

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)

Кафедра _____ Системотехніки _____
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

Розробка та дослідження рекомендаційної системи для сервісу з
підбору нерухомості _____
(тема)

Виконав: студент 2 курсу, групи ІТІм-18-1
Образков В. Є. _____
(прізвище, ініціали)

Спеціальність 6.050101 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна _____

Освітня програма Інформаційні
технології проектування _____
(повна назва освітньої програми)

Керівник _____ доц. Імангулова З.А. _____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри СТ

(підпис)

проф. Гребеннік І.В. _____
(прізвище, ініціали)

2019 р

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук
Кафедра _____ системного проектування
Освітньо-кваліфікаційний рівень другий (магістерський)
Спеціальність _____ 6.050101 Комп'ютерні науки
Тип програми _____ освітньо-професійна
Освітня програма _____ Інформаційні технології проектування

ЗАТВЕРДЖУЮ:
Зав. кафедри СТ _____
(підпис)
« _____ » _____ 2019 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

Студентові _____ Образкову Владиславу Євгенійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та дослідження рекомендаційної системи для сервісу з підбору нерухомості

затверджена наказом по університету від « 04 » листопада 2019 р. № 1627 Ст

2. Термін здачі студентом закінченої роботи: 18.12.2019

3. Вихідні дані до роботи(проекту): Науково-технічні публікації, дані статей, результати експериментальних досліджень, дані проектів щодо розробки рекомендаційних систем

4. Зміст пояснювальної записки (перелік питань, що потрібно розробити)
Вступ, 1 Огляд і аналіз сучасного стану проблеми що розглядається, 4.1 Поняття рекомендаційної системи, 4.2 Рекомендаційні системи на основі спільної фільтрації, 4.3 Рекомендаційні системи на основі вмісту, 4.4 Рекомендаційні системи засновані на знаннях, 2 Рекомендаційна система для сервісів зі «складними» предметними сферами, 2.1 Стратегія надання рекомендацій, 2.2 Математичний опис задачі, 3 Постановка задачі, 3.1 Формалізована постановка задачі., 3.2 Характеристика предметної області, 3.3 Характеристика комплексу задач, 3.3.1 Призначення комплексу задач, 3.3.2 Перелік об'єктів, при управлінні якими вирішується комплекс задач, 3.4 Вхідна інформація, 3.5 Вихідна інформація, 4 Проектування системи, 4.1 Опис системних вимог, 4.2 Опис функціональних вимог, 4.2 Опис архітектури системи, 5 Розробка інформаційного забезпечення, 6 Розробка програмного забезпечення та експериментальний аналіз, 6.1 Вибір мови програмування, 6.3 Опис алгоритмів інформаційної системи, 6.4 Експериментальний аналіз та наочні результати роботи рекомендаційної системи, Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, схем плакатів) _____

5.1 Схема алгоритму створення об'єви (1 Аркуш А4). 5.2 Схема алгоритму пошуку

об'яв(1 Аркуш А4). 5.3 Схема алгоритму роботи рекомендаційної системи (1 Аркуш А4). 5.4 Концептуальна діаграма інформаційної системи (1 Аркуш А4). 5.5 Діаграма декомпозиції функції «Система» публікації оголошень з пошуку-продажу нерухомості» (1 Аркуш А4). 5.6 Діаграма декомпозиції функції «Пошук оголошень» (1 Аркуш А4). 5.7 Діаграма декомпозиції функції «Використання рекомендаційної системи» (1 Аркуш А4). 5.8 Знімок екрану «Активне оголошення та рекомендації до нього(1 Аркуш А4).».

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доцент Імангулова З. А.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на дипломну роботу	04.11.19	виконано
2	Аналіз предметної галузі і постановка завдання	05.11.19 – 8.11.19	виконано
3	Дослідження і аналіз існуючих стратегій створення рекомендацій	9.11.19 – 14.11.19	виконано
5	Розробка стратегії надання рекомендацій, для сфер зі складним простором елементів	15.11.19 – 21.11.19	виконано
6	Розробка веб-сервісу з пошуку нерухомості з використанням розробленої стратегії надання рекомендацій	22.11.19 – 04.12.19	виконано
7	Експериментальний аналіз та тестування розроблених стратегії та сервісу	05.12.19 – 07.12.19.	виконано
8	Оформлення пояснювальної записки та текстових матеріалів	07.12.19 – 08.12.19	виконано
10	Попередній захист	09.12.19	виконано
11	Преставлення роботи в ДЕК	16.12.19	виконано

Дата видачі завдання _____

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доцент Імангулова З. А.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської атестаційної роботи: 115 с., 5 табл., 26 рис., 3 додатки, 26 джерел. Графічна частина атестаційної роботи містить: 8 плакатів.

РЕКОМЕНДАЦІЙНІ СИСТЕМИ, CONTENT-BASED FILTERING, COLLABORATIVE FILTERING, KNOWLEDGE BASED RECOMMENDERS

Об'єктом дослідження атестаційної роботи є веб-сервіси, що спеціалізуються на продажі чи наданні доступу кінцевому користувачеві до деякого контенту (продукту), які охоплюють предметні області, простір елементів яких є складним, з точки зору можливості отримання оцінок та/або має низьку частоту переглядів.

Предмет дослідження – алгоритми та методи створення рекомендацій, які застосовуються для прогнозування уподобань користувачів та надання їм пропозицій, згідно їх уподобань.

Мета атестаційної роботи – дослідження існуючих методів побудови рекомендацій на предмет можливості їх використання у сферах, простір елементів яких є складним та/або має низьку частоту переглядів та створення системи рекомендацій, для таких предметних областей.

Досягнення мети атестаційної роботи базується на комплексному використанні методів теорії фільтрації інформації, що стосуються створення рекомендаційних систем різних видів

Результат атестаційної роботи – стратегія надання рекомендацій, рекомендаційна система яка використовує розроблену стратегію, а також веб-сервіс з пошуку нерухомості, з вбудованою до нього рекомендаційною системою.

ABSTRACT

Explanatory note: 115 pages, 5 tables, 26 figures, 3 applications, 26 sources.
Graphic material 8 p.

**RECOMMENDER SYSTEMS, CONTENT-BASED FILTERING,
COLLABORATIVE FILTERING, KNOWLEDGE BASED RECOMMENDERS**

The object of the work is web services that specialize in selling or providing end-user access to certain content, covering subject areas of which the space of the elements is complex, in terms of being able to receive ratings and / or having a low frequency of views.

The subject of the work is the algorithms and methods of creating recommendations, which are used to predict users' preferences and to provide offers according to their preferences.

The purpose of the work is to investigate existing methods of constructing recommendations for the possibility of their use in areas whose element space is complex and has a low frequency of review and to create of a recommender system for such subject areas.

Methods of development – methods of information filtering theory related to the creation of recommendation systems.

The results of the work – strategy for making recommendations, the recommender system based on it, realty search service which uses a developed recommendation system.

ЗМІСТ

Перелік скорочень, умовних позначень, символів, одиниць і термінів	8
Вступ.....	9
1. Огляд і аналіз сучасного стану проблеми, що розглядається	12
1.1 Поняття рекомендаційної системи	12
1.2 Рекомендаційні системи на основі спільної фільтрації.....	15
1.3 Рекомендаційні системи на основі вмісту	21
1.4 Рекомендаційні системи засновані на знаннях.....	25
2 Рекомендаційна система для сервісів зі «складними» предметними сферами.....	28
2.1 Стратегія надання рекомендацій	28
2.2 Математичний опис задачі	32
3 Постановка задачі.....	38
3.1 Формалізована постановка задачі	38
3.2 Характеристика предметної області.....	38
3.3 Характеристика комплексу задач	40
3.3.1 Призначення комплексу задач.....	40
3.3.2 Перелік об'єктів, при управлінні якими вирішується комплекс задач	41
3.4 Вхідна інформація	41
3.5 Вихідна інформація	43
4 Проектування системи	45
4.1 Опис системних вимог	45
4.2 Опис функціональних вимог	46
4.2 Опис архітектури системи	51
5 Розробка інформаційного забезпечення	53
5.1 Вибір СУБД	53

5.2	Опис розробленої БД	53
6	Розробка програмного забезпечення та експериментальний аналіз	
	алгоритму створення рекомендацій	57
6.1	Вибір мови програмування	57
6.2	Опис алгоритмів інформаційної системи.....	60
6.3	Експериментальний аналіз та наочні результати роботи РС	63
	Висновки.....	69
	Перелік джерел посилання	71
	Додаток А. Графічні матеріали	74
	Додаток Б. Посібник користувача	83
	Додаток В. Текст програми	94

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

Collaborative filtering – колаборативна або спільна фільтрація;

Content-based filtering – фільтрація на основі вмісту або контентна фільтрація;

Memory based – фільтрація заснована на сусідстві;

Model based – фільтрація заснована на моделі;

MVC –Model-View-Controller – архітектурний шаблон;

РС – рекомендаційна система.

ВСТУП

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. [1, 2, 3] 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.

Колаборативна фільтрація у рекомендаційних системах базується на попередніх взаємодіях між користувачами та елементами, які використовуються для того, щоб створювати нові рекомендації. Такі взаємодії зберігаються у вигляді матриць оцінок «елементи-користувачі». Основна ідея полягає у тому, що цих попередніх оцінок достатньо для того щоб знаходити

схожих користувачів та/або схожі (близькі) елементи та створювати рекомендації на їх основі [1 – 3].

Методи фільтрації на основі вмісту засновані на описі елементів та профілі переваг користувача, що будується на попередніх взаємодіях користувача з елементами. Ця взаємодія зберігається у матрицю оцінок елементів, на основі якої потім будуються уподобання користувача. Основна ідея полягає у тому щоб пропонувати користувачу ті елементи, які найбільше відповідають його персональним уподобанням [1, 2, 3].

Існуючі підходи базуються на різних способах використання будь-чєї історії рейтингів, для створення рекомендацій. Але такої історії уподобань може не бути. Цілком можливо, що у користувача взагалі не буде оцінок і фактично кожен новий користувач буде мати таку проблему. Це називається проблемою холодного запуску, і це реальна проблема в проектуванні рекомендаційних систем, тому що важко робити прогнози, коли уподобання користувача невідомі. Іншою причиною недостатньої кількості оцінок, може бути те, що рекомендовані речі є дуже складними та комплексними. Такі предмети, як машини, нерухомість, ювелірні вироби купуються нечасто і можуть бути настільки складними, що оцінки для них можуть бути недоступні. Тому створення рекомендаційних систем, для сфер з обмеженою можливістю отримання оцінок є актуальною задачею та предметом активних досліджень.

Метою даної роботи є дослідження існуючих методів побудови рекомендацій на предмет можливості їх використання у сферах, простір елементів яких є складним та/або має низьку частоту переглядів та створення системи рекомендацій, для таких предметних областей.

Поставлена мета роботи обумовила наступні завдання дослідження:

- 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 рекомендаційної системи, що може використовуватися для подібних сфер;
- розробка веб-сервісу для однієї з таких предметних областей;
- реалізація та впровадження розробленої рекомендаційної системи до створеного сервісу.

Об'єктом дослідження є веб-сервіси, що спеціалізуються на продажі чи наданні доступу кінцевому користувачеві до деякого контенту (продукту) та охоплюють предметні сфери простір елементів яких є складним, з точки зору можливості отримання оцінок та/або має низьку частоту переглядів.

Предметом дослідження є алгоритми та методи створення рекомендацій, які застосовуються для прогнозування уподобань користувачів та надання їм пропозицій, згідно їх уподобань.

Методи дослідження. Досягнення мети атестаційної роботи базується на комплексному використанні методів теорії фільтрації інформації, що стосуються створення рекомендаційних систем різних видів.

1. ОГЛЯД І АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ, ЩО РОЗГЛЯДАЄТЬСЯ

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, рекомендаторами продуктів для сервісів, таких як Amazon, або рекомендаторами 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 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).



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.

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):

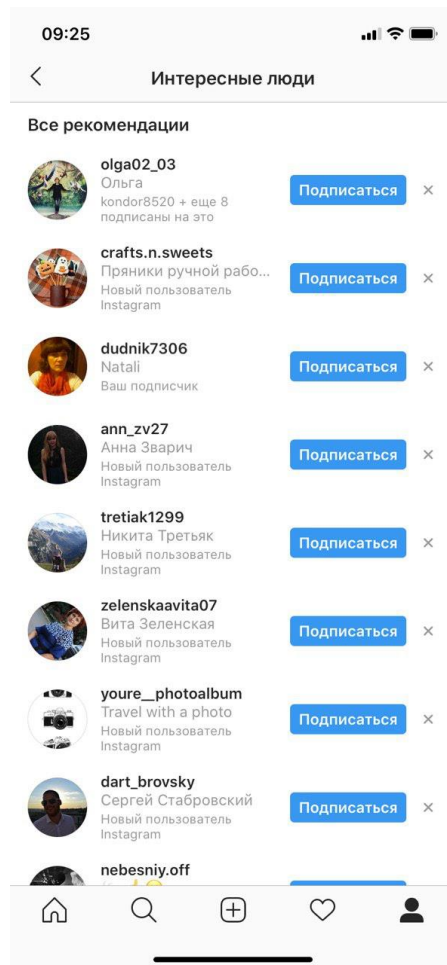


Рисунок 1.2 – Рекомендації у соціальній мережі Instagram



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);
- 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 користувачької 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 використовувана 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 байесовських 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 масштабністю 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, чії смаки схожі на переваги нашого цільового користувача [8]. Якщо раніше «користувач 1» і «користувач 2» проставили 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 Жану-П'єру сподобалися фільми «Повернення джедая» і «Імперія завдає удару у відповідь», а Джейсону сподобався фільм «Повернення джедая», то ми безумовно повинні підказати Джейсону і фільм «Імперія завдає удару у відповідь» (рисунок 1.4).

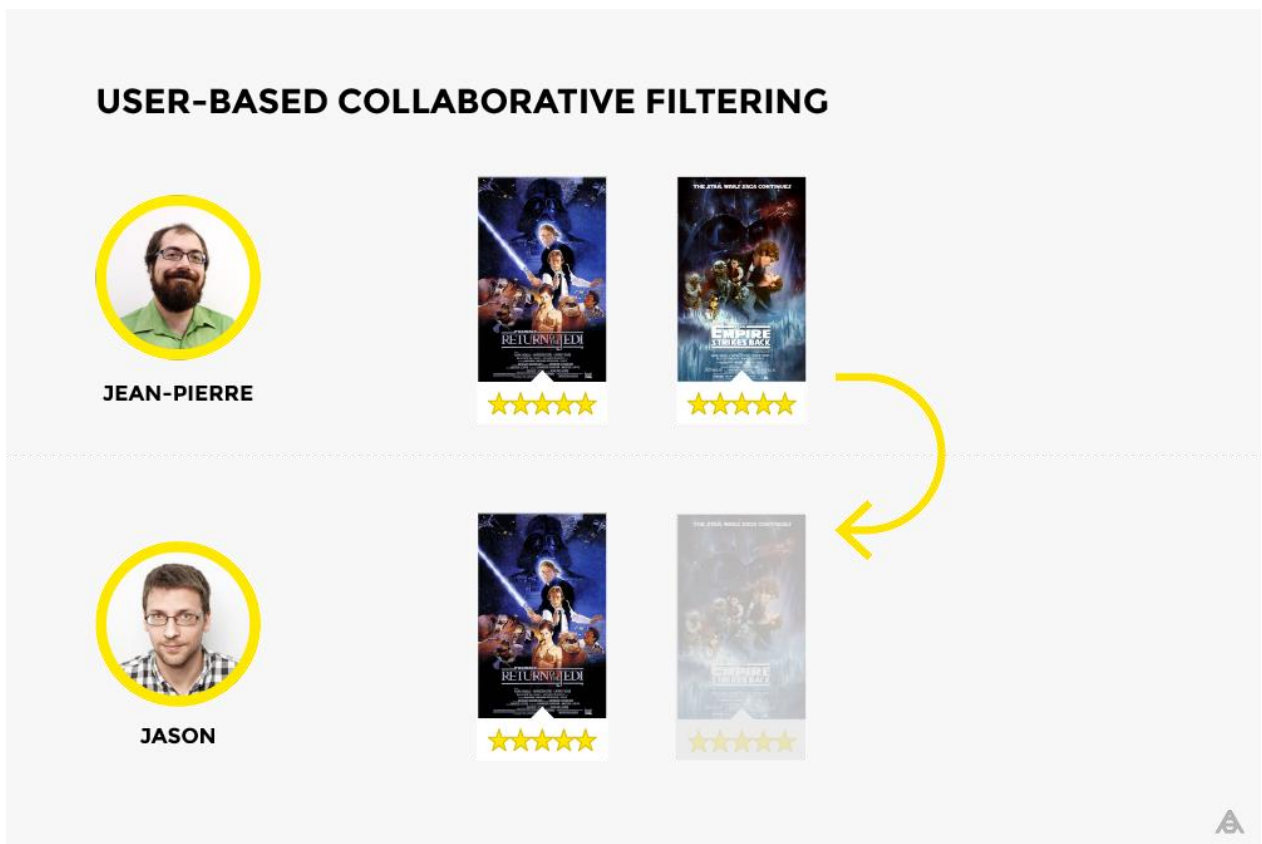


Рисунок 1.4 – Колаборативна фільтрація «користувач-користувач»

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.

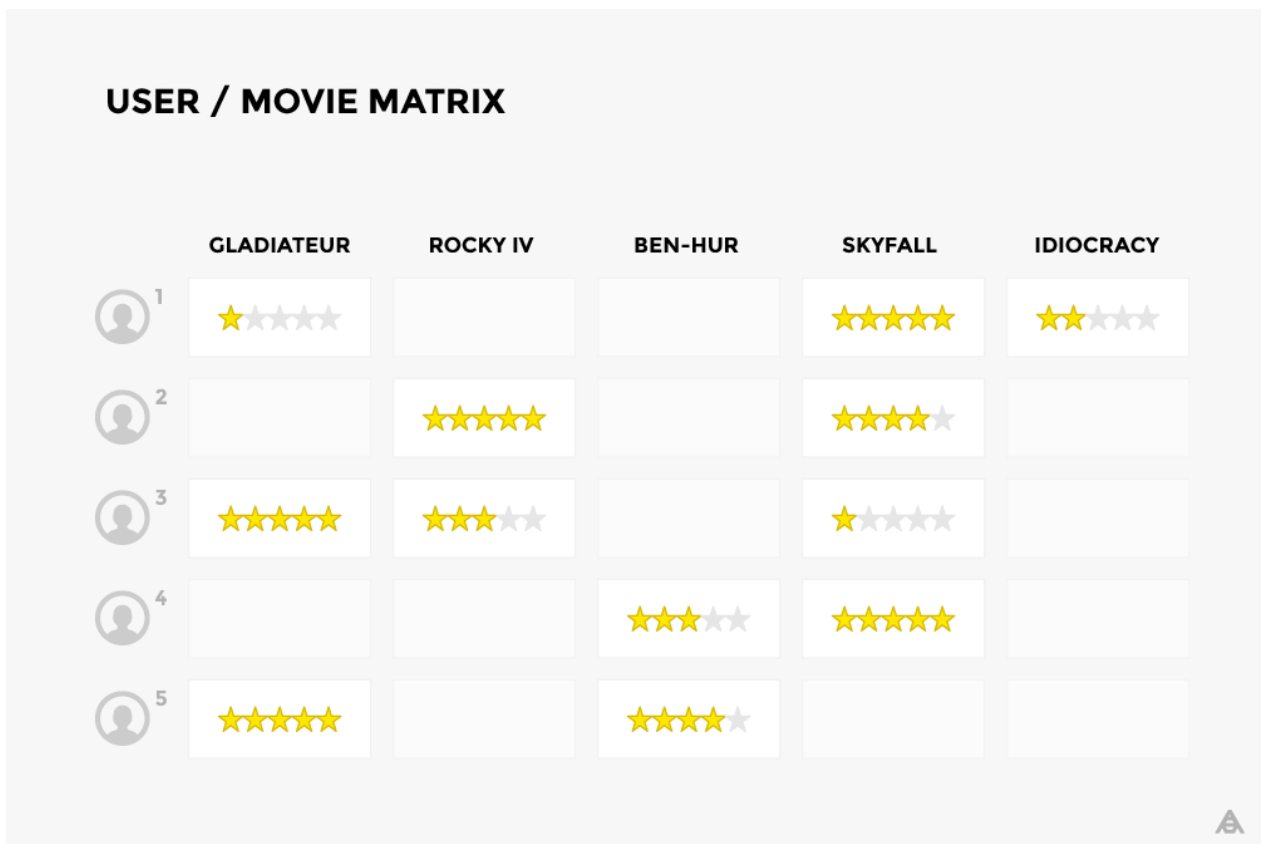


Рисунок 1.5 – Матриця оцінок «користувач-елемент»

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, Жаном-П'єру і Джейсону сподобалися фільми «Повернення джедая» і «Імперія завдає удару у відповідь». Можна зробити висновок, що більшості користувачів, які високо оцінили перший фільм, повинен сподобатися і другий. Таким чином, було б доречно запропонувати фільм «Імперія завдає удару у відповідь» Ларрі, якому сподобався фільм «Повернення джедая» (рисунок 1.6).

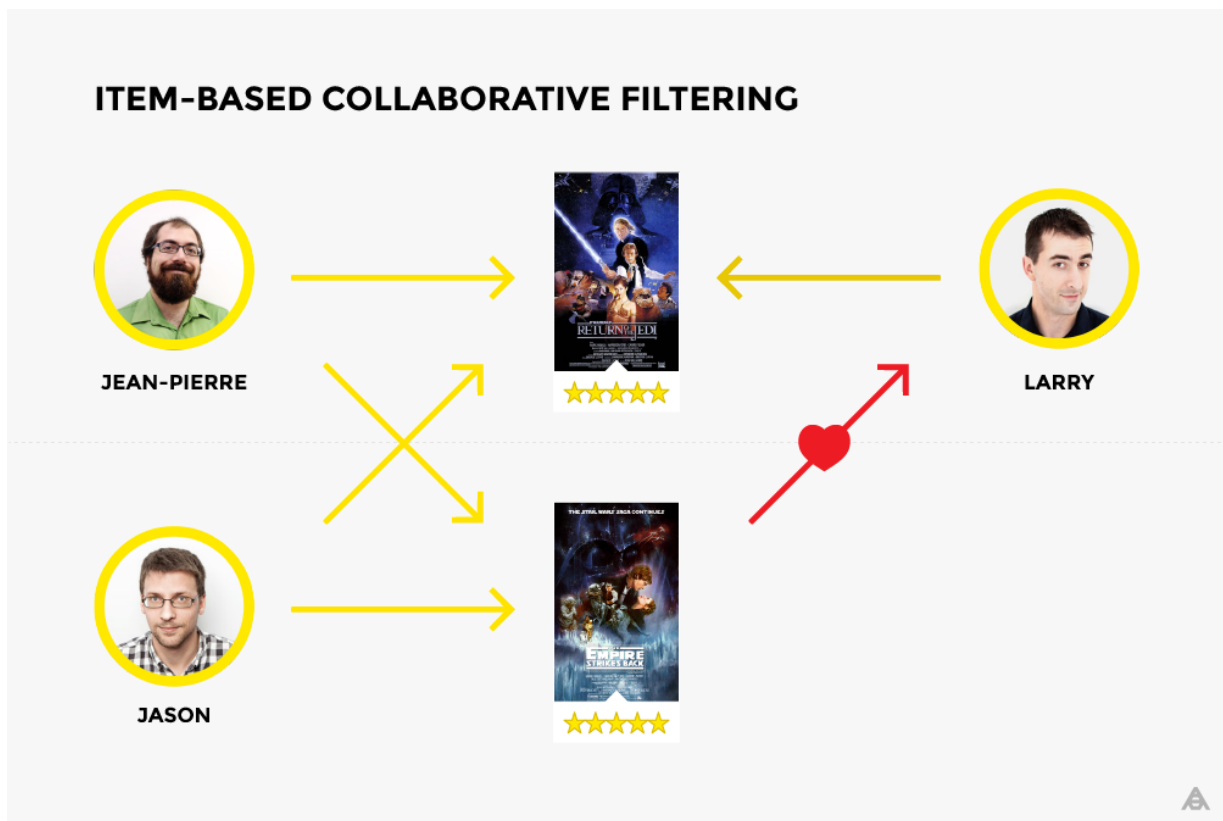


Рисунок 1.6 – Колаборативна фільтрація «елемент-елемент»

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 двокімнатну квартиру у певному районі з обмеженою максимальною вартістю оренди). Тому проблема розрідженості даних, що супроводжує сумісну фільтрацію стає більш загальною і унеможливорює 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].

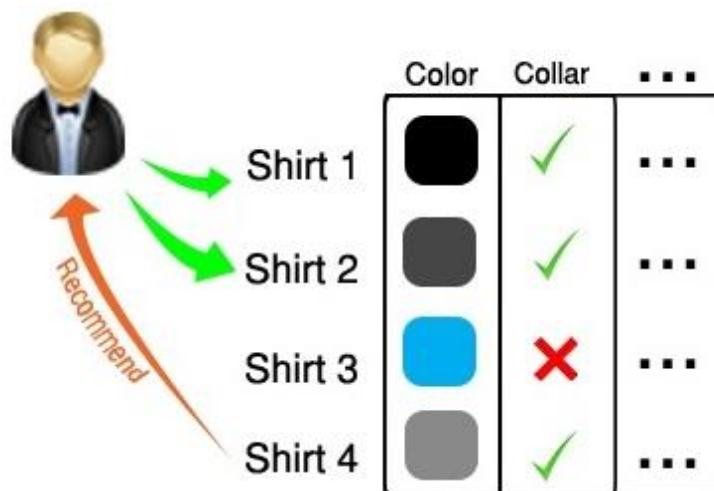


Рисунок 1.7 – Візуалізація підходу зі створенням профілю користувача і профілю елемента

Для виконання результативної 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].

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.

2 РЕКОМЕНДАЦІЙНА СИСТЕМА 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 напряму 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 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.

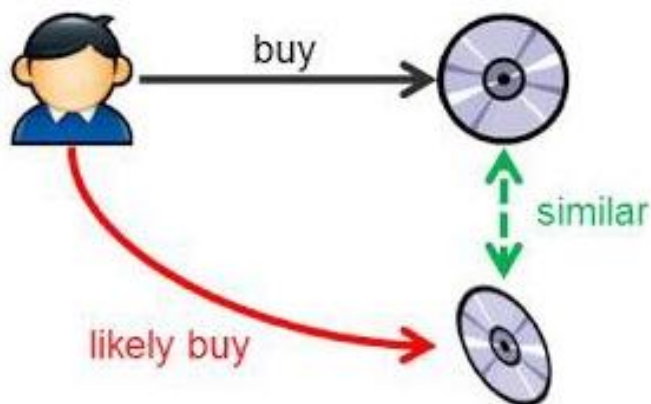


Рисунок 2.1 – Підхід з аналізом лише контенту

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 оголошення, що найбільш схожі на те, яке він наразі переглядає (активне оголошення). На другому етапі, коли користувач вже переглянув певну кількість оголошень, рекомендації базуються на групі з деякої кількості останніх переглянутих оголошень.

Група останніх переглянутих оголошень 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.

Временной ряд	1	2	3	4	5	6	7	8	9	...	n
1 окно	x1	x2	x3	x4	x5	y1					
2 окно		x1	x2	x3	x4	x5	y1				
3 окно			x1	x2	x3	x4	x5	y1			
...											

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 квартири у Києві, якщо при пошуку він обрав Харків, 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 відносну важливість атрибутів що описують елементи, та спосіб їх порівняння.

Проте відносну важливість атрибутів можна встановити і без участі експерта виходячи із законів здорового глузду і 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 Математичний опис задачі

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 :

$$\text{Score} = \sum w_i \times k_i, \quad (2.3)$$

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:

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

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 отримує 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 входу 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.

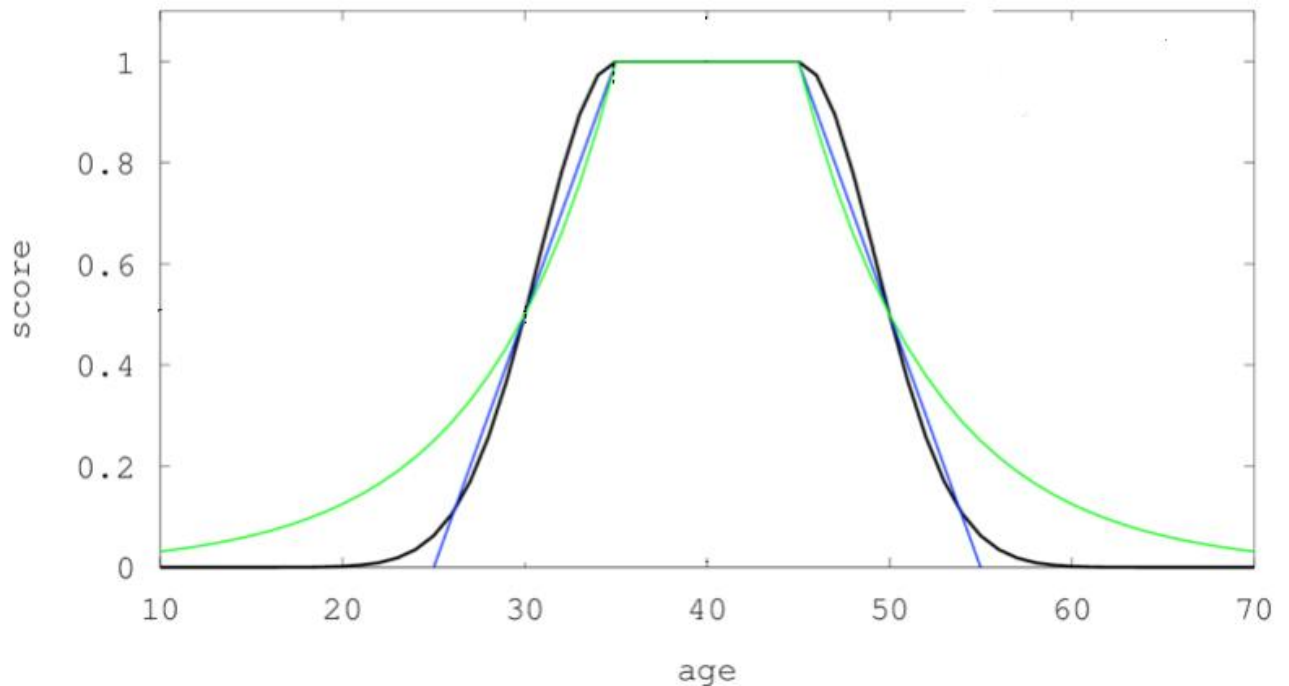


Рисунок 2.3 – графіки залежності оцінки (score) деякої величини в залежності від збільшення відстані до центра мас: зелений – експоненціальна залежність; синій – лінійна залежність; чорний – гауссіан

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.

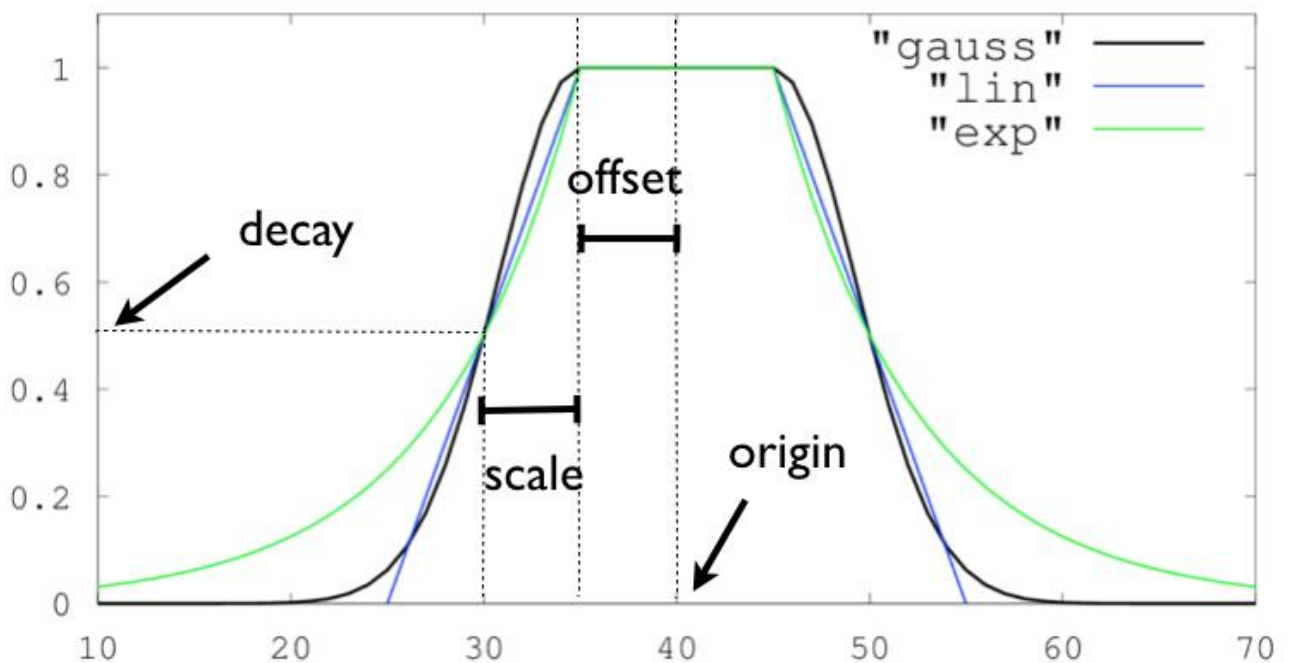


Рисунок 2.4 – Наочна інтерпретація параметрів затухання

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:

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

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 оцінюватись 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 розраховуватись наступним чином: $K_i = 0$.

Тобто в такому разі буде спостерігатися зменшення значення функції схожості.

3 ПОСТАНОВКА ЗАДАЧІ

3.1 Формалізована постановка задачі

Метою даної атестаційної роботи є:

- 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 рекомендаційної системи, що може використовуватися для подібних сфер, з використанням деяких аспектів розглянутих стратегій;
- розробка веб-сервісу для однієї з таких сфер;
- реалізація та впровадження розробленої рекомендаційної системи до створеного сервісу.

3.2 Характеристика предметної області

Задачу створення рекомендаційної системи для областей, що охоплюють «складні» предметні сфери будемо виконувати на прикладі сфери нерухомості.

В даний час існує безліч систем, які здійснюють посередницькі послуги між продавцем і покупцем на ринку нерухомості. Як правило, вони надають такий набір послуг:

- купівля-продаж нерухомості;
- оренда нерухомості;
- пошук продавців і покупців;
- юридичний супровід угоди.

Такі сервіси зазвичай складаються з клієнтського додатку, що представлений веб-інтерфейсом, що відповідає за отримання даних з сервера

для подальшої візуалізації в зручному вигляді для користувача і відправки даних на сервер, а також сервера, який приймає запити клієнта, обробляє і зберігає інформацію. Сервіси з пошуку нерухомості надають своїм користувачам наступні загальні можливості:

- перегляд та пошук оголошень за відповідними рубриками;
- фільтрація оголошень за різними критеріями;
- реєстрація користувачів у системі;
- додавання/видалення оголошень за відповідними рубриками;
- написання відгуків
- надання користувачу рекомендацій.

Приклад такої системи можна побачити на рисунках 3.1 – 3.2 Розроблюваний сервіс з пошуку нерухомості буде виконувати усі описані стандартні можливості та буде спеціалізуватися на продажі/оренді квартир будинків та гаражів.

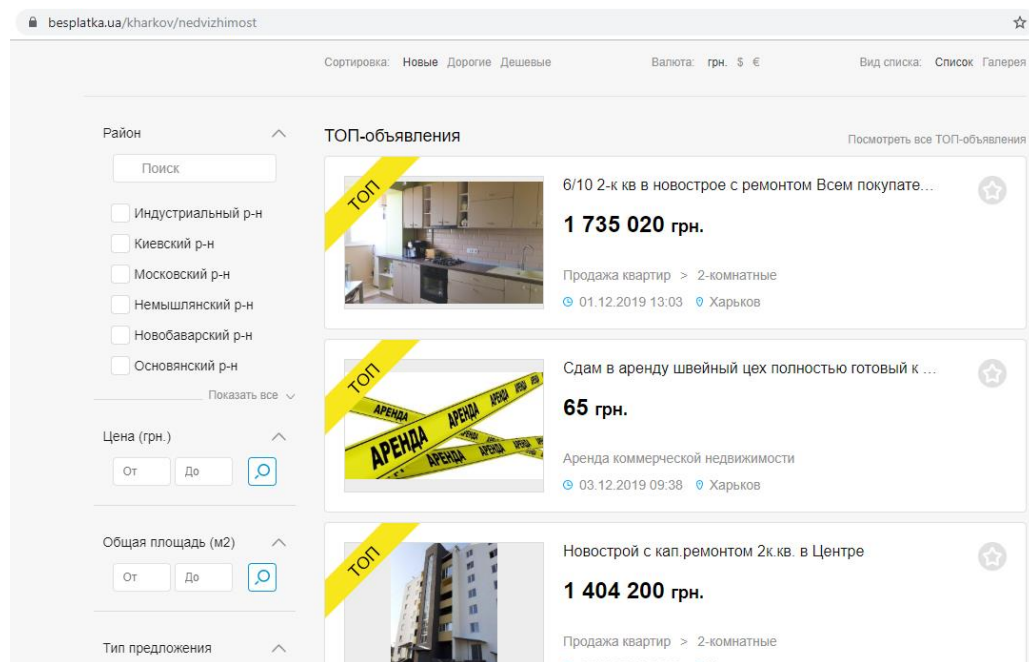


Рисунок 3.1 – Сторінка пошуку у сервісі з пошуку нерухомості «Бесплатка»

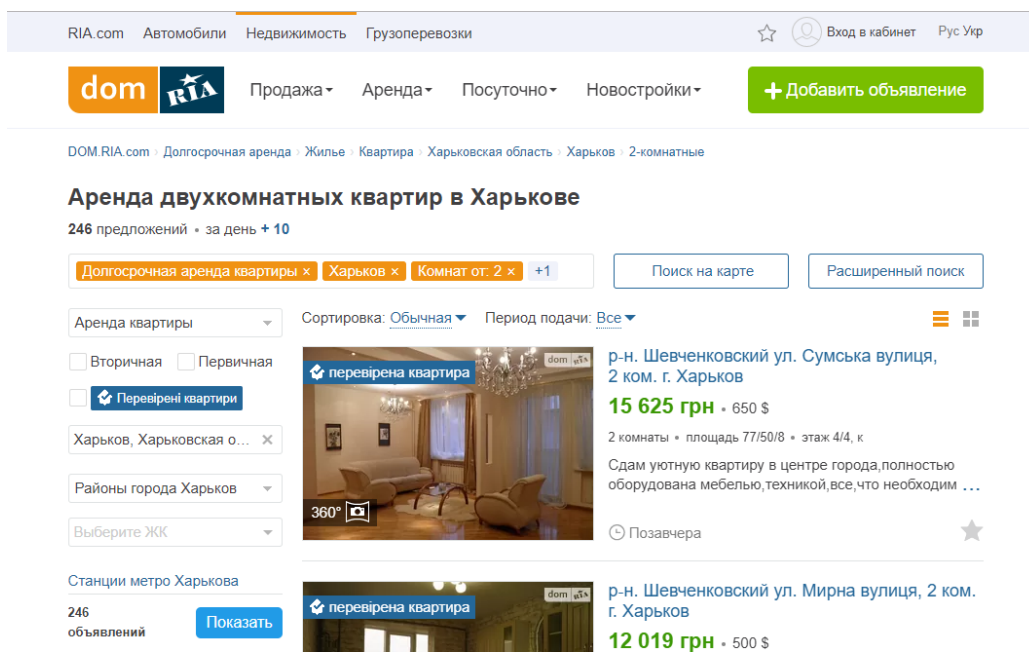


Рисунок 3.2 – Сторінка пошуку у сервісі з пошуку нерухомості «DomRIA»

3.3 Характеристика комплексу задач

3.3.1 Призначення комплексу задач

Рішення комплексу задач призначене для створення такої рекомендаційної системи, що відповідала б наступним вимогам:

- відсутність необхідності використання історії рейтингів;
- відсутність проблеми «холодного старту»;
- відсутність необхідності створення асоціативної моделі експертом предметної області;
- мінімізація «вузького місця» придбання знань стосовно предметної області;
- відповідність обмеженням, що встановлює користувач при пошуку;
- гнучкість прогнозувань, в залежності від зміни вектору уподобань користувача під час перегляду елементів.

3.3.2 Перелік об'єктів, при управлінні якими вирішується комплекс задач

Об'єктом, при управлінні яким вирішується комплекс задач дослідження, є веб-сервіс, з пошуку нерухомості, що спеціалізується на наданні користувачу доступу до кінцевого контенту, а саме до оголошень про продаж/оренду квартир чи будинків.

3.3.3 Тривалість і періодичність рішення

Дана задача може вирішуватися кожного разу, коли користувачі користуються можливістю пошуку деяких елементів.

3.4 Вхідна інформація

Вхідною інформацією є:

- тип уподобаного об'єкту;
- кількість характеристик, що описують об'єкт;
- вектор значень характеристик, що описують об'єкт та шкали в яких вони вимірюються;
- вектор вагових коефіцієнтів, що описують важливість характеристик;
- кількість оголошень, по яким буде будуватися прогноз;
- offset (зміщення) – відстань, на якій значення характеристики буде вважатися рівним аналогічній характеристиці уподобаного об'єкту, задається для кожної характеристики;
- scale (масштаб) – відстань від значення характеристики уподобаного об'єкту, яка визначає діапазон прийнятної (допустимої) схожості, задається

для кожної характеристики;

– decay (спад) – значення схожості в яке оцінюється значення характеристики на відстані масштабу від значення характеристики уподобаного об'єкту, задається для кожної характеристики;

– кількість рекомендованих елементів.

Перелік та опис вхідних повідомлень наведено в таблиці 3.1

Таблиця 3.1 – Перелік та опис вхідних повідомлень;

Ідентифікатор	Форма представлення	Термін і частота видачі
Тип уподобаного об'єкту	Дані вбудовані в програмний код, визначені на етапі проектування сервісу	Кожен раз при використанні сервісу з рекомендаційною системою
Кількість характеристик	Дані вбудовані в програмний код, визначені на етапі проектування сервісу	Кожен раз при використанні сервісу з рекомендаційною системою
Вектор значень характеристик	Дані що зберігаються в БД, визначені на етапі додавання об'єкту у систему	Кожен раз при використанні сервісу з рекомендаційною системою
Вектор вагових коефіцієнтів	Дані вбудовані в програмний код, визначені на етапі проектування РС	Кожен раз при використанні сервісу з рекомендаційною системою

Продовження таблиці 3.1

Ідентифікатор	Форма представлення	Термін і частота видачі
Кількість оголошень, по яким буде будуватися прогноз	Дані вбудовані в програмний код, визначені на етапі проектування РС	Кожен раз при використанні сервісу з рекомендаційною системою
Зміщення	Дані вбудовані в програмний код, визначені на етапі проектування РС	Кожен раз при використанні сервісу з рекомендаційною системою
Масштаб	Дані вбудовані в програмний код, визначені на етапі проектування РС	Кожен раз при використанні сервісу з рекомендаційною системою
Спад	Дані вбудовані в програмний код, визначені на етапі проектування РС	Кожен раз при використанні сервісу з рекомендаційною системою
Кількість рекомендованих елементів	Дані вбудовані в програмний код, визначені на етапі проектування сервісу	Кожен раз при використанні сервісу з рекомендаційною системою

3.5 Вихідна інформація

Результатом роботи рекомендаційної системи та вихідними даними є оголошення (у заданій кількості), що підбрала система рекомендацій, які певним чином відображаються у зовнішньому користувацькому інтерфейсі у

веб додатку.

Перелік та опис вихідних повідомлень наведено в таблиці 3.2

Таблиця 3.2 – Перелік та опис вихідних повідомлень

Ідентифікатор повідомлення	Термін і частота видачі	Отримувач
Значення функції схожості, розраховане на основі одного оголошення, що переглядається	Кожного разу при обчисленні функції схожості	Програмний код
Значення функції схожості, розраховане на основі групи оголошень, що переглядаються	Кожного разу при обчисленні функції схожості	Програмний код
Елемент користувацького інтерфейсу у якому розміщуються рекомендовані оголошення	Кожного разу при пошуку оголошень у веб-додатку	Користувач

4 ПРОЕКТУВАННЯ СИСТЕМИ

4.1 Опис системних вимог

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 нерухомості. Тому створена стратегія рекомендацій буде застосовуватися для веб-сервісу з пошуку нерухомості. Рекомендаційна система буде однією з обчислювальних частин цієї інформаційної системи, буде повністю в неї інтегрована та буде спеціалізуватися на пропонуванні користувачу оголошень, які схожі на ті, що він переглядає у поточний момент.

Сервіс призначений для створення і управління оголошеннями про купівлю-продаж нерухомості. 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.

4.2 Опис функціональних вимог

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 4.1 – 4.7.



Рисунок 4.1 – Концептуальна діаграма інформаційної системи

На рисунку 4.1 представлена концептуальна діаграма системи. На вхід подаються «Дані користувача», головна функція – «Система публікації об’яв з продажу-купівлі нерухомості». На виході ми отримуємо результат пошуку повідомлень, результат роботи рекомендаційної системи, результат створення оголошення 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 4.2 представлена декомпозиція функції «Система публікації оголошень про продаж-купівлю нерухомості» системи. Дану функцію ми декомпозуємо на наступні функції:

- реєстрація;
- створення оголошення;
- повідомлення про помилку;
- публікація;
- пошук оголошень.

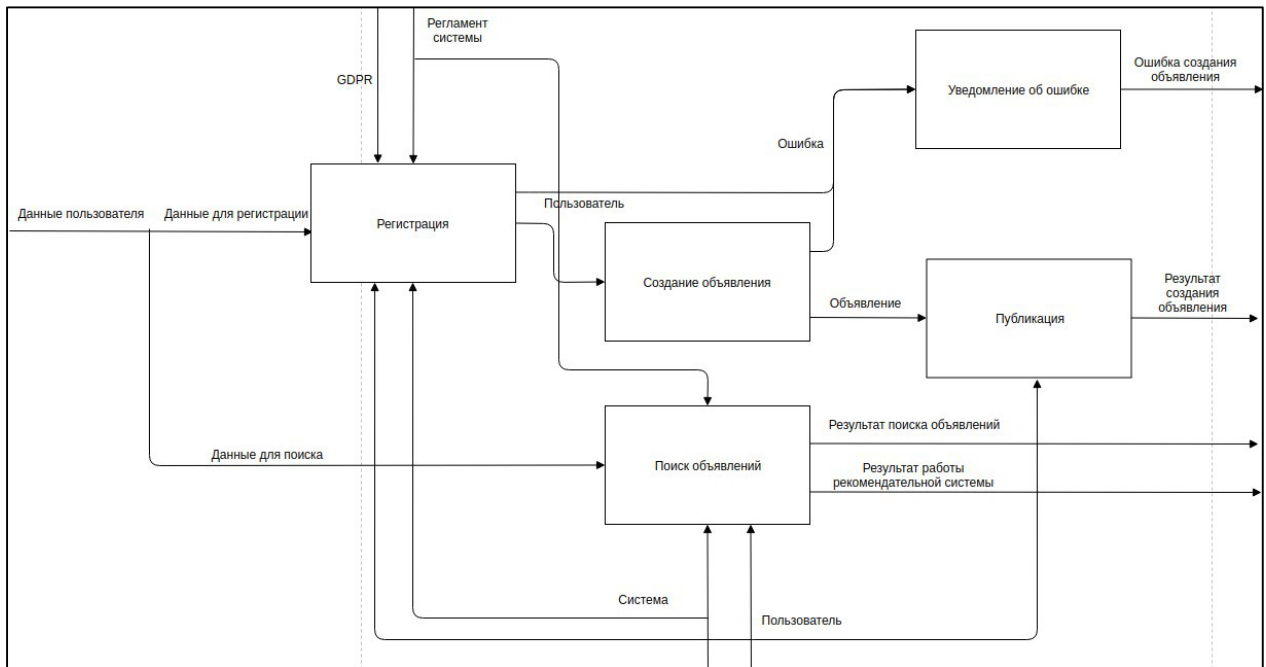


Рисунок 4.2 - Діаграма декомпозиції, що представляє функції системи

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 оголошення 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 4.3 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».

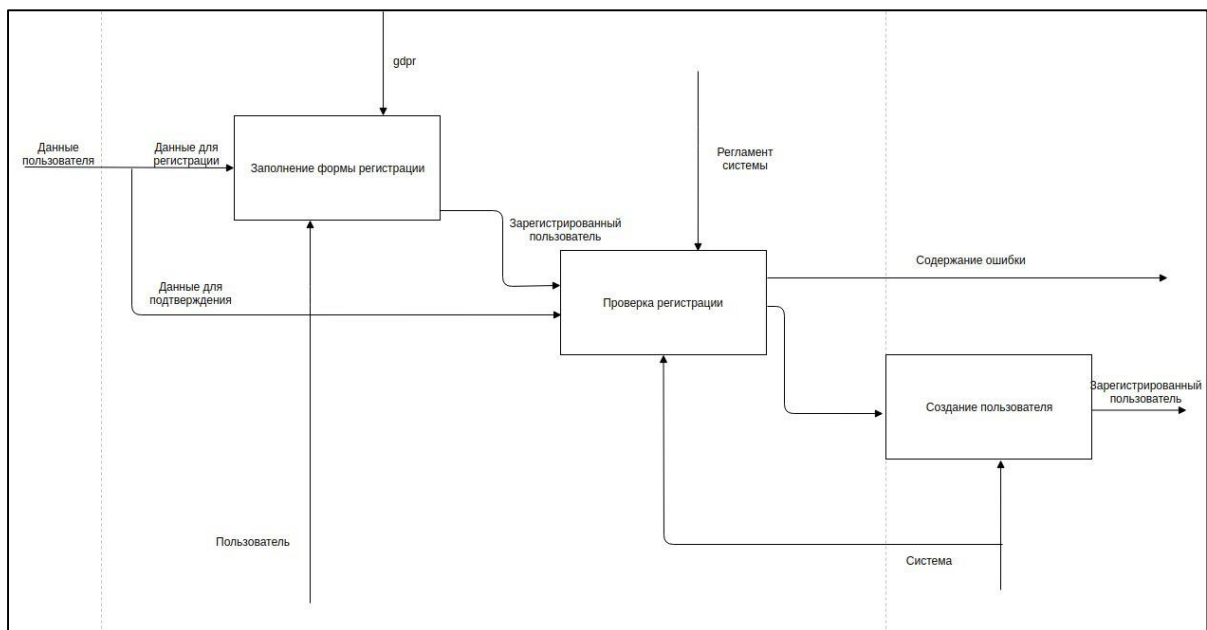


Рисунок 4.3 - Діаграма декомпозиції функції «Реєстрація» системи

На рисунку 4.4 представлена декомпозиція функції «Створення оголошення» системи. Ця функція декомпозується на 2 функції:

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

На вхід першої функції надходять дані про акаунт і зміст оголошення. Далі відбувається перевірка наданих даних з регламентом системи та на коректність введених даних. Якщо перевірка пройшла успішно на виході отримуємо підготовлені дані.

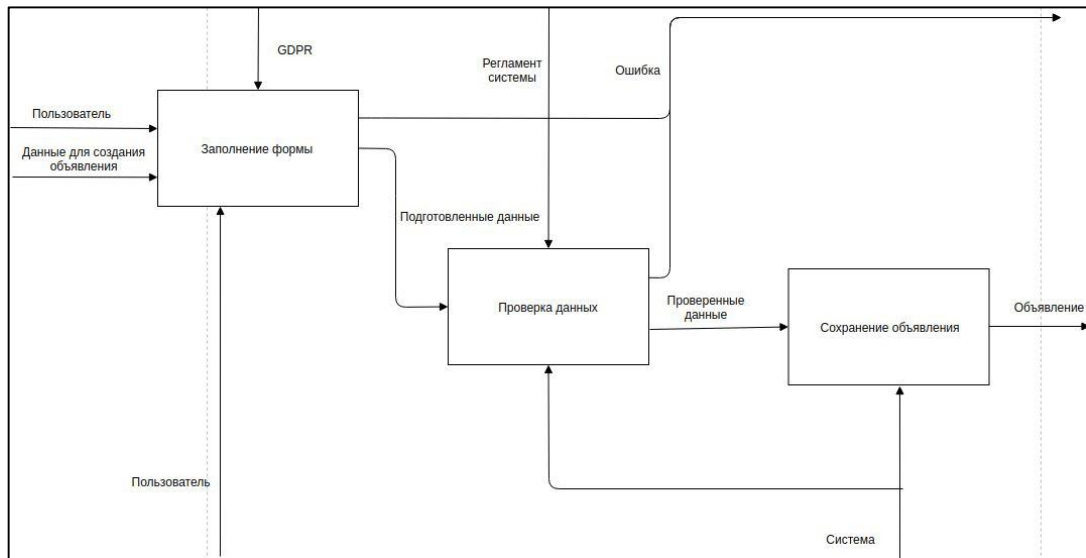


Рисунок 4.4 - Діаграма декомпозиції функції «Створення оголошення» системи

На рисунку 4.5 представлена діаграма декомпозиції функції «Пошук оголошень» системи. Функція складається з наступних функцій:

- встановлення необхідних обмежень;
- використання рекомендаційної системи;

Щоб встановити необхідні обмеження, необхідно ввести дані для пошуку

об'яв у пошукові фільтри. На виході будуть результати пошуку за обраними обмеженнями. Для роботи механізму рекомендаційної системи необхідно на вхід подати інформацію про характеристики активного оголошення, що обране у результаті пошуку.

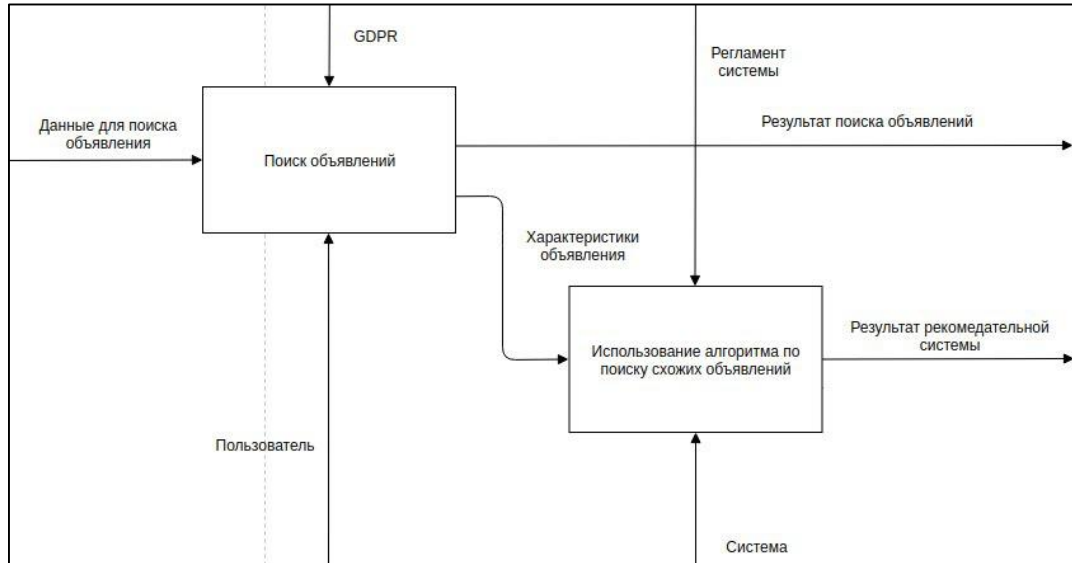


Рисунок 4.5 - Діаграма декомпозиції функції «Пошук оголошень» системи

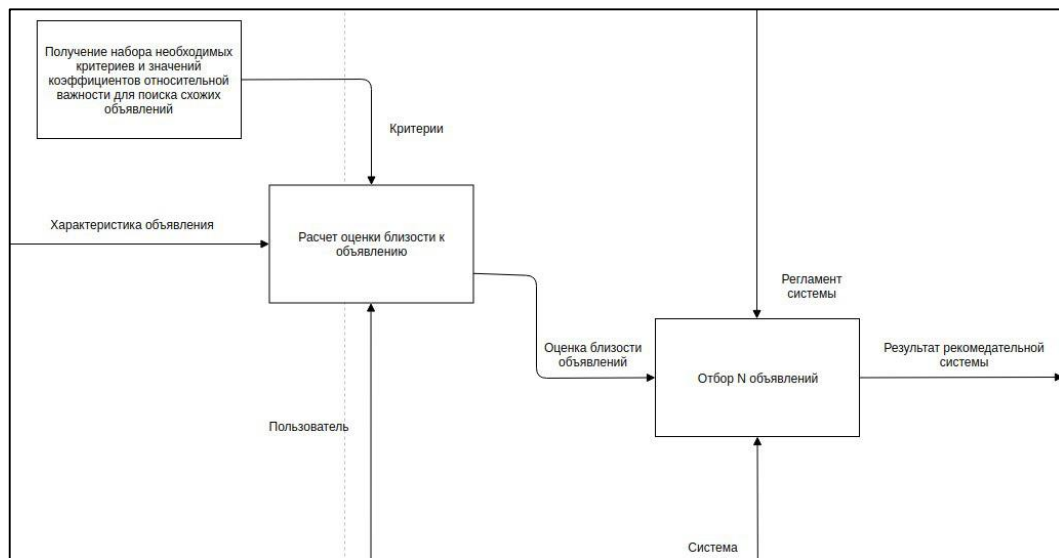


Рисунок 4.6 - Діаграма декомпозиції функції «Застосування РС»

На рисунку 4.6 представлена діаграма декомпозиції функції «Застосування рекомендаційної системи». Для її виконання спочатку необхідно отримати набір критеріїв, що описують заданий тип об'яви та значення їх відносної важливості, характеристику еталонного оголошення, та характеристики всіх оголошень 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 оголошень за типом нерухомості;
- редагування оголошень;
- написання відгуків у оголошенні;
- редагування профілю користувача;
- рейтинг у профілю;
- відображення місцезнаходження у оголошенні;
- відправка повідомлення на пошту користувачеві;
- фільтрація оголошень за різними критеріями;
- надання рекомендацій;
- відображення помилок.

4.2 Опис архітектури системи

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.

5 РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Вибір СУБД

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.

5.2 Опис розробленої БД

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 5.1

Пояснення зв'язків між сутностями в побудованій моделі даних наведені в таблиці 5.2.

Таблиця 5.1 – Сутності бази даних та їх опис

Ім'я сутності	Опис
Ad	Інформація про об'яву
PhotoAd	Фотографії об'яв
City	Список міст
Area	Список районів
Street	Список вулиць
User	Інформація про користувача
RecomendationScores	Результати розрахунків оцінок близькості для оголошень
ConfirmRegistration	Список аккаунтів, які очікують підтвердження за допомогою електронного листа
UserVotes	Дані про відгуки користувачів
PhotoUser	Фотографії користувачів
TypeProperty	Список статусів доступних типів об'яв

Таблиця 5.2 – Типи зв'язків в побудованій схемі даних

№	Тип сутності	Тип зв'язку	Тип сутності	Кардинальність
1	Ad	З'єднанано	PhotoAd	1:M
2	City	З'єднанано	Ad	1:M
3	Area	З'єднанано	Ad	1:M
4	Street	З'єднанано	Ad	1:M
5	User	З'єднанано	Ad	1:M
6	TypeProperty	З'єднанано	Ad	1:M
7	User	З'єднанано	ProfileScores	1:1
8	User	З'єднанано	ConfirmRegistration	1:1

Продовження таблиці 5.2

№	Тип сутності	Тип зв'язку	Тип сутності	Кардинальність
9	User	З'єднано	UserVotes	1:M
10	User	З'єднано	PhotoUser	1:M
11	City	З'єднано	Area	1:M
12	Area	З'єднано	Street	1:M

Схема бази даних зображена на рисунку 5.1.

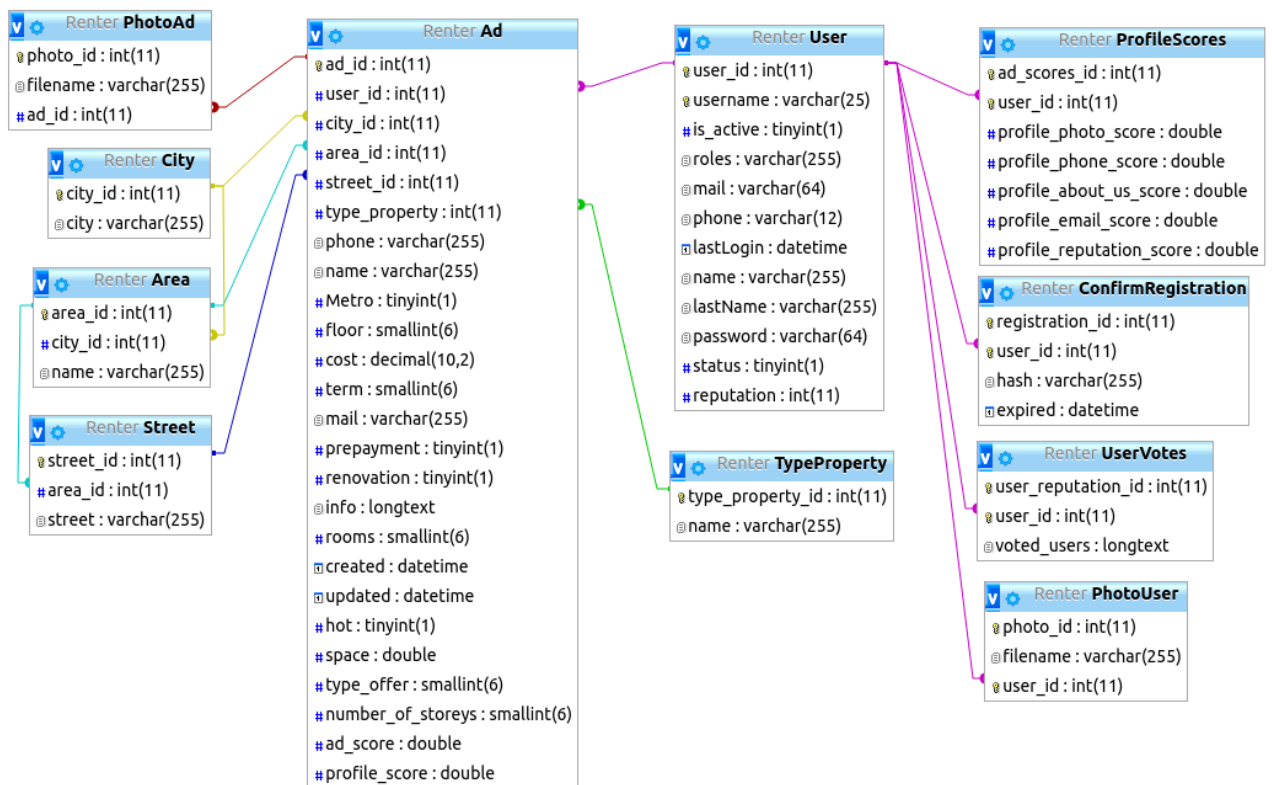


Рисунок 5.1 - Схема бази даних

6 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЕКСПЕРИМЕНТАЛЬНИЙ АНАЛІЗ АЛГОРИТМУ СТВОРЕННЯ РЕКОМЕНДАЦІЙ

6.1 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].

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.

6.2 Опис алгоритмів інформаційної системи

Нижче наведено опис та схеми роботи деяких основних алгоритмів сервісу.

На рисунку 6.1 зображено повний цикл для створення оголошення. Для того, щоб додати нове оголошення, 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 тип оголошення та заповнює усі необхідні поля, та тисне на кнопку «Додати». Якщо все зроблено вірно, то буде додане нове оголошення.

На рисунку 6.2 зображено схему алгоритму пошуку оголошень.

Щоб знайти необхідне оголошення, користувач має обрати необхідний йому тип об'яви та завдати критерії пошуку. Серед наведених об'яв обрати необхідну.

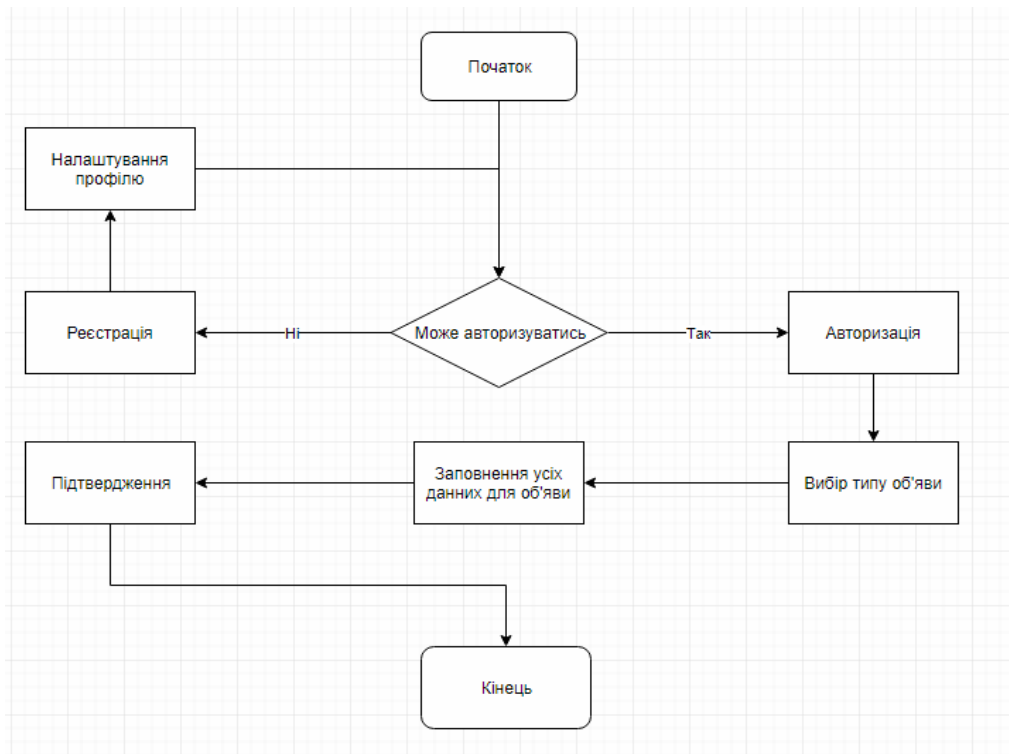


Рисунок 6.1 – Схема алгоритму створення оголошення

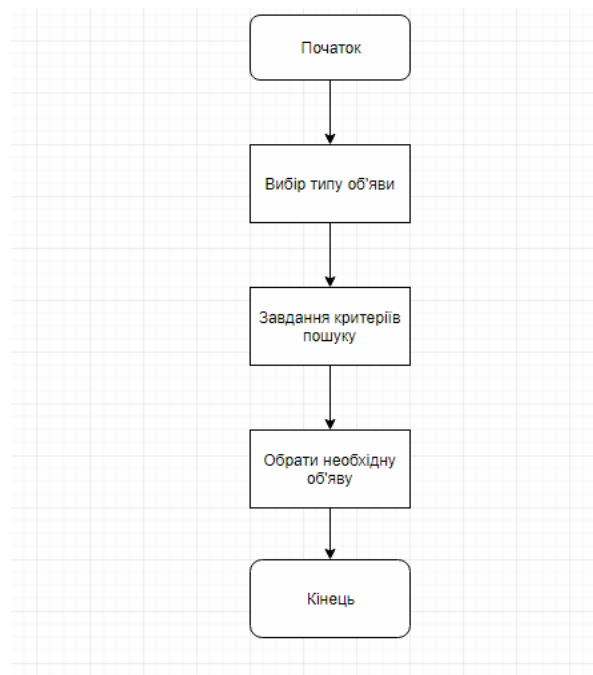


Рисунок 6.2 – Схема алгоритму для пошуку оголошень

На рисунку 6.3 зображено спрощену схему роботи рекомендаційної системи (РС).

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

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

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

При виконанні другого варіанту умови, останні n переглянутих оголошень (що обираються за методом «ковзаючого вікна») становлять «ідеальну точку», потім за формулами (2.3) та (2.5) відбувається розрахунок оцінок близькості інших оголошень до кожного з оголошень, які становлять центр, а далі за формулою (2.4) розраховується результуюча оцінка близькості, стосовно обраного центру та обирається k переможців.

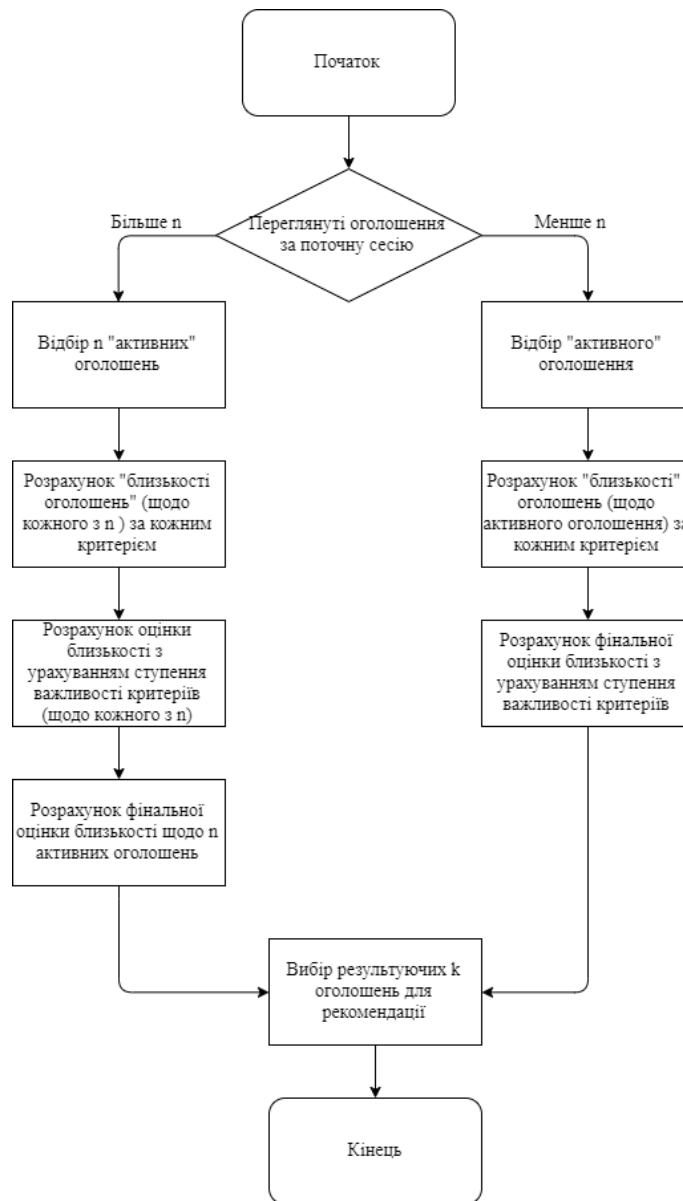


Рисунок 6.3 – Схема алгоритму роботи рекомендаційної системи

6.3 Експериментальний аналіз та наочні результати роботи РС

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 оголошень використовуються наступні критерії та їх відносна важливість:

- адреса – 0.3;
- адміністративний район;
- кількість кімнат – 0.15;
- ціна – 0.20;
- поверх – 0.025;
- згальна площа – 0.03;
- житлова площа -0.02;
- площа кухні – 0.05;
- тип квартири 0.025;
- тип стін – 0.025;
- тип пропозиції (від власника чи посередника) – 0.05;
- кількість поверхів – 0.05;
- площа ділянки 0.1;
- близькість до метро – 0.025;
- передплата – 0.

Значення параметрів для розрахунку близькості за формулою 2.5 для кожного критерію, що представлений у кількісній шкалі описані у таблиці 6.1.

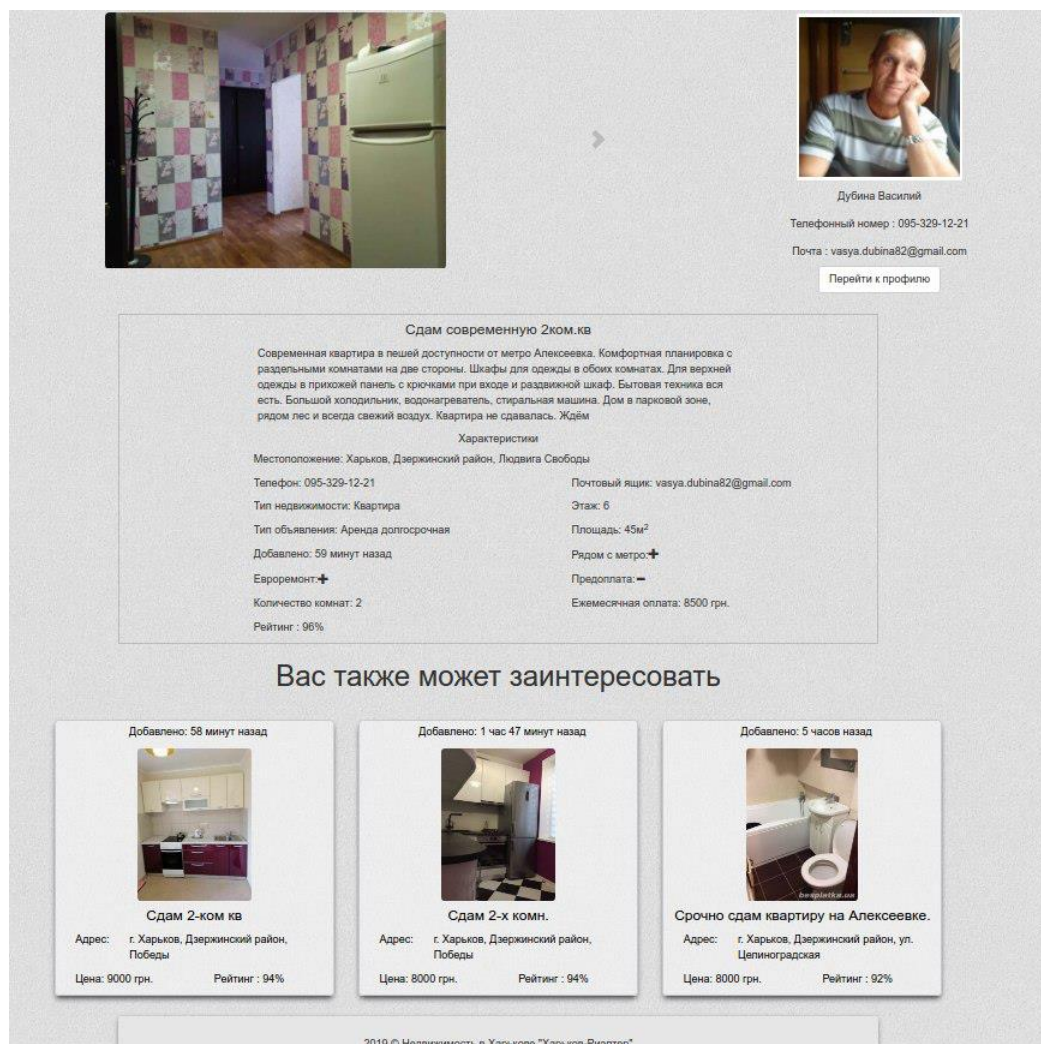
Таблиця 6.1 – Значення параметрів для розрахунку оцінки близькості

Назва критерію, одиниці вимірювання	Назви параметрів		
	Offset	Scale	Decay
Адреса (Геолокація), км	1	5	0,5
К-сть кімнат, шт	0	1	0,5
Ціна, грн	4% (відмінність від ціни)	15% (відмінність від ціни)	0,5
Поверх,	0	2	0,5
Загальна площа, м ²	6%	18%	0,5
Житлова площа, м ²	5%	12%	0,5
Площа кухні, м ²	4%	12%	0,5
К-сть поверхів, шт	0	1	0,5
Площа ділянки, м ²	5%	15%	0,5

На рисунках 6.4 – 6.7 можна побачити результати роботи рекомендаційної системи. Рисунки 6.4 – 6.5 відображають прогнози щодо одного оголошення, тому можна спостерігати невелику різницю у значеннях основних критеріїв, наприклад ціни, адреси та кількості кімнат. Тобто система рекомендує близькі, «шаблонні» оголошення, що добре відповідають відкритому оголошенню.

На рисунках 6.6 – 6.7 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 оголошення, що значно менше схожі на його попередню активність, проте також залишилося оголошення, що більше схоже на попередні рекомендації, це означає, що попередня активність, ще не повністю «стерта», або ж користувача цікавлять декілька варіантів, які він шукає паралельно тому різні оголошення і потрапляють до рекомендацій.



The screenshot displays a user profile for 'Дубина Василий' (Dubina Vasily) with contact information: phone 095-329-12-21 and email vasya.dubina82@gmail.com. Below the profile is a detailed listing for a modern 2-room apartment, including a description, location (Kharkiv, Dzerzhynskiy district), and key features like area (45m²) and monthly rent (8500 UAH). At the bottom, there are three smaller listings for similar properties, each with a photo, title, address, price, and rating.

Сдам современную 2ком.кв

Современная квартира в пешей доступности от метро Алексеевка. Комфортная планировка с раздельными комнатами на две стороны. Шкафы для одежды в обеих комнатах. Для верхней одежды в прихожей панель с крючками при входе и застёгнутой шкафа. Бытовая техника вся есть. Большой холодильник, водонагреватель, стиральная машина. Дом в парковой зоне, рядом лес и всегда свежий воздух. Квартира не сдавалась. Ждём

Характеристики

Местоположение: Харьков, Дзержинский район, Людвиг Свободы

Телефон: 095-329-12-21

Почтовый ящик: vasya.dubina82@gmail.com

Тип недвижимости: Квартира

Этаж: 6

Тип объявления: Аренда долгосрочная

Площадь: 45м²

Добавлено: 59 минут назад

Рядом с метро: +

Евроремонт: +

Предоплата: -

Количество комнат: 2

Ежемесячная оплата: 8500 грн.

Рейтинг: 96%

Вас также может заинтересовать

Добавлено: 58 минут назад

Сдам 2-ком кв

Адрес: г. Харьков, Дзержинский район, Победы

Цена: 9000 грн. Рейтинг: 94%

Добавлено: 1 час 47 минут назад

Сдам 2-х комн.

Адрес: г. Харьков, Дзержинский район, Победы

Цена: 8000 грн. Рейтинг: 94%

Добавлено: 5 часов назад

Срочно сдам квартиру на Алексеевке.

Адрес: г. Харьков, Дзержинский район, ул. Целиноградская


Цена: 8000 грн. Рейтинг: 92%

2019 © Недвижимость в Харькове "Харьков-Ризтлер"

Рисунок 6.4 – Активне оголошення

Вас также может заинтересовать


Добавлено: 58 минут назад



Сдам 2-ком кв

Адрес: г. Харьков, Дзержинский район, Победы
Цена: 9000 грн. Рейтинг: 94%


Добавлено: 1 час 47 минут назад



Сдам 2-х комн.

Адрес: г. Харьков, Дзержинский район, Победы
Цена: 8000 грн. Рейтинг: 94%


Добавлено: 5 часов назад




Срочно сдам квартиру на Алексеевке.

Адрес: г. Харьков, Дзержинский район, ул. Целиноградская
Цена: 8000 грн. Рейтинг: 92%

Рисунок 6.5 – Рекомендації до активного оголошення





Дибров Иван
Телефонный номер : 095-382-82-88
Почта : dibrov_i2017@gmail.com
[Перейти к профилю](#)

Сдается квартира


Квартира после евроремонта, новая сантехника, бытовая техника, мебель. Абсолютно все удобства для комфортного проживания.

Характеристики

Местоположение: Харьков, Шевченковский район, 23 августа	
Телефон: 095-382-82-88	Почтовый ящик: vasya.dubina82@gmail.com
Тип недвижимости: Квартира	Этаж: 4
Тип объявления: Аренда долгосрочная	Площадь: 52м ²
Добавлено: 2 часа назад	Рядом с метро: +
Евроремонт: +	Предоплата: -
Количество комнат: 2	Ежемесячная оплата: 8900 грн.
Рейтинг : 89%	

Вас также может заинтересовать


Добавлено: 25 минут назад



Сдам 2-ком квартира на 23 августа

Адрес: г. Харьков, Шевченковский район, 23 августа
Цена: 9000 грн. Рейтинг: 89%


Добавлено: 1 час 47 минут назад



Сдам 2-х комн. кв на 23 августа

Адрес: г. Харьков, Шевченковский район, 23 августа
Цена: 8000 грн. Рейтинг: 91%

Добавлено: 2 часа назад



Сдам современную 2ком.кв

Адрес: г. Харьков, Дзержинский район, ул. Целиноградская
Цена: 8500 грн. Рейтинг: 96%

Рисунок 6.6 – Активне оголошення

Вас также может заинтересовать

<p>Добавлено: 25 минут назад</p>  <p>Сдам 2-ком квартира на 23 августа</p> <p>Адрес: г. Харьков, Шевченковский район, 23 августа</p> <p>Цена: 9000 грн. Рейтинг : 89%</p>	<p>Добавлено: 1 час 47 минут назад</p>  <p>Сдам 2-х комн. кв на 23 августа</p> <p>Адрес: г. Харьков, Шевченковский район, 23 августа</p> <p>Цена: 8000 грн. Рейтинг : 91%</p>	<p>Добавлено: 2 часа назад</p>  <p>Сдам современную 2ком.кв</p> <p>Адрес: г. Харьков, Дзержинский район, ул. Целиноградская</p> <p>Цена: 8500 грн. Рейтинг : 96%</p>
---	---	--

Рисунок 6.7 – Рекомендації до активного оголошення

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. Стало зрозуміло, що проблема створення рекомендаційних систем зі складними предметними сферами є дуже актуальною, бо жодна з існуючих і добре досліджених стратегій не підходить для них, адже вони потребують знань про уподобання окремо взятого користувача чи групи користувачів. Для сервісів, які мають високу частоту оновлення елементів та/або низьку частоту їх перегляду, створення такої бази знань, на основі якої можна було б давати релевантні рекомендації, є складною задачею, бо такі особливості функціонування подібних сервісів тягнуть за собою непрогнозованість уподобань користувачів та велику розрідженість матриць оцінок елементів.

Після аналізу основних стратегій по створенню рекомендацій на предмет їх застосування для вирішення поставленої задачі, були виділені їх сильні та слабкі сторони, а також відібрані аспекти та ідеї, на основі яких була розроблена стратегія рекомендацій, яка підходить для рекомендування елементів зі складних предметних сфер. Ця стратегія базується на контентних рекомендаціях та рекомендаціях на основі знань. Вона створює рекомендації на основі лише опису товарів, слідкуючи при цьому, за активністю користувачів та поступово змінюючи вектор рекомендацій, якщо уподобання користувача змінюються за час перегляду, при цьому стратегія не передбачає збереження інформації про уподобання користувача за попередні проміжки часу. Це зроблено через особливості предметної області, бо низька частота переглядів не гарантує створення релевантної бази знань, адже існує велика імовірність того, що уподобання користувачів будуть відрізнятися при кожному такому перегляді.

У якості прикладу складної предметної сфери була обрана сфера нерухомості. Тому у якості об'єкту дослідження були обрані системи, що спеціалізуються на пошуку нерухомості, був проведений їх аналіз та виділені їх основні можливості та принципи функціонування. Була вирішена задача проектування інформаційного забезпечення однієї з таких систем та виконана програмна реалізація додатку, що спеціалізується на пошуку нерухомості з використанням, реалізацією та впровадженням розробленої стратегії по створенню рекомендацій. Наочні результати роботи розробленої рекомендаційної системи, показали, що вона здатна створювати релевантні рекомендації і може використовуватися для різноманітних сервісів, що охоплюють складні предметні сфери.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Jannach D., Zanker M., Felfernig A., Friedrich G. Recommender Systems: An Introduction. Cambridge University Press (2010).
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. Р. Берк, Системы рекомендаций, основанные на знаниях, Энциклопедия библиотечных и информационных наук. С: 180-200, 2000.
14. Y. Salem, J. Hong, W. Liu. History-Guided Conversational Recommendation, World Wide Web (WWW), 2014.
15. А. Фелфернигу и Р. Бёрку, Системы рекомендаций на основе ограничений: Технологии и проблемы исследований. АСМ Международная конференция по электронной торговле. С: 17-26, 2008.
16. 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.
17. Salter, J.; Antonopoulos, N. (January 2006). CinemaScreen Recommender Agent: Combining Collaborative and Content-Based Filtering. IEEE Intelligent Systems.2006
18. 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.
19. Windowing meyhod. [Электронный ресурс]. Режим доступа: <https://wiki.loginom.ru/articles/windowing-method.html>
20. Метод взвешенной суммы критериев в анализе многокритериальных решений. [Электронный ресурс]. Режим доступа: <https://bijournal.hse.ru/data/2013/10/11/1281327437/5>.
21. Маклаков С.В. Моделирование бизнес-процессов с ВРwin 4.0 – М.: Диалогмифи, 2002. – 151 с.
22. Черемных С.В., Семёнов И.О., Ручкин В.С. Структурный анализ систем: IDEF – технологии – М.:Финансы и статистика, 2003. – 208 с.

23. Wikipedia [Электронный ресурс]. Model-View-Controller. Режим доступа:<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>.

24. P. DuBois, MySQL Cookbook: Solutions for Database Developers and Administrators. O'Reilly Media; 3 edition, 2012 – С. 866.

25. М. Зандстра PHP: Objects, Patterns, and Practice. «Ozon», 3-издание, 2011. – С. 576.

26. Wikipedia [Электронный ресурс]. JS. Режим доступа: <https://en.wikipedia.org/wiki/JavaScript>.