

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра комп'ютерних наук

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему «Інформаційна технологія рекомендаційного сервісу для інтернет-магазину з використанням машинного навчання»

Виконав: студент 2 курсу,
групи 1КН-19м
спеціальності 122 «Комп'ютерні науки»
Пеньков В.С.

Керівник: к.т.н., ст. викл. Озеранський В.С.

Рецензент: д.т.н., проф. Романюк О.Н.

Вінниця 2020

ЗАТВЕРДЖУЮ

_____ Червяков І.М.

«__» _____ 2020 р.

ЗАТВЕРДЖУЮ

Зав. каф. КН, д.т.н., проф.

_____ А.А. Яровий

«__» _____ 2020 р.

ЗАВДАННЯ

на магістерську кваліфікаційну роботу на здобуття кваліфікації магістра наук зі спеціальності: 122 – «Комп'ютерні науки»

08-22.МКР.008.19.000.ПЗ

Магістранта групи 1КН-19м Пенькова Владислав Сергійовича

Тема магістерської кваліфікаційної роботи: «Інформаційна технологія рекомендаційного сервісу для інтернет-магазину з використанням машинного навчання»

Вхідні дані: мова програмування – скриптова, повинна забезпечувати можливість маніпулювання даними і управління базами даних, сервер яких підтримує HTTP протокол; база даних ; час зберігання історії користувача – 1 місяць.

Короткий зміст частин магістерської кваліфікаційної роботи:

1. Графічна: Схема алгоритму створення користувача, алгоритм пошук товару, алгоритму роботи рекомендаційної системи, концептуальна діаграма інформаційної технології.

2. Текстова (пояснювальна записка): Вступ, Огляд і аналіз стану проблеми що розглядається, Поняття рекомендаційної системи, Рекомендаційні системи на основі спільної фільтрації, Рекомендаційні системи на основі вмісту, Рекомендаційні системи засновані на знаннях, Розробка рекомендаційної системи для сервісів зі «складними» предметними сферами, Стратегія надання рекомендацій, Математичний опис задачі, Висновок, Проектування системи, Розробка системних вимог, Розробка функціональних вимог, Розробка архітектурних вимог, Висновок, Програмна реалізація інформаційної технології з розробки та дослідження рекомендаційного сервісу для інтернет магазину з використанням технологій машинного навчання, Вибір СУБД, Розробка БД, Вибір мови програмування, Розробка алгоритмів інформаційної технології, тестування та аналіз результатів роботи рекомендаційної системи, Висновок, Економічна частина, Висновки, Список використаної літератури. Додатки.

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз сучасних підходів до створення рекомендаційних систем. Постановка задач дослідження			Аналітичний огляд літературних джерел, задачі досліджень, розділ 1
2	Розробка та проектування рекомендаційної системи			Математичні моделі, розділ 2, 3
3	Програмна реалізація розробленої інформаційної технології, тестування та оцінка параметрів			База даних, Інтернет ресурси, Алгоритми роботи РС розділ 4
4	Економічна частина			розділ 5
5	Апробація та/або впровадження результатів дослідження			тези доповідей
6	Оформлення пояснювальної записки, графічного матеріалу та презентації			Пояснювальна записка, графічний матеріал, презентація

Консультанти з окремих розділів магістерської кваліфікаційної роботи

1. Науковий керівник _____ канд. техн. наук, ст. викл. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)
 “ ____ ” _____ 20__ р. _____ В.С. Озеранський

ініціали та прізвище

2. Економічна частина _____ канд. екон. наук, доц. кафедри ЕПВМ
(підпис) наук. ступінь, вчене звання (посада)
 _____ Бальзан М.В.

ініціали та прізвище

“ ____ ” _____ 20__ р.

Дата попереднього захисту роботи “ ____ ” _____ 20__ р.

Рецензент _____ д. т. н., професор, завідувач кафедри ПЗ
(підпис) наук. ступінь, вчене звання (посада)
 _____ О. Н. Романюк

ініціали та прізвище

Завдання видав науковий керівник _____ канд. техн. наук, ст. викл. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)
 _____ В.С. Озеранський

ініціали та прізвище

“ ____ ” _____ 20__ р.

Завдання отримав магістрант _____ В.С. Пеньков

_____ (підпис) _____ ініціали та прізвище
 “ ____ ” _____ 20__ р.

АННОТАЦІЯ

В даній роботі представлено огляд існуючих рекомендаційних систем, удосконалено існуючий підхід до створення рекомендаційних систем використовуючи метаматичні засоби. Обґрунтовується необхідність використання технологій при вирішенні задач рекомендаційних систем.

Обґрунтовується вибір методу та програмного середовища для розробки та реалізації даної системи та алгоритму прийняття рішень. Представлена покрокова формалізація загальної моделі, яка ідентифікує моменти входу та виходу в системі.

На основі створеної моделі розроблено інформаційну технологію аналізу даних, яка практично реалізована у вигляді рекомендаційної системи для інтернет ресурсу в середовищі PHP з використанням технологій машинного навчання.

ABSTRACT

This work provides an overview of existing recommendation systems, improves the existing approach to the creation of recommendation systems using metamatic tools. The necessity of using technologies in solving the problems of recommendation systems is substantiated.

The choice of method and software environment for the development and implementation of this system and decision-making algorithm is substantiated. A step-by-step formalization of the general model is presented, which identifies the moments of entry and exit in the system.

Based on the created model, the information technology of data analysis was developed, which is practically implemented in the form of a recommendation system for an Internet resource in the PHP environment using machine learning technologies.

ЗМІСТ

ВСТУП.....	8
1 ОГЛЯД ТА АНАЛІЗ СУЧАСНОГО СТАНУ НАДАННЯ РЕКОМЕНДАЦІЙ В ІНТЕРНЕТ-МАГАЗИНІ	11
1.1 Поняття рекомендаційної системи	11
1.2 Рекомендаційні системи на основі спільної фільтрації	13
1.3 Рекомендаційні системи на основі вмісту	19
1.4 Рекомендаційні системи засновані на знаннях	22
1.5 Висновок	24
2 РОЗРОБКА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДЛЯ СЕРВІСІВ ЗІ «СКЛАДНИМИ» ПРЕДМЕТНИМИ СФЕРАМИ	25
2.1 Стратегія надання рекомендацій	25
2.2 Математична модель надання рекомендацій для користувачів інтернет- магазину	28
2.3 Висновки	35
3 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РЕКОМЕНДАЦІЙНОГО СЕРВІСУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ	36
3.1 Розробка системних вимог для рекомендаційного сервісу	36
3.2 Розробка функціональних вимог для рекомендаційного сервісу	37
3.3 Розробка архітектурних вимог для рекомендаційного сервісу	42
3.4 Висновки	43
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РЕКОМЕНДАЦІЙНОГО СЕРВІСУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ.....	46
4.1 Вибір СУБД для рекомендаційного сервісу.....	46
4.2 Розробка БД для рекомендаційного сервісу.....	46
4.3 Вибір мови програмування	48
4.4 Розробка алгоритмів функціонування інформаційної технології рекомендаційного сервісу для інтернет-магазину з використанням машинного навчання».....	52

	7
4.5 Тестування та аналіз результатів роботи рекомендаційного сервісу для інтернет-магазину з використанням машинного навчання»	55
4.6 Висновок	60
5 ЕКОНОМІЧНА ЧАСТИНА.....	61
5.1 Оцінювання комерційного потенціалу розробки.....	61
5.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи	62
5.3 Прогнозування комерційних ефектів від реалізації результатів розробки.....	65
5.4 Розрахунок ефективності вкладених інвестицій та період їх окупності...	67
5.5 Висновок	70
ВИСНОВКИ.....	71
ЛІТЕРАТУРА.....	72
ДОДАТКИ.....	75
Додаток А Інструкція користувача.....	76
Додаток Б Лістинг програми.....	78
Додаток В Графічні матеріали	106
Додаток Г Акт впровадження	122

ВСТУП

Актуальність теми дослідження. Останнім часом кількість інтернет-платформ, що пропонують користувачам різноманітні товари збільшується з експоненціальною швидкістю, а рекомендаційні системи стають незамінними компонентами будь-якої подібної веб-системи. Різноманітні інтернет-магазини, електронні дошки оголошень, соціальні мережі, платформи з аудіо- та відеоконтентом, комерційні організації, що мають великі каталоги товарів змогли отримати вигоду від подібних систем. Адже орієнтація у великих каталогах товарів надзвичайно складна і потенційного користувача може втомити робота з такою платформою або він не зможе знайти саме ті елементи, які йому потрібні, навіть з урахуванням пошукових фільтрів, а рекомендаційні системи дозволяють вирішити подібні проблеми.

У загальному розумінні рекомендаційні системи — це активні системи фільтрації інформації, які намагаються надати користувачу інформаційні елементи (кіно, телебачення, музика, новини, веб-сторінки), у яких він зацікавлений. Наразі існує дві основні стратегії по створенню рекомендаційних систем: спільна (колаборативна) фільтрація та фільтрація на основі контенту. А також гібридні підходи, що використовують обидві стратегії у певній мірі. Зазвичай великі сервіси, такі як Netflix, YouTube чи Amazon, які мають величезні клієнтські бази і каталоги товарів, використовують колаборативну фільтрацію або гібридний підхід, хоча при цьому рекомендаційні системи засновані на контенті також можуть використовуватися.

Зв'язок роботи з науковими програмами, планами, темами.

Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка спеціалізованих засобів штучного інтелекту на основі інтелектуального аналізу даних та машинного навчання».

Метою даної роботи є покращення показника якості надання рекомендацій рекомендаційною системою для використання в інтернет магазинах з різними

предметними областями що несе за собою збільшення прибутку для підприємства.

Задачі дослідження. Для досягнення поставленої мети необхідно розв'язати такі задачі:

- огляд існуючих стратегій по створенню рекомендаційних систем;
- аналіз можливості застосування розглянутих стратегій в рамках сфер що мають складний (комплексний) простір елементів, який використовується нечасто;
- розробка математичної моделі для обробки обраних товарів для надання рекомендацій;
- удосконалення інформаційної технології рекомендаційного сервісу для інтернет-магазину.
- створення алгоритму та методики надання рекомендацій на базі розробленої математичної моделі;
- програмна реалізація інформаційної технології рекомендаційного сервісу для інтернет-магазину з використанням машинного навчання;
- тестування та аналіз результатів роботи сервісу.

Об'єктом дослідження є процес надання рекомендацій користувачам інтернет-магазину з використанням машинного навчання.

Предметом дослідження є програмні засоби створення рекомендацій, які застосовуються для прогнозування уподобань користувачів та надання їм пропозицій, згідно їх інтересів.

Методи дослідження - метод статистичної обробки даних, методи прикладної математики, методи збору та фільтрації цифрових даних.

Наукова новизна одержаних результатів полягає в наступному:

- удосконалено інформаційну технологію надання рекомендацій, що відрізняється від існуючих якістю та швидкістю надання рекомендацій, яка поєднує в собі декілька існуючих підходів та базується на методах машинного навчання, що розширює функціональні можливості рекомендаційних сервісів та підвищує можливість заробітку для інтернет магазину.

Практичне значення одержаних результатів полягає у наступному:

1. Удосконалено стратегію надання рекомендацій в інтернет магазині.
2. Розроблено алгоритм функціонування інформаційної системи рекомендацій з використанням технологій машинного навчання.
3. Розроблено програмний засіб для рекомендацій користувачеві та вбудовано його в он-лайн магазин.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими, та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок магістранта. Всі результати, подані у магістерській кваліфікаційній роботі, отримані автором самостійно. У роботі, опублікованій у співавторстві, автору належать такі результати: [1] – розглянуто існуючі методи проектування інформаційної технології рекомендаційного сервісу.

Апробація результатів роботи. Результати досліджень було апробовано на всеукраїнській науково-практичній Інтернет-конференції студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2021)».

Публікації. За основними результатами досліджень опубліковано тези доповіді [1].

1 ОГЛЯД ТА АНАЛІЗ СУЧАСНОГО СТАНУ НАДАННЯ РЕКОМЕНДАЦІЙ В ІНТЕРНЕТ-МАГАЗИНІ

1.1 Поняття рекомендаційної системи

Рекомендаційні системи – представляють собою підклас систем фільтрації інформації, це програми та сервіси, що намагаються передбачити які об’єкти (фільми, музика, новини, товари, веб-сайти) будуть цікаві користувачеві, спираючись на деяку інформацію (профіль користувача чи опис об’єкту). Ці програми вдосконалили способи взаємодії між сервісом та відвідувачем, тому що замість того, щоб надавати статичну інформацію, користувач отримує інтерактивні можливості [4].

Рекомендаційні системи використовуються у найрізноманітніших областях і найчастіше вважаються генераторами списків відтворення для відео- і музичних сервісів, таких як Netflix, YouTube і Spotify, рекоменаторами продуктів для сервісів, таких як Amazon, або рекоменаторами контенту для платформ соціальних мереж, таких як Facebook і Twitter [5, 6]. Найчастіше такі системи використовуються у комерційних сервісах. Рекомендаційні системи є корисною альтернативою пошуковим алгоритмам, оскільки вони допомагають користувачам виявляти елементи, які в іншому випадку, вони могли б і не знайти. Для інтернет-магазинів це в принципі надзвичайно важлива функція та один з небагатьох способів якісно працювати. Рекомендації в даному випадку не є звичайною додатковою опцією платформи, вони забезпечують зручність навігації користувача по веб-ресурсу. Якщо електронний каталог містить більше 20 000 найменувань продукції, орієнтація вже представляється непомірно важкою, що ж говорити, якщо товарів мільйони.

Існує безліч конкретних прикладів використання рекомендаційних систем найрізноманітніших видів. Рекомендації на відомому онлайн-гіпермаркеті Amazon виглядають наступним чином (рисунок 1.1).

Your Recently Viewed Items and Featured Recommendations

Best Sellers



Рисунок 1.1 – Рекомендації товарів в онлайн-гіпермаркеті Amazon

Як показує практика, не тільки онлайн-магазини використовують подібний прийом. Соціальні мережі також не відстають. На рисунку 1.2 наведено приклад із соціальної мережі Instagram.

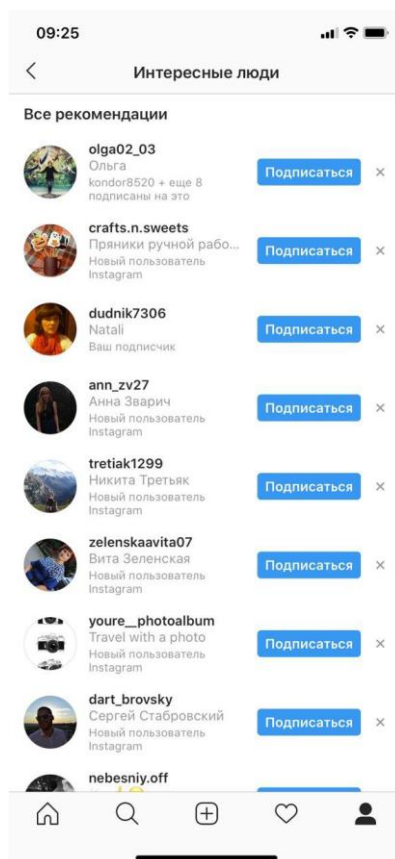


Рисунок 1.2 – Рекомендації у соціальній мережі Instagram

Також подібні прийоми легко можна побачити на різних соціальних платформах, порталах, присвячених літературі, подорожам, на новинних ресурсах, словом - майже всюди. Ця методика дійсно дуже популярна. Веб-ресурс Кинопоиск – ще один доступний приклад (рисунок 1.3).



Рисунок 1.3 – Рекомендації у веб-каталозі присвяченому фільмам Кинопоиск

Наразі існують три основні та найбільш використовувані стратегії по створенню рекомендаційних систем [1 – 3]:

- content-based filtering (фільтрація на основі вмісту / контентна фільтрація);
- collaborative filtering (спільна фільтрація);
- knowledge-based;
- гібридний підхід – використовує обидві основні стратегії у певній мірі.

Далі будуть детально розглянуті принципи роботи цих стратегій та проаналізована можливість вирішення поставленого завдання дослідження за їх допомогою.

1.2 Рекомендаційні системи на основі спільної фільтрації

Спільна фільтрація – це трьохступеневий процес, що починається зі збору користувацької інформації, потім будується матриця для розрахунку асоціацій і, нарешті, дається дуже вірогідна рекомендація. Її основне припущення полягає в

наступному: ті, хто однаково оцінював будь-які предмети в минулому, схильні давати схожі оцінки інших предметів і в майбутньому [7]. Наприклад, за допомогою колаборативної фільтрації музичний додаток здатен прогнозувати, яка музика сподобається користувачу, маючи неповний список його переваг (симпатій і антипатій). Прогнози складаються індивідуально для кожного користувача, хоча використовується інформація зібрана від багатьох учасників.

Спільна фільтрація поділяється на два основних підходи:

– memory based (заснований на сусідстві) – існує два основні різновиди цього методу: на основі користувачів, що займаються пошуком, і на основі елементів, що утворюють ту чи іншу категорію;

– model based (заснований на моделі) – даний підхід надає рекомендації, вимірюючи параметри статистичних моделей для оцінок користувачів, побудованих за допомогою таких методів як, метод байєсовських мереж, кластеризації, латентної семантичної моделі, такі як сингулярне розкладання, імовірнісний латентний семантичний аналіз, прихований розподіл Діріхле і марківський процес прийняття рішень на основі моделей. Моделі розробляються з використанням інтелектуального аналізу даних, алгоритмів машинного навчання, щоб знайти закономірності на основі навчальних даних. Цей підхід є більш комплексним і дає більш точні прогнози, так як допомагає розкрити латентні фактори, що пояснюють спостережувані оцінки. Даний підхід має ряд переваг. Він обробляє розріджені матриці краще, ніж підхід заснований на сусідстві, що в свою чергу допомагає з масштабільністю великих наборів даних [7].

Розглянемо детальніше підхід заснований на сусідстві. Існує два основні методи «підбору сусідів»: «користувач-користувач», «продукт-продукт».

Для створення нових рекомендацій для активного користувача метод «користувач-користувач» намагається знайти користувачів із найбільш схожим «профілем взаємодій» (таких користувачів називають найближчими сусідами) для пропонування продуктів, що є найбільш популярними серед цієї групи користувачів. Тобто ідея, на якій ґрунтується такий метод – шукати користувачів, чий смак схожий на переваги нашого цільового користувача [8]. Якщо раніше «користувач 1» і «користувач 2» поставили схожі оцінки для

декількох фільмів, то ми вважаємо, що смаки у них подібні, і за рейтингами тих чи інших фільмів, проставленим «користувачем 1», можемо вгадати невідомі рейтинги «користувача 2».

Наприклад, якщо Гомеру сподобалась така їжа як Піца, Пончик та Чіпси, а Пітеру сподобались Пончик і Піца, то ми безумовно повинні підказати Пітеру таку їжу як Чіпси (рисунок 1.4).

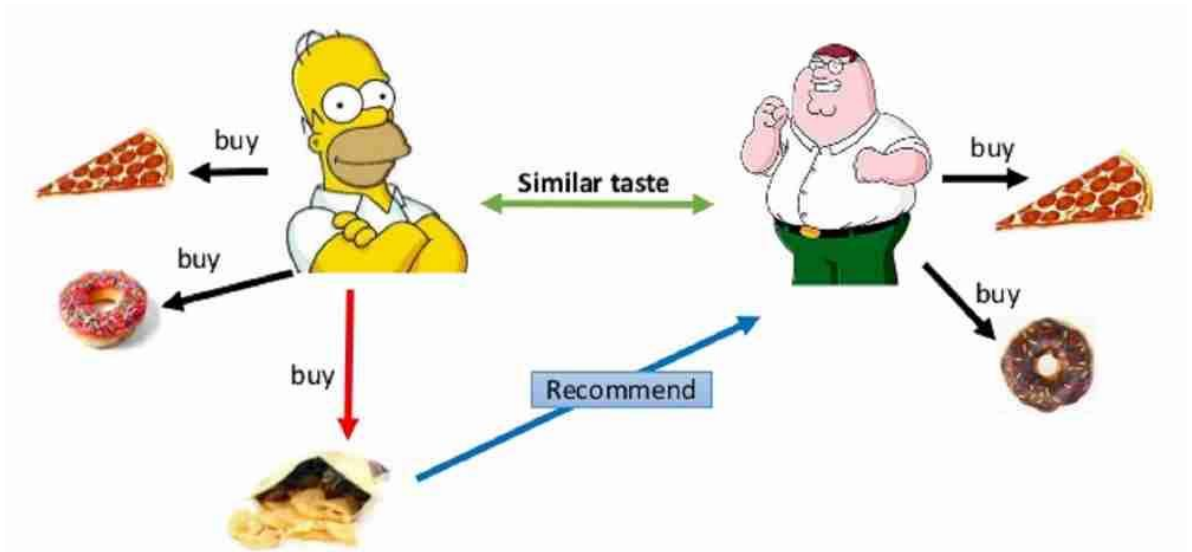


Рисунок 1.4 – Колаборативна фільтрація «користувач-користувач»

Уподобання користувачів записуються до матриці «користувачі-елементи» (рисунок 1.5), на основі якої і знаходять найближчих сусідів.

У такій матриці кожен рядок відповідає користувачу, а кожен стовпець – елементу. Тобто кожен рядок матриці становить вектор користувацького уподобання стосовно представлених елементів (при цьому якщо користувач, ще не оцінив той чи інший елемент, відповідний елемент матриці залишається пустим). Далі обчислюється міра схожості (за однією з обраних метрик) між активним користувачем та іншими користувачами, яка полягає у наступному: два користувача зі схожими оцінками на одному і тому ж наборі елементів мають визнаватися схожими. На основі розрахованих мір схожості формуються групи користувачів яким пропонуються елементи найбільш популярні серед них, обираючи лише ті, які активний користувач ще не оцінив.










		M items			
					
N users		5		1	
				3	
			5		
				2	
					5

Рисунок 1.5 – Матриця оцінок «користувач-елемент»

Метод «продукт-продукт» полягає у тому, що користувачу потрібно рекомендувати елементи, схожі на ті, які він уподобав раніше. Тобто схожість розраховується між елементами, а не користувачами. При цьому два елементи вважаються схожими, якщо більшість користувачів взаємодіяли з обома схожим чином [9]. Наприклад, Тіму і Емі сподобалися такі продукти як «Тістечко» і «Морозив». Можна зробити висновок, що більшості користувачів, які високо оцінили перший продукт, повинен сподобатися і другий. Таким чином, було б доречно запропонувати продукт «Морозиво» Джону, якому сподобався продукт «Тістечко» (рисунок 1.6).

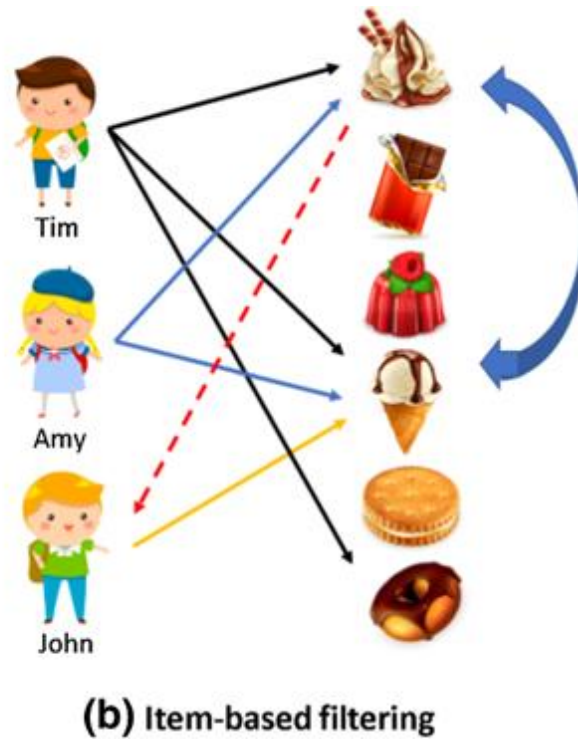


Рисунок 1.6 – Колаборативна фільтрація «елемент-елемент»

Цей підхід також використовує матрицю представлену на рисунку 1.5. При цьому розглядаються стовпці, які представляють собою вектор взаємодії елемента з кожним користувачем. Для надання рекомендації активному користувачу обираються елементи, що він найбільше уподобав, вони представляються як вектори взаємодій із кожним користувачем. Далі обчислюється міра схожості між цими елементами та усіма іншими та обираються найбільш схожі, які з'являться потім у рекомендаціях.

Метод «користувач-користувач» базується на пошуку схожих користувачів на основі взаємодій з продуктами. Якщо кожен користувач взаємодіє з невеликою кількістю продуктів, метод стає достатньо чутливим до будь-яких збережених взаємодій. З іншого боку, якщо фінальна рекомендація базується лише на збережених взаємодіях користувачів із схожими інтересами дозволяє отримати більш персональний результат (низька похибка).

Метод «продукт-продукт», навпаки, базується на пошуку схожих продуктів на основі взаємодій користувачів з ними. Загалом, якщо багато користувачів взаємодіяли з продуктом, пошук сусідів є значно менш чутливим до одиничних взаємодій.

Аналогічно, дії будь-яких користувачів (навіть тих, що достатньо відрізняються від обраного) розглядаються у рекомендації, роблячи цей метод менш персональним (більша похибка). Таким чином, цей підхід є менш персональним ніж підхід «користувач-користувач», але є більш надійним.

В ході аналізу підходів спільної фільтрації був виявлений ряд її ключових недоліків:

- розрідженість даних – більшість комерційних рекомендаційних систем заснована на великій кількості даних (товарів), в той час як більшість користувачів не ставить оцінки товарам. В результаті цього матриця «предмет-користувач» виходить дуже великою і розрідженою, що викликає особливі проблем при обчисленні рекомендацій;

- масштабованість – зі збільшенням кількості користувачів в системі з'являється проблема масштабованості. Наприклад, маючи 10 мільйонів покупців і мільйон предметів складність алгоритму колаборативної фільтрації є дуже високою, а сам алгоритм вже занадто складний для розрахунків;

- холодний старт – нові предмети або користувачі представляють велику проблему для рекомендаційних систем. Такі предмети, ще не мають оцінок користувачів, тому вони не будуть потрапляти до рекомендацій поки матриця «предмет-користувач» не буде наповнена певною кількістю значень. Однак проблему з наданням рекомендації для нового користувача полягає у тому ж: користувач ще не має матриці переваг, а тому його не можливо віднести до жодної з груп [10, 11].

Враховуючи суть, переваги та недоліки описаної стратегії, для вирішення поставленого завдання по створенню рекомендаційної системи для сервісів зі «складною», з точки зору надання рекомендацій, сферою застосування, вона не може бути застосована з наступних причин:

- активність користувачів на таких сервісах буде тимчасовою, а матриці «елемент-користувач» будуть дуже розрідженими, адже користувач буде оцінювати лише певну обмежену групу товарів (наприклад смартфони флагмани з певним типом екрану, процесору та інших технічних характеристик у певному ціновому діапазоні). Тому проблема розрідженості даних, що супроводжує

сумісну фільтрацію стає більш загальною і унеможлиблює використання цього методу;

– специфічність обраної предметної сфери не дозволяє створювати навіть відносно постійні профілі користувачів. Існує дуже мала імовірність того, що користувач буде декілька разів шукати дуже схожий за параметрами елемент, а спрогнозувати його наступну активність неможливо. Отже основне припущення, що допускає спільна фільтрація у рамках такої системи не дійсне для кожного конкретного користувача. Тобто групи користувачів, що формуються, будуть непостійними, і оцінки такої групи буде складно використовувати для прогнозування переваг нового користувача;

– проблема «холодного старту» не може існувати у рамках задачі, що розглядається, адже тут спостерігається постійна «текучість» активних елементів.

1.3 Рекомендаційні системи на основі вмісту

Методи фільтрації на основі вмісту засновані на описі елемента та профілі переваг користувача. Ці методи найкраще підходять для ситуацій, коли є відомі дані про об'єкт (ім'я, місце розташування, опис тощо.), але не про користувача. Контент-орієнтовані рекомендації розглядають рекомендації як специфічну для користувача проблему класифікації і створюють класифікатор для симпатій і антипатій користувача на основі характеристик продукту.

У цій системі ключові слова використовуються для опису елементів, а профіль користувача створюється для визначення типу елемента, який подобається цьому користувачеві. Іншими словами, ці алгоритми намагаються рекомендувати елементи, які схожі на ті, які користувач любив в минулому або вивчає в даний час. Зокрема, різні елементи-кандидати порівнюються з елементами, раніше оціненими користувачем, і рекомендуються найбільш підходящі елементи. Цей підхід має своє коріння в пошуку інформації та дослідженнях інформаційної фільтрації.

Ключова проблема з фільтрацією на основі контенту полягає в тому, чи здатна система дізнатися користувальницькі переваги з дій користувачів щодо одного джерела контенту і використовувати їх в інших типах контенту. Наприклад, рекомендація новинних статей на основі перегляду новин корисна, але буде набагато корисніше, коли на основі перегляду новин можна рекомендувати музику, відео, продукти тощо. Для подолання цієї проблеми, більшість заснованих на контенті рекомендаційних систем використовують гібридні системи, що містять у собі елементи обох підходів.

Даний підхід використовує опис або атрибути елементів, з якими взаємодівав користувач, для рекомендації схожих елементів. В такому випадку залежність створюється тільки на основі попереднього вибору користувача, що робить цей метод надійним, щоб уникнути проблеми холодного запуску [2].

Припустимо, деякий користувач дивиться фільм певного жанру, йому будуть рекомендуватися фільми з відповідним жанром. Назва, рік випуску, режисер, акторський склад, іншими словами контент, опис фільму, також допомагають ідентифікувати схожий за контентом фільм. Візуалізацію цього підходу можна побачити на рисунку 1.7.

При такому підході зміст продукту вже оцінюється на основі уподобань користувача (профіль користувача), а значення атрибутів елемента - це неявні функції, які будуть використовуватися для створення профілю елемента. Потім оцінка предмета прогнозується з використанням обох профілів, і можуть бути зроблені рекомендації [2].

Для виконання результируючої рекомендації необхідно створити профілі користувача та елементів. Для цього є дві групи вхідних даних:

- таблиця уподобань користувача (rating table): демонструє відношення користувач-елемент;
- профіль користувача (item profile): демонструє відношення елемент-атрибут, набір значень атрибутів елемента.

Тоді на основі цих даних створюється профіль користувача (user profile), який демонструє відношення користувач-атрибут, тобто маємо набір значень атрибутів елемента деякого типу, яким найбільше віддає перевагу користувач[3].

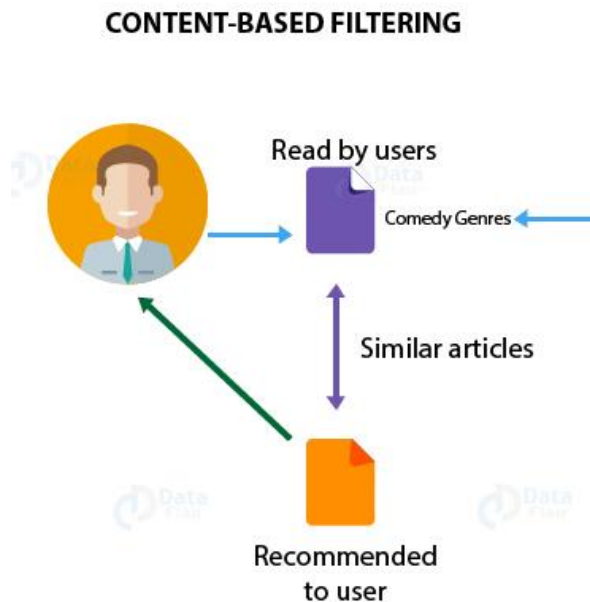


Рисунок 1.7 – Візуалізація підходу зі створенням профілю користувача і профілю елемента

Тепер, маючи профілі користувача та елементів, можна передбачити оцінку для елементів, відносно вподобань даного користувача.

Даний підхід має ряд переваг відносно спільної фільтрації:

- незалежність користувача: для спільної фільтрації потрібна оцінка інших користувачів, щоб знайти схожість між користувачами, а потім дати рекомендацію. Замість цього метод, заснований на контенті, повинен тільки проаналізувати елементи і профіль користувача для рекомендації;

- прозорість: метод спільної фільтрації дає рекомендацію, бо деякі невідомі користувачі мають такий же смак, як ви, але метод, заснований на контенті, рекомендує елементи тільки на основі їх опису;

- відсутність холодного старту: на відміну від спільної фільтрації, можна пропонувати нові елементи, перш ніж вони будуть оцінені значним числом користувачів.

Також цей підхід ряд суттєвих недоліків:

- обмежений аналіз контенту: якщо контент не містить достатньо інформації для точного розрізнення елементів, рекомендація буде не зовсім остаточною;

– надспеціалізація: метод заснований на контенті забезпечує граничний ступінь новизни, оскільки він повинен відповідати особливостям профілю та елементів. Цілком досконала фільтрація на основі вмісту може не запропонувати нічого нового;

– новий користувач: коли недостатньо інформації для створення надійного профілю користувача, рекомендацію неможливо надати правильно.

Незважаючи на переваги даного методу стосовно спільної фільтрації він також не придатний для вирішення поставленої задачі адже для його коректної роботи як і для методів спільної фільтрації необхідна можливість створення надійного користувацького профілю на основі рейтингів , що у рамках об'єкту дослідження є неможливим.

1.4 Рекомендаційні системи засновані на знаннях

Системи рекомендацій, засновані на знаннях (експертні рекомендаційні системи), являють собою особливий тип систем рекомендацій, які засновані на явних знаннях про асортимент товарів, користувацьких перевагах і критеріях рекомендацій (тобто, який товар слід рекомендувати, в якому контексті) [12].

Ці системи застосовуються в сценаріях, де основні підходи, такі як спільна фільтрація і фільтрація на основі контенту, не можуть бути застосовані. Системи рекомендацій, засновані на знаннях, добре підходять для складних областей, де предмети купуються не дуже часто, таких як квартири і автомобілі [13]. Іншими прикладами предметних областей, релевантних для систем рекомендацій, заснованих на знаннях, є фінансові послуги і туризм. Системи, засновані на рейтингах, не працюють добре в цих областях через малу кількість доступних рейтингів. Крім того, в доменах зі складними предметами, клієнти хочуть явно вказати свої переваги (наприклад, «максимальна ціна автомобіля - X»). У цьому контексті система рекомендацій повинна враховувати обмеження: наприклад, слід рекомендувати тільки ті фінансові послуги, які підтримують інвестиційний період, зазначений клієнтом. Жоден з цих аспектів не підтримується такими підходами, як спільна фільтрація і фільтрація на основі контенту. Ще однією

перевагою, заснованих на знаннях, рекомендаційних систем є відсутність проблем з холодним запуском.

Існує декілька основних ключових ідей які використовуються для створення таких рекомендаційних систем:

– «діалог» – системи рекомендацій, засновані на знаннях, часто є діалоговими, тобто користувацькі уподобання і переваги виявляються в рамках циклу зворотного зв'язку [14]. Основною причиною діалогового характеру цих систем рекомендацій є складність предметної області, в якій часто неможливо сформулювати всі вимоги користувача одночасно. Крім того, уподобання не відомі точно на початку, але можуть бути виявлені в рамках сеансу рекомендацій;

– «обмеження» – зворотний зв'язок з користувачем може бути реалізований у вигляді відповідей на питання, які обмежують набір відповідних елементів. [15]. Прикладом такого питання є «Якому типу системи об'єктивів ви віддаєте перевагу: фіксовані або змінні об'єктиви?». На технічному рівні такі сценарії рекомендацій можуть бути реалізовані на основі систем рекомендацій, заснованих на обмеженнях що встановлюються користувачем під час виконання пошукового запиту (пошуковий фільтр). В такому разі рекомендаційна система є надбудовою над обмеженнями, що встановив користувач, яка на основі встановлених асоціативних правил робить пошук відповідних варіантів. Необхідність у такій надбудові є коли простір елементів є складним і його звичайна фільтрація (пошуковий фільтр) призведе до втрати деяких варіантів;

– «критика» – користувацький зворотний зв'язок зазвичай надається в термінах «критичних зауважень», які вказують запити на зміну, що стосуються елемента, в даний час рекомендованого користувачеві. [16] Критика потім використовується для рекомендації наступного «кандидата». Приклад критики в контексті сценарію рекомендації цифрової камери: «Я хотів би мати таку камеру, але з більш низькою ціною». Це приклад «критичної оцінки», яка представляє запит на зміну одного атрибута елемента.

Найбільшим недоліком такого підходу є вузьке місце набуття знань, викликане необхідністю явного визначення рекомендацій [12]. Адже для

створення такої системи рекомендацій необхідна участь експертів предметної галузі, які побудують систему асоціативних правил, на яких будуть базуватися рекомендації. Такий спосіб знань є найбільш затратним з точки зору фінансів, людських ресурсів та часу. До того ж при такому підході кожна така система рекомендацій буде унікальною і її не можливо буде застосувати до іншої предметної області без повторної побудови нових асоціативних правил.

1.5 Висновок

Під час роботи над розділом проаналізовано предметну область, розглянуто: основні стратегії по створенню рекомендаційних систем які можливо використати при побудові рекомендаційної системи для інтернет магазину.

В результаті стратегій визначено їхні характеристики, основні критерії, переваги та недоліки, область використання, розглянуто основні проблеми, які виникають при розробці подібних систем та технології за допомогою яких, можна їх усунути.

2 РОЗРОБКА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДЛЯ СЕРВІСІВ ЗІ «СКЛАДНИМИ» ПРЕДМЕТНИМИ СФЕРАМИ

2.1 Стратегія надання рекомендацій

Аналіз основних стратегій по створенню рекомендаційних систем показав, що вони не можуть бути застосовані напряду для вирішення поставленої задачі через необхідність використання та зберігання історії рейтингів. А рекомендаційні системи засновані на знаннях, хоча і призначені для роботи з такими предметними сферами, не є універсальними через необхідність явного збору знань для кожної предметної області, та явній побудові асоціативних правил взаємодії з залученням експертів предметної області.

Можна зробити висновок, що існує необхідність у розробці стратегій та способів прогнозування уподобань користувачів, які могли б працювати зі складними предметними сферами, не потребуючи при цьому явного збору знань для кожної такої сфери окремо. Тобто така стратегія має відповідати наступним двом пунктам:

- незалежність від історії збережених рейтингів;
- незалежність логіки побудови прогнозів від предметної сфери.

Деякі аспекти розглянутих методів можна використати та об'єднати між собою, для створення системи рекомендацій що здатна працювати зі складними предметними областями та мінімізує вузьке місце у способі отримання знань.

По перше, можна частково застосувати метод спільної фільтрації заснований на контенті, але в цьому разі буде аналізуватися лише опис товарів без побудови профілю користувача [17]. Стратегія полягає у наступному: користувачу пропонуються об'єкти, які максимально схожі на вподобаний ним об'єкт, при цьому «вподобаним» вважається об'єкт, що переглядається користувачем у даний момент. Тобто в основу такої рекомендації лягає так званий «метод ідеальної точки», ідеальною точкою при цьому і є об'єкт, що вподобав користувач (рисунок 2.1). Такий підхід дозволить знаходити елементи,

які не зовсім відповідають пошуковому запиту (фільтру) користувача, що є першочерговою задачею.

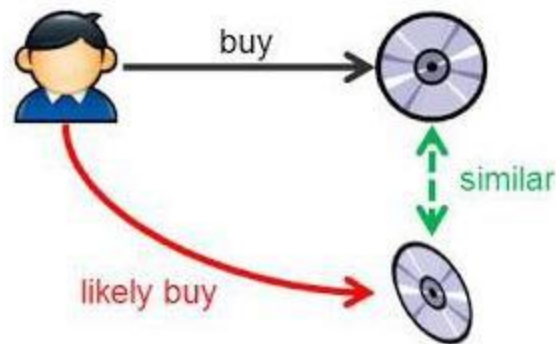


Рисунок 2.1 – Підхід з аналізом лише контенту

Для знаходження «однаковості» елементів необхідно використовувати різноманітні метрики подібності для порівняння значень атрибутів елементів. При цьому для кожного атрибута необхідно знати дві речі: його відносну важливість, та вигляд функції яка описує подібність двох значень цього атрибуту.

По друге, можна використати ідею «діалогу», яка стосується експертних рекомендаційних систем, для організації зворотного зв'язку з користувачем. Тобто уподобання користувача будуть формуватися в рамках поточного сеансу перегляду елементів, а рекомендації будуть формуватися шляхом аналізу опису тих елементів, що становлять його уподобання [18].

Тоді створення рекомендацій буде становити двоступеневий процес. На першому етапі користувачу пропонуються товари, що найбільш схожі на те, яке він наразі переглядає (активний товар). На другому етапі, коли користувач вже переглянув певну кількість товарів, рекомендації базуються на групі з деякої кількості останніх переглянутих товарів.

Група останніх переглянутих товарів формується методом «ковзаючого вікна». Це алгоритм трансформації, що дозволяє сформувати з членів часового ряду набір даних, який може служити навчальною множиною для побудови моделі прогнозування [19]. Під «вікном» у даному випадку розуміється часовий інтервал, що містить набір значень, які використовуються для формування

навчального прикладу. У процесі роботи алгоритму вікно послідовно зміщується на одиницю спостереження, при цьому «забуваючи» попередні спостереження (рисунок 2.2). Такий підхід чудово реалізує ідею «діалогу», враховуючі поточні уподобання користувача та враховуючи їх зміну шляхом «забування» старих елементів.

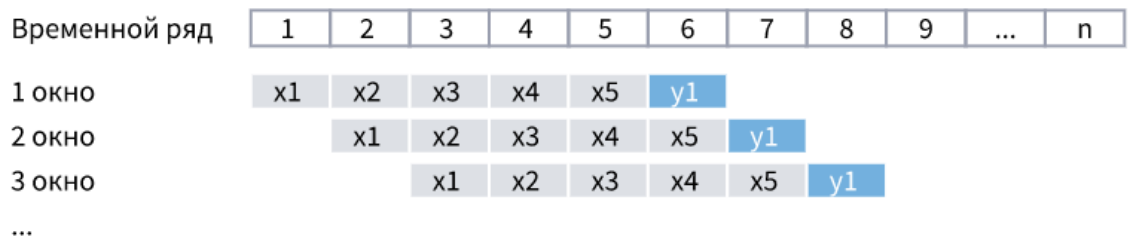


Рисунок 2.2 – Метод «ковзаючого вікна»

По-третє, можна застосувати ідею «обмежень» шляхом зменшення простору елементів, що пропонуються у якості рекомендацій, відповідно до основних обмежень у пошуковому запиті. Наприклад, не має сенсу пропонувати користувачеві товари із категорії Телевізори, якщо при пошуку він обрав категорію Ноутбуки, навіть при ідеальній відповідності іншим обмеженням. При цьому повного обмеження відповідно до усіх вимог користувача відбуватися не повинно, щоб не створити проблему з утратою деяких елементів.

Описана стратегія має наступні ключові переваги, відносно вирішуваної проблеми:

- відсутність необхідності у створенні профілю користувача, адже система опирається лише на опис елементів;
- вирішується проблема холодного старту, адже елементи не залежать від розміру відношення «користувач-елемент»;
- уподобання користувача виявляються у реальному часі;
- присутня значна доля універсальності, адже немає необхідності для створення асоціативних правил пошуку для кожної предметної області.

Проте така стратегія буде мати і свої недоліки: оскільки подібна система рекомендацій у повній мірі опирається на опис елементів, для правильного

розрізнення елементів, необхідно створити дуже детальний та змістовний опис кожного елемента, а також обов'язковою умовою є необхідність максимального заповнення інформацією профілю об'єкту.

Також такий підхід не буде універсальним у повній мірі адже експерту предметної області все ж необхідно надати такій системі рекомендацій відносно важливості атрибутів що описують елементи, та спосіб їх порівняння.

Проте відносно важливості атрибутів можна встановити і без участі експерта виходячи із принципів функціонування предметної області, або виходячи із обмежень що встановив користувач під час виконання пошукового запиту. А вигляди функцій близькості можна встановлювати вже на етапі впровадження рекомендаційної системи, виходячи із набору стандартних метрик, що існують наразі для характеристик, що представлені у тих чи інших шкалах.

Також суттєвим обмеженням є те, що така система передбачає, що рекомендації мають відбуватися на множині однакових об'єктів, що характеризуються певним набором однакових характеристик. Тобто рекомендація буде працювати лише в межах обраної рубрики, рекомендуватися буде лише той самий тип елементів.

2.2 Математична модель надання рекомендацій для користувачів інтернет-магазину

Запропонована стратегія передбачає, що кожен елемент характеризується вектором значень деяких визначених характеристик.

Для кількісної оцінки схожості деякого елемента з «еталонним» використовується метод зваженої суми критеріїв [20]. Для цього кожен елемент необхідно представити вектором критеріїв:

$$K=\{k_1,k_2,k_3,k_4\dots k_n\}, \quad (2.1)$$

де k_i – кількісна оцінка схожості i -тої характеристики елемента з i -тою характеристикою «еталонного» елемента.

Відносна важливість кожної з характеристик виражена вектором вагових коефіцієнтів:

$$W = \{w_1, w_2, w_3, w_4 \dots w_n\}, \quad (2.2)$$

де w_i – відносна важливість i -тої характеристики елемента:

$$\sum w_i = 1.$$

Результуюча оцінка схожості елемента розраховується наступним чином :

$$Score = \sum w_i \times k_i \quad (2.3)$$

Для оцінки на основі групи елементів, оцінки (2.3) нового елемента, що розраховуються відносно кожного елемента у межах групи переглянутих, підсумовуються між собою для отримання кінцевого результату:

$$Score^* = \sum_{i=1}^N Score_i, \quad (2.4)$$

де N – кількість переглянутих елементів.

В цьому випадку подібний розрахунок можна розглядати як голосування переглянутих сторінок за непереглянуті, та сторінка яка має найбільші значення голосів (2.3) відносно елементів у групі і отримає найвищий бал.

Оцінка близькості елементів відбувається окремо за кожною характеристикою з подальшим формуванням вектору (2.1).

Для характеристик, що представлені у інтервальній оцінці схожості характеристики елемента аналогічній характеристиці еталонного елемента буде розраховуватися за допомогою функцій затухання (або функцій розподілу).

Функції затухання оцінюють певну величину за допомогою функції, що затухає в залежності від збільшення відстані між значенням цієї величини та певним заданим джерелом (центром мас). Така оцінка схожа на запит, щодо входження деякого значення в заданий діапазон, з подальшим призначенням кількісної оцінки цьому значенню в залежності від близькості до центра діапазону.

Використовують три основні залежності (форми) розпаду (рисунок 2.3):

- гауссіан (розпад за функцією Гауса);
- експоненціальний розпад;
- лінійний розпад.

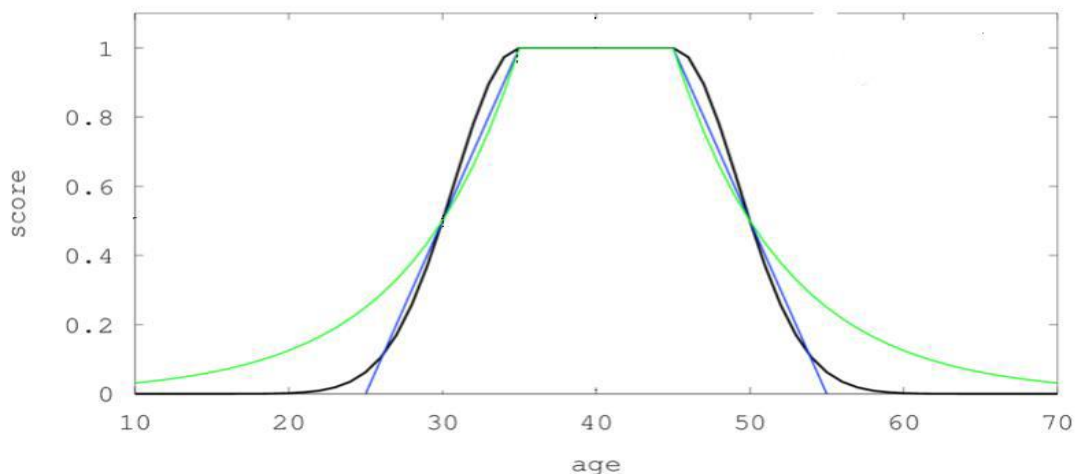


Рисунок 2.3 – графіки залежності оцінки (score) деякої величини в залежності від збільшення відстані до центра мас: зелений – експоненціальна залежність; синій – лінійна залежність; чорний – гауссіан

Для того щоб використати таку функцію для оцінки характеристик необхідно для кожної з них визначити значення наступних параметрів (рисунок 2.4):

- origin (центр) – значення еталонного оголошення;

- offset (зміщення) – відстань, на якій значення характеристики буде вважатися рівним еталонному значенню;
- scale (масштаб) – відстань від центру (+ зміщення) яка визначає діапазон прийнятної (допустимої) схожості;
- decay (спад) – значення схожості в яке оцінюється значення характеристики на відстані масштабу від обраного центру.

Параметри Scale та Decay разом встановлюють швидкість спаду залежності як видно з рисунку 2.4.

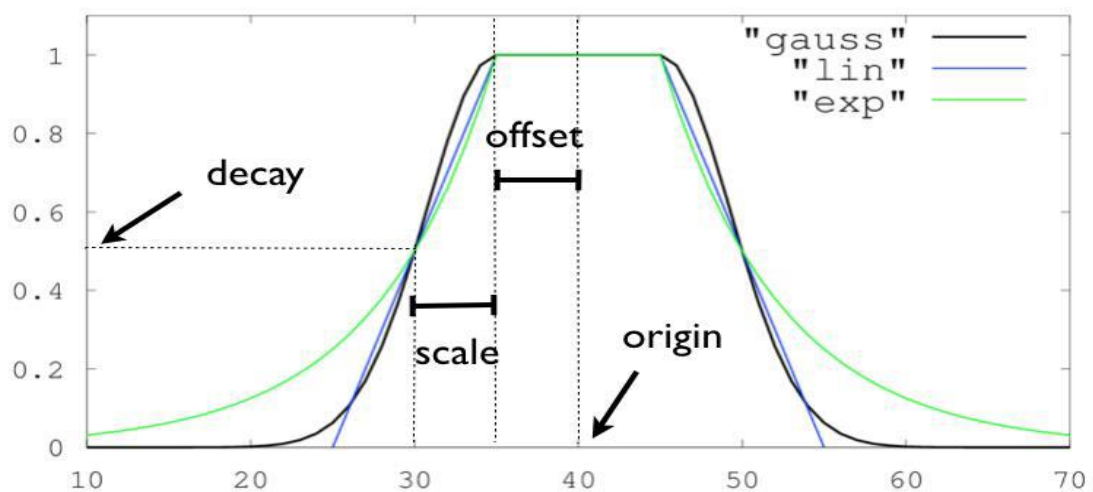


Рисунок 2.4 – Наочна інтерпретація параметрів затухання

В рекомендаційній системі, що розробляється в атестаційній роботі, використовується гаусівська залежність, оскільки лінійна залежність дуже примітивна і встановлює пропорціональне зменшення оцінки в процесі віддалення від центру, а експоненціальна залежність має дуже крутий графік залежності поряд із центром з тенденцією до більшої пологості на його кінцях, що означатиме дуже стрімке зменшення значень оцінок в межах обраного масштабу, та більш полого – за його межами, а в даному разі необхідна зворотна залежність, яку і надає гаусіан.

Тоді затухання для певної характеристики деякого елемента можна визначити наступним чином:

$$K_i = \exp \left(- \frac{(|value_i - origin| - offset)^2}{2 \times \left(- \frac{scale^2}{2 \times \ln(decay)} \right)} \right), \quad (2.5)$$

де K_i – кількісна оцінка схожості i -тої характеристики елемента з i -тою характеристикою «еталонного» елемента, приймає значення в діапазоні $[0;1]$;

$value_i$ – значення i -тої характеристики елемента;

$origin$ – значення i -тої характеристики еталонного елемента;

$offset$ – відстань, на якій значення характеристики буде вважатися рівним еталонній;

$scale$ – відстань від центру (+ зміщення) яка визначає діапазон прийнятної (допустимої) схожості;

$decay$ – значення схожості в яке оцінюється значення характеристики на відстані $scale$ від $origin$.

Для характеристик, що представлені у номінальних шкалах можливо встановити лише подібність або відмінність об'єктів. Тому подібність значень таких характеристик буде підраховуватися наступним чином:

– $K_i=1$, якщо значення відповідних характеристик однакові;

– $K_i=0$, якщо значення відповідних характеристик різні.

Тобто в такому разі йдеться про повне співпадіння значення, що буде оцінюватися максимальним балом, або про відсутність співпадіння, яке оцінюється нулем.

Якщо значення деякої характеристики відсутнє, то оцінка схожості цієї характеристики елемента аналогічній характеристиці еталонного елемента буде розраховуватися наступним чином: $K_i = 0$.

Тобто в такому разі буде спостерігатися зменшення значення функції схожості.

2.3 Класифікаційний метод машинного навчання «k-найближчих сусідів»

Метод найближчих сусідів - алгоритм класифікації, заснований на визначенні подібності між об'єктами. У цьому алгоритмі класифікується об'єкт отримує той клас, об'єктів якого більше серед найближчих до нього в навчальній вибірці.

У методі найближчих сусідів (k-nearest neighbors, K-NN) об'єкту привласнюється той клас, членів якого більше серед найближчих до нього об'єктів.

Позначимо за X – об'єкти, а Y – класи об'єктів. $X^1 = (x_i, y_i)_{i=1}^l$ - навчальна вибірка, що складається р пар об'єкт і його клас, с – число класів.

Для визначення міри близькості об'єктів задається функція відстані:

$$p: X \times X \rightarrow \{1, c\}.$$

Об'єкти є схожими, якщо значення цієї функції маленьке. В данній роботі задачі в якості опції відстані використовується Евклидова відстань:

$$p(x^1, x^2) = \sqrt{\sum_{i=1}^n (x_i^1 - x_i^2)^2},$$

де $x^1 = (x_1^1, \dots, x_n^1)$, $x^2 = (x_1^2, \dots, x_n^2)$ – об'єкти вибірки.

Для довільного об'єкта v його клас $k(v)$ визначається як:

$$k(v) = \arg \max_{y \in Y} \sum_{i=1}^l [y(x_i) = y] w(i, v),$$

Де $y(x_i)$ позначає клас об'єкта в навчальній вибірці, l число найближчих об'єктів в навчальній вибірці, $w(i, v)$ - вага i -го сусіда при класифікації об'єкта v . В даному випадку $w(i, v) = 1, i = 1, l$.

В алгоритмі найближчих сусідів гостро стоїть питання у визначенні цього параметра адже від його вибору залежать результати класифікації.

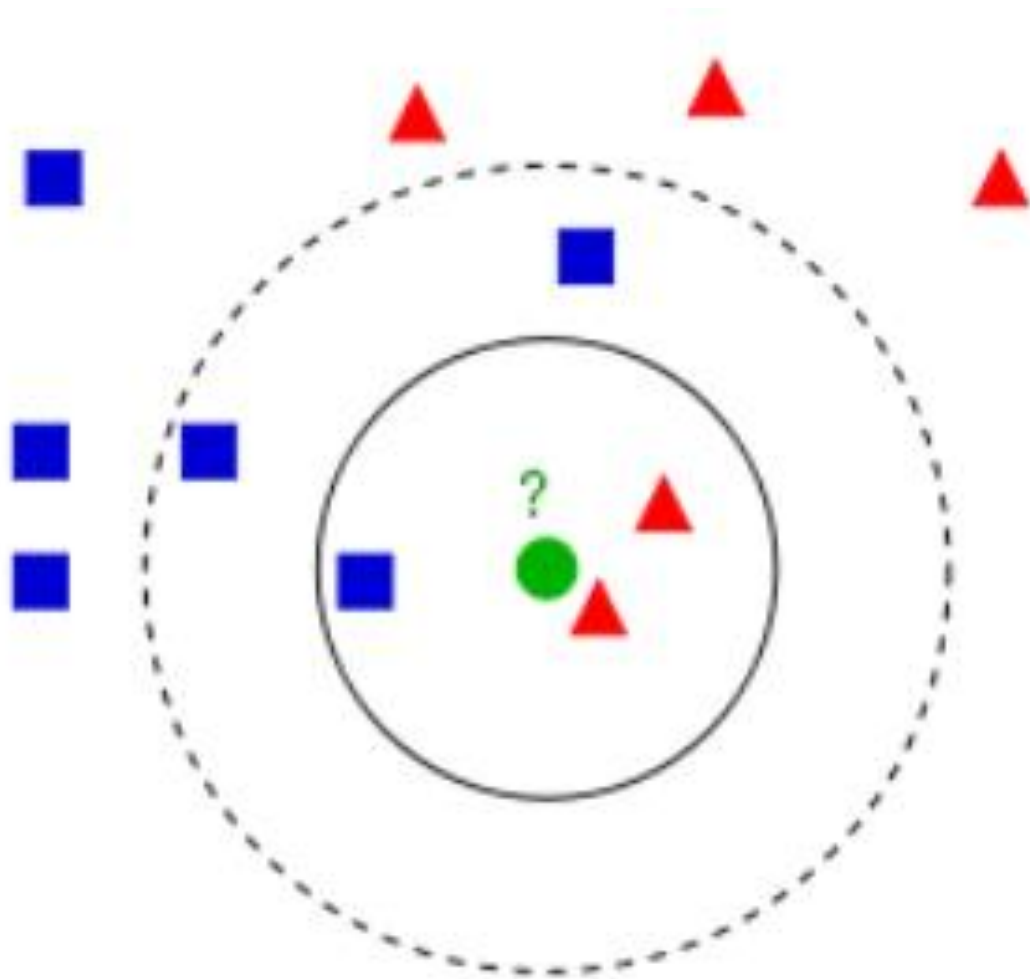


Рисунок 2.5 – Метод k-NN

З Рисунок 2.5 видно, що в залежності від вибору k зелене коло може бути класифіковане як червоний трикутник у випадку $k = 3$, або як синій квадрат, якщо $k = 5$.

Якщо $k = 1$, то алгоритм видає неправильні результати при наявності об'єктів-викидів: алгоритм помиляється як на самих об'єктах-викидах, так і на найближчих до них сусідів. У разі, коли $k = l$, всім об'єктам класифікації присвоюється один і той же клас, що теж неправильно. На практиці параметр k підбирається ковзаючим контролем.

2.4 Висновок

В данному розділі розроблено стратегій та способи прогнозування уподобань користувачів, які могли б працювати зі складними предметними сферами. Для машинного навчання системи надання рекомендацій використано метод найближчих сусідів. Було запропоновано поєднати два базових підходи до створення рекомендаційних систем що вирішує ряд важливих проблеми. Розроблена математична модель.

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РЕКОМЕНДАЦІЙНОГО СЕРВІСУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

3.1 Розробка системних вимог для рекомендаційного сервісу

Основне призначення будь-якої рекомендаційної системи – прогнозування уподобань та надання рекомендацій її кінцевим користувачам. Такі системи можуть використовуватися або бути частиною будь-яких сервісів, що спеціалізуються на наданні користувачам доступу до деякого контенту.

У якості прикладу складної предметної сфери була обрана сфера торгівлі через Інтернет. Тому створена стратегія рекомендацій буде застосовуватися для веб-сервісу з пошуку товарів. Рекомендаційна система буде однією з обчислювальних частин цієї інформаційної системи, буде повністю в неї інтегрована та буде спеціалізуватися на пропонуванні користувачу позиції, які схожі на ті, що він переглядає у поточний момент.

Сервіс призначений для створення і управління позицій товару в інтернет магазині. Виходячи з цілей системи, до неї висувуються наступні системні вимоги.

Система має складатися з клієнтської частини, яка відповідає за отримання даних з сервера для подальшої візуалізації в зручному вигляді для користувача і відправки даних на сервер, а також сервера, який має приймати запити клієнта, обробляти їх, зберігати інформацію про користувацьку активність та прогнозувати клієнтські уподобання, шляхом виконання алгоритмів стратегії рекомендацій. Так як такого типу системи отримують велику кількість запитів, то вона повинна бути стійкою до відмов. Потрібно розробити систему так, щоб у майбутньому можливо було її масштабувати роблячи як найменш змін. Система не має бути прив'язана до однієї платформи, повинен бути розроблений програмний інтерфейс додатку.

3.2 Розробка функціональних вимог для рекомендаційного сервісу

Для визначення функціональних вимог до системи була розроблена її функціональна модель з використанням стандарту IDEF0 [21, 22]. У рамках цієї методології був створений набір діаграм, що описують систему, визначають її межі та основні процеси. Вони представлені на рисунках 3.1 – 3.7.



Рисунок 3.1 – Концептуальна діаграма інформаційної технології

На рисунку 3.1 представлена концептуальна діаграма системи. На вхід подаються «Дані користувача», головна функція – «Інтернет магазин з продажу товарів». На виході ми отримуємо результат пошуку товару, результат роботи рекомендаційної системи, результат створення товару чи повідомлення про помилку під час його створення. Механізмами в даному випадку є «Користувач» і «Система». У ролі керуючих стрілок виступають «Регламент системи», а також «GDPR».

Для подальшої розробки системи необхідно провести декомпозицію першого рівня, щоб отримати більш детальну інформацію про вимоги до

системи. На рисунку 3.2 представлена декомпозиція функції «Система публікації оголошень про товар» системи. Дану функцію ми декомпозуємо на наступні функції:

- реєстрація;
- створення товару;
- повідомлення про помилку;
- публікація;
- пошук товарів.

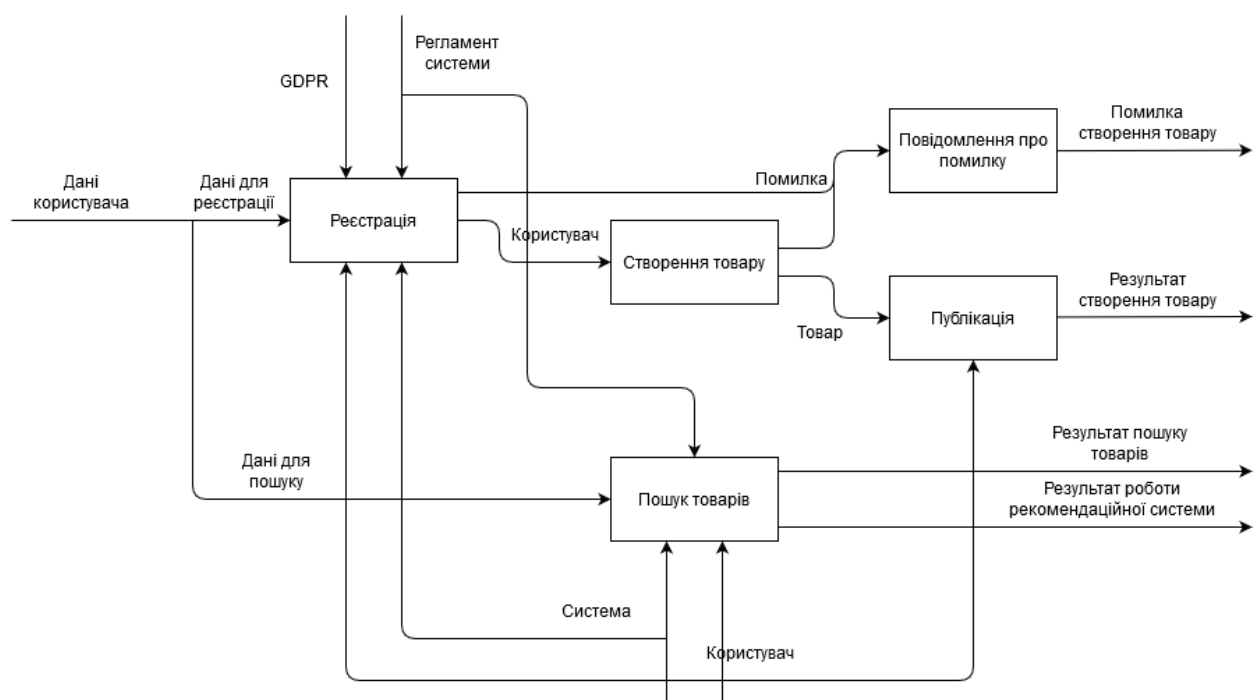


Рисунок 3.2 – Діаграма декомпозиції, що представляє функції системи

Для виконання функції «Реєстрація» необхідні такі дані, як логін, пароль, поштова адреса. Далі, можливе виконання функції «Створення оголошення», на вхід якої надходять дані зазначені в змісті оголошення, а також інформація про аккаунт користувача. Якщо створення товарів пройшло успішно, то виконується функція «Публікація», інакше відбувається перехід до функції «Повідомлення про помилку». Для виконання функції «Пошук оголошень» необхідні критерії для пошуку, при цьому виконувати функцію реєстрації не обов'язково.

На рисунку 3.3 представлена декомпозиція функції «Реєстрація» системи. Ця функція декомпозується на 3 функції:

- заповнення форми реєстрації;
- перевірка реєстрації;
- створення користувача.

Під час виконання функції «Заповнення форми реєстрації» користувач вказує дані для свого акаунту. Після виконується функція «Перевірка реєстрації», де проходять перевірку дані, введені користувачем. У разі помилки на вихід подається її змістовний опис, інакше система створить нового користувача у функції «Створення користувача».

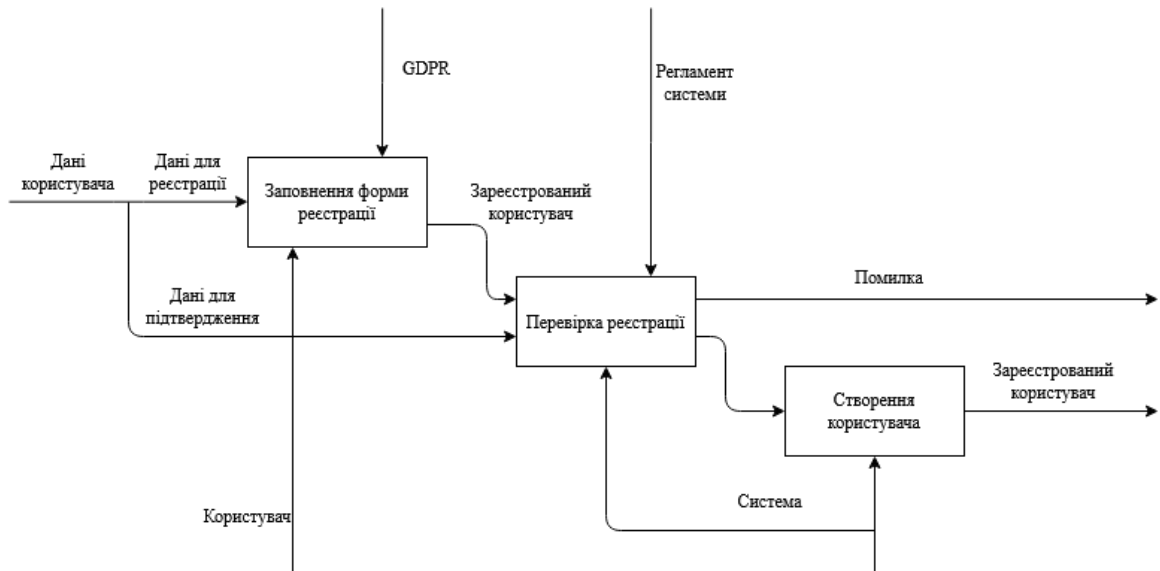


Рисунок 3.3 – Діаграма декомпозиції функції «Реєстрація» системи

На рисунку 3.4 представлена декомпозиція функції «Створення товару» системи. Ця функція декомпозується на 2 функції:

- заповнення форми;
- перевірка даних;
- збереження товару.

На вхід першої функції надходять дані про акаунт і опис товару. Далі відбувається перевірка наданих даних з регламентом системи та на коректність

введених даних. Якщо перевірка пройшла успішно на виході отримуємо підготовлені дані.

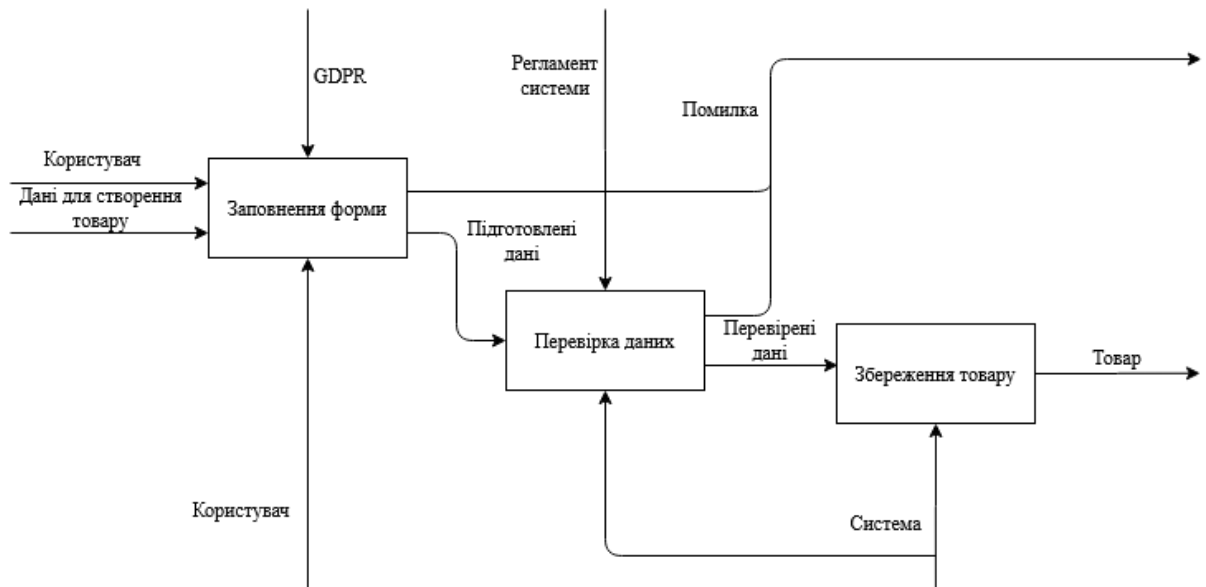


Рисунок 3.4 – Діаграма декомпозиції функції «Створення товару» системи

На рисунку 3.5 представлена діаграма декомпозиції функції «Пошук товарів» системи. Функція складається з наступних функцій:

- встановлення необхідних обмежень;
- використання рекомендаційної системи;

Щоб встановити необхідні обмеження, необхідно ввести дані для пошуку товарів у пошукові фільтри. На виході будуть результати пошуку за обраними обмеженнями. Для роботи механізму рекомендаційної системи необхідно на вхід подати інформацію про характеристики товару, що обраний у результаті пошуку

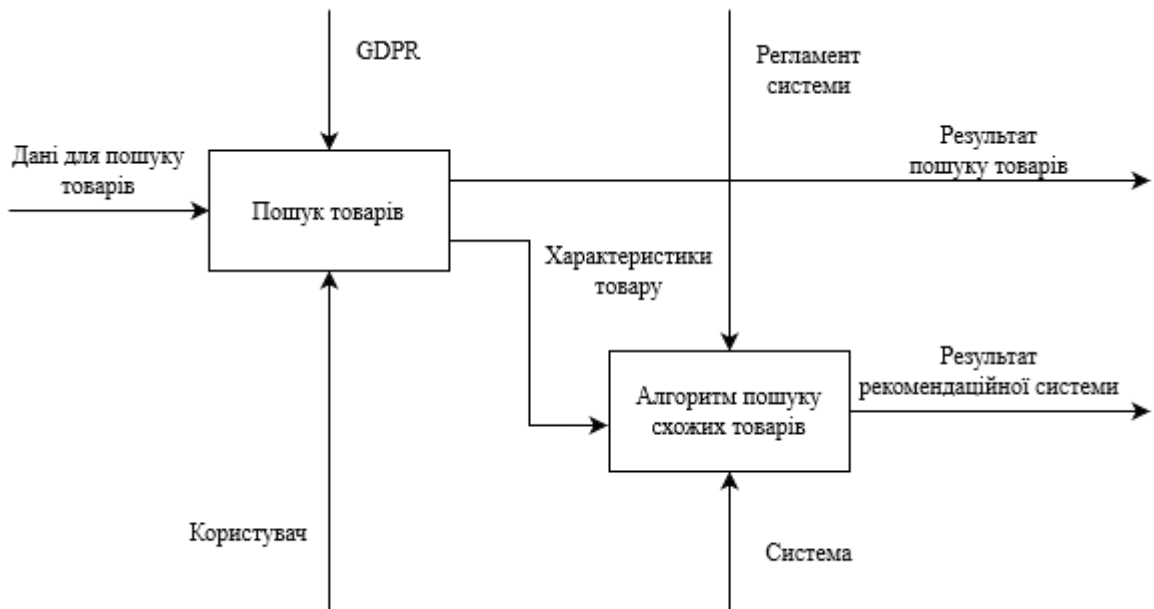


Рисунок 3.5 – Діаграма декомпозиції функції «Пошук товарів» системи

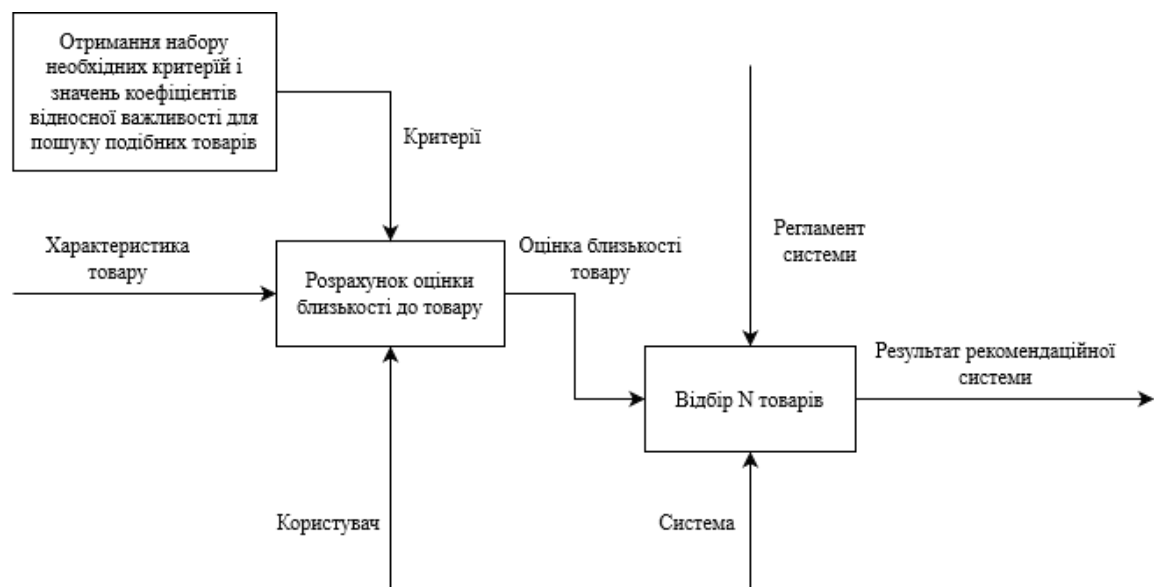


Рисунок 3.6 – Діаграма декомпозиції функції «Застосування рекомендаційної системи»

На рисунку 3.6 представлена діаграма декомпозиції функції «Застосування рекомендаційної системи». Для її виконання спочатку необхідно отримати набір критеріїв, що описують заданий тип товарів та значення їх відносної важливості, характеристику еталонного товару, та характеристики всіх товарів для яких буде розраховуватися значення близькості.

Таким чином, проведене функціональне моделювання та аналіз отриманих діаграм дозволили уточнити функціональні вимоги до розроблюваної системи:

- реєстрація нових користувачів;
- перевірка введених даних;
- створення нових товарів за типом категорії;
- редагування товарів;
- написання відгуків до товарів;
- редагування профілю користувача;
- рейтинг у профілю;
- відправка повідомлення на пошту користувачеві;
- фільтрація товарів за різними критеріями;
- надання рекомендацій;
- відображення помилок.

3.3 Розробка архітектурних вимог для рекомендаційного сервісу

Відповідно до встановлених системних і функціональних вимог для розроблюваної системи була обрана клієнт-серверна архітектура, при якій взаємодія модулів відбувається по протоколу HTTP шляхом передачі запитів від клієнта до сервера. У якості архітектурного шаблону сервісу була обрана «MVC» модель.

Шаблон проектування MVC передбачає поділ даних програми, призначеного для користувача інтерфейсу і керуючої логіки на три окремих компоненти: модель, представлення, контролер – таким чином, що модифікація кожного компонента може здійснюватися незалежно [23].

Під моделлю, зазвичай розуміється частина, яка містить в собі функціональну бізнес-логіку програми. Модель повинна бути повністю незалежна від інших частин продукту. Модельний шар нічого не повинен знати про елементи дизайну, і яким чином він буде відображатися.

В обов'язки представлення входить відображення даних отриманих від моделі. Однак, представлення не може безпосередньо впливати на модель. Можна сказати, що воно отримує доступ «тільки на читання» до даних.

Представлення має наступні ознаки:

- у представленні реалізується відображення даних, які виходять від моделі будь-яким способом;
- у деяких випадках, представлення може мати код, який реалізує деяку бізнес-логіку.

Прикладами представлення можуть бути: HTML-сторінка, WPF форма, Windows Form.

Контролер інтерпретує дії користувача, сповіщаючи модель про необхідність змін. Контролер забезпечує «зв'язок» між користувачем та системою, він направляє дані від користувача до системи и навпаки, використовує модель и представлення для реалізації необхідної дії.

3.4 Використання метода машинного навчання «к-найближчих сусідів»

Спільні системи фільтрації використовують дії користувачів, щоб рекомендувати інші продукти. Загалом, вони можуть бути як на основі користувачів, так і на основі елементів. Елементний підхід, як правило, кращий за підхід, орієнтований на користувача. Підхід, орієнтований на користувача, часто важче масштабувати через динамічну природу користувачів, тоді як елементи, як правило, не сильно змінюються, і підхід, заснований на елементах, часто можна обчислити в автономному режимі та обслуговувати без постійного повторного навчання.

Для реалізації спільної фільтрації на основі елементів, KNN є ідеальною моделлю для переходу, а також дуже доброю базою для розробки рекомендаційної системи. KNN - це непараметричний метод лінивого навчання. Він використовує базу даних, у якій точки даних розділені на кілька кластерів, щоб зробити висновок щодо нових зразків.

KNN не робить жодних припущень щодо базового розподілу даних, але він покладається на схожість елементів елементів. Коли KNN робить висновок про товар в магазині, KNN обчислює "відстань" між цільовим товаром (рисунок 3.7) та кожним іншим товаром у своїй базі даних, потім він ранжирує свої відстані та повертає найкращі K-товари найближчих сусідів як найбільш подібні рекомендації щодо товарів.

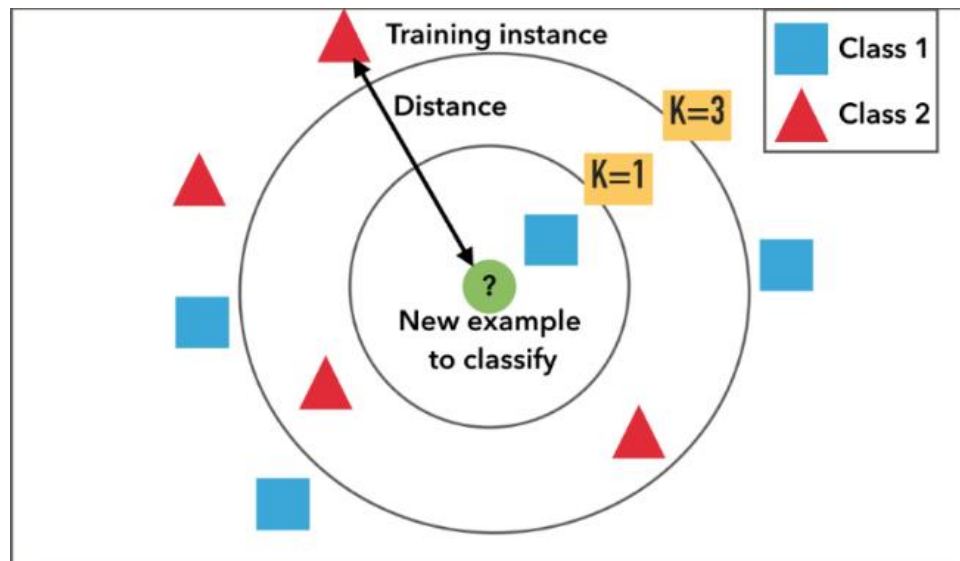


Рисунок 3.7 – Приклад класифікації об'єкту методом KNN

По-перше, нам потрібно перетворити фрейм даних оцінок товару у відповідний формат, який може використовуватися моделлю KNN. Необхідно щоб дані були в масиві m на n , де m - кількість товарів, а n - кількість користувачів. Щоб змінити фрейм даних рейтингів товарів, ми переведемо фрейм даних у широкий формат із товарами як рядками та користувачами як стовпцями. Тоді заповнимо відсутні спостереження 0, оскільки будемо виконувати операції лінійної алгебри (обчислення відстані між векторами). Назвемо цей новий набір даних "набір даних про особливості товару".

Наш набір даних параметрів товару - надзвичайно розріджена матриця форми $13\ 500 \times 113\ 291$. Ми точно не хочемо передавати всі дані з переважно 0 в типі даних float32 до KNN. Для більш ефективного обчислення та зменшення розміру пам'яті нам потрібно перетворити значення фрейму даних на розріджену матрицю (рисунок 3.8).

```

from scipy.sparse import csr_matrix# pivot ratings into shop_items features
df_shop_items_features = df_ratings.pivot(
    index='shop_itemId',
    columns='userId',
    values='rating'
).fillna(0)# convert dataframe of shop items features to scipy sparse matrix
mat_shop_features = csr_matrix(df_shop_items_features.values)

```

```

In [31]: df_shop_features.head(5)
Out[31]:

```

	userId	4	5	10	14	15	18	19	26	31	34	...	283199	283204	283206	283208	283210	283215	283219	283222	283224	283228
productId	1	4.0	0.0	5.0	4.5	4.0	0.0	0.0	0.0	5.0	0.0	...	5.0	0.0	0.0	4.5	0.0	4.0	4.0	0.0	0.0	4.5
2	4.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	4.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0

Рисунок 3.8 – Розріджена матриця

Зараз наші навчальні дані мають дуже високу розмірність. Ефективність KNN постраждає від надмірної розмірності, якщо вона використовує "евклідову відстань" у своїй цільовій функції. Евклідова відстань не є корисною у великих розмірах, оскільки всі вектори майже рівновіддалені від вектора пошукового запиту (особливості цільового товару). Натомість ми будемо використовувати подібність косинусів для пошуку найближчого сусіда.

3.5 Висновок

В данному розділі були поставлені ряд вимог такі як системні, функціональні та архітектурні що повинні бути виконані для функціонування інформаційної технології. Було розглянуто особливості реалізації методу машинного навчання «к-найближчих сусідів» у рекомендаційних системах.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РЕКОМЕНДАЦІЙНОГО СЕРВІСУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

4.1 Вибір СУБД для рекомендаційного сервісу

Для розробки сервісу у якості СУБД була обрана MySQL.

MySQL – це одна з найпоширеніших СУБД. Гнучкість MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Ця база даних пропонує досить багато інструментів для розробки додатків. MySQL вважається гарним рішенням для малих і середніх застосувань [24].

Можливості сервера MySQL:

- простота у встановленні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн;
- швидкість обробки даних – спрощення деяких стандартів дозволяє MySQL значно збільшити продуктивність у порівнянні з PostgreSQL, який при простих операціях читання може значно уповільнити сервер;
- наявність простої і ефективної системи безпеки;
- масштабованість – MySQL легко працює з великими обсягами даних та легко масштабується.

4.2 Розробка БД для рекомендаційного сервісу

Так як MySQL має реляційний тип баз даних, то потрібно спроектувати схему бази даних, яка повністю буде покривати наші вимоги. Для реляційних баз даних необхідно виконати нормалізацію. Кінцевою метою нормалізації є зменшення потенційної суперечливості інформації, збереженої в базі даних.

Форми нормалізацій, які ми будемо застосовувати:

– перша нормальна форма – відношення відповідає першій формі тоді, коли на перетині кожного стовпця і кожного рядка знаходяться тільки елементарні значення атрибутів і не містяться групи, що повторюються;

– друга нормальна форма – відношення знаходиться у другій формі тоді, коли воно знаходиться у першій нормальній формі та кожен описовий атрибут функціонально повно залежить від первинного ключа;

– третя нормальна форма – відношення знаходиться у третій формі тоді, коли воно виконує обмеження другої нормальної форми та усі описові атрибути відношення взаємно незалежні і повністю залежать від первинного ключа, тобто кожний описовий атрибут не тразитивно залежить від ключа [24].

Розроблювана БД буде приведена до 3-ї нормальної форми, що забезпечить такі властивості даних:

– наявність основного ключа: мінімальний набір колонок, які ідентифікують запис;

– уникнення повторень даних, що можуть зустрічатись різну кількість разів в різних записах;

– атомарність даних – кожен атрибут повинен мати лише одне значення, а не множину значень.

Для збереження даних про об'яви, даних користувача та іншу необхідну інформацію для використання системи – визначені сутності бази даних, які описані в таблиці 4.1

Пояснення зв'язків між сутностями в побудованій моделі даних наведені в таблиці 4.2.

Таблиця 4.1 – Сутності бази даних та їх опис

Ім'я сутності	Опис
Products	Інформація про товар
Product_Photos	Фотографії до товару
Product_Categories	Категорії товарів

Продовження таблиці 4.1

User	Користувачі
User_Comments	Коментарі користувачів
Payment_Systems	Системи оплати
Cities	Міста
Areas	Райони
Streets	Вулиці
Recommendation_Scores	Результати розрахунків близькості для товарів

Таблиця 4.2 – Типи зв'язків в побудованій схемі даних

№	Тип сутності	Тип зв'язку	Тип сутності
	Products	1:Б	Product_Photos
	Products	1:1	Product_Categories
	Products	1:Б	User_Comments
	Users	1:Б	User_Comments
	Areas	1:Б	Users
	Cities	1:Б	Users
	Streets	1:Б	Users
	Cities	1:Б	Areas
	Areas	1:Б	Streets
	Recommendation_Scores	1:1	Product

4.3 Вибір мови програмування

Для розробки серверної частини була обрана мова PHP. Це мова загального призначення з відкритим вихідним кодом. PHP спеціально сконструйований для веб-розробників і його код може впроваджуватися безпосередньо в HTML. На

сьогоднішній день PHP є найбільш поширеною мовою веб-програмування. Переважна більшість сайтів і веб-сервісів в інтернеті написані за допомогою PHP.

До переваг PHP відносять наступні:

- для всіх найбільш поширених операційних систем (Windows, MacOS, Linux) є свої версії пакетів розробки на PHP, а це значить, що є можливість створювати веб-сайти на будь-якій з цих операційних систем;
- PHP може працювати у зв'язці з різними веб-серверами: Apache, Nginx, IIS;
- простота і легкість освоєння;
- PHP схожий на мову C, тому, знаючи C або одну з мов з сі-подібним синтаксисом, буде просто опанувати PHP;
- PHP підтримує роботу з великою кількістю систем баз даних (MySQL, MSSQL, Oracle, Postgre, MongoDB та інші);
- поширеність хостингових послуг і їх дешевизна. Як правило, хостингові компанії розміщують веб-сайти на PHP на веб-серверах Apache або Nginx, які працюють на одній з операційних систем сімейства Linux. І веб-сервери, і операційні системи на базі Linux безкоштовні, що знижує загальну вартість використання хостингу;
- постійний розвиток, PHP продовжує розвиватися, виходять нові версії, які адаптують мову програмування до нових реалій [25].

Серед фреймворків вибір припав на Symfony, який має велике ком'юніті та відмінну документацію. Symfony – вільний PHP фреймворк для швидкої розробки веб-додатків і виконання рутинних завдань веб-програмістів. Розробка і підтримка фреймворку спонсорується французькою компанією Sensio. Symfony складається з набору не пов'язаних між собою компонентів, які можна використовувати повторно в проектах.

Переваги:

- потужна екосистема навколо фреймворка, з хорошим співтовариством і безліччю розробників;
- хороша і постійно оновлювана документація для всіх версій фреймворка;

- безліч різних компонентів для повторного використання;
- пропонує механізм функціональних і модульних тестів для знаходження помилок у веб-додатку;
- підходить для складних і навантажених веб-проектів , електронної комерції.

Недоліки: незважаючи на хорошу документацію, фреймворк є складним для вивчення.

Для розробки клієнтської частини була обрана мова програмування JavaScript. Це мультипарадигмова мова програмування. Підтримує об'єктно-орієнтований, імперативний і функціональний стилі. JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить в браузерах як мова сценаріїв для додання інтерактивності веб-сторінок. Основні архітектурні риси: динамічна типізація, слабка типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу [26]. Переваги:

- потужна екосистема навколо фреймворка, з хорошим співтовариством і безліччю розробників;
- хороша і постійно оновлювана документація для всіх версій фреймворка;
- безліч різних компонентів для повторного використання;
- пропонує механізм функціональних і модульних тестів для знаходження помилок у веб-додатку;
- підходить для складних і навантажених веб-проектів , електронної комерції.

Недоліки: незважаючи на хорошу документацію, фреймворк є складним для вивчення.

У якості фреймворка для клієнтської частини був обраний React. Це JavaScript-бібліотека з відкритим вихідним кодом для розробки призначених для користувача інтерфейсів. React може використовуватися для розробки односторінкових і мобільних додатків. Його мета – надати високу швидкість, простоту і масштабованість розроблюваним додаткам.

Клієнтська частина використовує API, яке реалізоване на backend, для отримання та відправки даних. Зв'язок відбувається завдяки HTTP-запитам. Використовуються наступні види запитів:

- PUT – для оновлення;
- GET – для отримання;
- DELETE – для видалення;
- POST – для створення;
- PATCH — для часткового оновлення;
- OPTIONS - для опису параметрів з'єднання з цільовим ресурсом.

Аутентифікація відбувається завдяки JSON Web Token (JWT). JWT – це відкритий стандарт (RFC 7519) для створення токенів доступу, заснований на форматі JSON. Як правило, використовується для передачі даних для аутентифікації в клієнт-серверних додатках. Токени створюються сервером, підписуються секретним ключем і передаються клієнту, який в подальшому використовує даний токен для підтвердження своєї особи. Завдяки такому способу спілкування із сервером є можливість використовувати різноманітні платформи для додатку:

- мобільну;
- браузерну;
- десктопну.

Також відсутня прив'язка до backend частини, що дає можливість переписувати додаток, використовуючи інший фреймворк або взагалі іншу мову програмування.

У якості веб сервера виступає Nginx. NGINX – програмне забезпечення, написане для UNIX-систем. Основне призначення – самостійний HTTP-сервер, або, як його використовують частіше, фронтенд для високонавантажених проєктів. Можливе використання NGINX як поштового SMTP / IMAP / POP3-сервера, а також зворотного TCP проксі-сервера.

4.4 Розробка алгоритмів функціонування інформаційної технології рекомендаційного сервісу для інтернет-магазину з використанням машинного навчання»

Нижче наведено опис та схеми роботи деяких основних алгоритмів сервісу.

На рисунку 4.1 зображено повний цикл для створення оголошення. Для того, щоб додати товару, користувач повинен мати свій акаунт. Для цього він повинен авторизуватись, або якщо він не зареєстрований, то йому необхідно створити новий акаунт. Після авторизації він обирає категорію товару та заповнює усі необхідні поля, та тисне на кнопку «Додати». Якщо все зроблено вірно, то буде доданий новий товар.

На рисунку 4.2 зображено схему алгоритму пошуку товарів.

Щоб знайти необхідний товар, користувач має обрати необхідну категорію товарів та завдати критерії пошуку. Серед наведених товарів обрати необхідний.

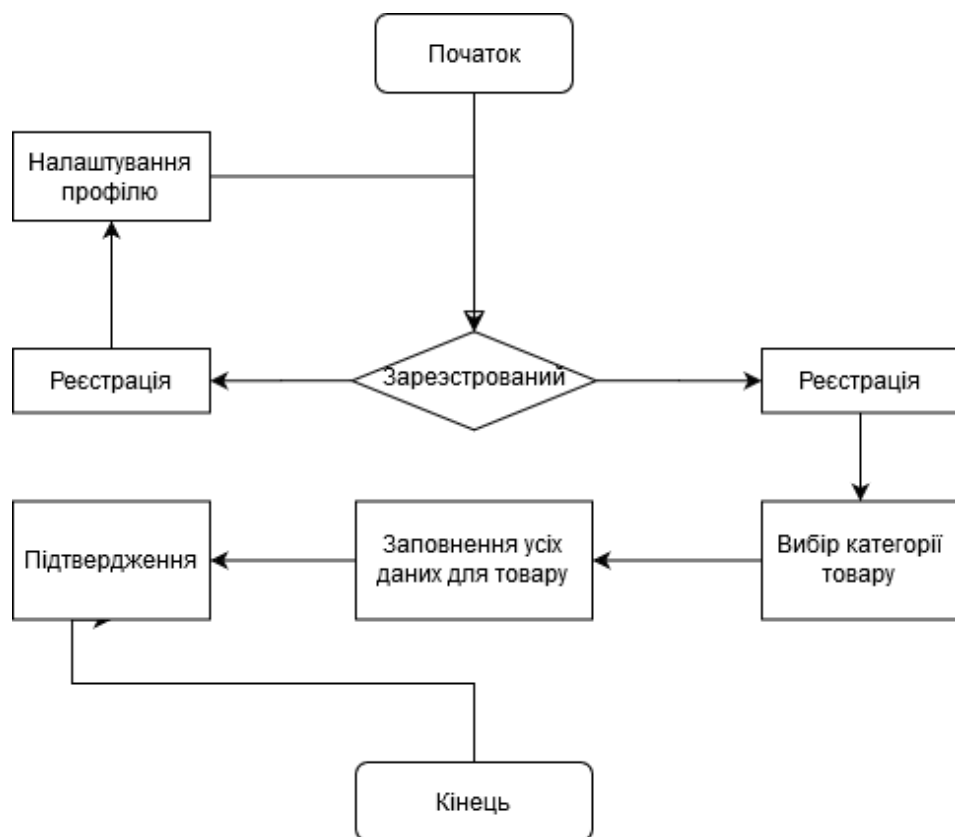


Рисунок 4.1 – Схема алгоритму створення товару



Рисунок 4.2 – Схема алгоритму для пошуку товарів

На рисунку 4.3 зображено спрощену схему роботи рекомендаційної системи (РС).

Для відображення результатів роботи РС користувач має обрати один з товарів, що пропонує система після виконання пошуку. На початку алгоритм перевіряє кількість товарів, що вже були переглянуті користувачем, за поточну сесію. Результати цієї перевірки можуть призвести до двох наступних варіантів:

- кількість товарів виявляється менше заданої кількості n ;
- кількість товарів виявляється більше заданої кількості n .

При виконанні першого варіанту умови, обраний товар обирається в якості «ідеальної» точки, потім за формулами (2.3) та (2.5) відбувається розрахунок оцінок близькості інших товарів до обраного центру та обирається k переможців.

При виконанні другого варіанту умови, останні n переглянутих товарів (що обираються за методом «ковзаючого вікна») становлять «ідеальну точку», потім за формулами (2.3) та (2.5) відбувається розрахунок оцінок близькості інших товарів до кожного з товарів, які становлять центр, а далі за формулою (2.4)

розраховується результуюча оцінка близькості, стосовно обраного центру та обирається k переможців.

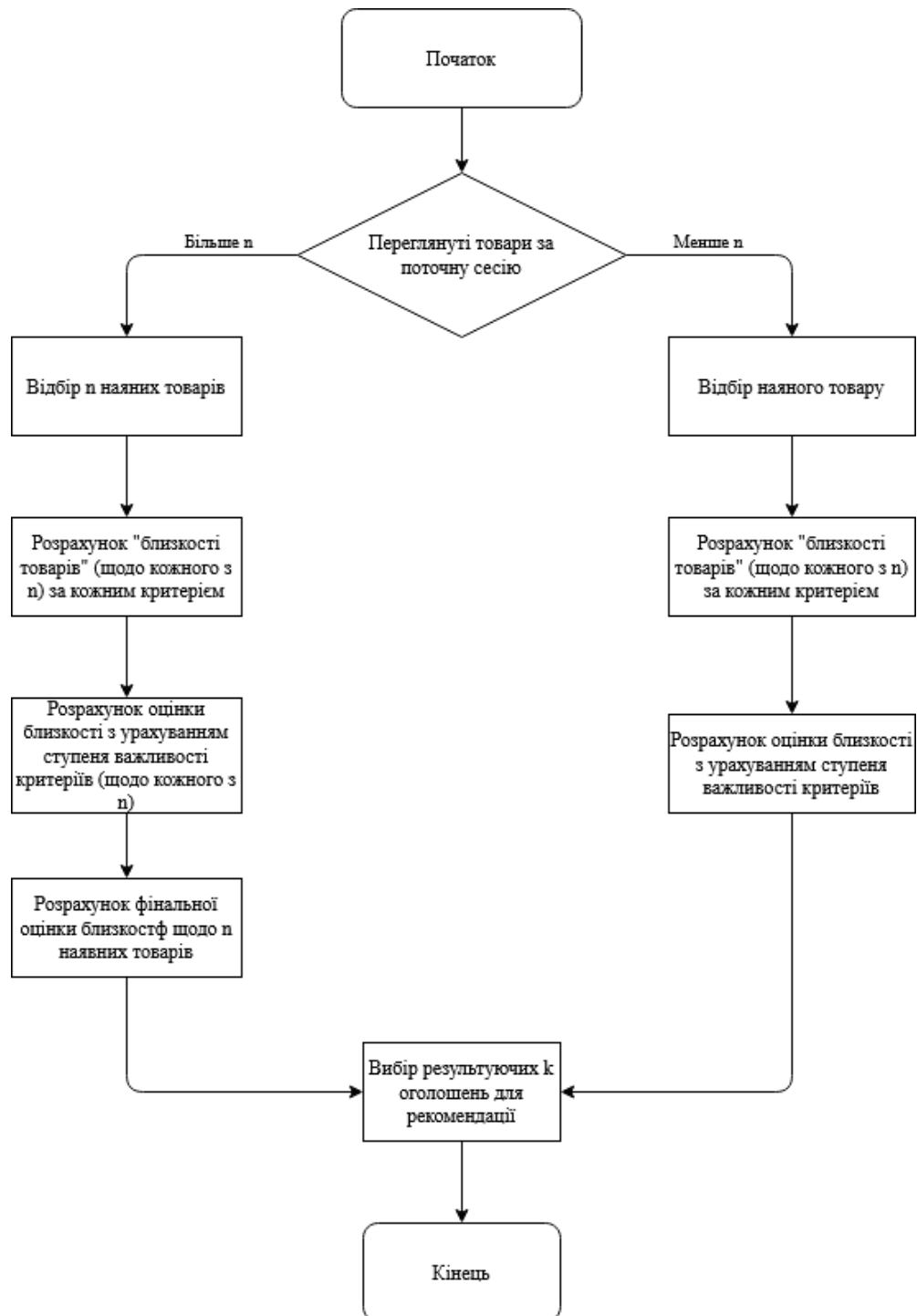


Рисунок 4.3 – Схема алгоритму роботи рекомендаційної системи

4.5 Тестування та аналіз результатів роботи рекомендаційного сервісу для інтернет-магазину з використанням машинного навчання»

Експериментальна частина даної роботи полягає у підборі таких параметрів створеної стратегії рекомендацій, які давали б найкращі результати по прогнозуванню уподобань користувача. Результати рекомендацій при цьому порівнюються евристично, тобто шляхом зіставлення різних варіантів кінцевих переможців та визначення того, що найкраще відображає користувацькі потреби.

Підбір параметрів відбувається шляхом аналізу законів функціонування предметної області, переваг середньостатистичного користувача (особиста думка такого користувача представляється при цьому експертом предметної області), які він надає тим чи іншим критеріям під час пошуку.

Значення n , що дорівнює кількості елементів, які становлять «ковзаюче вікно» дорівнює трьом, проте воно може відрізнятись в залежності від предметної сфери рекомендованих продуктів.

У межах РС для товарів використовуються наступні критерії та їх відносна важливість:

- категорія – 0.3;
- опис – 0.2;
- ціна – 0.15;
- країна походження – 0.025;
- знижка – 0.03;
- платіжна система -0.02;
- виробник – 0.05;
- кредит 0.025;
- рейтинг коментарів – 0.075;
- колір – 0.05;
- рейтинг - 0.125.

Значення параметрів для розрахунку близькості за формулою (2.5) для кожного критерію, що представлений у кількісній шкалі описані у таблиці 4.3.

Таблиця 4.3 – Значення параметрів для розрахунку оцінки близькості

Назва критерію	Назви параметрів		
	Offset	Scale	Decay
Ціна	4%	15%	0.5
Категорія	0%	6%	0.5
Опис	5%	20%	0.5
Рік	1	3	0.5
Рейтинг	5%	20%	0.5
Знижка	10%	50%	0.5
Рейтинг коментарів	5%	20%	0.5

На рисунках 4.4 – 4.7 можна побачити результати роботи рекомендаційної системи. Рисунки 4.4 – 4.5 відображають прогнози щодо одного товару, тому можна спостерігати невелику різницю у значеннях основних критеріїв, наприклад ціни, виробника та серії. Тобто система рекомендує близькі, «шаблонні» товари, що добре відповідають відкритому оголошенню.

На рисунках 4.6 – 4.7 можна спостерігати значно помітнішу різницю, між прогнозованими варіантами. Це результат застосування принципу «діалогу». Система поступово «забуває» попередню активність користувача, та запам'ятовує поточну. Так, на цих рисунках бажання користувача змінилися, тому до рекомендації потрапили нові товари, що значно менше схожі на його попередню активність, проте також залишився товар, що більше схожий на попередні рекомендації, це означає, що попередня активність, ще не повністю «стерта», або ж користувача цікавлять декілька варіантів, які він шукає паралельно тому різні товари і потрапляють до рекомендацій.

Смартфони і телефони / Смартфони / Apple iPhone 11 Pro 512GB



Apple iPhone 11 Pro 512GB

ID товару: 09754

35 499 грн

Колір Золотий



Купити

Основні Характеристики:

Виробник: **Apple**

Серія: **11**

Рік: **2019**

Діагональ екрану: **5,8"**

Роздільна здатність екрану: **2436x1125**

Роздільна здатність камери: **12Mp**

Кількість ядер: **6 ядер**

Рисунок 4.4 – Приклад товару



Рисунок 4.5 – Рекомендації до товару

Смартфони і телефони / Смартфони / Samsung Galaxy S20 Ultra 12/128Gb (G988)



Samsung Galaxy S20 Ultra 12/128Gb (G988)

ID товару: т30953

33 999 грн

Колір Чорний



Купити

Основні Характеристики:

Виробник: **Samsung**

Серія: **20**

Рік: **2020**

Діагональ екрану: **6,9"**

Роздільна здатність екрану: **3200x1440**

Роздільна здатність камери: **108Мп**

Кількість ядер: **8 ядер**

Рисунок 4.6 – Приклад товару



Рисунок 4.7 – Рекомендації до товару

Для тесту проведемо порівняння роботи сервісів аналогів з роботою рекомендаційного сервісу власної розробки.

Kameleoon - система вивчає поведінку користувача на сайті, його дії, контекст, джерела, з яких він прийшов, зіставляє інформацію з даними з CRM і сегментує відвідувачів по 40 критеріям. Грунтуючись на отриманій інформації система адаптує зміст сторінок сайту відповідно до очікувань і переваг відвідувачів. Kameleoon також може в реальному часі видавати тригерні повідомлення (індивідуальні пропозиції, нагадування і т.д.), розсилати персоналізовані листи. Алгоритми інтелектуального орієнтування працюють з

окремими товарами або товарними категоріями, вибираючи з аудиторії сайту тих відвідувачів, які придбають це товари з найбільшою ймовірністю.

Retail Rocket - платформа аналізує поведінку користувача на сайті і формує персональні товарні рекомендації, які можуть розміщуватися на будь-якій сторінці сайту - від головної до чекаута, включаючи картки відсутніх товарів і сторінку 404. Для підключення до сервісу досить розмістити на сайті спеціальний трекінг-код і вивантажити товари в форматі YML. Крім персоналізації будь-яких розділів сайту, платформа RetailRocket пропонує тригерні розсилки. У кожному з них підставляються саме ті товари, які переглядав користувач, а також персональні рекомендації до них. Всі ключові метрики по розсилках доступні в особистому кабінеті.

Таблиця 4.4 – Результати порівняння роботи готових рішень на ринку відносно власної розробки

	Конверсія	Середній чек	Дохід на відвідувача
Retail Rocket	-4.3%	-2.27%	-5.25%
Kameleoon	-2.7%	-0.86%	-3.56%
Середнє значення	-3.5%	-1.565	-4.405%

Таблиця 4.5 – Результати порівняння роботи готових рішень на ринку відносно власної розробки

	Схожість рекомендацій	Час надання рекомендації
Retail Rocket	0.86	1100ms
Kameleoon	0.9	1050ms
Власна розробка	0.93	870ms

За результатом тестування можна вважати приріст зроблених замовлень на 3.5 %, середній чек на 1.5% а дохід збільшений у перерахунку на одного

користувача на 4.4%. Ці значення були отримані як середнє арифметичне з роботи сервісів аналогів.

Також, ще важливими показниками являються «Схожість рекомендації» та «Час надання рекомендації». За результатами тестування, схожість рекомендацій власної розробки на 5.15% більше ніж в конкурентів. Також було виявлено, що швидкість генерації рекомендацій на 7.65% більше, що зумовлено роботою рекомендаційної систем безпосередньо на одному сервері з сайтом в той час як конкуренти працюють через API.

4.6 Висновок

У четвертому розділі було проаналізовано та обрано мову і середовище програмування, базу даних та її тип, спроектовано програмні засоби надання рекомендацій для сервісів з продажу товарів. Створено відповідний веб додаток для тестування рекомендаційної системи. Розроблено структуру бази даних та алгоритми роботи інформаційної. Виконано тестування рекомендаційної систем системи за допомогою конкретного прикладу.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів. Такими експертами будуть Перевозніков С.І. та Яровий А.А.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями за 5-ти бальною шкалою.

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 5.1.

Таблиця 5.1 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта	
	1. Перевозніков С.І.	2. Яровий А.А.
	Бали, виставлені експертами:	
1	4	4
2	3	3
3	4	4
4	4	3
5	3	4
6	3	4
7	3	3
8	4	4
9	4	3
10	3	3
11	3	4
12	4	4
Сума балів	СБ ₁ = 43	СБ ₂ = 43
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{2} = 43$	

Отже, з отриманих даних таблиці 5.1 видно, що нова розробка має високий рівень комерційного потенціалу.

5.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи

Для розробки нового програмного продукту необхідні такі витрати.

Основна заробітна плата для розробників визначається за формулою (5.1):

$$Z_o = \frac{M}{T_p} \cdot t, \quad (5.1)$$

де M - місячний посадовий оклад конкретного розробника;

T_p - кількість робочих днів у місяці, $T_p = 22$ дні;

t - число днів роботи розробника, $t = 60$ днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 5.2.

Таблиця 5.2 – Розрахунки основної заробітної плати

Працівник	Оклад M , грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	5000	227.27	10	2272.73
Інженер-програміст	7500	340.9	60	20454.55
Всього:				22727.28

Розрахуємо додаткову заробітну плату:

$$Z_{\text{дод}} = 0,1 \cdot 22727,28 = 2272,72 \text{ (грн.)}$$

Нарахування на заробітну плату операторів НЗП розраховується як 37,5-40% від суми їхньої основної та додаткової заробітної плати:

$$H_{\text{зп}} = (Z_o + Z_p) \cdot \frac{\beta}{100}, \quad (5.2)$$

$$H_{\text{зп}} = (22727,28 + 2272,72) \cdot \frac{38,5}{100} = 5500,22 \text{ (грн.)}$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$A = \frac{Ц \cdot N_a}{100} \cdot \frac{T}{12}, \quad (5.3)$$

де Ц – балансова вартість обладнання, грн;

N_a – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

T – Термін використання (T=3 міс.).

Таблиця 5.3 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
Персональний комп'ютер	18000	25	2	750
Всього:				750

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_1^n N_i \cdot Ц_i \cdot K_i, \quad (5.4)$$

де n – кількість комплектуючих;

N_i - кількість комплектуючих i-го виду;

$Ц_i$ – покупна ціна комплектуючих i-го виду, грн;

K_i – коефіцієнт транспортних витрат (прийmemo $K_i = 1,1$).

Таблиця 5.4 - Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Безпроводна мишка	шт.	450	1	450
Додатковий монітор	шт.	5000	1	5000
Мережевий фільтр	шт.	150	1	150
Блокнот	шт.	90	1	90
Ручка	шт.	10	1	10
Всього з урахуванням транспортних витрат				5700

Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi} ; \quad (5.5)$$

де V – вартість 1кВт-години електроенергії ($V=1.6$ грн/кВт);

Π_k – установлена потужність комп'ютера ($\Pi=0,1$ кВт);

Π_m – установлена потужність монітора ($\Pi=0,05$ кВт);

Φ – фактична кількість годин роботи комп'ютера ($\Phi=540$ год.);

K_{Π} – коефіцієнт використання потужності ($K_{\Pi} < 1$, $K_{\Pi} = 0,65$).

$$V_e = 1,6 \cdot 0,15 \cdot 540 \cdot 0,65 = 84.24 \text{ (грн.)}$$

Розрахуємо інші витрати $V_{ін}$.

Інші витрати I_b можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$V_{ін} = (1..3) \cdot (Z_o + Z_p). \quad (5.6)$$

Отже, розрахуємо інші витрати:

$$V_{ін} = 1.5 * (22727.28 + 2272.72) = 37500 \text{ (грн.)}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$V = Z_o + Z_d + H_{зп} + A + K + V_e + I_b$$

$$V = 22727.28 + 2272.72 + 5500.22 + 750 + 5700 + 84.24 + 37500 = 74535.46 \text{ (грн.)}$$

Розрахуємо загальну вартість наукової роботи $V_{заг}$ за формулою:

$$B_{\text{заг}} = \frac{B_{\text{ін}}}{\alpha} \quad (5.7)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$B_{\text{заг}} = \frac{74535.46}{1} = 74535.46$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\beta} \quad (5.8)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{78659.24}{0,85} = 87688.78 \text{ (грн.)}$$

5.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спрогнозуємо отримання прибутку від реалізації результатів нашої розробки. Зростання чистого прибутку можна оцінити у теперішній вартості грошей. Це забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності .

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}} \Delta N)_i \quad (5.9)$$

де $\Delta\Pi_{\text{я}}$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$\Pi_{\text{я}}$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки товарообіг збільшиться на 75 процентів (що автоматично спричинить збільшення чистого прибутку підприємства на 50-75 процентів в залежності від вартості товару, далі будемо використовувати середнє значення 62.5%), а кількість користувачів, які будуть користуватись торговим сервісом збільшиться: протягом першого року – на 250 користувачів, протягом другого року – на 300 користувачів, протягом третього року – 200 користувачів. Статистика інтернет магазину до впровадження інформаційної технології складала 1000 активних покупців які в середньому приносили 340 гривень прибутку.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту $\Delta\Pi_1$ протягом першого року складатиме:

$$\Delta\Pi_1 = (340 \cdot 0.625) \cdot 1000 + (340 \cdot 1.625) \cdot 250 = 350625 \text{ грн.}$$

Протягом другого року:

$$\Delta\Pi_2 = \Delta\Pi_1 + (340 \cdot 1.625) \cdot 300 = 516375 \text{ грн.}$$

Протягом третього року:

$$\Delta\Pi_3 = \Delta\Pi_2 + (340 \cdot 1.625) \cdot 200 = 626875 \text{ грн.}$$

Теперішню вартість інвестицій PV , що можуть бути вкладені в розроблену нами інтелектуальну систему, можна розрахувати за формулою:

$$PV = [(1 \dots 5) \times 3B],$$

де $(1 \dots 5)$ – коефіцієнт, який враховує можливі додаткові витрати інвестора на можливе впровадження нашої розробки (оренда, підготовка персоналу, реклама тощо).

Для нашого випадку отримаємо:

$$PV = (1 \dots 5) \times 3B = 2.5 \times 87688.78 = 219221.94 \text{ (грн.)}$$

5.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $E_{\text{абс}}$ вкладених інвестицій розраховується за формулою:

$$E_{\text{абс}} = (ПП - PV), \quad (5.10)$$

де $ПП$ – приведена вартість всіх можливих чистих прибутків від реалізації розробки, грн;

PV – теперішня вартість інвестицій, $PV = 219221.94$ грн.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, рисунок 5.1.



Рисунок 5.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Розрахуємо вартість чистих прибутків за формулою:

$$ПП = \sum_1^m \frac{\Delta\Pi_t}{(1+\tau)^t} \quad (5.11)$$

де $\Delta\Pi_t$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки.

Отже, розрахуємо вартість чистого прибутку:

$$ПП = \frac{350625}{(1+0,1)^1} + \frac{516375}{(1+0,1)^2} + \frac{626875}{(1+0,1)^3} = 1216486.7 \text{ (грн.)}$$

Тоді розрахуємо $E_{\text{абс}}$:

$$E_{abc} = 1216486.7 - 219221.94 = 997264.76 \text{ грн.}$$

Оскільки $E_{abc} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_B за формулою:

$$E_B = \sqrt[T]{1 + \frac{E_{abc}}{PV}} - 1 \quad (5.12)$$

де E_{abc} – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій $PV = 3B$, грн;

T – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_B = \sqrt[3]{1 + \frac{997264.76}{219221.94}} - 1 = 0.77 \text{ або } 77\%$$

Далі, розраховану величина E_B порівнюємо з мінімальною (бар'єрною) ставкою дисконтування τ_{\min} , яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування τ_{\min} визначається за формулою:

$$\tau = d + f,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2020 році в Україні $d = 0,2$;

f – показник, що характеризує ризикованість вкладень, величина $f = 0,1$.

$$\tau = 0,2 + 0,1 = 0,3$$

Оскільки $E_B = 77\% > \tau_{\min} = 0,3 = 30\%$, то інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ розраховується за формулою:

$$T_{\text{ок}} = \frac{1}{E_B}$$

$$T_{\text{ок}} = \frac{1}{0.77} = 1,3 \text{ року}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

5.5 Висновок

В даному розділі було здійснено оцінювання комерційного потенціалу розробки інформаційної технології з розробки та дослідження рекомендаційного сервісу для інтернет магазину з використанням технологій машинного навчання.

Проведено технологічний аудит з залученням двох експертів. Аналіз експертних даних показав, що рівень комерційного потенціалу розробки вище середнього. Дослідження комерційного потенціалу розробки підтвердило, що програмний продукт за своїми характеристиками випереджає аналогічні програмні продукти і є перспективною розробкою. Він має кращі функціональні показники, а тому є конкурентоспроможним товаром на ринку.

Згідно із розрахунками всіх статей витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи загальна вартість витрат на розробку і впровадження складає 87688.78 грн.

Розрахована абсолютна ефективність вкладених інвестицій в сумі 997264.76 грн свідчить про отримання прибутку інвестором від впровадження програмного продукту у діяльність підприємства.

Щорічна ефективність вкладених в наукову розробку інвестицій складає 77%, що вище за мінімальну бар'єрну ставку дисконтування, яка складає 30%. Це означає потенційну зацікавленість інвесторів у фінансуванні розробки.

Термін окупності складає 1.3 року, що також свідчить про доцільність фінансування.

Усе це, узятє разом, забезпечує прийняття рішення про доцільність виготовлення нового продукту.

ВИСНОВКИ

В ході аналізу сучасного стану досліджуваної проблеми, пов'язаної з використанням рекомендаційних систем, були розглянуті основні підходи до їх створення, що використовуються сучасними інтернет-платформами.

Здійснено порівняльний аналіз можливості застосування розглянутих стратегій в рамках сфер що мають складний (комплексний) простір елементів, який використовується нечасто.

Розроблено математичну модель надання рекомендацій для користувачів інтернет-магазину.

Удосконалено інформаційну технологію надання рекомендацій, що відрізняється від існуючих стратегією надання рекомендацій, яка поєднує в собі декілька існуючих підходів та базується на методах машинного навчання, що розширює функціональні можливості рекомендаційних сервісів та підвищує можливість заробітку для інтернет магазину. Створено алгоритми функціонування рекомендаційного сервісу.

Виконано програмну реалізацію інформаційної технології рекомендаційного сервісу для інтернет-магазину з використанням машинного навчання у вигляді веб-додатку.

Проведено тестування та аналіз результатів роботи створеного сервісу.

За результатами тестування, схожість рекомендацій власної розробки на 5.15% більше ніж в конкурентів. Також було виявлено, що швидкість генерації рекомендацій на 7.65% більше, що зумовлено роботою рекомендаційної систем безпосередньо на одному сервері з сайтом в той час як конкуренти працюють через API.

Отже, всі завдання, які було поставлено вирішено у повному обсязі, мету роботи досягнуто.

ЛІТЕРАТУРА

1. Озеранський В.С., Пеньков В.С. «Рекомендаційні системи» в Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2020)», Вінниця, 2020. [Електронний ресурс]. Режим доступу <https://conferences.vntu.edu.ua/index.php/mn/mn2021/paper/view/11031> Дата звернення: грудень, 2020
2. P. Lops, M. Gemmis, G. Semeraro Recommender Systems Handbook Springer-Verlag, 2011.
3. C. C. Aggarwal, Recommender Systems: The Textbook. Springer. С: 8-37, 2016.
4. Introduction to recommender systems [Електронний ресурс]. Режим доступу: <https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>
5. K. Falk, Practical Recommender Systems 1st Edition. Manning Publications С:150-174, 2019
6. Recommender Systems in Practice [Електронний ресурс]. Режим доступу: <https://towardsdatascience.com/recommender-systems-in-practice-cef9033bb23a>
7. Xiaoyuan Su, Khoshgoftaar T. M. A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence, 2009.
8. U., K., Viswanathan. Optimization of User Based Collaborative Filtering. LAP LAMBERT Academic Publishing. С:7-22, 2019
9. S., Berkovsky, I., Cantador, Domonkos T. Collaborative Recommendations: Algorithms, Practical Challenges and Applications. World Scientific Publishing Co, 2019
10. B., Jesús; O., Fernando; A.; Bernal, A collaborative filtering approach to mitigate the new user cold start problem. 2012
11. Wikipedia [Електронний ресурс]. Cold start (Computing). Режим доступу: [https://en.wikipedia.org/wiki/Cold_start_\(computing\)](https://en.wikipedia.org/wiki/Cold_start_(computing)).

12. Wikipedia [Электронный ресурс]. Knowledge-based recommender systems. Режим доступа: https://en.wikipedia.org/wiki/Knowledge-based_recommender_system.
13. Y. Salem, J. Hong, W. Liu. History-Guided Conversational Recommendation, World Wide Web (WWW), 2014.
14. А. Фелфернигу и Р. Бёрку, Системы рекомендаций на основе ограничений: Технологии и проблемы исследований. АСМ Международная конференция по электронной торговле. С: 17-26, 2008.
15. Л. Чен и П. Пу. Рекомендации на основе критики: обзор и новые тенденции. «Моделирование пользователей и адаптированное к пользователю взаимодействие» (UMUAI), С: 125-150, 2012.
16. Salter, J.; Antonopoulos, N. (January 2006). CinemaScreen Recommender Agent: Combining Collaborative and Content-Based Filtering. IEEE Intelligent Systems.2006
17. T., Chen, W., Li Han, H., D., Wang, Y., X., Zhou; B., X., B., Y., Zang, Content Recommendation System Based on Private Dynamic User Profile. International Conference on Machine Learning and Cybernetics, 2014.
18. Windowing meyhod. [Электронный ресурс]. Режим доступа: <https://wiki.loginom.ru/articles/windowing-method.html>
19. Метод взвешенной суммы критериев в анализе многокритериальных решений. [Электронный ресурс]. Режим доступа: <https://bijournal.hse.ru/data/2013/10/11/1281327437/5>.
20. Маклаков С.В. Моделирование бизнес-процессов с BPwin 4.0 – М.: Диалогмифи, 2002. – 151 с.
21. Черемных С.В., Семёнов И.О., Ручкин В.С. Структурный анализ систем: IDEF – технологии – М.:Финансы и статистика, 2003. – 208 с.
22. Wikipedia [Электронный ресурс]. Model-View-Controller. Режим доступа:<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>.
23. P. DuBois, MySQL Cookbook: Solutions for Database Developers and Administrators. O'Reilly Media; 3 edition, 2012 – С. 866.

24. М. Зандстра PHP: Objects, Patterns, and Practice. «Ozon», 3-издание, 2011. – С. 576.
25. Wikipedia [Электронный ресурс]. JS. Режим доступа: <https://en.wikipedia.org/wiki/JavaScript>.