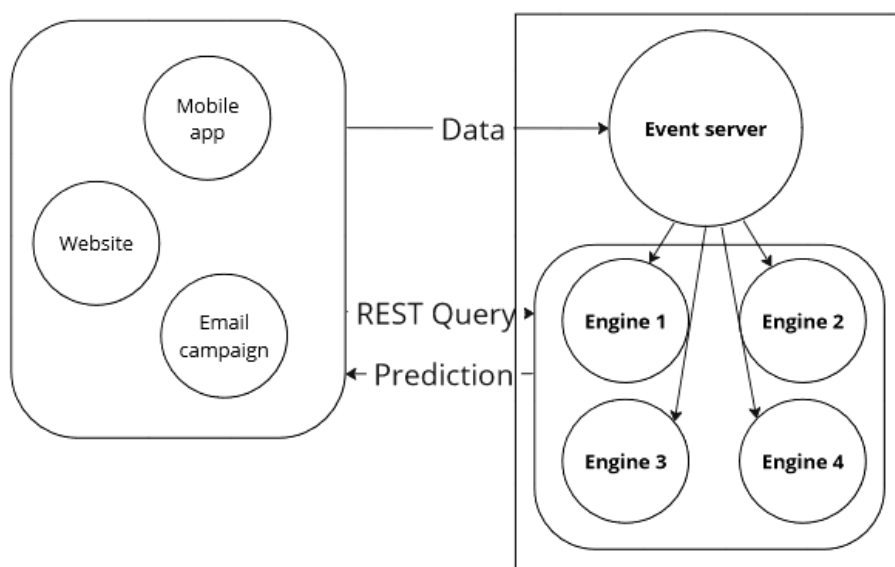


Рис. 2.1. Схематичне представлення архітектури системи

Після збору даних, вони в основному служать двом цілям, а саме - даних механізму(ам) для навчання та оцінки моделі та систематизація даних для аналізу.

Як і сервер бази даних, сервер подій може розміщувати кілька програм. Дані для кожної програми розділені унікальним `app_name`. Після запуску сервера подій можна надсилати дані певному `app_name`, визначеному ключем доступу, через його Event API із запитом HTTP або за допомогою одного з SDK. У деяких особливих випадках може знадобитися, щоб механізм читав дані з іншого сховища даних замість сервера подій.

Engine відповідає за прогнозування. Він містить один або кілька алгоритмів машинного навчання. Механізм зчитує навчальні дані та створює

прогностичну модель (моделі). Потім він розгортається як веб-сервіс. Розгорнутий механізм відповідає на прогнозовані запити від вашої програми через REST API у режимі реального часу. Галерея шаблонів PredictionIO пропонує шаблони механізмів для всіх видів завдань машинного навчання[19]. За допомогою цих шаблонів можна легко створити один або кілька механізмів. Компоненти шаблону, а саме джерело даних, засіб підготовки даних, алгоритм(и) і обслуговування, можна налаштувати відповідно до ваших конкретних потреб. Схематичне представлення механізму engine наведено на (рис. 2.2).

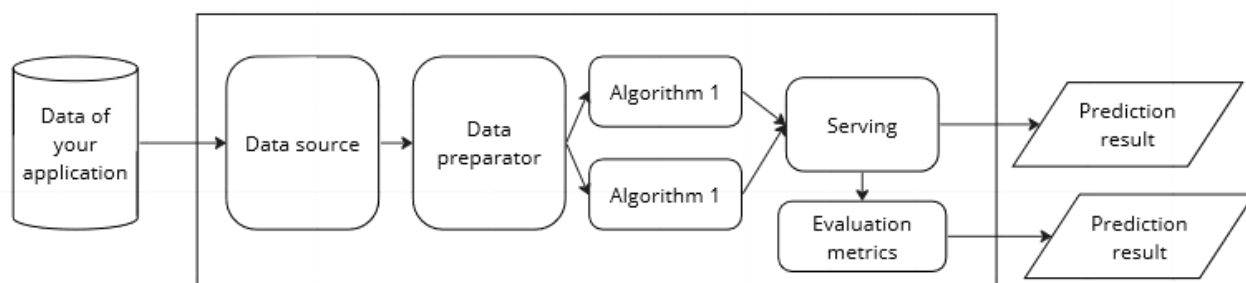


Рис. 2.2. Схематичне представлення механізму engine

Мінімальні необхідні системні вимоги включають в себе наявні встановлені застосунки - Java SE Development Kit 8 або вище, PostgreSQL 9 або вище, Elasticsearch 1.7.6 або вище.

Наступним кроком для створення нового engine є перегляд галереї шаблонів PredictionIO, де можна знайти шаблони для всіх видів завдань машинного навчання та обрати той, який найкраще відповідає завданню. Щоб завантажити шаблон, [необхідно перейти до галереї шаблонів](#), обрати шаблон engine та натиснути його назву. Ви будете перенаправлені до репозиторію GitHub шаблону. Склонуйте його на локальну машину, щоб налаштувати його далі.

Далі необхідно створити новий `app_name` та зберегти Access Key для доступу. Після цього, використовуючи SDK під платформу необхідно розробити колекцію даних для навчання моделі.

Сервер подій може збирати та зберігати довільні події. На початку проекту рекомендується зібрати якомога більше даних. Пізніше ви можете виключити дані, які не стосуються прогнозової моделі у Data Preparator. Модель рекомендацій на основі колаборативної фільтрації працює за шаблоном загальним шаблоном даних – user – action – item. Це значить, що необхідно збирати дані про користувачів (створити записи користувачів в базі та зібрати їх ідентифікатори), дані про елементи з їх атрибутами та активності користувача, що направлені на елементи. де користувачі та елементи мають пов'язані з ними властивості. Наприклад, для персоналізованої рекомендації книг потрібно зібрати деякі події:

Користувач 1 придбав продукт X

Користувач 2 переглянув продукт Y

Користувач 1 додав продукт Z у кошик

Модель взаємодії зображена на (рис 2.3)

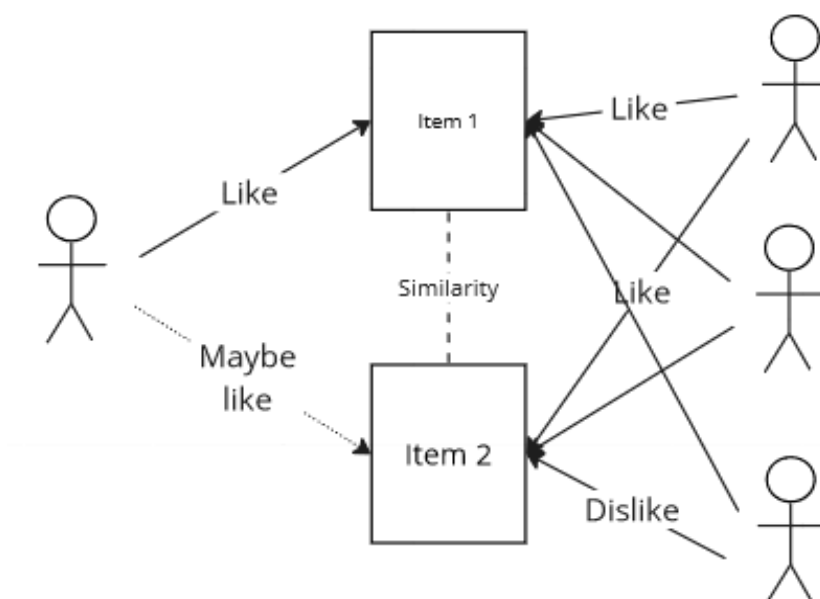


Рис. 2.3. Модель взаємодії користувачів при колаборативній фільтрації

Властивості (атрибути) користувача можуть бути статтю, віком, місцезнаходженням тощо. Властивості елемента можуть бути жанром, автором та іншими атрибутами, які можуть бути пов'язані з уподобаннями користувача. Збір даних значно відрізняється залежно від вашої програми та вашої мети прогнозування.

Алгоритм генерування пропозицій повинен включати збір даних про наявні товарні позиції та збір даних про взаємодію користувача з товаром. Дані взаємодії користувача з товаром можуть бути довільними і залежати від специфіки проекту та мети прогнозування, але основні події - це перегляд товару, додавання до кошику та оформлення замовлення. Відповідно до шаблону Universal Recommender це “view”, “add-to-cart”, “purchase”. Алгоритм взаємодій користувача та товарної пропозиції виглядає наступним чином (рис. 2.4)

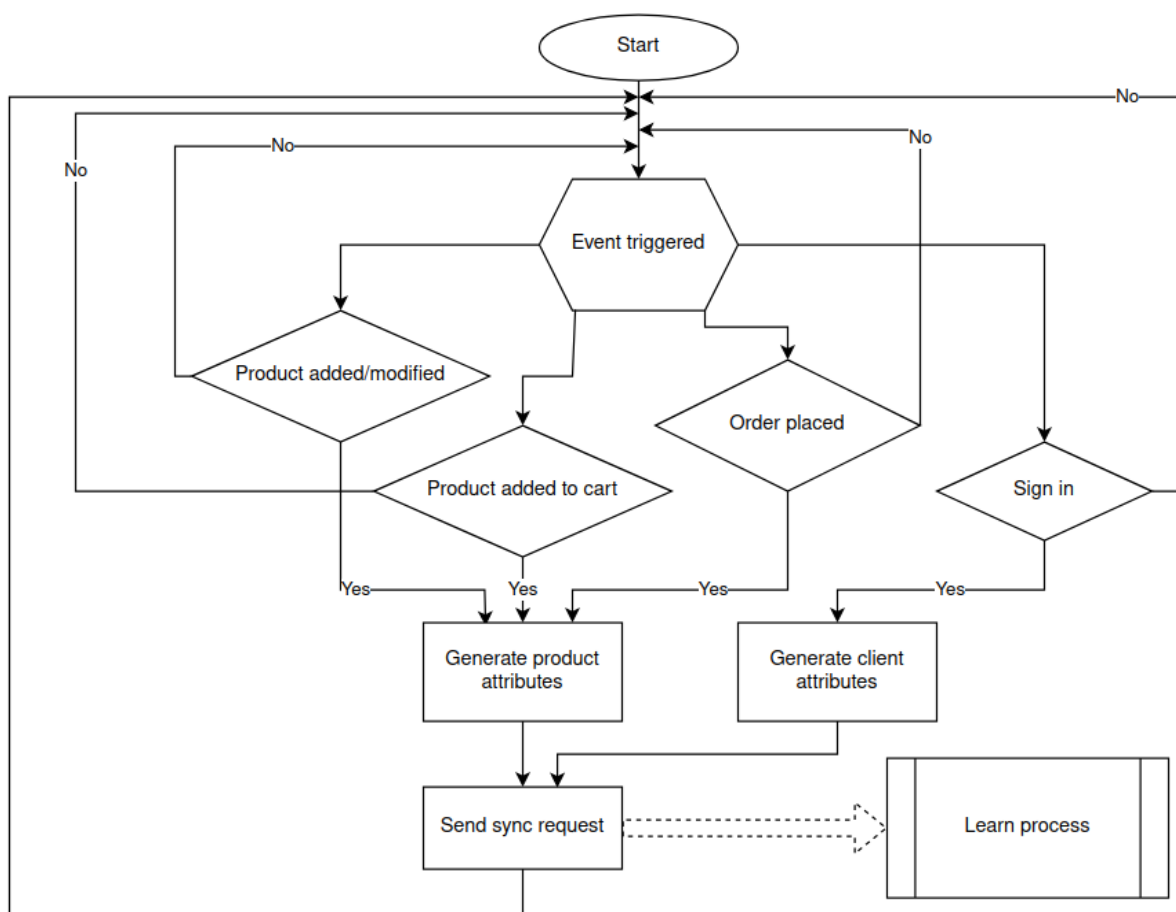


Рис 2.4. Алгоритм взаємодій користувача та товарної пропозиції

Таким чином проведений аналіз роботи бібліотеки та набір даних для його роботи показав, що найбільш ефективною та доцільною для реалізації додатку в межах магістерського дослідження є використання саме бібліотеки PredictionIO, яка повністю задовольняє виконання завдання кваліфікаційної роботи.

2.2. Постановка завдання

Shopware – це платформа для створення інтернет-магазинів та є популярною серед західних підприємств які працюють у сфері e-commerce. Недоліком платформи є те, що вона не має системи персоналізованих рекомендацій. Максимум, що можна реалізувати існуючим функціоналом – це налаштувати динамічні списки товарів за заданими правилами. Проте такі списки жорстко прив'язані до певної категорії товарів, групи користувача, списку товарів у кошику, тощо.

Приклад можливих правил представлений на (рис 2.5). Наповнення категорії каталогу на основі динамічної групи товарів представлено на (рис 2.6).

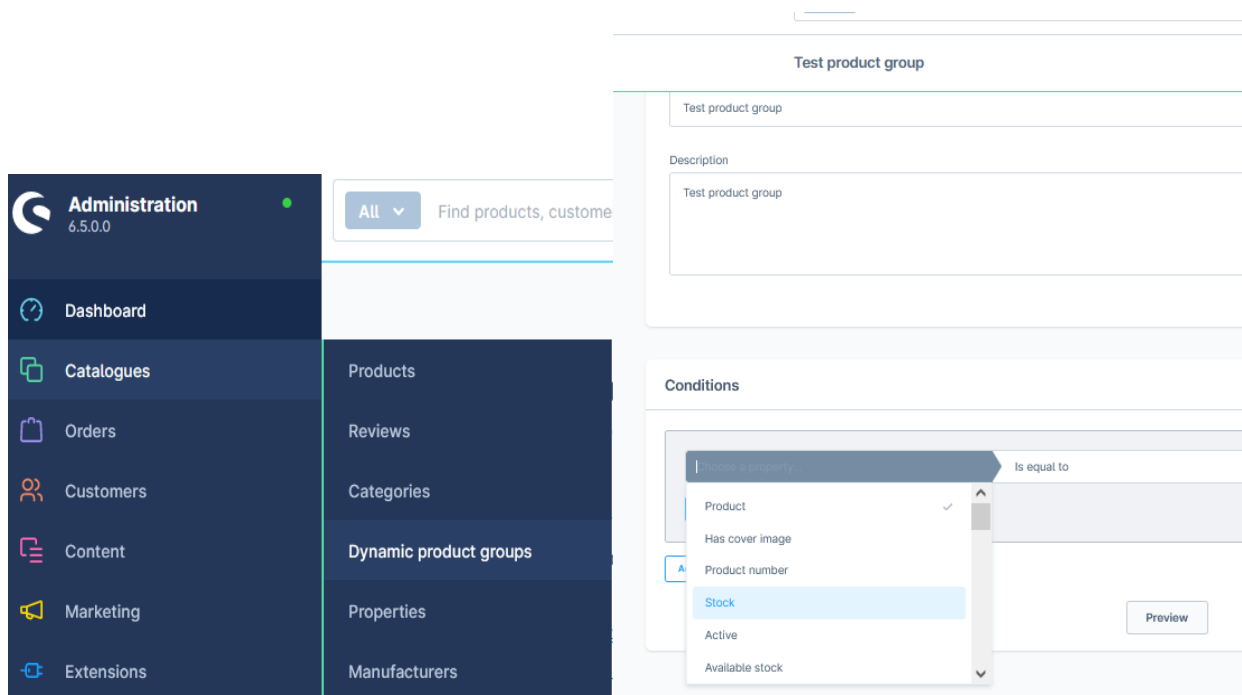


Рис 2.5. Налаштування динамічної групи товарів

Використовуючи пошук на маркеті подібних додатків для даної системи також не вдалося знайти. Саме тому було прийняте рішення щодо створення необхідного додатку власноруч. Додаток має бути простим, реалізовувати необхідні події для формування бази інтересів користувачів, а також необхідно враховувати атрибути товарів, які створюють різноманітність товарного асортименту, такі як розмір, колір, дизайн і т.д.

Name	Manufacturer
Main product with properties (Size: L)	Shopware Fashion
Main product with properties (Size: M)	Shopware Fashion
Main product with advanced prices	Shopware Freetime
Main product with properties (Size: XL)	Shopware Fashion
Main product	Shopware Fashion
Variant product (Colour: Blue Size: M)	Shopware Fashion
Main product with properties (Size: S)	Shopware Fashion
Main product, free shipping with highlighting	Shopware Freetime
Variant product (Colour: Red Size: XL)	Shopware Fashion

Рис 2.6. Наповнення категорії динамічним списком товарів

Саме тому було прийняте рішення щодо створення необхідного додатку власноруч. Додаток має бути простим, реалізовувати необхідні події для формування бази інтересів користувачів, а також необхідно враховувати атрибути товарів, які створюють різноманітність товарного асортименту, такі як розмір, колір, дизайн і т.д.

Попередньо було розглянуто тенденції сучасної роботи популярних ритейл платформ та визначено стратегію забезпечення фінансово-економічної безпеки підприємства і можливим шляхом реалізації стратегії. То ж сформуємо

формалізований список задач які необхідно вирішити в ході виконання роботи для формування актуального релевантного списку пропозицій користувачу. Серед задач є:

- 1) створення моделі збору та аналізу даних наявних пропозицій на основі визначеного алгоритму;
- 2) створення моделі взаємодії користувача і продуктів
- 3) визначення метрик товарів/користувачів які будуть піддаватися аналізу;
- 4) створення програмного додатку взаємодії інтернет-магазину та системи рекомендацій, що реалізує передачу метрик від веб-сайту до системи рекомендацій;
- 5) реалізація пошуку рекомендацій для користувача на основі зібраних та переданих метрик;
- 6) візуалізація запропонованих пропозицій користувачу.

Наступним етапом є проектування моделі взаємодії додатку з користувачами для наступної програмної реалізації.

2.3 Проектування моделі взаємодії додатку з користувачами

Взаємодія ритейлера як клієнта підприємства з користувачами інтернет ресурсами починається з дії, яка спонукає залучити людей до взаємодії з ритейл системою. Такою дією може бути, наприклад, створення реклами продукту чи послуги та її поширення на цільову аудиторію за допомогою глобальних систем, дослідження поведінки користувачів[20]. Ці системи аналізують поведінку користувачів на основі унікального ідентифікатора та можуть зробити висновок стосовно віку, статі, захопленнями людей. Пропонування реклама цільовій аудиторії також являється результатом роботи алгоритмів рекомендацій, але на глобальному рівні. Відмінністю таких систем є те, що вони гарантують врівноваження ваги пропозицій від сотень ритейлерів та дають змогу реалізувати їхню конкурентоспроможність. Схема взаємодії з глобальними системами рекомендацій наведена на (рис.2.7).

Далі в роботу необхідно включити систему рекомендацій на рівні ритейлера, який буде генерувати свої унікальні пропозиції в залежності від асортименту, цінової політики, акцій, бонусів, авторитету і т.д.

Для оптимальної роботи локальної системи рекомендацій необхідно вивчити поведінку користувача шляхом отримання уточнення атрибутів взаємодії відвідувачів сайту та товарів/послуг які зацікавили відвідувача. Це може бути як отримання базової інформації такої як стать, вік, місце проживання під час реєстрації, так і вподобання користувача шляхом дослідження переходів між товарами та інтерес до супутніх товарів шляхом дослідження корзини, здійснених транзакцій, або переглянутих товарів чи списку вподобань.

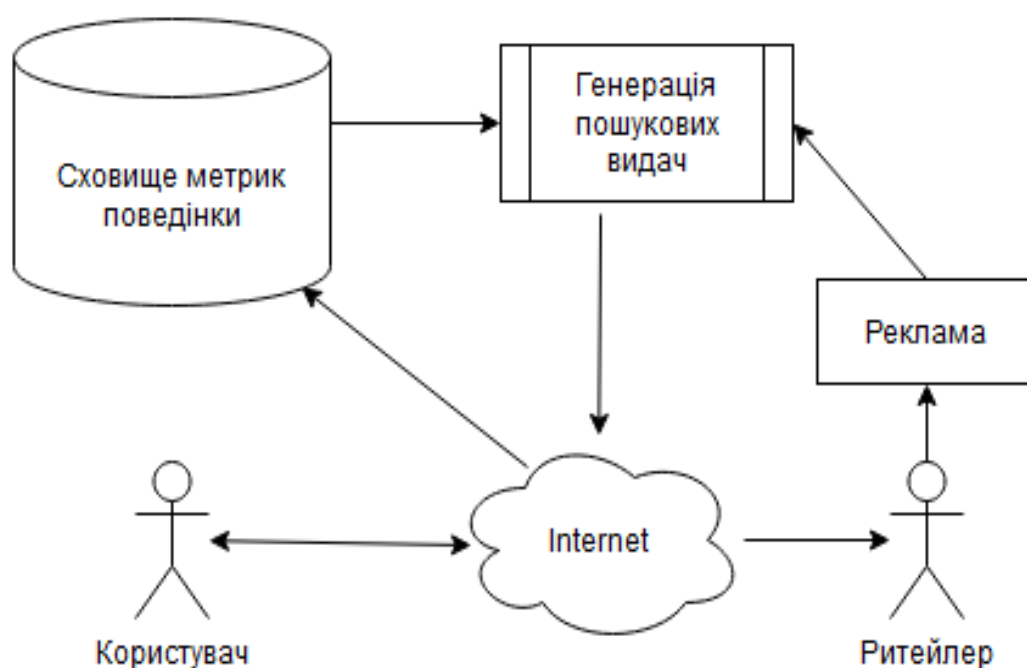


Рис. 2.7. Схема взаємодії з глобальними системами рекомендацій

Також, опціонально, можна отримати інформацію про користувача за допомогою зовнішніх агрегаторів, таких як соціальні мережі, пошуковики і т.д [21]. Схема взаємодії з локальною системою рекомендацій наведена на (рис. 2.8).

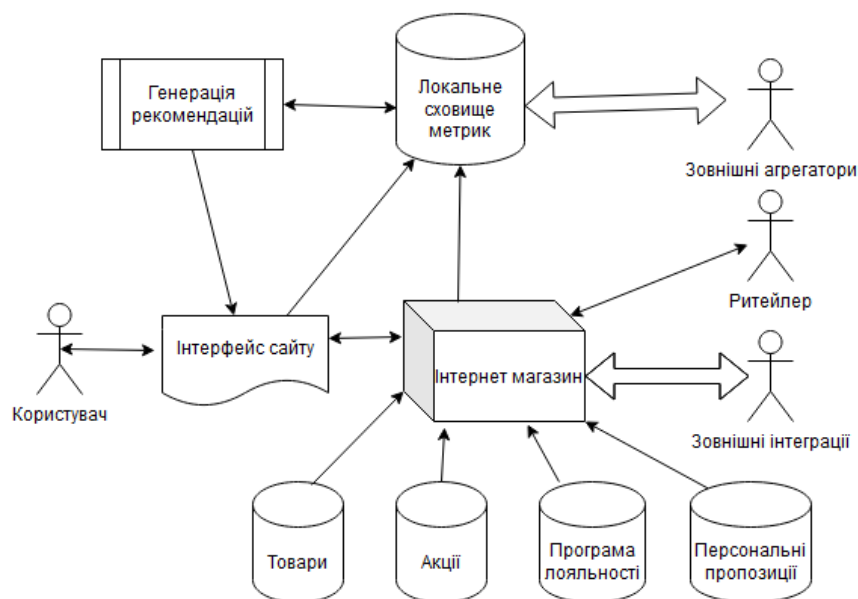


Рис. 2.8. Схема взаємодії з локальною системою рекомендацій

Відповідно до схеми взаємодії можна побачити, що нам доступні декілька шляхів збору інформації про взаємодію користувача та пропозицій. Це можуть бути як внутрішні події, генеровані веб-платформою, так і зовнішні інтегратори, інформацію про взаємодію з яким потенційно можна імпортувати в систему рекомендацій. На даному етапі ми не обмежені кількістю джерел даних, але обмежимося внутрішніми подіями, оскільки інтеграція зовнішніх інтеграторів вимагає додаткових комунікаційних інтерфейсів з зовнішніми системами.

Висновки до другого розділу

1. В даному розділі ознайомилися з бібліотекою генерацій персоналізованих рекомендацій PredictionIo та було проведено аналіз функціональних блоків бібліотеки, механізму збору даних взаємодії користувачів та продуктів, визначено шаблон функціонування та необхідні дані для його роботи, механізму навчання та генерації рекомендацій. В результаті аналізу обрали дану бібліотеку як основний інструмент для роботи з додатком.

2. Виявлено недолік популярної платформи для створення інтернет-магазинів, а саме відсутність реалізації динамічної системи персоналізованих рекомендацій.

3. В результаті проведеного аналізу атрибутів, необхідних для реалізації адаптивних алгоритмів, сформовано формалізований список задач, які необхідно виконати для створення веб-додатку з реалізацією адаптивного алгоритму генерування релевантних пропозицій користувачам інтернет магазину.

4. Проведено проектування моделі взаємодії додатку з користувачами.

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ

3.1 Створення програмного додатку

Створення програмного додатку починається з створення Plugin (Bundle), який буде реалізовувати взаємодію між компонентами фреймворку Shopware та SDK predictionIo.

Відповідно до попереднього розділу, необхідно передати до системи predictionIo інформацію про сутність клієнта, всіх продуктів та події взаємодії між клієнтами та продуктами.

Для передачі інформації про продукти використаємо стандартна подія Shopware ProductIndexerEvent. Ця подія відпрацьовує тоді коли відбувається індексація всіх продуктів системи. Таким чином ми можемо перерахувати всі товари які існують в системі та за допомогою сабскрайбера передати відповідну інформацію до бази predictionIo. Скорочений лістинг сабскрайбера індексації продуктів наведено нижче.

```
<?php
namespace Flexzt\PredictionRecommendation\Components\Communication\Subscriber;

use Flexzt\PredictionRecommendation\Components\Business\Recommendation\PredictionEventModelInterface;
use Flexzt\PredictionRecommendation\Components\Business\Recommendation\PredictionModelInterface;
use Shopware\Core\Content\Product\Events\ProductIndexerEvent;
use Shopware\Storefront\Page\Product\ProductPageLoadedEvent;
use Symfony\Component\EventDispatcher\EventSubscriberInterface;

class ProductSubscriber implements EventSubscriberInterface
{
    public function __construct(
        protected PredictionEventModelInterface $predictionEventModel,
        protected PredictionModelInterface $predictionModel
    ) { }

    public static function getSubscribedEvents(): array
    {
        return [ ProductIndexerEvent::class => 'onProductIndexing' ];
    }

    public function onProductIndexing(ProductIndexerEvent $event): void
```

```

{
    $eventClient = $this->predictionEventModel->getEventClient();

    foreach ($event->getIds() as $id) {
        $eventClient->deleteItem($id);
        $eventClient->setItem( $id, [ 'itemType' => 'product' ] );
    }
}
}
}

```

Даний сабскрайбер викликає екземпляр класу EventClient для комунікації з API predictionIo та передає ідентифікатор кожного товару (id) та атрибути. На даний момент це просто ідентифікатор сутності (product), але для вирішення конкретної задачі необхідно доповнити атрибути товару відповідно до специфіки товарного асортименту (колір, розмір, ширина, висота, модуль автомобіля до якого підходить запчастина і тд.). Атрибутами товару було запропоновано обрати властивості товару, які наповнюються з адміністративної частини вебсайту та описують комбінацію унікальних властивостей товару. Кількість таких властивостей необмежена та може змінюватися динамічно. Після збереження товару інформація про зміни автоматично синхронізується з сервісом PredictionIo.

Наступним кроком є створення сабскрайбера для передачі даних про активного користувача. Для цього використаємо стандартна подія CustomerLoginEvent. Дана подія викликається тоді, коли користувач залогінився до особистого кабінету і його можна однозначно ідентифікувати. Скорочений лістинг сабскрайбера наведений нижче.

```

<?php
namespace Flexzt\PredictionRecommendation\Components\Communication\Subscriber;

use Flexzt\PredictionRecommendation\Components\Business\Recommendation\PredictionEventModelInterface;
use Flexzt\PredictionRecommendation\Components\Business\Recommendation\PredictionModelInterface;
use Shopware\Core\Checkout\Customer\Event\CustomerLoginEvent;
use Shopware\Storefront\Page\Product\ProductPageLoadedEvent;
use Symfony\Component\EventDispatcher\EventSubscriberInterface;

class CustomerSubscriber implements EventSubscriberInterface
{

```

```

public function __construct(
    protected PredictionEventModelInterface $predictionEventModel,
    protected PredictionModelInterface $predictionModel
) { }

public static function getSubscribedEvents(): array
{
    return [
        CustomerLoginEvent::class => 'onCustomerLogin',
        ProductPageLoadedEvent::class => 'onProductPageLoaded',
    ];
}

public function onCustomerLogin(CustomerLoginEvent $event): void
{
    $this->predictionEventModel->getEventClient($event->getSalesChannelId())
        ->setUser( $event->getCustomer()->getId() );
}

public function onProductPageLoaded(ProductPageLoadedEvent $event): void
{
    $salesChannelContext = $event->getSalesChannelContext();
    if (!$salesChannelContext->getCustomer()) return;

    $product = $event->getPage()->getProduct();
    $this->predictionEventModel->getEventClient($salesChannelContext->getSalesChannelId())
        ->recordUserActionOnItem(
            'view',
            $salesChannelContext->getCustomerId(),
            $product->getId(),
            [
                'itemType' => 'product',
                'variant' => $product->getProductNumber()
            ]
        );
}
}

```

Подібно до попереднього сабскрайбера, кастомер сабскрайбер створює екземпляр класу `EventClient` для комунікації та викликається функція створення клієнта з відповідними атрибутами. Аналогічно до попереднього випадку, атрибути мають бути специфічними відповідно до поставленої задачі.

Останнім кроком для формування набору даних взаємодії є створення сабскрайбера для передачі інформації про події, що виникають в ході відвідування вебсайту клієнтом. До таких подій можна віднести перегляд товару, додавання товару в кошик та його покупка. Звісно, кількість подій може і має бути більшою, але список необхідних подій також формується відповідно до поставленої задачі. Скорочений лістинг сабскрайбера додавання товару до кошика та покупки наведено нижче.

```
<?php
namespace Flexzt\PredictionRecommendation\Components\Communication\Subscriber;

use Flexzt\PredictionRecommendation\Components\Business\Recommendation\PredictionEventModelInterface;
use Flexzt\PredictionRecommendation\Components\Business\Recommendation\PredictionModelInterface;
use Shopware\Core\Checkout\Cart\Event\BeforeLineItemAddedEvent;
use Shopware\Core\Checkout\Cart\Event\CheckoutOrderPlacedEvent;
use Shopware\Core\Checkout\Cart\LineItem\LineItem;
use Shopware\Core\Checkout\Order\Aggregate\OrderCustomer\OrderCustomerEntity;
use Shopware\Core\Checkout\Order\Aggregate\OrderLineItem\OrderLineItemCollection;
use Symfony\Component\EventDispatcher\EventSubscriberInterface;

class OrderSubscriber implements EventSubscriberInterface
{
    public function __construct(
        protected PredictionEventModelInterface $predictionEventModel,
        protected PredictionModelInterface $predictionModel
    ) { }

    public static function getSubscribedEvents(): array
    {
        return [
            CheckoutOrderPlacedEvent::class => 'onOrderPlaced',
            BeforeLineItemAddedEvent::class => 'onLineItemAdded'
        ];
    }

    public function onOrderPlaced(CheckoutOrderPlacedEvent $event): void
    {
        $customer = $event->getOrder()->getOrderCustomer();
        $lineItems = $event->getOrder()->getLineItems();
        $eventClient = $this->predictionEventModel->getEventClient($event->getSalesChannelId());
```

```

foreach ($lineItems->filterByType(LineItem::PRODUCT_LINE_ITEM_TYPE) as $purchasedProduct) {
    $seventClient->recordUserActionOnItem(
        'purchase',
        $customer->getCustomerId(),
        $purchasedProduct->getReferencedId(),
        [ 'itemType' => 'product', 'variant' => $purchasedProduct->getPayload()['productNumber'] ]
    );
}
}

public function onLineItemAdded(BeforeLineItemAddedEvent $event): void
{
    $salesChannelContext = $event->getSalesChannelContext();
    if (!$salesChannelContext->getCustomer()) return;
    if ($event->getLineItem()->getType() !== LineItem::PRODUCT_LINE_ITEM_TYPE) return;
    $seventClient = $this->predictionEventModel->getEventClient(
        $salesChannelContext->getSalesChannelId()
    );
    $seventClient->recordUserActionOnItem(
        'add-to-cart',
        $salesChannelContext->getCustomerId(),
        $event->getLineItem()->getReferencedId(),
        [ 'itemType' => 'product' ]
    );
}
}

```

Даний сабскрайбер викликає специфічний метод взаємодії клієнтів з товаром. Атрибути також присутні. Серед них можуть бути номер варіанту замовленого продукту, кількість, місце доставки, сума замовлення і т.д. Також формується специфікою поставленого завдання.

Для пошуку релевантних пропозицій сервіс PredictionIo пропонує API інтерфейс взаємодії за допомогою класу EngineClient. Після виклику метода `sendQuery()` з описом необхідних параметрів пошуку ми отримуємо масив релевантних товарів, які за вектором подібності найбільше відповідають інтересам користувача. В якості параметрів пошуку достатньо передати тільки ідентифікатор користувача (його id в системі).

3.2 Створення середовища роботи додатків та їх взаємодії

Як було вказано раніше, predictionIo може працювати на окремому веб-сервері та обслуговувати декілька app для довільних суміжних задач. Також можна побудувати кластер для збільшення продуктивності системи. Через це варто відділити даний додаток в окреме мікросервісне середовище незалежно від серверу вебсайту. Для реалізації мікросервісної архітектури використаємо Docker як простий та швидкий в налаштуваннях додаток для контейнеризації. Лістинг налаштувань контейнерів докер для вебсайту наведений в додатку Г. Лістинг налаштування контейнерів докера для predictionIo наведено в додатку Г.

Дана конфігурація створює контейнери pio, adminer, postgres, elasticsearch.

Для взаємодії між контейнерами об'єднаємо їх в одну мережу та додамо мережу взаємодії з веб платформою. При цьому контейнер веб платформи вже повинен бути створеним та запущеним.

```
shopware | DOCKWARE: testing and starting Apache...
shopware | Syntax OK
shopware | * Restarting Apache httpd web server apache2
shopware | ...done.
shopware | -----
shopware | -----
shopware |
shopware | WOH000, dockware/dev:6.5.1.0 IS READY :) - let's get started
shopware | -----
shopware | DOCKWARE CHANGELOG: /var/www/CHANGELOG.md
shopware | PHP: PHP 8.1.18 (cli) (built: Apr 14 2023 04:39:44) (NTS)
shopware | Apache DocRoot: /var/www/html/public
shopware | URLs (if you are using a custom domain, make sure its available using /etc/hosts or other approaches)
shopware | ADMINER URL: http://localhost/adminer.php
shopware | MAILCATCHER URL: http://localhost/mailcatcher
shopware | PIMPHYLOG URL: http://localhost/logs
shopware | SHOP URL: http://localhost
shopware | ADMIN URL: http://localhost/admin
shopware |
shopware | What's new in this version? see the changelog for further details
shopware | https://www.shopware.com/de/changelog/
shopware |
shopware |
```

Рис. 3.1. Запуск контейнера веб платформи

Для запуску контейнерів виконаємо команду об'єднання всіх шарів контейнерів:

```
docker-compose -f docker-compose.yml -f docker-compose.networks.yml -f
adminer/docker-compose.base.yml -f pgsql/docker-compose.base.yml -f
```

elasticsearch/docker-compose.base.yml -f elasticsearch/docker-compose.meta.yml -f pgsql/docker-compose.event.yml -f pgsql/docker-compose.model.yml up. Ця команда завантажить всі необхідні компоненти для створення контейнерів макросервісного середовища системи рекомендацій.

Після білда контейнера можемо зайти всередину контейнера pio для налаштування. По замовчуванню в файлі Dockerfile вже прописані основні команди для встановлення необхідних компонентів, то ж залишилось тільки створити app, додати ключ доступу до налаштувань та збілдити та навчити додаток.

```
pio_1 | [INFO] [Management$] Apache Spark is installed at /usr/share/spark-2.2.2-bin-hadoop2.7
pio_1 | [INFO] [Management$] Apache Spark 2.2.2 detected (meets minimum requirement of 1.6.3)
pio_1 | [INFO] [Management$] Inspecting storage backend connections...
pio_1 | [INFO] [Storage$] Verifying Meta Data Backend (Source: ELASTICSEARCH)...
elasticsearch_1 | [2023-06-09T14:32:49,528][INFO ][o.e.c.m.MetaDataCreateIndexService] [pio_meta] creating index, cause [api], templates [], shards [5]/[1], mappings []
elasticsearch_1 | [2023-06-09T14:32:49,734][INFO ][o.e.c.m.MetaDataMappingService] [pio_meta/F161cscoT262EkmPmL3UtQ] create_mapping [engine_instances]
elasticsearch_1 | [2023-06-09T14:32:49,768][INFO ][o.e.c.m.MetaDataMappingService] [pio_meta/F161cscoT262EkmPmL3UtQ] create_mapping [sequences]
elasticsearch_1 | [2023-06-09T14:32:49,799][INFO ][o.e.c.m.MetaDataMappingService] [pio_meta/F161cscoT262EkmPmL3UtQ] create_mapping [evaluation_instances]
elasticsearch_1 | [2023-06-09T14:32:49,840][INFO ][o.e.c.m.MetaDataMappingService] [pio_meta/F161cscoT262EkmPmL3UtQ] create_mapping [apps]
elasticsearch_1 | [2023-06-09T14:32:49,880][INFO ][o.e.c.m.MetaDataMappingService] [pio_meta/F161cscoT262EkmPmL3UtQ] create_mapping [accesskeys]
pio_1 | [INFO] [Storage$] Verifying Model Data Backend (Source: PGSQL)...
pio_1 | [INFO] [Storage$] Verifying Event Data Backend (Source: PGSQL)...
pio_1 | [INFO] [Storage$] Test writing to Event Store (App Id 0)...
pio_1 | [INFO] [Management$] Your system is all ready to go.
pio_1 | [INFO] [Management$] Creating Event Server at 0.0.0.0:7070
elasticsearch_1 | [2023-06-09T14:32:53,598][INFO ][o.e.c.m.MetaDataMappingService] [pio_meta/F161cscoT262EkmPmL3UtQ] create_mapping [channels]
pio_1 | [INFO] [HttpListener] Bound to /0.0.0.0:7070
pio_1 | [INFO] [EventServerActor] Bound received. EventServer is ready.

o.hornytskyi@dev-ua-nb187:~/PhpstormProjects/Shopware6.5/predictionio/docker$ docker exec -it docker_pio_1 bash
root@6a0fa376bfef:/templates# cd universal-recommender/
root@6a0fa376bfef:/templates/universal-recommender#
```

Рис. 3.2. Запуск контейнерів системи рекомендацій

Тепер виконаємо команди з докерфайлу. Вони виконуються автоматично при створенні та запуску докер контейнерів.

```
#RUN pio app new apisearch
#RUN pio app list

#RUN cd /templates
#RUN git clone https://github.com/actionml/universal-recommender.git
#RUN cd universal-recommender

#RUN apt-get install python
RUN curl https://bootstrap.pypa.io/pip/2.7/get-pip.py -o get-pip.py
RUN python get-pip.py
RUN pip install predictionio datetime

RUN mkdir -p /root/.sbt/launchers/0.13.13/
```

```
RUN wget https://repo.typesafe.com/typesafe/ivy-releases/org.scala-sbt/sbt-launch/0.13.13/sbt-launch.jar -O
/root/.sbt/launchers/0.13.13/sbt-launch.jar
```

```
#RUN python examples/import_handmade.py --access_key {{ token }}
```

```
#RUN cp engine.json.spark-tuning engine.json
```

```
#RUN pio build --verbose -- --driver-memory 6g
```

```
#RUN pio train -- --driver-memory 6g
```

```
#RUN pio deploy -- --driver-memory 6g
```

```
EXPOSE 7070
```

```
EXPOSE 8000
```

Дані команди виконують завантаження та встановлення необхідної версії python, шаблону системи та виконують їх конфігурацію. Як видно з (рис. 3.3), був створений новий аплікейшн та новий ключ доступу доданий до нього. Цей ключ доступу важливий для того, щоб система рекомендацій знала до якого саме аплікейшена звертається веб платформа, а також використовується в цілях безпеки даних. Пізніше він буде доданий в конфігурацію додатку.

```
root@6a0fa376bfef:/templates/universal-recommender# pio app list
[INFO] [Pio$]      Name | ID |      Access Key | Allowed Event(s)
[INFO] [Pio$] Finished listing 0 app(s).
root@6a0fa376bfef:/templates/universal-recommender# pio app new apisearch
[INFO] [App$] Initialized Event Store for this app ID: 1.
[INFO] [Pio$] Created a new app:
[INFO] [Pio$]      Name: apisearch
[INFO] [Pio$]      ID: 1
[INFO] [Pio$] Access Key: IrCGSpKKxauQFAiBxTYk6Hzy0HgQr7FM0SMHsVUv9VroxLUKMrts5D0sCMK33FR
root@6a0fa376bfef:/templates/universal-recommender#
```

Рис. 3.3. Відображення списку створених аплікейшенів

Рис. 3.4. Конфігурація веб додатку

Для зручності користування та обмеження кількості необхідних пропозицій в конфігурацію додатку було також додано значення максимальної кількості необхідних пропозицій. Це дасть можливість не обтяжувати сторінку надмірною кількістю пропозицій, а обмежитися тільки найбільш актуальними.

Наступним кроком буде проведення індексації всіх товарів веб платформи. Додаток створений таким чином, що при запуску вбудованого механізму індексування товарів буде виконуватися індексація (синхронізація) всіх товарів в систему рекомендацій з усіма атрибутами – варіантами товарів та навчання системи.

☐ Modify	id	event	entitytype	entityid	targetentitytype	targetentityid	
☐ edit	4ff6699b8ae04525b9928a9db69ffd5b	\$set	item	3ac014f329884b57a2cce5a29f34779c	NULL	NULL	{"itemType": "product"}
☐ edit	826a30bd6bb449e7b17f041ac10c25ce	\$set	item	11dc680240b04f469ccba354cbf0b967	NULL	NULL	{"itemType": "product"}
☐ edit	cb32d5e370084e1eb331faf48326c096	\$delete	item	2a88d9b59d474c7e869d8071649be43c	NULL	NULL	{}
☐ edit	4a885dfa5e034364bad17bb711aafd2c	\$delete	item	3ac014f329884b57a2cce5a29f34779c	NULL	NULL	{}
☐ edit	e8e58975d2bf4a32b0a98fcf12628674	\$delete	item	11dc680240b04f469ccba354cbf0b967	NULL	NULL	{}
☐ edit	c780c026a8e644118463ac671d49056f	\$delete	item	03ac1d11d0e9415bba9cda1a52f6b032	NULL	NULL	{}
☐ edit	15ba7cb8a64d4e8e8eaf5ae5dd25abd	\$set	item	0d1c5fc140744225a442e951f8159978	NULL	NULL	{"itemType": "product"}
☐ edit	cc2676eb1dff447499c6f1941a350e66	\$set	item	4e4772de2344fd682f25ceca984357d	NULL	NULL	{"itemType": "product"}
☐ edit	2921b891f8414ffca3e4ea13f364f583	\$set	item	2a88d9b59d474c7e869d8071649be43c	NULL	NULL	{"itemType": "product"}
☐ edit	bb6cb340aa8b48d0a8aa2853c7675c2d	\$delete	item	1901dc5e888f4b1ea4168c2c5f005540	NULL	NULL	{}
☐ edit	2667bd87840b4c56aa371f3621dacf0f	\$delete	item	c241129d01b24e3d96ba618ba0d3325d	NULL	NULL	{}
☐ edit	3a03db08aabb4b79bcd8c5d66d1b5a5	\$delete	item	0d1c5fc140744225a442e951f8159978	NULL	NULL	{}
☐ edit	78d90703bb684571b84ff4cd3ea45faf	\$set	item	c326bf6e981f452db358549474c34131	NULL	NULL	{"itemType": "product"}
☐ edit	cd8c840022ce45d1bf6948a6dd6081fb	\$delete	item	c326bf6e981f452db358549474c34131	NULL	NULL	{}
☐ edit	2fc0c49581554561aeca43cfc4f9f5fd	\$set	item	0d65487f3e1c4d6ab4c318723f7a0200	NULL	NULL	{"itemType": "product"}
☐ edit	31da42e411e64815bd2c7a43224810b2	\$delete	item	0d65487f3e1c4d6ab4c318723f7a0200	NULL	NULL	{}
☐ edit	6a65c92eaf8e4320ab72af890ac165de	\$set	item	c7bca22753c84d08b6178a50052b4146	NULL	NULL	{"itemType": "product"}
☐ edit	4f01b22ccf1c441ca2ecbdfbd081eb7	\$delete	item	c7bca22753c84d08b6178a50052b4146	NULL	NULL	{}
☐ edit	3c762ca602774a47b375d0daf669a35b	\$set	item	c241129d01b24e3d96ba618ba0d3325d	NULL	NULL	{"itemType": "product"}
☐ edit	52c6e53fb02442fe87fe4b10e685ae56	\$set	item	1901dc5e888f4b1ea4168c2c5f005540	NULL	NULL	{"itemType": "product"}
☐ edit	ad872bae0df04316a53c47d17e130e19	\$set	item	f68e68b680e2420b9f9386284cf323b7	NULL	NULL	{"itemType": "product"}
☐ edit	2a9af4d87fd740fbb94d5423b1d18bb	\$set	item	03ac1d11d0e9415bba9cda1a52f6b032	NULL	NULL	{"itemType": "product"}
☐ edit	2d915b068f524cd897b8c92c61ab6	\$delete	item	f68e68b680e2420b9f9386284cf323b7	NULL	NULL	{}
☐ edit	d54bca15a5c4081b72d8b73106fbd3	\$delete	item	4e4772de2344fd682f25ceca984357d	NULL	NULL	{}

Рис. 3.5. Результат індексації товарів в базі даних

Навчання системи відбувається за допомогою консольної команди “`rio train -- --driver-memory 6g`”. В даному випадку додатковим параметром вказується обмеження кількості використаної пам’яті процесі навчання. Якщо цей параметр не вказати, то буде використовуватися вся доступна пам’ять, що може призвести до її закінчення, оскільки процес навчання дуже ресурсоємний процес.

```

[INFO] [Engine$] [warn] Strategy 'concat' was applied to a file
[INFO] [Engine$] [warn] Strategy 'discard' was applied to 46 files
[INFO] [Engine$] [info] Assembly up to date: /templates/universal-recommender/target/scala-2.11/universal-recommender-assembly-0.7.3-deps.jar
[INFO] [Engine$] [success] Total time: 2 s, completed Jun 9, 2023 2:55:37 PM
[INFO] [Engine$] Compilation finished successfully.
[INFO] [Engine$] Looking for an engine...
[INFO] [Engine$] Found universal-recommender-assembly-0.7.3-deps.jar
[INFO] [Engine$] Found universal-recommender_2.11-0.7.3.jar
[INFO] [Engine$] Build finished successfully.
[INFO] [Pio$] Your engine is ready for training.
root@6a0fa376bfef:/templates/universal-recommender#

[INFO] [RootSolverFactory$] Unable to create class OMPMMul: falling back to java version
[INFO] [URModel] ES fields[10]: List(purchase, expires, countries, view, available, id, categories, popRank, defaultRank, date)
[INFO] [EsClient$] Create new index: urindex_1686323243366, items, List(purchase, expires, countries, view, available, id, categories, p
, true), purchase -> (keyword, true), popRank -> (float, false), view -> (keyword, true))
[INFO] [EsClient$] Number of ES connections for saveToEs: 16
[INFO] [Engine$] org.apache.predictionio.data.storage.NullModel does not support data sanity check. Skipping check.
[INFO] [Engine$] EngineWorkflow.train completed
[INFO] [Engine] engineInstanceId=AYigsk5m_uyKIJabg321
[INFO] [CoreWorkflow$] Inserting persistent model
[INFO] [CoreWorkflow$] Updating engine instance
[INFO] [CoreWorkflow$] Training completed successfully.
[INFO] [AbstractConnector] Stopped Spark@5bcec67e{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
root@6a0fa376bfef:/templates/universal-recommender#

```

Рис. 3.6. Навчання системи

Після проведення навчання необхідно виконати команду “`pio deploy -- --driver-memory 6g`” для початку опрацювання запитів на генерацію рекомендацій. В даному випадку було також обмежено кількість пам’яті для використання.

```

[INFO] [URAlgorithm]

Init URAlgorithm

App name           apisearch
ES index name      urindex
ES type name       items
RecsModel          all
Event names        List(purchase, view, product-view, add-to-cart)

Random seed        -1598876574
MaxCorrelatorsPerEventType  50
MaxEventsPerEventType  500
BlacklistEvents    List(purchase)

User bias          1.0
Item bias          1.0
Max query events   100
Limit              20

Rankings:
popular            Some(popRank)

[INFO] [MasterActor] Undeploying any existing engine instance at http://0.0.0.0:8000
[WARN] [MasterActor] Nothing at http://0.0.0.0:8000
[INFO] [HttpListener] Bound to /0.0.0.0:8000
[INFO] [MasterActor] Engine is deployed and running. Engine API is live at http://0.0.0.0:8000.

```

Рис. 3.7 Запуск системи для збору даних та генерації рекомендацій

Після запуску команди на обробку запитів система готова для пошуку персоналізованих рекомендацій по критеріям пошуку. Критерії пошуку можемо передавати в якості параметрів методу “sendQuery()” класу EngineClient. В якості параметра для пошуку по замовчуванню будемо використовувати параметр “user“, що вказує на те для якого користувача мають генеруватися рекомендації.

3.3 Тестування додатку

Для тестування додаток був встановлений на діючий проект та проведена індексація всіх товарів для наповнення бази системи рекомендації. Також були увімкнені обробники подій для перегляду товару, додавання його до кошика, реєстрації користувача в систему для створення векторів вподобань.

Після запуску середовища виконаємо створення елементів для рекомендацій та двох користувачів. Перший буде генерувати події для товарів, корзини та покупки, а другий буде користуватися результатом рекомендацій на основі досвіду першого користувача. Скріншоти створених користувачів наведені на (рис 3.8).

The image displays two identical user profile pages side-by-side, representing two different users: test11 and test22. Each page has a sidebar on the left with the following links: Overview, My profile (highlighted), Addresses, Wishlist, My orders, and Newsletter. Below these links is a Logout button. The main content area is titled 'My profile' and includes a sub-header 'Check your personal data.' Below this is a 'Personal data' section with a form. The form has two columns: 'First name*' and 'Last name*'. For test11, the first name is 'test11' and the last name is 'Gornitskiy11'. For test22, the first name is 'test22' and the last name is 'Gornitskiy22'. Below the form, there is a note: 'Fields marked with asterisks (*) are required.' and a 'Save changes' button.

Рис. 3.8. Приклад створених користувачів

Далі перший користувач має згенерувати декілька подій взаємодії з товарами. Для цього перший користувач переглядає декілька товарів, додає їх в кошик та оформляє замовлення з одним з переглянутих товарів. При цих діях додаток збирає інформацію про взаємодію та передає її в систему збору даних за допомогою методу “recordUserActionOnItem()” класу EventClient.

Всі події взаємодії фіксуються в базі даних та будуть в подальшому використовуватися для розрахунку та створення рекомендацій іншим користувачам. Приклад створених подій наведені на (рис 3.9)..

☐ Modify	id	event	entitytype	entityid	targetentitytype	targetentityid	pr
☐ edit	e29572b166e44c6bbbc7acfb9889deb9	purchase	user	a2cc6421d91d4919a0f482c11e669c83	item	aab4bf9048a048c784344fb5254f2e8c	{"itemType": "product", "variant": "SWDEMO10005.6"}
☐ edit	c1b587915a394ee393969d2ed31cb555	add-to-cart	user	a2cc6421d91d4919a0f482c11e669c83	item	c241129d01b24e3d96ba618ba0d3325d	{"itemType": "product"}
☐ edit	722a15bbadee45e1bdd0ba6f34f670c9	view	user	a2cc6421d91d4919a0f482c11e669c83	item	c241129d01b24e3d96ba618ba0d3325d	{"itemType": "product", "variant": "SWDEMO10005.6"}
☐ edit	dc8438865a8341469d381ed768bc7f05	view	user	a2cc6421d91d4919a0f482c11e669c83	item	03ac1d11d0e9415bba9cda1a52f6b032	{"itemType": "product", "variant": "SWDEMO10005.5"}
☐ edit	722a15bbadee45e1bdd0ba6f34f670c9	view	user	a2cc6421d91d4919a0f482c11e669c83	item	c241129d01b24e3d96ba618ba0d3325d	{"itemType": "product", "variant": "SWDEMO10005.6"}

Рис. 3.9. Приклад створених подій

Аналогічні події генеруються для другого та всіх інших користувачів. Важливою умовою для тестування є те, що користувачі мають генерувати події з максимально подібними атрибутами. Користувачі також мають бути побідними у своїх атрибутах. Створені події в базі даних представлені на (рис 3.10).

☐ Modify	id	event	entitytype	entityid	targetentitytype	targetentityid	
☐ edit	8ac62a4f80ae46f6a748a7d2b10007dd	\$set	user	743b27e231c942ea99e794c9e626ec25	NULL	NULL	{}
☐ edit	e29572b166e44c6bbbc7acfb9889deb9	purchase	user	a2cc6421d91d4919a0f482c11e669c83	item	aab4bf9048a048c784344fb5254f2e8c	{"itemType": "product", "variant": "SWDEMO10005.6"}
☐ edit	c1b587915a394ee393969d2ed31cb555	add-to-cart	user	a2cc6421d91d4919a0f482c11e669c83	item	c241129d01b24e3d96ba618ba0d3325d	{"itemType": "product"}
☐ edit	722a15bbadee45e1bdd0ba6f34f670c9	view	user	a2cc6421d91d4919a0f482c11e669c83	item	c241129d01b24e3d96ba618ba0d3325d	{"itemType": "product", "variant": "SWDEMO10005.6"}

Рис. 3.10. Приклад створених подій

Створені події беруть участь в уточненні коефіцієнтів подібності та покращенні релевантних пропозицій користувачам. Можемо земулювати запит до сервісу рекомендацій та передати в якості параметра пошуку ідентифікатор другого користувача. Результат запиту зображено на (рис. 3.11)

```

www-data@847837a40c74:~/html$ curl -H "Content-Type: application/json" -d '{"user": "a2cc6421d91d4919a0f482c11e669c83", "num": 4 }' http://pio:8000/queries.json
{"itemScores":[{"item": "Iphone 5", "score": 0.0}, {"item": "Galaxy", "score": 0.0}, {"item": "Iphone 4", "score": 0.0}, {"item": "Iphone 6", "score": 0.0}]}www-data@847837a40c74:~/html$

www-data@847837a40c74:~/html$ curl -H "Content-Type: application/json" -d '{"user": "743b27e231c942ea99e794c9e626ec25", "num": 4 }' http://pio:8000/queries.json
{"itemScores":[{"item": "aab4bf9048a048c784344fb5254f2e8c", "score": 1.9616584}, {"item": "Iphone 5", "score": 0.0}, {"item": "Galaxy", "score": 0.0}, {"item": "Iphone 4", "score": 0.0}]}www-data@847837a40c74:~/html$

```

Рис. 3.11. Результат запиту

Саме даний запит використовується для отримання рекомендацій в додатку. Результатом запиту є відповідь у форматі json, яка містить відомості

про рекомендації у вигляді серіалізованого (представленого у вигляді рядка) масиву. Перший елемент масиву відповідає за найбільш релевантну рекомендацію.

Для візуального відображення рекомендованих товарів було додано HTML шаблон для представлення інформації про товари у вигляді анімованої каруселі. Елемент каруселі містить зображення товару, назву, ціну товару, а також інформацію про можливі знижки.

Припустимо, що перший користувач оформить одне чи два замовлення. В такому випадку події оформлення замовлення будуть мати найвищий пріоритет, оскільки дана подія призводить до транзакції.

Hello, test11 Gornitskiy11

- Overview
- My profile
- Addresses
- Wishlist
- My orders
- Newsletter

Logout

Orders

Your recent orders:

Order: 07/12/2023 Open

Order number: 1212456

Shipping status	Payment status	Payment method	Shipping method
Open	Paid	IDEAL	Home delivery

Product	Quantity	Unit price	Subtotal
 Brody Hoodie Color: Bright Sage Green Size: 14	1	€33.50	€33.50

Product	Quantity	Unit price	Subtotal
 Nora Blazer Color: Fawn Size: 40	1	€205.00	€205.00

Рис. 3.12. Транзакції оформлені першим користувачем

Відтак, для другого користувача дані товари будуть найбільш релевантними рекомендаціями, оскільки за основу рекомендацій обраний Collaborative filtering алгоритм. Якщо тепер переглянути рекомендації першому та другому користувачам, то можна побачити наступне.

Shopping bag updated

Shopping bag



Finest Hoodie
Color: Black | Size: 12

€77.00
X 1

Related products

The following products provide an alternative or addition to you.



NIKKIE
€155.00- €62.00
SLEEVELESS DRESS WITH BUTTONS



NIK&NIK
€72.00
DEWEE AD DRESS



KATE MOSS X NIKKIE
€145.00
HIGH DRIE VERAN I FATHED SHORTS WITH

Summary

Total	€77.00
Shipping costs	€1.95
The shipping costs will be recalculated in the next step	
Grand total	€78.95
Net total	€65.25
21% VAT	€13.70

Enter discount code ...

PROCEED TO CHECKOUT



- Easy to return
- Quick post-payment with Riverty

Shopping bag



Ellery Top
Color: Black/Marrow | Size: 38

€92.00
X 1

Related products

The following products provide an alternative or addition to you.



NIK&NIK
€67.00- €33.50
CROPPED HOODIE



FIFTH HOUSE
€205.00
SATIN LOOK SINGLE-BREASTED BLAZER



NIKKIE
€155.00- €62.00
SLEEVELESS DRESS WITH BUTTONS

Summary

Total	€92.00
Shipping costs	€2.95
The shipping costs will be recalculated in the next step	
Grand total	€94.95
Net total	€79.13
20% VAT	€15.82

Enter discount code ...

PROCEED TO CHECKOUT



- Easy to return
- Quick post-payment with Riverty

Рис. 3.13. Порівняння рекомендацій першому та другому користувачу

Відповідно до рисунку (рис 3.13) можна побачити, що другому користувачу пропонуються речі, які замовив перший користувач з вищим пріоритетом. За ними слідує товар, однаковий для обох користувачів. Це означає, що для них однакові коефіцієнти подібності, тобто не визначено подій взаємодії користувачів з цими товарами, або ж відмінності в коефіцієнтах мінімальні.

Висновки до третього розділу

1. Під час практичної реалізації програмного додатку було створено мультисервісну систему для збору інформації про всі можливі товарні пропозиції веб сайту та їх атрибутів. Загалом було додано 88 тисяч 399 товарів з їхніми атрибутами.

2. Було розроблено механізм збору інформації про уподобання користувачів шляхом фіксації переглянутих товарів, додавання товарів до кошика конкретним користувачем. Дана інформація дає змогу побудувати вектори подібності уподобань користувачів шляхом навчання системи релевантних пропозицій. В тестовому режимі було створено два незалежні користувачі які переглядали довільні товари на веб сайті. Загалом було переглянуто 34 товари та оформлено 3 замовлення першим користувачем, а також переглянуто 12 товарів другим користувачем.

3. Результатом навчання системи було згенеровано список релевантних пропозицій на сторінці кошика та візуалізовано у вигляді анімованої каруселі. Таке розташування було обране як одне з пріоритетних на основі аналізу можливих положень на популярних ритейлсторах. Метою такого розташування списку релевантних пропозицій є спонукання клієнтів збільшення вартості кошика додаванням супутніх товарів або альтернативним товаром.

В прикладі було обрано товарів на суму 94.95 євро, а за умови використання альтернативного товару вартість кошика могла б змінитися/збільшитися до 128.55 євро з урахуванням акційної пропозиції, або 205 євро в разі заміни, що на 26,2% та 115,8% більше відповідно. (приведена вартість рекомендованих товарів).

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було проведено аналіз теоретичних основ побудови алгоритмів рекомендацій на основі контенту та колабораційних алгоритмів. Для цього було вирішено наступні завдання:

1. Проведений аналіз предметної області та поглиблене вивчення тематики дозволив визначити основні задачі, що необхідні для реалізації веб-додатку для генерування релевантних пропозицій користувачам інтернет магазину. В результаті проведеного аналізу методів та засобів реалізації алгоритмів рекомендацій були визначені рекомендаційні блоки, що допомагають вирішити поставлені завдання.

2. Виявлено, що, якщо додати рекомендації в набір товарів для продажу, продавці можуть у середньому збільшити кількість конверсій до 40% за тієї самої відвідуваності. Визначено, що поєднання каталогів з привабливими пропозиціями і унікальними рішеннями для аудиторій, ритейлери можуть збільшити обсяги продажів потенційним клієнтам на 94%, а рентабельність витрат на рекламу зменшити на 28%.

3. Було проведено аналіз роботи успішних ритейл платформ, запропоновано стратегію фінансово-економічної безпеки підприємства та розглянуто моделі створення персоналізованих рекомендацій користувачу.

4. Було проаналізовано атрибути, необхідні для реалізації алгоритму, функціональні та нефункціональні вимоги до системи, розроблено моделі взаємодії клієнтів, ритейлерів та системи;

5. Визначено, що оптимальною архітектурою для реалізації буде MVC, обрано необхідний та достатній набір програмних продуктів для реалізації поставлених задач. Розглянуто алгоритми основних процесів системи та наведено детальний розгляд головних аспектів реалізації web-додатку.

6. Розроблено web -додаток для перевірки роботи рекомендацій на основі колаборативного алгоритму, описано порядок встановлення і налаштування

додатку та проведено тестування системи. Програмний додаток базується на використанні створеного Plugin (Bundle), який реалізує взаємодію між компонентами фреймворку Shopware та SDK predictionIo. Під час практичної реалізації програмного додатку було створено мультисервісну систему для збору інформації про всі можливі товарні пропозиції веб сайту та їх атрибутів. Загалом було додано 88 тисяч 399 товарів з їхніми атрибутами.

7. Було розроблено механізм збору інформації про уподобання користувачів шляхом фіксації переглянутих товарів, додавання товарів до кошика конкретним користувачем. Дана інформація дає змогу побудувати вектори подібності уподобань користувачів шляхом навчання системи релевантних пропозицій. В тестовому режимі було створено два незалежні користувачі які переглядали довільні товари на веб сайті. Загалом було переглянуто 34 товари та оформлено 3 замовлення першим користувачем, а також переглянуто 12 товарів другим користувачем.

8. Результатом навчання системи було згенеровано список релевантних пропозицій на сторінці кошика та візуалізовано у вигляді анімованої каруселі. Таке розташування було обране як одне з пріоритетних на основі аналізу можливих положень на популярних ритейлсторах. Метою такого розташування списку релевантних пропозицій є спонукання клієнтів збільшення вартості кошика додаванням супутніх товарів або альтернативним товаром.

В прикладі було обрано товарів на суму 94.95 євро, а за умови використання альтернативного товару вартість кошика могла б змінитися/збільшитися до 128.55 євро з урахуванням акційної пропозиції, або 205 євро в разі заміни, що на 26,2% та 115,8% більше відповідно. (приведена вартість рекомендованих товарів).

9. В результаті проведеної роботи було вирішено поставлену задачу та розроблено додаток для генерації персоналізованих релевантних пропозицій. Реалізований програмний комплекс готовий до використання.

10. В ході роботи отримані наступні висновки щодо роботи алгоритмів:
- Рекомендації неможливі без порівняння певних абстрактних величин.

- Рекомендації взаємозалежні від атрибутів переданих для аналізу.
- Рекомендації можуть змінюватися з часом – це динамічна система, разом з користувачами, які можуть змінювати свій «кластер» вподобань при появі нових подій, які вони генерують, нових користувачів та нових елементів рекомендації.
- оцінивши роботу додатку за тестовими даними було виявлено, що сума кошика може збільшитися до 115,8%.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Сухий О. Л. Алгоритми пошуку в інформаційних системах: методичні рекомендації [Електронний ресурс] / О. Л. Сухий, В. М. Міленін, В. М. Тарадайнік. – К., 2015. – 2,0 д.а. – Режим доступу до ресурсу: https://lib.iitta.gov.ua/704951/1/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4.%20%D1%80%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D0%B0%D1%86%D1%96%D1%97_%D0%A1%D1%83%D1%85%D0%B8%D0%B9%202015.pdf.
2. Горницький О. В. Дослідження психоемоційного стану інтернет магазину / Горницький О. В., Вакалюк Т. А. // ІІ Всеукраїнська науково-технічна конференція «Комп'ютерні технології: інновації, проблеми, рішення». – 2023.
3. Conway J. How to Optimise Your Product Mix with Consumer Trends and E-Commerce Data [Електронний ресурс] / Jared Conway. – 2023. – Режим доступу до ресурсу: <https://go.euromonitor.com/white-paper-230302-Optimising-Product-Mix-with-Consumer-Trends-and-E-Commerce.html>.
4. Полторак К. А. Аналіз особливостей поведінки споживачів в мережі інтернет. [Електронний ресурс] / К. А. Полторак // Національний технічний університет України КПУ, м.Київ. – 2015. – Режим доступу до ресурсу: http://www.economy.nayka.com.ua/pdf/11_2015/99.pdf.
5. Brainsolis. Portal of Social Business and Social Media [Електронний ресурс] – Режим доступу до ресурсу: www.briansolis.com.
6. Online shopping personalization [Електронний ресурс] – Режим доступу до ресурсу: <https://www.invespcro.com/blog/online-shopping-personalization/>.
7. Improvement of Personalized Recommendation Algorithm Based Content-boosted Collaborative Filtering Algorithm. [Електронний ресурс] – Режим доступу до ресурсу: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=595c713b80cb0264a6a8e51c666243ee562070a2>.

8. Момот Т. В. Фінансово-економічна безпека : стратегічна аналітика та аудиторський супровід : монографія / Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова ; за заг. ред. Т. В. Момот. – Харків : ХНУМГ ім. О. М. Бекетова, 2015. – 340 с.
9. Melville P. Content-Boosted Collaborative Filtering for Improved Recommendations. National Conference on Artificial Intelligence / P. Melville, R. Mooney, R. Nagarajan. – Edmonton, Canada: Edmonton, Canada, 2016. – 192 с. – (AAAI-2002). – (AAAI).
10. Мазурік О. Ю. Порівняльний аналіз моделей оцінювання в рекомендаційних системах. Системний аналіз та інформаційні технології / О. Ю. Мазурік. – Київ: НТУУ «КПІ», 2016. – 113 с.
11. Recommendation Systems: Content-Based Filtering [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/mlearning-ai/recommendation-systems-content-based-filtering-e19e3b0a309e>.
12. Linden G. Item-to-Item Collaborative Filtering / G. Linden, B. Smith, J. York. – Los Alamitos, CA, USA: IEEE Internet Computing, 2003.
13. Калініна І. О., Дослідження ефективності методів класифікації при прогнозуванні в задачах машинного навчання / Калініна І. О., Гожий О. П. // Управління розвитком складних систем, Київ. – 2021. – № 46. – С. 173 – 180.
14. Polynomial Regression (2018), [Електронний ресурс] – Режим доступу до ресурсу: <https://towardsdatascience.com/polynomial-regression-bbe8b9d97491>.
15. Sheremet O. I. Metod opornyih vektorov (Support-vector machine (SVM)). Mathematical modeling / O. I. Sheremet, O. V. Sadovoi. // Mathematical modeling. – 2013. – №28. – С. 13–17.
16. Нові функції, що дають змогу підвищити попит за допомогою оголошень Discovery [Електронний ресурс] – Режим доступу до ресурсу: <https://support.google.com/google-ads/answer/13404405?hl=uk>.

17. Про атрибуцію на основі даних [Електронний ресурс] – Режим доступу до ресурсу: <https://support.google.com/google-ads/answer/6394265?sjid=2946638495239299172-EU>.
18. PredictionIO Python SDK - Official manual [Електронний ресурс] – Режим доступу до ресурсу: <https://pypi.org/project/PredictionIO/>.
19. PredictionIo - Official manual [Електронний ресурс] – Режим доступу до ресурсу: <https://predictionio.apache.org/start/>.
20. Bell R. Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights [Електронний ресурс] / R. Bell, Y. Koren // IEEE International Conference on Data Mining. – 2007. – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/document/4470228>.
21. Create production-grade machine learning models with TensorFlow. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tensorflow.org/?hl=en>.