Міністерство освіти й науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ”
Катедра інформатики факультету інформатики

Кваліфікаційна робота

освітній ступінь – бакалавр

на тему: “**ДОСЛІДЖЕННЯ МЕТОДОЛОГІЇ РОЗРОБКИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ**”

Виконав студент
4-го року навчання
122 “Комп’ютерні Науки”
Макарець Андрій Олександрович
Керівник курсової роботи
ас. Курочкін А.В.
Рецензент ст. в. Салата К.В.
Кваліфікаційна робота захищена з оцінкою
_____
Секретар ЕК _____
(підпис)
“___” _____ 2023 року

Київ 2023

Міністерство освіти й науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри інформатики,

к.ф.-м.н.доц. Гороховський С.С.

(підпис)

„____”_____2010 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на кваліфікаційну роботу

студенту Макарцю Андрію факультету Інформатики 4 року навчання

ТЕМА «ДОСЛІДЖЕННЯ МЕТОДОЛОГІЇ РОЗРОБКИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ»

Зміст ТЧ до кваліфікаційної роботи:

Індивідуальне завдання

Зміст

Анотація

1 Вступ

2 Підходи до реалізації рекомендаційних систем

3 Приклади застосування РС

4 Порівняльний аналіз підходів

5. Практична частина

6. Висновки

Список літератури

Дата видачі „___” _____ 2010 р. Керівник Курочкін А.В. _____

(підпис)

Завдання отримав _____

(підпис)

Графік підготовки кваліфікаційної роботи до захисту

№ п/п	Назва етапу роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на кваліфікаційну роботу	10.2022	
2.	Огляд літератури за темою роботи	02.2023	
3.	Написання вступу	02.2023	
4.	Написання основної текстової частини	03-05.2023	
5.	Написання програмної частини	07.05.2023	
6.	Надання роботи керівнику для перевірки	08.05.2023	
7.	Попередній захист	12.05.2023	
8.	Коригування роботи, перевірка на плагіат	22.05.2023	
9.	Захист кваліфікаційної роботи	30.05.2023	

ЗМІСТ

Анотація

Тема курсової роботи: дослідження методології розробки рекомендаційних систем

Студент Макарець Андрій Олександрович

Рік навчання, спеціальність, факультет: 4-й рік навчання (бакалаврська програма), комп’ютерні науки, факультет інформатики.

Науковий керівник Курочкін А.В.

Мета роботи полягає в аналізі наявних підходів до реалізації рекомендаційних систем, їх порівнянні та реалізації системи

Ключові слова: рекомендаційна система, користувач, елемент, контент-базована, колаборативна фільтрація, рекомендаційна корисність

1. ВСТУП

Рекомендаційні системи є важливим інструментом для бізнесу та користувачів, що надає персоналізовані рекомендації щодо товарів, послуг та інших об'єктів на основі відомостей про користувача та інших факторів. Розробка та вдосконалення рекомендаційних систем вимагає вивчення методології їхньої розробки та використання різноманітних алгоритмів для забезпечення оптимальної якості рекомендацій. У цій дипломній роботі буде проведено дослідження методології розробки рекомендаційних систем та її застосування в різних сферах, а також розглянуті найбільш популярні алгоритми та їхні переваги та недоліки. Результати дослідження допоможуть покращити якість рекомендаційних систем та забезпечити більш ефективну їхню розробку.

Зважаючи на висхідну кількість даних, які збираються та зберігаються, рекомендаційні системи стають все більш потужним та важливим інструментом для бізнесу та користувачів. Їхнє використання дозволяє ефективно розв’язувати завдання, пов'язані зі збільшенням продажів, збільшенням задоволеності користувачів та зменшенням витрат.

Однак, розробка рекомендаційних систем вимагає досить складної методології, що охоплює використання різноманітних методів машинного навчання, статистичних методів та експертних знань. Крім того, розробка рекомендаційної системи потребує розуміння бізнес-процесів та потреб користувачів, що додатково ускладнює процес розробки.

Актуальність:

- Системи рекомендацій відіграють важливу роль на таких популярних та відомих Інтернет-сайтах як Amazon.com, Yahoo, YouTube, Netflix, Last.fm та IMDb. Деякі з них ми розглянемо далі. Крім того, багато медіакомпаній зараз розробляють і розгортають РС як частину послуг, які вони надають своїм абонентам[3]

- У сучасному світі, де значна кількість даних генерується щоденно та ми стикаємося з обсягом інформації, що невпинно зростає, рекомендаційні системи забезпечують користувачам індивідуальні рекомендації на основі їх історії взаємодії з системою, що полегшує процес знаходження потрібної інформації та підвищує рівень задоволеності користувачів.

- Розробка та впровадження алгоритмів надання рекомендацій може суттєво збільшити прибутки інформаційних систем такого характеру, що було не раз доведено на практиці. Так, наприклад, 2/3 всіх фільмів, що було переглянули користувачі Netflix, було знайдено саме завдяки алгоритмам рекомендацій, а продажі товарів, рекомендованих алгоритмами Amazon, складають приблизно 35% всього доходу[6]

- Технологія вибору квитка в залізничному сполученні покладена на клієнта, тоді як автоматичний підбір може зекономити час та покращити результат(буде продемонстровано далі на прикладі розробленої системи).

Наслідком актуальности є той факт, що **у** вищих навчальних закладах по всьому світу, наприклад в Київському національному університеті імені Тараса Шевченка викладаються предмети з Рекомендаційних систем, навчальні посібники з РС дуже популярні на конференції з інформатики; публікуються книги щодо технологій РС[2];

Мета дослідження:

Розробка рекомендаційних систем має свої складнощі, такі як великий обсяг даних, проблеми з точністю рекомендацій, необхідність постійної аналітики та оновлення моделей. Тому дослідження методології розробки рекомендаційних систем має на меті забезпечити якість та ефективність роботи цих систем. Побудована система базуватиметься на означених побажаннях клієнта, а не історії, що попередньо існує, на відміну від більш звичних рекомендаційних підходів.

Об’єкт дослідження: розробка рекомендаційної системи з підбору залізничних квитків.

Предмет дослідження: методи та алгоритми розробки рекомендаційних систем, їх застосування.

2.ПІДХОДИ ДО РЕАЛІЗАЦІЇ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

2.1.Основні відомості

Рекомендаційні системи(далі РС) - це програмні рішення, які надають користувачам персоналізовані рекомендації стосовно продуктів, послуг або інших об'єктів, в яких вони можуть бути зацікавлені в певному контексті. Наприклад, рекомендаційна система може рекомендувати фільми, які вам можуть сподобатися, на основі ваших попередніх переглядів або популярності серед інших користувачів зі схожими інтересами.

Рекомендаційні системи - це інструменти, які допомагають розв’язати проблему інформаційного перенасичення, а самі - допомагають користувачам знайти відповідну інформацію в об'ємному масиві даних, доступних в мережі Інтернет.

Реалізація рекомендаційних систем - це складна задача, яка вимагає використання різних підходів та методів машинного навчання, обробки даних та аналізу. У цій роботі розглянемо основні підходи до реалізації рекомендаційних систем та їх переваги та недоліки.

Виникнення та розвиток рекомендаційних систем доводиться на кінець минулого століття[5].

Історія рекомендаційних систем почалася в 1990-х роках, коли науковці почали досліджувати методи автоматизованої рекомендації товарів в електронній торгівлі. Одним із перших важливих результатів у цьому напрямку було запровадження алгоритму колаборативної фільтрації (Collaborative filtering), який заснований на ідеях, що користувачі, які споживають подібні товари, ймовірно, мають схожі смакові уподобання.

Перші рекомендаційні системи почали з'являтися в 1990-х роках. У цей час інтернет був ще молодим та малорозвиненим, тому не було великої кількості даних, які можна було б використати для створення рекомендацій.

Найбільш відомим прикладом ранньої рекомендаційної системи є система, розроблена в 1994 році компанією Firefly Network, яка допомагала зберігати музичні композиції, що відповідали особистим музичним вподобанням користувача. Ця система аналізувала інформацію про те, які композиції сподобалися користувачеві раніше, та робила рекомендації на основі цих даних.

Згодом, з появою соціальних мереж, які дозволили користувачам поділитися своїми вподобаннями та відомостями про себе, стали доступні більш об'ємні дані, які можна було б використати для рекомендацій. Наприклад, у 2002 році була створена рекомендаційна система Amazon, яка використовувала інформацію про покупки користувачів для рекомендаційних товарів.

Пізніше науковці розробили інші методи для рекомендацій, такі як засновані на контенті (Content-based), які засновані на аналізі властивостей товарів і їх порівнянні з відомими смаковими уподобаннями користувача, або гібридні методи, які об'єднують у деяких підходах.

У своїй найпростішій формі персоналізовані рекомендації пропонуються у вигляді ранжованих списків предметів. Виконуючи це рейтингування, РС намагаються передбачити, які продукти чи послуги є найбільшслушними, виходячи з уподобань і обмежень користувача. Завершивши таке обчислювальне завдання, РС збирає від користувачів їхні переваги, які виражаються явно, скажімо оцінки продуктів, або випливають з інтерпретації дій користувача. Наприклад, РС може розглядати навігацію до конкретної сторінки продукту як неявний знак переваги товарів, показаних на цій сторінці

Розробка РС розпочалася з досить простого спостереження: люди часто покладаються на рекомендації, надані іншими, при прийнятті рутинних щоденних рішень[7].

2.2.Функції рекомендаційних систем

Основні функції рекомендаційних систем:

1.Збір даних: РС збирають дані про користувачів, їхній профіль, історію взаємодії з системою, відгуки, рейтинги, поведінку та інші параметри.

2.Аналіз даних: РС аналізують зібрані дані за допомогою алгоритмів машинного навчання та статистичних методів для визначення інтересів та поведінки користувачів.

3.Розуміння контексту: РС враховують контекст, такий як час дня, день тижня, локація та інші параметри, які можуть вплинути на вибір продукту або послуги.

4.Прогнозування: на основі зібраних даних та аналізу прогнозують, які продукти або послуги можуть бути корисними або сподобатися користувачам.

5.Ранжування: ранжують пропозиції на основі оцінок та ймовірності того, що користувач зробить покупку або взаємодію з продуктом.

6.Персоналізація: РС намагаються зробити пропозиції більш персоналізованими, враховуючи індивідуальні потреби, інтереси та поведінку кожного користувача.

7.Апдейт пропозицій: постійно оновлюють свої пропозиції на основі нових даних та змін в поведінці користувачів.

8.Підтримка взаємодії: можуть допомогти користувачам знайти співробітників або партнерів на основі їхніх інтересів та потреб.

9.Забезпечення релевантности: намагаються забезпечити максимальну релевантність пропозицій користувачам для збільшення їхньої задоволеності та зменшення кількості відмов.

10.Регулювання рекомендацій: можуть дозволяти користувачам вибирати, які типи пропозицій вони бажають отримувати та наскільки часто.

11.Захист даних: мають забезпечувати безпеку та захист даних користувачів, які вони збирають та обробляють.

Конверсія – це дія, яка має для вас цінність, як-от здійснення онлайн-покупки або зв'язку з вашою компанією через мобільний телефон, що відбувається після взаємодії з рекламою чи точною інформацією (наприклад, натискання текстової реклами чи перегляду відеореклами). Коефіцієнт конверсії розраховується як відношення кількості конверсій до загальної кількості взаємодій. [9].

Системи рекомендацій можна використовувати в широкому діапазоні програм, включаючи електронну комерцію, соціальні мережі, онлайн-рекламу та розважальні платформи. Деякі з ключових переваг використання систем рекомендацій – це збільшення залученості користувачів, вищі коефіцієнти конверсії, покращення задоволеності клієнтів і зниження витрат на пошук.

Таким чином рекомендаційні системи допомагають підлаштовувати контент під юзерів та бути конкурентним у відповідній галузі послуг.

2.3 Рекомендаційна корисність

Щоб реалізувати свою основну функцію, визначити корисні елементи для користувача, РС має передбачити, що він вартий рекомендації. Узагальнюючи, для цього система повинна мати можливість передбачити корисність деяких з них або, принаймні, порівняти корисність деяких елементів, а потім вирішити, які елементи рекомендувати на основі цього порівняння. Крок передбачення може не бути чітким[1].

Щоб проілюструвати крок передбачення розглянемо найпростіший неперсоналізований алгоритм рекомендацій суть якого полягатиме в тому, щоб

рекомендувати найпопулярніші пісні, для простоти припускаючи узагальнено відсутність іншої інформації. За корисність візьмемо популярність пісні. Дійсно, чим популярніша пісня, тим ймовірніше вона сподобається користувачеві. Такий алгоритм є прикладом алгоритму **колаборативної фільтрації.**

Нехай функція R(u,i)- ступінь корисності елемента i для користувача u. Тоді завдання алгоритму РС для всіх пар (u,i) передбачити значення ф-ї R, іншими словами обчислити $\hat{R}(u,i)$. Така дія зазвичай виконується для певного користувача, званого активним. Отже, обчисливши це передбачення для користувача u щодо набору елементів, тобто $\hat{R}(u,i_0),\ldots, \hat{R}(u,i_n)$ система рекомендуватиме елементи $ij_0,\ldots,ij_к$ ($k \leq n$).

Деякі рекомендаційні системи не повністю оцінюють корисність перш ніж дати рекомендацію, тобто вони можуть застосувати деякі евристики, щоб припустити, що елемент корисний для користувача. Це характерно, наприклад, для орієнтованих на знання (knowledge-based) систем. Ці передбачення корисності обчислюються за допомогою спеціальних алгоритмів, що будуть розглянуті далі використовувати різного роду знання про користувачів, предмети та саму корисну функцію.

2.4 Класифікація основних підходів

2.4.1 Контент-базовані

Контент-базовані(Content-based): система навчена рекомендувати елементи, подібні до тих, що їм користувач надавав перевагу в минулому. Схожість предметів обчислюється на основі їх спільності - характеристики, пов'язаної з порівнянням предметів. Наприклад, якщо поставив фільму високу оцінку, який належить до комедійного жанру, то система може навчитись навчитися рекомендувати інші фільми цього жанру. Для цього використовуються методи обробки природньої мови, аналізу зображень та інші

технології. Далі ми розглянемо деякі реалізації подібних систем, а в 4 розділі – переваги та недоліки.

Системи, що реалізують підхід рекомендацій на основі вмісту, аналізують набір документів та/або описи предметів, попередньо оцінених користувачем, і створюють модель або профіль інтересів користувача на основі особливостей оцінюваних цим користувачем об’єктів.

Профіль є структурованим представленням інтересів користувача, прийнятим для рекомендації нових цікавих речей. Процес рекомендації в основному полягає в порівнянні атрибутів профілю користувача з атрибутами об’єкта вмісту. Результатом є оцінка релевантності, яка відображає рівень інтересу користувача до цього об’єкта.Якщо профіль точно відображає вподобання користувача, це має величезну перевагу для ефективности процесу доступу до інформації. Наприклад, його можна використовувати для фільтрування результатів пошуку, вирішуючи, чи зацікавлений користувач у певній вебсторінці, чи ні, і, у негативному випадку, запобігання її відображенню[10].

Процес рекомендацій виконується в три кроки, кожен з яких обробляється окремим компонентом системи:

-Аналізатор змісту: якщо інформація не має структури (наприклад, текст), необхідна деяка попередня обробка даних для отримання відповідної структурованої інформації. Основна функція компонента - це представлення вмісту елементів, що надходять із джерел інформації у формі, придатній для наступних етапів обробки.

- Profile learner (вивчальник профілю): цей модуль збирає дані, що відповідають уподобанням користувача і намагається узагальнити ці дані, щоб побудувати профіль користувача. Зазвичай стратегія узагальнення реалізується за допомогою технік машинного навчання, які здатні вивести модель інтересів користувача, починаючи з елементів, які сподобалися або не сподобалися раніше. Наприклад, PROFILE LEARNER рекомендувача веб-сторінок може

реалізувати релевантний метод зворотного зв’язку, у якому навчальна техніка поєднує вектори позитивних і негативних прикладів у прототип-вектор, що представляє профіль користувача. Навчальними прикладами є вебсторінка, на які було отримано позитивний чи негативний відгук від користувача[11];

- **Компонент фільтрування**(filtering component) — цей модуль використовує профіль користувача, щоб пропонувати релевантні елементи, порівнюючи представлення профілю з елементами. Результатом є двійкове або безперервне оцінювання релевантності (обчислене за допомогою деяких показників подібності), в останньому випадку утворюється ранжований список потенційно цікавих предметів.

2.4.2 Колаборативна фільтрація

Найпростіша та оригінальна реалізація цього підходу рекомендує користувачеві елементи, які інші користувачі з подібними смаками вподобали в минулому. Схожість у смаках двох користувачів розраховується на основі схожість історії рейтингів користувачів (тобто як вони оцінювали елементи в минулому). Спільна фільтрація є найпопулярнішою і широко застосовуваною технікою в РС[12]. Одним із методів реалізації є метод на основі сусідства[1]. Методи сусідства зосереджені на зв’язках між елементами або, альтернативно, між користувачами. Підходи на основі спільної фільтрації використовують інформацію про інтереси користувачів та їхню історію взаємодії з системою для рекомендації продуктів. Наприклад, якщо користувач А та користувач Б мають схожі інтереси та обирали подібні продукти в минулому, то система може рекомендувати користувачу А продукти, які обрав користувач Б.

Існує два основних типи спільної фільтрації: на основі користувачів і на основі елементів(user-based and item-based).

User-based Collaborative Filtering:

Колаборативна фільтрація на основі користувачів працює, знаходячи користувачів, схожих на активного користувача, на основі їхніх оцінок або вподобань, і рекомендує елементи, які цим подібним користувачам сподобалися, але з якими активний користувач ще не взаємодіяв. Подібність між користувачами обчислюється за допомогою метрики відстані, наприклад косинусної подібності або коефіцієнта кореляції Пірсона. Потім рекомендовані елементи сортуються на основі їхніх прогнозованих оцінок, які обчислюються з використанням середньозважених оцінок подібних користувачів.

$$\hat{r}_{u,i} = \frac{\sum_{v \in N_u} w_{u,v} \cdot r_{v,i}}{\sum_{v \in N_u} w_{u,v}}$$

Рис. 2.1

Де:

r(v,i) рейтинг елемента j для користувача u;

w(u,v) подібність між користувачами u та v;

N_u- множина користувачів, подібних до користувача u;

$\hat{R}_{(u,i)}$ -прогнозований рейтинг елемента i для користувача u;

Колаборативна фільтрація на основі елементів працює, знаходячи елементи, схожі на ті, з якими взаємодіяв активний користувач, на основі їх спільної зустрічальності або подібності, і рекомендує ці схожі елементи. Подібність між елементами обчислюється за допомогою тих же метрик відстані. Наприклад, за косинусною подібністю[13] W(A,B)=:

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

Рис. 2.2

$$\hat{r}_{u,i} = \frac{\sum_{j \in I_u} w_{i,j} \cdot r_{u,j}}{\sum_{j \in I_u} w_{i,j}}$$

Рис. 2.3

r(u,i) рейтинг елемента і для користувача u;

w(i,j) подібність між елементами і та j;

I_u – множина елементів, з якими взаємодіяв користувач u

$\hat{R}_{(u,i)}$ -прогнозований рейтинг елемента і для користувача u;

Спільна фільтрація як на основі користувачів, так і на основі елементів має свої переваги та недоліки. Спільна фільтрація на основі користувачів краще підходить для випадків, коли користувачі мають рідкісні оцінки, а кількість користувачів менша за кількість елементів. З іншого боку, спільне фільтрування на основі елементів краще підходить для випадків, коли елементи мають рідкісні оцінки, а кількість елементів менша за кількість користувачів.

В останні роки спостерігається значний дослідницький інтерес до проблеми виявлення спроб провести атаки на системи рекомендацій, наприклад, для отримання несанкціонованої вигоди з просування певного продукту.

З наведених формул випливає, що на прогнозовану оцінку сильніше впливають елементи, ступінь подібності яких є більшою, і менше впливають елементи, ступінь подібності яких нижче. Цей висновок привів до спрощеного алгоритму – алгоритму k-найближчих сусідів – який часто використовується для прогнозування оцінки[6]. Концепція алгоритму k найближчих сусідів відбувається в тому, щоб обмежити кількість елементів (користувачів), що враховуються, до k. Алгоритм k-means зменшує кількість обчислень, але зберігає точність, подібну до бажаних результатів. Цей алгоритм буде розглянуто в пункті 2.5.

2.4.3.Демографічні РС

Цей тип системи рекомендує товари на основі демографічного профілю користувача. Припускається, що повинні бути різні рекомендації для різних демографічних ніш. Багато вебсайт використовують прості та ефективні рішення персоналізації на основі демографічних показників. Наприклад, користувачі надсилаються на певні вебсайт залежно від їхньої мови чи країни. Або пропозиції можуть бути налаштовані відповідно до віку користувача. Поки ці підходи були досить популярними в маркетинговій літературі, було відносно мало власне дослідження РС демографічних систем[14].Отже, рекомендаційний алгоритм ідентифікує користувачів, демографічно схожих на u. Передбачення для рекомендованого елемента екстраполюється з того, як подібні користувачі оцінили цей продукт, і наскільки вони схожі на u.

Наприклад: «Для дітей дуже привабливе, вимагає низьких знань, вимагає певної серйозності та візит досить короткий. Це дуже привертає увагу, і має

високу історичну цінність». У цій системі рекомендації запропоновані як структурований огляд, класифікація місць для відвідування відповідно до їх придатності для різних типів мандрівників (наприклад, діти, інваліди). Потім користувачі можуть додати ці пункти до свого маршруту[1].

2.4.4.РС, базовані на знаннях

Системи, засновані на знаннях, рекомендують елементи на основі знань конкретної області про те, як певні функції предмета відповідають потребам і вподобанням користувачів і, зрештою, наскільки елемент корисний для користувача. Відомі системи рекомендацій, засновані на знаннях, базуються на конкретних випадках (кейсах). У цих системах функція подібности оцінює, наскільки користувачеві потрібно (опис проблеми) відповідати рекомендаціям (розв'язання проблеми). Тут оцінка подібності може бути прямо інтерпретованою як корисність рекомендації для користувача.

Системи на основі обмежень є ще одним типом базованих на знаннях РС. З точки зору знань, що використовуються, обидві системи схожі. Основна відмінність полягає в способі обчислення рішень.РС на основі випадків визначають рекомендації на основі показників подібності, тоді як РС, засновані на обмеженнях, переважно використовують попередньо визначені бази знань, які містять чіткі правила щодо того, як пов’язувати вимоги клієнтів із характеристиками предмета. У системі рекомендацій на основі знань створюється база знань, яка містить інформацію про рекомендовані елементи. Ця база знань може бути створена експертами, які мають глибоке розуміння елементів, або вона може бути автоматично згенерована за допомогою методів машинного навчання.

На початку впровадження системи, засновані на знаннях, зазвичай працюють краще за інші, але якщо вони не оснащені навчальними компонентами, то можуть бути перевершеними іншими методами, які,

наприклад, можуть використовувати взаємодію людини/комп'ютера, як колаборативна фільтрація[1].

Традиційні підходи до рекомендацій (колаборативне, контент-базоване) добре підходять для рекомендацій якісних і смакових продуктів, наприклад книги, фільми чи новини. Однак для таких речей, як автомобілі, комп'ютери, квартири чи фінансові послуги - не найкращий вибір. Наприклад, квартири не купують дуже часто, що робить неможливим збирати численні рейтинги для одного конкретного елементу (саме такі оцінки вимагають алгоритми колаборативних рекомендацій). Крім того, користувачі додатків на основі РС не будуть задоволені рекомендаціями базованими на багаторічних уподобаннях елементів. Технології рекомендацій, засновані на знаннях, допомагають подолати ці виклики використовуючи чіткі вимоги користувачів[15]. Саме оригінальне застосування системи такого типу буде мною запропоноване.

2.4.5.РС, базовані на спільноті

Цей тип системи рекомендує елементи на основі вподобань друзів користувачів. Цей прийом файно ілюструє приказка «Скажи мені, хто ти друзі є, і я скажу тобі хто ти». Докази свідчать про те, що люди схильні більше покладатися на рекомендації своїх друзів, ніж на рекомендації від подібних, але анонімних осіб. Це спостереження в поєднанні з популярністю відкритих соціальних мереж, що невпинно зростає, викликає збільшення інтересу до системи, базованих на спільноті, або, як їх ще називають, соціальних РС. Цей тип РС моделює та отримує інформацію про соціальні стосунки та вподобання друзів користувача. Рекомендація базується на оцінках, наданих друзями. Фактично ці РС ідуть в ногу з розвитком соціальних мереж.

Дослідження в цій галузі все ще перебувають на ранній стадії та результати щодо продуктивності систем неоднозначні. Наприклад, в [17] повідомляють, що загалом рекомендації, засновані на соціальних мережах, не

точніші, ніж ті, що випливають із традиційних CF підходів, за певними винятками.

Системи рекомендацій на основі спільнот можна розділити на два основні типи: соціальні та групові.

Соціальні рекомендаційні системи використовують аналіз соціальних мереж, щоб ідентифікувати користувачів, які схожі один на одного на основі їхніх соціальних зв'язків або взаємодії. Ці системи аналізують структуру соціальних мереж користувачів і дають рекомендації на основі поведінки та вподобань користувачів, які мають соціальні зв'язки. Наприклад, система соціальних рекомендацій щодо фільмів може давати рекомендації на основі оцінок і рецензій на фільми, зроблених друзями користувача або людьми в його соціальній мережі.

Системи рекомендацій на основі груп використовують алгоритми кластеризації для ідентифікації груп користувачів, які мають схожі переваги чи поведінку. Ці системи аналізують моделі поведінки та вподобання користувачів і групують їх разом на основі їх подібності. Потім система дає рекомендації на основі поведінки та вподобань групи. Наприклад, система групових рекомендацій щодо музики може давати рекомендації на основі поведінки слухачів і уподобань користувачів, які об'єднані разом, оскільки мають схожі музичні смаки.

2.4.6.Інші типи РС

Підходи на основі навчання з підкріпленням

Підходи на основі навчання з підкріпленням (reinforcement learning) використовують методи машинного навчання для навчання системи рекомендацій шляхом максимізації деякого функціонала винагороди. Наприклад, система може навчатися на підставі того, які продукти користувач купує після отримання рекомендації.

Системи рекомендацій на основі навчання з підкріпленням (RL) — це тип системи машинного навчання, яка вчиться рекомендувати предмети або дії на основі зворотного зв’язку, отриманого з середовища.

У контексті рекомендаційних систем середовище представляє користувачів та їхню взаємодію з системою, тоді як елементи або дії представляють продукти або послуги, які рекомендуються. Мета системи рекомендацій на основі RL полягає в тому, щоб максимізувати певний сигнал винагороди (наприклад, задоволеність користувачів або дохід), навчившись рекомендувати елементи, які призводять до вищих винагород.

У системі рекомендацій на основі RL система спочатку не знає, які елементи рекомендувати яким користувачам. Система починає з дослідження простору можливих рекомендацій і спостереження за відповідями користувачів на кожну рекомендацію. На основі цих відповідей система оновлює свої знання про вподобання користувачів і очікувану винагороду за кожну рекомендацію.

У міру того, як система продовжує взаємодіяти з навколишнім середовищем, вона поступово дізнається, які рекомендації призводять до більшої винагороди, а які ні. Ці знання використовуються для надання кращих рекомендацій у майбутньому, таким чином максимізуючи загальну винагороду, отриману системою.

Рекомендаційні системи на основі RL виявилися ефективними в таких сферах, як онлайн-реклама, електронна комерція та рекомендації вмісту. Однак вони також створюють проблеми, такі проблема холодного запуску(Потрібна велика кількість даних та час для навчання ефективних моделей.) та масштабованість процесу навчання

Підходи на основі глибинного навчання використовують нейронні мережі з багатьма шарами для побудови складних моделей рекомендаційних систем.

Рекомендації на основі контексту

У цьому підході система враховує контекст, у якому користувач звертається до продукту. Наприклад, якщо користувач знаходиться в кінотеатрі, система може рекомендувати фільми, які доступні для перегляду в цей час та місці. Також система може враховувати контекст відносно звичок та інтересів користувачів. Контекст користувача, коли він шукає рекомендацію можна використовувати для кращої персоналізації виведення системи. Для прикладу, у часовому контексті рекомендації щодо відпустки взимку мають дуже відрізнятися від тих, що надаються влітку.

2.4.7.Гібридні РС

Ці РС засновані на комбінації вищезазначених та інших технік. Гібридна система, що поєднує два методи намагається використовувати переваги одного для усунення недоліків іншого. Наприклад, методи КФ страждають від проблем з новими предметами, тобто вони не можуть рекомендувати предмети, які ще не мають рейтингів. Це не обмежує підходи, засновані на змісті(контент-базовані), оскільки передбачення для нових елементів базується на їхньому описі (характеристиках), які зазвичай легко доступний.

2.5Алгоритм k-means

Процес пошуку паралелей у кількох методах створюється за допомогою лінійних методів, тому збільшення кількості порівнянь негативно позначається на продуктивності системи. Одним із підходів до розв'язання проблеми порівнянь є методи. кластеризації, наприклад k-середні. Описані процедури дозволять вам розділити набір на окремі компоненти-підмножини, що складаються з елементів, близьких один до одного. Після такого розбою достатньо запитати систему в межах параметрів відповідно до кластера, тому що подібність між елементами кластера найбільша, і тому надані рекомендації це буде найбільш ефективним у відповідному сенсі.

Метод к середніх дозволяє розбити дані на кластери, щоб зменшити кількість порівнянь. Об’єкти групуються за своїми характеристиками, після чого пошук можна проводити лише серед об’єктів одного кластеру[6].

K-means — це алгоритм кластеризації, мета якого — розділити набір точок даних на K кластерів, де K — попередньо визначена кількість кластерів. Алгоритм роботи наступний:

1.Ініціалізуйте K центроїдів випадковим чином із точок даних або будь-яким іншим способом.

2.Призначте кожну точку даних найближчому центроїду на основі евклідової(або іншої) відстані між точкою даних і центроїдом.

3.Перерахуйте центроїди кожного кластера, взявши середнє значення всіх точок даних, призначених цьому кластеру.

4.Повторюйте кроки 2 і 3 до збіжності, яка виникає, коли центроїди більше не рухаються або досягається максимальна кількість ітерацій.

Результатом роботи алгоритму є K кластерів, де кожен кластер представлений своїм центроїдом.

Приклад класифікації точок методом к-середніх:

```
def knn(x, y, k, dots, classes):
    distances = np.zeros(len(dots))
    for i in range(len(dots)):
        distances[i] = math.sqrt((x - dots[i][0]) ** 2 + (y - dots[i][1]) ** 2)
    closest = list(map((lambda j: classes[j]), sort_pos(distances, k)))  #
список класів k-сусідів
    return max(set(closest), key=closest.count)  # найбільший клас
```

Робота методу:

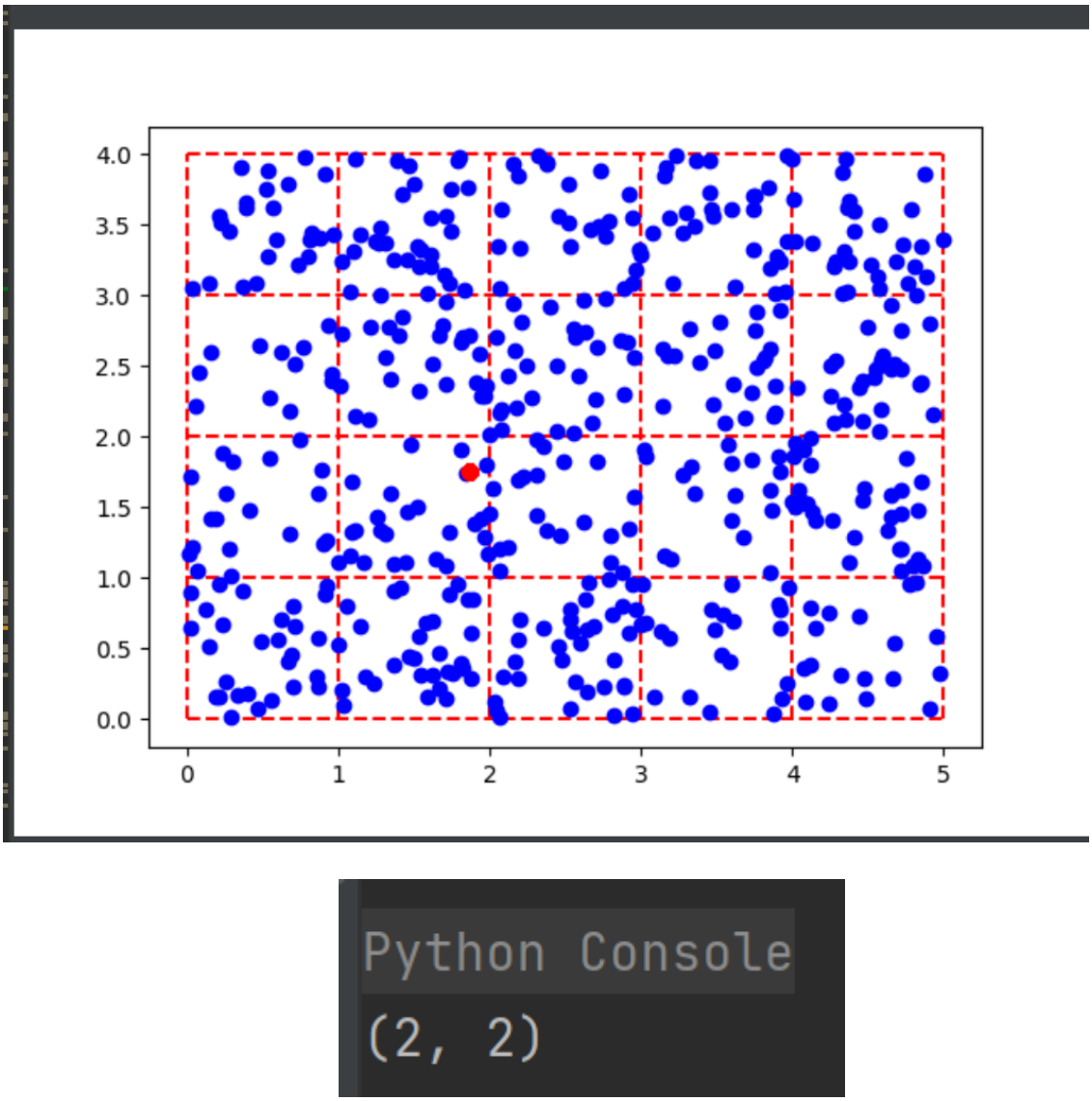


Рис. 2.4

2.6.Деякі інші підходи до порівняння даних

Будь-які методи класифікація та кластеризації першочергово базуються на означенні відповідної подібности або міри дистанції. Одна з таких мір, вона ж косинусна подібність, набула широкого використання, зокрема, в підходах з використанням глибинного навчання. Її суть описана в 2.4.2. Втім, такий підхід є однобоким, бо враховує лише кут між векторами, зовсім нехтуючи їх відстанню. Дійсно, вектори(0.1,0.1,0.1) та (100000, 100000, 100000) при

обчисленні косинусної подібности за її визначенням матимуть подібність 1, що в деякому контексті абсурдно. Не підходить це і в умовах запропонованої мною задачі, оскільки кожен елемент вектора являє собою певну відносну міру, з абсолютною важливістю її значення (детальніше в практичній частині – розділ 5). У зв'язку з цим вирішив застосувати іншу міру подібности.

Для евклідових просторів існує базова формула обчислень відстані: Евклідова відстань. Таким чином[1]:

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}$$

Рис. 2.5

Де x,y – вектори певних порівняних величин.

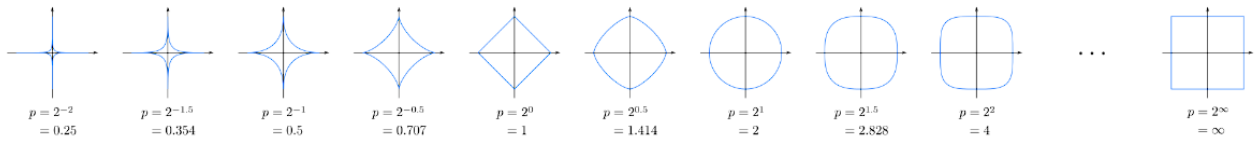
```
def euclidean(x, y):
    return np.sqrt(np.sum((x - y) ** 2))
```

Існує також Манхеттенська відстань, де замість піднесення до квадрата та знаходження кореня використовується просто модуль між величинами. Для узагальнення обох формул була запропонована метрика Мінковського:

$$d(x,y) = (\sum_{k=1}^{n}|x_k - y_k|^r)^{\frac{1}{r}}$$

Рис. 2.6

Дійсно, ця формула набуває значення відстані Евкліда при r = 2 та Манхеттенської при r= 1. Деякі інші значення відносно p візуалізовано на зображенні нижче[21]:



2.7 Висновки

В розділі розглянуто основні відомості про рекомендаційні системи, їх функції, методології розробки; деякі підходи до нормалізації та порівняння даних. В той час як «класичні» контент-базовані рекомендаційні системи та РС базовані на колаборативній фільтрації мають успіх для більшості задач, в деяких випадках більш специфічні РС можуть бути більш ефективними.

У розділі "Підходи до реалізації рекомендаційних систем" було проаналізовано різні підходи до розробки рекомендаційних систем. Зокрема, було розглянуто колаборативний та контентний підходи, гібридні підходи, а також методи, що базуються на зведенні до задачі класифікації.

Під час дослідження було виявлено, що різні підходи мають свої переваги та недоліки, що більш детально розглянуто в розділі 4.

3 ПРИКЛАДИ ЗАСТОСУВАННЯ РС

3.1 SaaS

В [20] наведено не вичерпний, але найширший перелік зі знайдених мною рекомендаційних систем. Спершу розглядаються Software-as-a-service (ПЗ як сервіс, далі SaaS) рекомендаційні системи. Такі системи зазвичай є із закритим кодом(не опен-сорсними). Програмне забезпечення як послуга (SaaS) є одним із трьох основних компонентів хмарних обчислень разом з іншими двома: платформа як послуга (PaaS) та інфраструктура як послуга (IaaS). SaaS працює поверх PaaS, який, своєю чергою, працює поверх IaaS[22].

Підсумовуючи, системи SaaS пропонують багато переваг, включаючи низькі накладні витрати, чіткі шляхи інтеграції та постійний розвиток і вдосконалення. Однак їхня розробка стикається з багатьма проблемами, такими як: багатокористувацький доступ, зберігання й обробку даних, а також забезпечення безпеки конфіденційних даних клієнтів на віддалених серверах.

Кілька компаній пропонують закриті рекомендаційні системи SaaS, зокрема Peerius, Strands, SLI Systems і Gravity R&D. Крім того, існують такі інструменти, як ParallelDots, Amazon Machine Learning, Azure ML, IBM Watson, Recombee та Segmentify, які надають платформи машинного навчання для моделювання даних і створення прогнозів.

Інші спеціалізовані системи рекомендацій SaaS включають Dressipi Style Adviser, Sajari, Mr. DLib, Rumo та Froomle, які обслуговують конкретні галузі, такі як одяг, пошук, академічні бібліотеки, розваги, новини та компанії електронної комерції відповідно.

Загалом, SaaS Recommender Systems є життєздатним варіантом для компаній, яким потрібні персоналізовані рекомендації без великих початкових

інвестицій і технічного досвіду, необхідних для створення внутрішньої системи.

3.2 Системи з відкритим кодом

У[22] наведено приклади безлічі систем рекомендацій із відкритим кодом із різними функціями, алгоритмами та технологіями. Проте, багато з цих РС не підтримуються активно чи взагалі не підтримуються.Найпопулярніші й активні системи рекомендацій із відкритим кодом побудовані на таких мовах програмування, як Python, Java, C++ і Go(golang).

Багато РС із відкритим вихідним кодом використовують методи спільної (колаборативної) фільтрації, які використовуються в системах рекомендацій завдяки їхній ефективності у передбаченні переваг користувача. Деякі системи побудовані на платформах великих даних, таких як Apache Spark і Hadoop, які забезпечують масштабованість і обробку великих наборів даних. Методи глибокого навчання також є популярними, як це видно в таких проєкт, як Alibaba EasyRec. Комерційна підтримка доступна для деяких РС, що свідчить про те, що вони використовуються у виробничих середовищах і мають потужну базу користувачів.

Наявність систем рекомендацій із відкритим вихідним кодом дозволяє розробникам налаштовувати та адаптувати ці системи відповідно до своїх конкретних потреб і варіантів використання, що потенційно скорочує витрати та час розробки, необхідні для створення системи рекомендацій з нуля.

3.3 РС у великих компаніях

Останніми роками системи рекомендацій стають дедалі популярнішими, і великі компанії використовують їх, щоб забезпечити більш персоналізовану та привабливу взаємодію з користувачем. Ці системи довели свою високу

ефективність у збільшенні утримання та продажів, що зрештою допомагає компаніям досягти їхніх бізнес-цілей.

Одним із таких прикладів є YouTube, який використовує складну двоетапну систему рекомендацій. Перший етап, який називається мережею створення кандидатів, забезпечує широку персоналізацію за допомогою спільної фільтрації. Це передбачає отримання невеликої підмножини відео із великої кількості на основі історії активности користувача на YouTube і схожости між користувачами, вираженої у вигляді ідентифікаторів переглядів відео, маркерів пошукового запиту, демографічних даних тощо.

З іншого боку, Netflix відстежує, що дивляться його користувачі та як вони взаємодіють із платформою, проводячи поточні A/B-тести та вимірюючи довгострокові показники задоволеності, щоб покращити свої рекомендації. Замість того, щоб просто перевіряти ідеї на основі історичних даних або прогнозувати рейтинги, Netflix прагне забезпечити персоналізований рейтинг, створення сторінок, пошук, вибір зображень, обмін повідомленнями тощо.

Facebook із величезним обсягом даних не може покладатися на традиційні підходи машинного навчання, які базуються на вибірці. Натомість він використовує найсучаснішу модель рекомендацій із відкритим вихідним кодом (DLRM) у поєднанні з відкритими фреймворками Facebook PyTorch і Caffe2. Спільна фільтрація та підходи на основі прогнозної аналітики допомагають надавати рекомендації на основі вподобань і взаємодії людей зі схожими смаками, дозволяючи користувачам виявляти найбільш релевантні елементи через детальні частини рекомендованого вмісту.

Instagram підтримує розділ рекомендацій Explore завдяки постійним дослідженням, які проводить команда Facebook AI. Використовуючи доменно-спеціальну мову, оптимізовану для отримання кандидатів у системах рекомендацій під назвою IGQL, Instagram використовує механізм пошуку найближчих сусідів Facebook, FAISS, щоб надавати своїм користувачам

налаштований канал рекомендованого вмісту. Рекомендації базуються на рекомендаціях сеансу та взаємодії подібних облікових записів, а не на фільтрації всього вмісту, доступного на платформі.

Нарешті, TikTok має повністю налаштовану стрічку вмісту під назвою «Для вас», яка враховує взаємодію користувачів, наприклад відео, які сподобалися та надіслані, підписані облікові записи, опубліковані коментарі, створений/завантажений вміст і «індикатор інтересу» (тобто час перегляду). . Система рекомендацій машинного навчання також враховує інформацію про відео, таку як субтитри, звуки, хештеги, тип пристрою, налаштування облікового запису, мовні параметри, країну, геолокацію тощо. Хоча система рекомендацій TikTok оптимізована для продуктивності, результати можуть бути менш точними, ніж на інших платформах. Вміст з облікових записів, які мають більше підписників, також отримує поштовх на платформі.

Попри відмінності в підходах, усі ці компанії використовують механізми рекомендацій для просування різних типів вмісту, що зрештою призводить до кращого, довшого та глибшого залучення користувачів, а також збільшення продажів і доходу.

3.4 WordToVec

В [24] розглянуто алгоритм Word2Vec, що дозволяє векторизувати текстові дані, що буває дуже корисно для рекомендаційних систем, а також реалізацію рекомендаційної системи, що рекомендує книги на основі їх опису.

Алгоритми word2vec використовують контекст для створення числових представлень слів, що призводить до схожих векторів для слів, які з'являються в одному контексті. Процес створення векторів слів заснований на таких кроках:

1.Створення пар даних у форматі [вхідне слово, вихідне слово], де кожне слово представлене у вигляді двійкового вектора фіксованої довжини n. У

цьому векторі значення і кодується одиницею на позиції і та нулем на всіх інших позиціях (це називається one-hot кодування).

2.Створення моделі, яка отримує на вхід one-hot вектори та видає на вихід такі ж one-hot вектори.

3.Визначення функції втрат, яка оцінює правильність передбачення вихідного слова та використовується для оптимізації моделі.

4.Оцінка якості моделі, переконання, що схожі слова мають схожі векторні представлення[25].

У статті[24] обговорюється використання вбудованих слів, зокрема середнього Word2Vec і TF-IDF Word2Vec, для створення механізму рекомендацій на основі вмісту. Система рекомендує книги користувачам за схожістю їх описів.

Word2Vec — це модель нейронної мережі з одним прихованим шаром, який передбачає сусідні слова для кожного слова в реченні. Вагові коефіцієнти, отримані прихованим шаром, можна використовувати для передбачення слів, які представляють слова як вектори в D-вимірному просторі. Word2Vec здатний фіксувати семантичне значення та зв’язки. У статті пояснюється, як конвертувати описи книг у вектори за допомогою Word2Vec. Подібність між описами книг можна виміряти за допомогою косинусної подібності. Результати рекомендацій: у статті представлено 5 найкращих рекомендацій щодо книг із використанням описаних методів. Показано, що метод TF-IDF Word2Vec надає потужніші рекомендації порівняно із середнім Word2Vec. Реальні системи рекомендацій є більш надійними та досконалими, а їх оцінка та вибір залежать від A/B-тестування та конкретної сфери діяльності.

Загалом у статті представлено огляд систем рекомендацій на основі вмісту, що використовують Word2Vec, і підкреслюється використання середнього Word2Vec і TF-IDF Word2Vec для рекомендацій книг на основі їхніх описів.

3.5 Висновки

Системи SaaS пропонують багато переваг, такі як низькі накладні витрати, легка інтеграція та постійний розвиток і вдосконалення, проте розробка систем SaaS може зіткнутися з проблемами, такими як багатокористувацький доступ, зберігання та обробка даних, а також забезпечення безпеки конфіденційних даних клієнтів на віддалених серверах.

Існує багато систем рекомендацій з відкритим вихідним кодом, які пропонують різні функції, алгоритми та технології, багато з цих систем підтримують методи колаборативної фільтрації, методи глибокого навчання та використовують платформи великих даних для обробки великих наборів даних. Наявність систем з відкритим вихідним кодом дозволяє розробникам налаштовувати й адаптувати ці системи під свої потреби, що зменшує витрати та час розробки.

Великі компанії, такі як YouTube, Netflix, Facebook, Instagram і TikTok, використовують системи рекомендацій для забезпечення персоналізованої взаємодії з користувачами та досягнення своїх бізнес-цілей. Вони використовують різні ідеї та алгоритми.

4 ПОРІВНЯЛЬНИЙ АНАЛІЗ ПІДХОДІВ

Існує кілька підходів до розробки систем рекомендацій, кожна з яких має свої переваги та обмеження. Основні з них було розглянуто в розділі 2. Пригадаймо деякі з поширених підходів і порівняймо їх:

Фільтрування на основі вмісту:

Фільтрація на основі вмісту рекомендує елементи користувачам на основі їхніх попередніх уподобань і атрибутів елементів.

Переваги:

Зменшена проблема холодного запуску: системи, засновані на вмісті, ефективні у розв’язанні проблеми холодного запуску, яка стосується складності надання рекомендацій для нових елементів або користувачів із невеликим чи відсутнім даними. Аналізуючи характеристики елементів, ці системи можуть давати рекомендації, навіть якщо доступна інформація обмежена.

Ефективність при роботі зі змістовною інформацією, такою як тексти, зображення або аудіо- та відеофайли.

Надає персоналізовані рекомендації на основі вподобань користувача.

Може запропонувати прозорість у рекомендаціях, враховуючи чіткі характеристики предметів.

Добре працює, коли доступні дані користувача обмежені.

Недоліки:

Схильний до обмеженої різноманітності рекомендацій.

Вимагає чітко визначених функцій для айтемів та алгоритмів обробки контенту.

Можуть виникнути проблеми з рекомендаціями нових або незнайомих речей.

Обмежена ефективність при роботі зі складними об'єктами, такими як музика або послуги, які складаються з багатьох складових частин.

Відсутність можливості врахування контексту рекомендації, наприклад, місця, часу чи настрою користувача(вирішується гібридними підходами).

Отже, цей підхід корисний, коли доступні дані користувача обмежені, і він може добре впоратися з проблемою холодного запуску. Однак фільтрація на основі вмісту, як правило, має обмежену різноманітність рекомендацій.

Спільна фільтрація:

Спільну фільтрацію можна класифікувати на два типи: спільне фільтрування на основі користувачів і на основі елементів. Спільна фільтрація на основі користувачів визначає схожих користувачів і рекомендує елементи, які сподобалися цим схожим користувачам. Спільна фільтрація на основі елементів зосереджена на схожості елементів і пропонує елементи, схожі на ті, до яких користувач уже виявив інтерес.

Переваги:

Ефективність при роботі з великими обсягами даних та складними продуктами.

Можливість врахування контексту рекомендації та персональних інтересів користувачів.

Може надавати випадкові рекомендації, визначаючи схожих користувачів або елементи.

Не вимагає явних характеристик предметів або знання домену.

Може отримувати довгострокові рекомендації, враховуючи інтереси.

Недоліки:

Страждає від нестачі даних, особливо коли мало оцінок або взаємодій.

Проблема "бульбашки фільтрації", коли система може рекомендувати тільки продукти, які вже були вибрані користувачем в минулому, і не дає можливості відкривати нові об'єкти.

Під час роботи з великими наборами даних можуть виникнути проблеми з масштабованістю.

Труднощі в роботі з новими користувачами або елементами з обмеженими історичними даними(тобто обмеженою кількістю взаємодій із системою в минулому).

Обробляти нових користувачів РС важко, оскільки немає даних, які можна використовувати для надання персоналізованих рекомендацій. Подібним чином, для нових елементів системам спільної фільтрації не вистачає даних про взаємодію користувачів, щоб точно оцінити їх якість або релевантність.

Отже, спільна фільтрація може надати випадкові рекомендації та не вимагає явних характеристик айтемів. Однак вона часто страждає від розрідженості даних, проблеми холодного запуску та ефекту «міхура фільтра».

Гібридні підходи:

Гібридні системи рекомендацій поєднують кілька підходів, щоб подолати обмеження окремих методів.

Переваги:

Використовує сильні сторони кількох підходів, поєднуючи їх стратегії рекомендацій.

Може надати більш точні та різноманітні рекомендації порівняно з окремими методами.

Знімає обмеження окремих підходів, компенсувавши їх слабкі сторони.

Пропонує гнучкість в адаптації до різних доменів і сценаріїв рекомендацій.

Недоліки:

Підвищена складність проєктування, впровадження та обслуговування системи.

Потрібна інтеграція та координація між різними алгоритмами рекомендацій.

Може вводити додаткові обчислювальні витрати та вимоги до ресурсів.

Проблеми у визначенні оптимального поєднання та зважування різних підходів.

Отже, гідридні РС використовують переваги різних методів для надання більш точних і різноманітних рекомендацій. Наприклад, гібридна система може використовувати колаборативну фільтрацію для захоплення налаштувань користувача та контент-базовану фільтрацію для покращення рекомендацій за допомогою функцій елементів. Змішані підходи часто більш ефективні, але можуть бути складними для реалізації та підтримки.

Підходи, що ґрунтуються на знаннях:

Рекомендаційні системи, засновані на знаннях, використовують знання домену та чіткі уподобання користувача для створення рекомендацій. Зазвичай вони включають явні дані користувача або ж бази знань.

Переваги:

Включає чіткі вподобання користувача.

Може надавати пояснення щодо рекомендацій, підвищуючи довіру та розуміння користувачів.

Ефективно в областях, де явні дані користувача є вирішальними (наприклад, експертні системи).

Може обробляти складні обмеження рекомендацій і вимоги до домену.

Недоліки:

Дуже залежить від наявності та точності знань предметної області.

Обмежена масштабованість при роботі з великою кількістю елементів або користувачів.

Складнощі в охопленні та впровадженні нових уподобань і смаків користувачів.

Отже, підходи, що ґрунтуються на знаннях, є кращими в тих сферах, де вирішальними є чіткі вподобання користувачів, які можуть надати пояснення для їхніх рекомендацій. Однак вони значною мірою покладаються на дані користувача та обмежені доступністю та точністю знань предметної області.

Контекстно-залежні підходи:

Контекстно-залежні системи рекомендацій враховують контекстні фактори, такі як час, місцеперебування, пристрій і поведінка користувача, щоб надавати персоналізовані рекомендації. Слід зазначити, що цей підхід застосовується разом з іншими в гібридних системах.

Переваги:

Підбирає рекомендації на основі контекстуальних факторів, підвищуючи релевантність і своєчасність.

Адаптується до різних ситуацій і потреб користувачів.

Може надавати рекомендації щодо певного місця, часу або пристрою(більша гнучкість).

Покращує взаємодію з користувачем, враховуючи середовище користувача.

Недоліки:

Потребує додаткового збору даних і обробки контекстної інформації.

Проблеми в отриманні точних і актуальних контекстних даних.

Підвищена складність моделювання та обробці контекстних факторів.

Потенційні проблеми конфіденційності, пов’язані з отриманням і використанням контексту користувача.

Отже, включаючи контекстну інформацію, ці системи можуть адаптувати рекомендації до різних ситуацій і потреб користувачів. Контекстно-залежні підходи підвищують релевантність і своєчасність рекомендацій, але можуть вимагати додаткового збору даних і обчислювальних ресурсів.

Демографічна фільтрація:

Демографічна фільтрація – це ще один підхід, який використовується в системах рекомендацій, який враховує демографічну інформацію, таку як вік, стать, місцеперебування та професія, для надання рекомендацій.

Переваги:

Простий і зрозумілий у застосуванні, оскільки він спирається на доступні демографічні дані.

Може бути ефективним у певних сценаріях, коли демографічні фактори сильно впливають на вподобання (наприклад, рекомендації щодо віку).

Допомагає персоналізувати рекомендації на основі демографічних сегментів, орієнтуючись на певні групи користувачів.

Може стати відправною точкою для нових користувачів з обмеженими даними взаємодії, розв'язувати проблему холодного запуску.

Недоліки:

Покладається виключно на демографічну інформацію, яка може не повністю відображати індивідуальні переваги та смаки.

Може призвести до стереотипів або узагальнень на основі демографічних характеристик, що призведе до менш різноманітних рекомендацій.

Ігнорує потенціал для індивідуальних варіацій у межах демографічних груп, що призводить до менш точних рекомендацій.

Вимагає точних і актуальних демографічних даних, які може бути складно отримати або зберегти.

Отже, загалом демографічне фільтрування може бути корисним компонентом гібридної системи рекомендацій або в поєднанні з іншими підходами для надання персоналізованих рекомендацій. Однак покладання виключно на демографічну інформацію має обмеження та може не врахувати тонкощі індивідуальних уподобань і смаків. Це часто ефективніше в поєднанні контент-базованою фільтрацією, колаборативною фільтрацією або іншими передовими методами для підвищення точності та різноманітності рекомендацій.

Важливо зауважити, що переваги та недоліки, перераховані вище, є загальними спостереженнями та можуть відрізнятися залежно від конкретних деталей впровадження, характеристик набору даних і областей застосування.

5. ПРАКТИЧНА ЧАСТИНА

5.1 Описова постановка задачі

Користувач хоче дістатися з пункту А в пункт Б, в конкретному випадку потягом, та має певні побажання щодо поїздки. Рекомендаційна система, отримавши відповідні дані від користувача пропонує сполучення, що є найбільш відповідними до запиту. Такий алгоритм рекомендацій було обрано, оскільки предметна область не передбачає можливості використання алгоритмів колаборативної фільтрації, бо нема принципу вибору схожими користувачами схожих маршрутів поїздок. Так само не передбачається організація поїздок згідно з відгуками на попередні поїздки – дійсно, люди їдуть туди, куди їм необхідно та обирають транспорт згідно зі своїми потребами та можливостей. Втім, відгуки на якість рекомендацій може бути використано для корекції роботи системи.

5.2 Технічна постановка задачі

Дано: - Вектор характеристик, обраних користувачем V= {v0, v1,..,vn}, де vj– певна характеристика, обрана користувачем, n – загальна кількість характеристик, j=0,1,…,n; 2.

Вектор характеристик для певного потяга U= {u0, u1, ,..,un}, де ui- певна характеристика потяга; i =0, 1, …, n; n – загальна кількість характеристик;

Знайти: вектори характеристик потягів, що мають найменший коефіцієнт подібності.

5.3Практична реалізація

5.3.1Використані інструменти розробки

Для розробки відповідної програми з рекомендаціями мною були використані засоби мови програмування Java, графічної бібліотеки JavaFX, як найкращої та сучаснішої графічної бібліотеки для даної мови, що має ряд переваг відносно Swing та реляційна база даних MySQL(відповідно мова запитів SQL), програмні засоби для роботи з ними.

Java — це високорівнева об’єктноорієнтована мова програмування, розроблена таким чином, щоб мати якомога менше залежностей реалізації. Це мова програмування загального призначення, призначена для того, щоб дозволити програмістам написати один раз і запустити будь-де(WORA) платформонезалежна - це означає, що скомпільований код Java може працювати на всіх платформах, які підтримують Java, без необхідності повторної компіляції. [18] Програми Java зазвичай скомпільовані в байт-код, який може працювати на будь-якій віртуальній машині Java (JVM) незалежно від базової архітектури комп'ютера. Синтаксис Java подібний до C і C++(сі-подібний синтаксис), але є менш низькорівневою.

Java FX – є сучасним набором інструментів для створення додатків на основі мови програмування Java. JavaFX був розроблений для заміни застарілого Swing, в якому відсутні багато сучасних функцій. Тому у нього набагато більше можливостей для десктопної розробки, ніж у Swing, що підтверджує порівняння двох технологій:

JavaFX	Swing
Підходить для розробки насичених клієнтських додатків із сучасним інтерфейсом користувача	Бібліотека для розробки стандартних GUI
Чистий код	Багато застарілих фіч
Вбудована підтримка MVC	Недостатня підтримка MVC
Підтримується та розробляється спільнотою; постійно впроваджуються нові фічі та покращення	Нові фічі не додаються
Підтримка стилів CSS	Створення стилів лише за допомогою коду
Вбудований API для роботи з багатопоточністю	Немає вбудованого API для роботи з багатопоточністю
FXML для декларативного створення макетів	Не підтримує декларативне створення макетів
Вбудована підтримка 3D графіки	Потрібні додаткові API для 3D
Підтримує прив'язку властивостей (при зв'язуванні компонентів зміна в одному компоненті автоматично відбивається на іншому)	Не підтримує прив'язку властивостей
Від'єднаний від JDK, починаючи з Java 11	Входить до складу JDK

Рис. 5.1

5.3.2Схема бази даних

Для збереження даних, згідно яких виконується рекомендаційний пошук, була розроблена база даних, схема якої представлена на UML-діаграмі нижче:

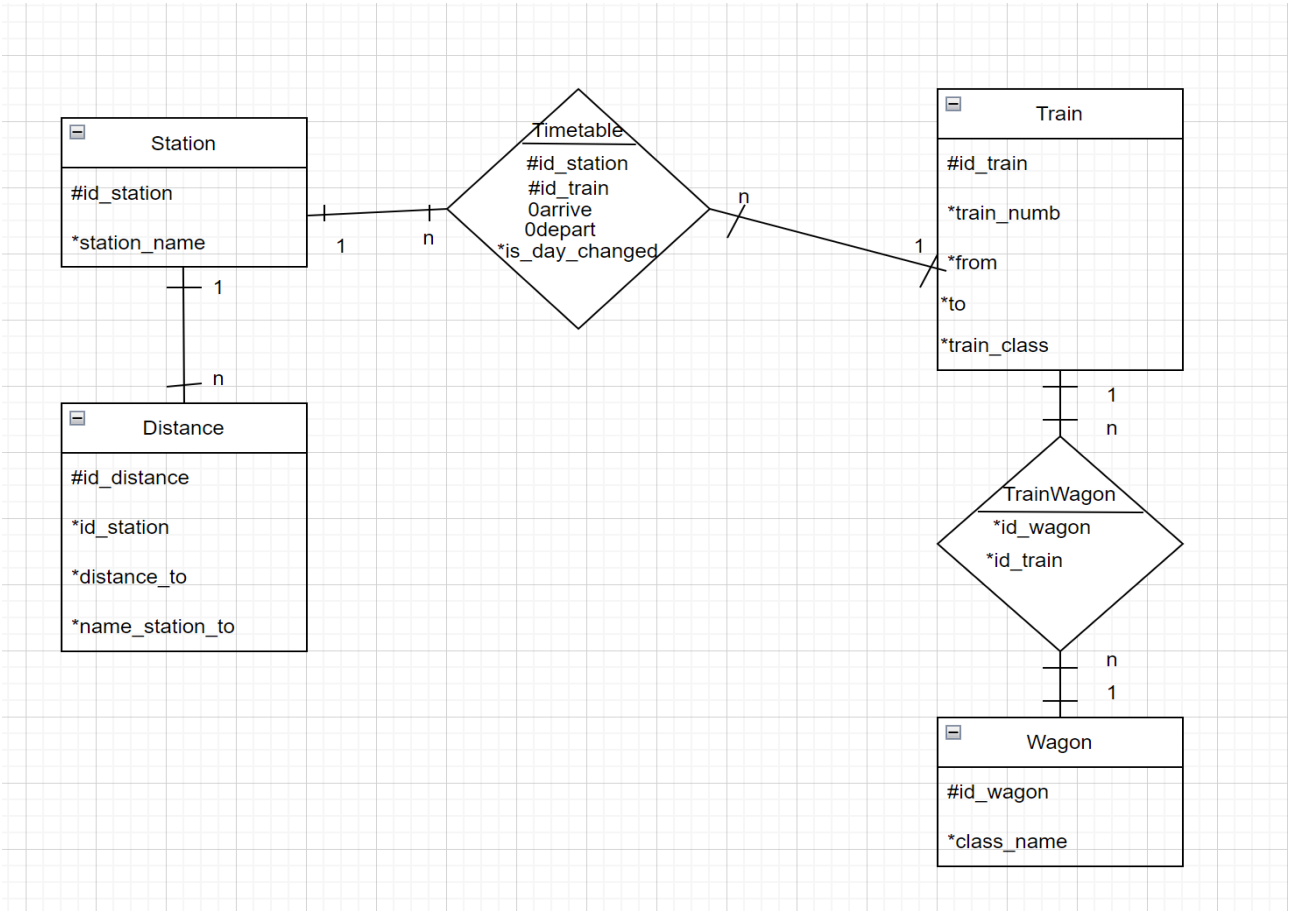


Рис.5.2

Для креслення схеми використано draw.io, який має певні переваги:

- Безплатний доступ.
- Декілька варіантів доступу(браузер, плагін, програма).
- Широкі можливості.
- Різноманітність форматів.
- Інтеграція із сервісами(GitHub, GitLab, Dropbox, Google-диск, OneDrive).
-

Головні сутності(таблиці, реляції):

-Train – зберігає інформацію про потяг, його номер, клас та маршрут прямування;
-Wagon – перелік типів вагонів потяга;

-Station – перелік станцій;

-Distance – перелік відстаней між станціями, використовується для обчислення вартости проїзду згідно визначеної формули.

Сутності TimeTable та TrainWagon реалізовують зв’язки багато до багатьох. Timetable також зберігає інформацію про час прибуття та відправлення потяга. Поле is_day_changed - індикатор переходу доби під час руху потяга, що необхідно для коректного пошуку напрямку потягів.

5.3.3 Опис програми:

Програма побудована за стандартною архітектурою трирівневих застосунків, тобто, має рівень роботи з базою даних, він же Repository або DAO – data access object, рівень бізнес-логіки Service та рівень представлення даних – Controller.

Також програма містить модельки, .fx, .css файли тощо.

Структуру програми продемонстровано на ілюстрації нижче:

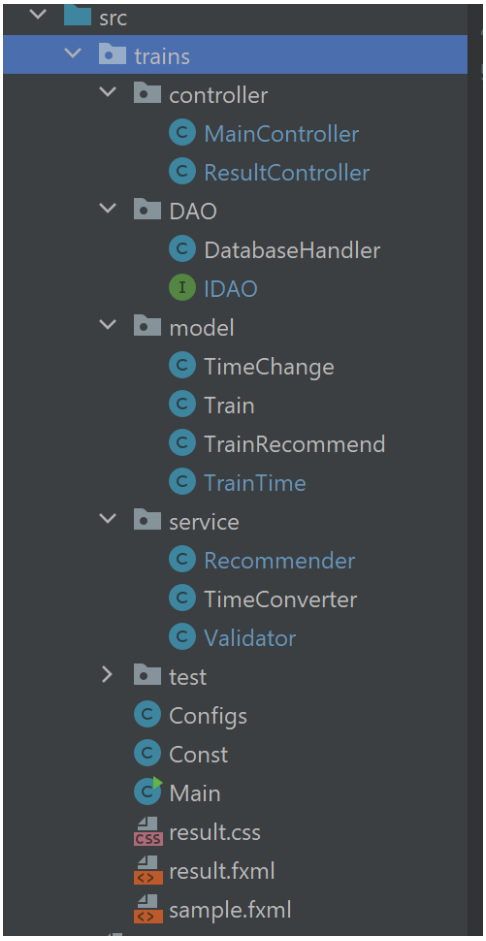


Рис. 5.3

Коротко опишу їх суть. Класи MainController та ResultController відповідають за відображення даних на відповідних сторінках.

DatabaseHandler реалізує операції роботи з базою даних(read-операції, бо зміна стану бази даних пересічним користувачем не передбачена умовою задачі). Цей клас реалізує інтерфейс IDAO. Відповідно, класи вищого рівня (сервісів) залежать від інтерфейсу. Це програмне рішення дозволяє в разі потреби(наприклад зміни бази даних) просто написати нову реалізацію цього інтерфейсу, без переписування решти класів.

Сервіс Recommender – ключова частина програми. Містить програмну реалізацію рекомендацій та виконує технічну постановку задачі, описану в пункті 5.2. Сервіс TimeConverter виконує різноманітні операції з конвертацією часу, та роботою з ним. Сервіс Validator відповідає за валідацію вхідних даних.

Нижче продемонстровану діаграму класів, згенеровану можливостями IntelliJ Ide:

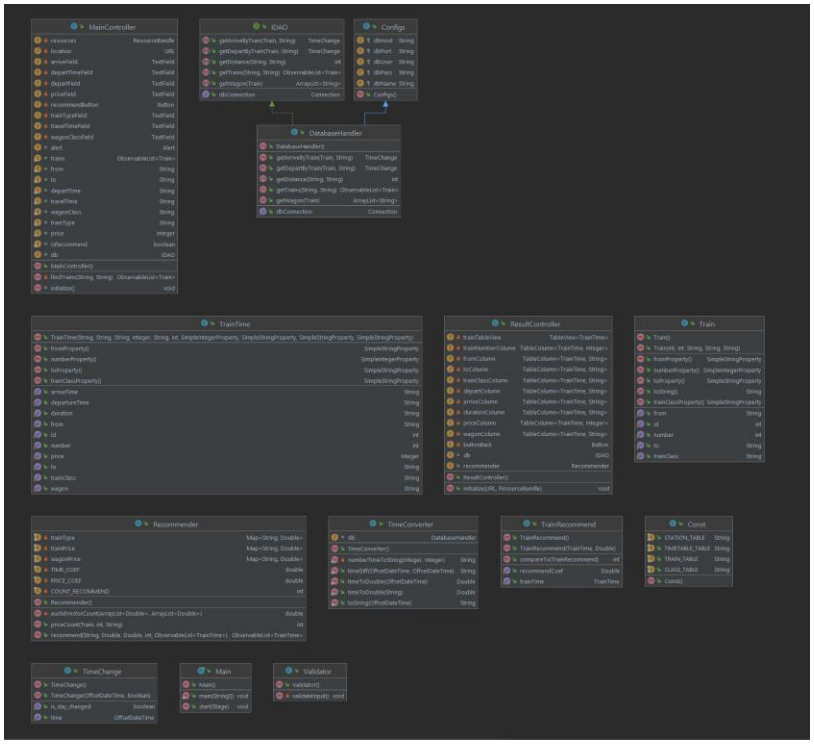


Рис. 5.4

5.4 Використання програми

При запуску програма виглядає наступним чином:

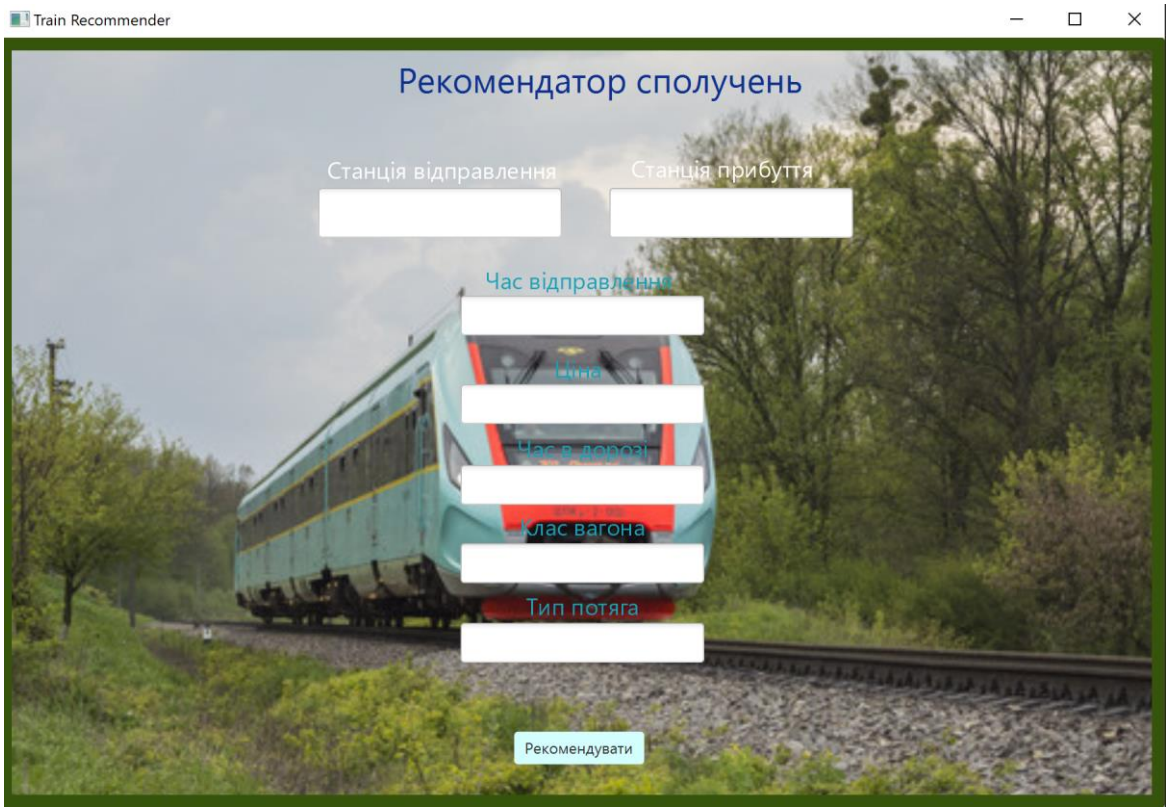


Рис. 5.5

Далі користувач вводить бажані характеристики для рекомендованих сполучень, наприклад:

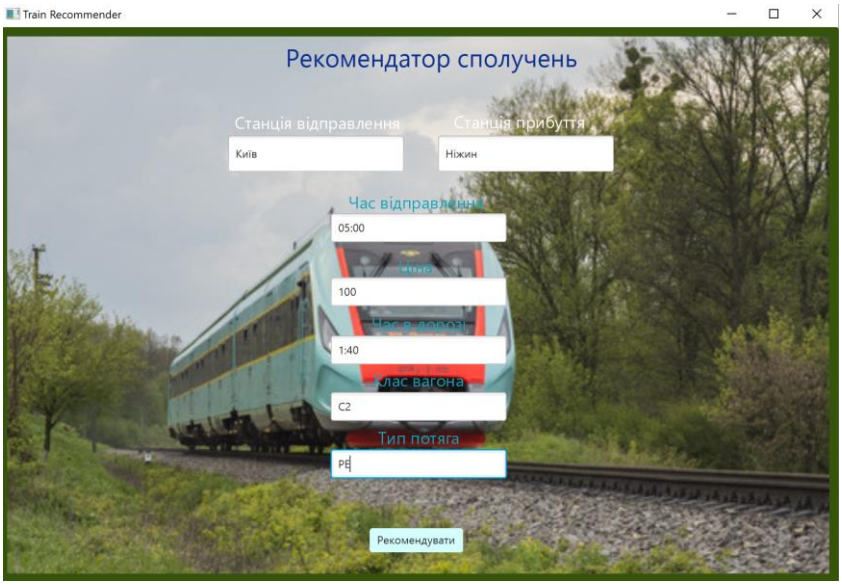
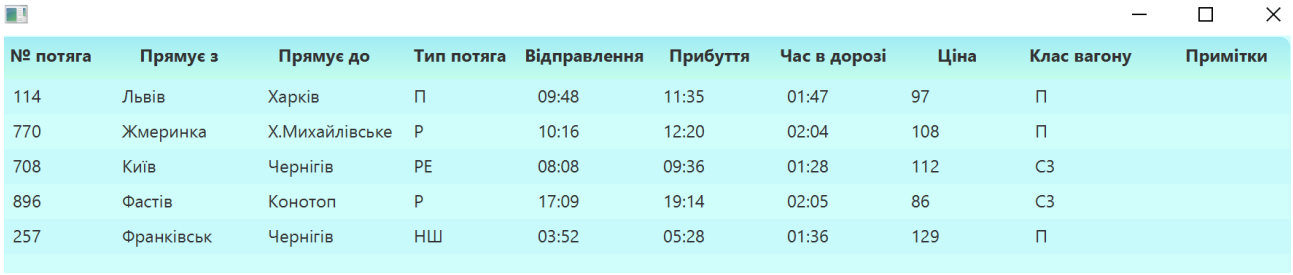


Рис. 5.6

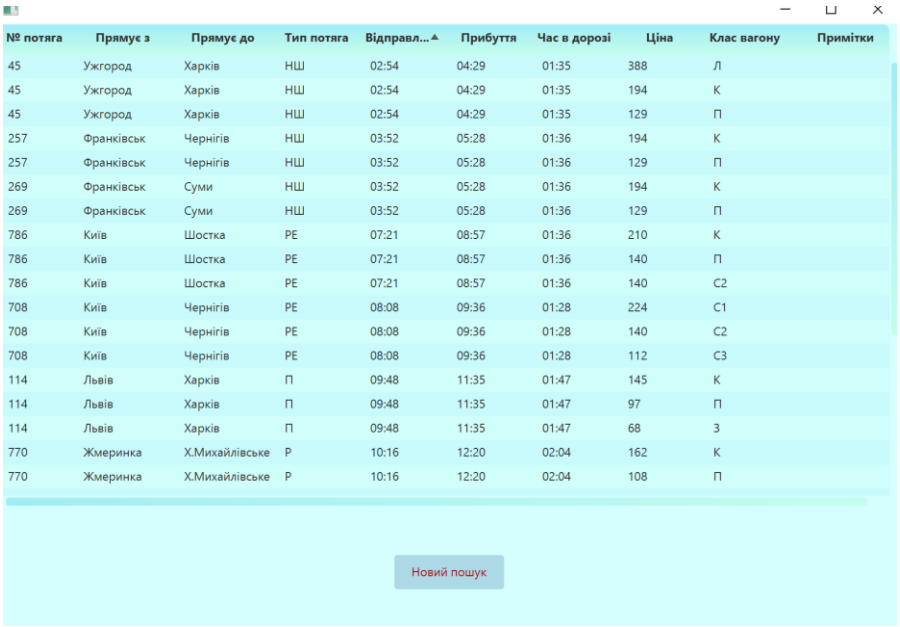
І система запропонує потяги, що максимально підходять, тобто ті, для яких рекомендаційна корисність є вищою. Така система є різновидом рекомендаційної системи базованої на знаннях – однозначно визначених користувачем даних. Недоцільність використання «класичних» вміст-базованого підходів або підходу спільної фільтрації аргументовано в пункті 5.1 Для вказаного прикладу маємо результат:



№ потяга	Прямує з	Прямує до	Тип потяга	Відправлення	Прибуття	Час в дорозі	Ціна	Клас вагону	Примітки
114	Львів	Харків	П	09:48	11:35	01:47	97	П	
770	Жмеринка	Х.Михайлівське	Р	10:16	12:20	02:04	108	П	
708	Київ	Чернігів	РЕ	08:08	09:36	01:28	112	С3	
896	Фастів	Конотоп	Р	17:09	19:14	02:05	86	С3	
257	Франківськ	Чернігів	НШ	03:52	05:28	01:36	129	П	

Рис. 5.7

Якщо вводити окремі дані, то буде обрахований результат, базований на введених користувачем даних. Якщо ввести тільки пункт відправлення та прибуття, то випаде перелік всіх наявних за маршрутом потягів та вагонів в них, наприклад, для того ж напрямку:



№ потяга	Прямує з	Прямує до	Тип потяга	Відправл...	Прибуття	Час в дорозі	Ціна	Клас вагону	Примітки
45	Ужгород	Харків	НШ	02:54	04:29	01:35	388	Л	
45	Ужгород	Харків	НШ	02:54	04:29	01:35	194	К	
45	Ужгород	Харків	НШ	02:54	04:29	01:35	129	П	
257	Франківськ	Чернігів	НШ	03:52	05:28	01:36	194	К	
257	Франківськ	Чернігів	НШ	03:52	05:28	01:36	129	П	
269	Франківськ	Суми	НШ	03:52	05:28	01:36	194	К	
269	Франківськ	Суми	НШ	03:52	05:28	01:36	129	П	
786	Київ	Шостка	РЕ	07:21	08:57	01:36	210	К	
786	Київ	Шостка	РЕ	07:21	08:57	01:36	140	П	
786	Київ	Шостка	РЕ	07:21	08:57	01:36	140	С2	
708	Київ	Чернігів	РЕ	08:08	09:36	01:28	224	С1	
708	Київ	Чернігів	РЕ	08:08	09:36	01:28	140	С2	
708	Київ	Чернігів	РЕ	08:08	09:36	01:28	112	С3	
114	Львів	Харків	П	09:48	11:35	01:47	145	К	
114	Львів	Харків	П	09:48	11:35	01:47	97	П	
114	Львів	Харків	П	09:48	11:35	01:47	68	З	
770	Жмеринка	Х.Михайлівське	Р	10:16	12:20	02:04	162	К	
770	Жмеринка	Х.Михайлівське	Р	10:16	12:20	02:04	108	П	

Новий пошук

Рис. 5.8

Широко реалізована система валідації даних, яка виправляє введені користувачем дані(наприклад, 1:50 сприймається як і 01:50) або виводить інформацію про необхідність корекції, наприклад:

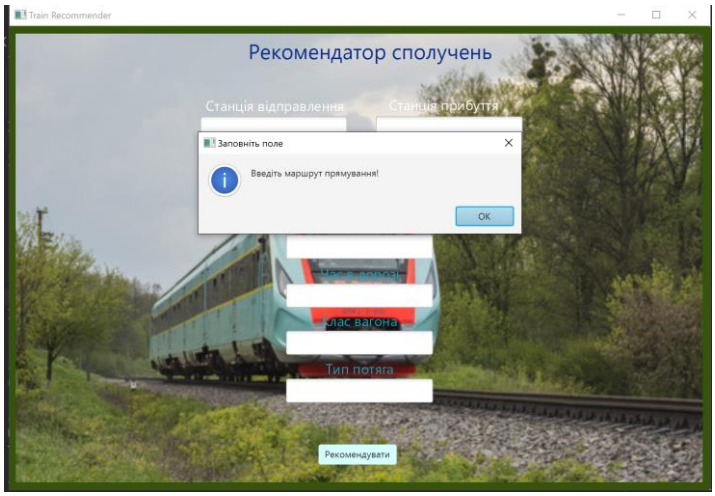


Рис. 5.9

Можливі доопрацювання додатка:

-роширення функціоналу на інші види сполучень;

-інтеграція з базою даних Укрзалізниці в разі надання апі;

-впровадження принципу рекомендацій для пошуку маршрутів з пересадкою не вважаю доцільним, оскільки зазвичай їх є дуже обмежена кількість та основним принципом підбору слугує час на пересадку. Втім програма пропонуватиме станцію пересадки в разі відсутности прямих сполучень, тоді пошуки через проміжну точку прямування можна здійснити звичайним чином.

Висновки

В результаті написання курсової роботи було досліджено актуальність та області застосування рекомендаційних систем, зокрема навчальних, розглянуто основні підходи до проєктування рекомендаційних систем, такі як:

- Контент-базовані. Цей підхід корисний, коли доступні дані користувача обмежені, і він може добре впоратися з проблемою холодного запуску. Однак фільтрація на основі вмісту, як правило, має обмежену різноманітність рекомендацій.

-Колаборативна(спільна) фільтрація - працює наступним чином: якщо користувач А та користувач Б мають схожі інтереси та обирали подібні продукти в минулому, то система може рекомендувати користувачу А продукти, які обрав користувач Б. Колаборативна фільтрація може надати нестандартні рекомендації та не вимагає явних характеристик елементів, проте вона часто страждає від розрідженості даних, проблеми холодного запуску та ефекту «міхура фільтра».

Тому використовують інші підходи до проєктування РС, такі як демографічні РС, РС базовані на знаннях. Гібридні РС дають можливість зібрати переваги різних типів рекомендаційних систем та позбутись недоліків, проте, є складнішими в розробці та підтримці.

Системи SaaS є популярним варіантом для компаній, що шукають персоналізовані рекомендації без великих початкових витрат і технічного досвіду. Вони пропонують переваги, такі як низькі накладні витрати, легка інтеграція та постійний розвиток і вдосконалення. Однак розробка таких систем може стикатися з проблемами багатокористувацького доступу, зберігання й обробки даних, а також забезпечення безпеки конфіденційних даних на віддалених серверах.

Наявність РС з відкритим вихідним кодом дозволяє розробникам налаштовувати й адаптувати ці системи відповідно до своїх потреб і варіантів використання. Більшість розглянутих РС побудовані засобами мов Багато з цих систем побудовані на мовах програмування, таких як Python, Java, C++ і Go

Рекомендаційні системи можуть мати значний вплив на економіку бізнесу. З одного боку, вони сприяють збільшенню продажів, залученню нових клієнтів та збереженню наявний. З іншого боку, вони допомагають зменшити витрати на маркетинг та рекламу, оскільки можуть бути спрямовані точніше та ефективніше.

Використання рекомендаційних систем також породжує етичні питання та може посягати на приватну інформацію. Важливо розробляти та застосовувати етичні стандарти та політики для розв’язку цих питань. Також важливо приділяти увагу безпеці систем, оскільки, вони можуть бути вразливими до деяких видів атак.

Рекомендаційні системи зазвичай використовують аналітичні методи для визначення ефективності своїх рекомендацій. Шляхом збору даних про взаємодію користувачів з рекомендаціями, можна оцінити, наскільки користувачі задоволені рекомендаціями та чи вони призводять до певних дій, таких як покупки або перегляди. Ця аналітика дозволяє покращити алгоритми та параметри рекомендаційних систем, щоб забезпечити кращі результати. Для оптимізації роботи спроєктованої мною рекомендаційної системи важливо отримувати та обробляти відгуки користувачів.

У роботі розглянуто основні відомості про рекомендаційні системи, їх функції, методології розробки; деякі підходи до нормалізації та порівняння даних. В той час як «класичні» контент-базовані рекомендаційні системи та РС базовані на колаборативній фільтрації мають успіх для більшості задач, в деяких випадках більш специфічні РС можуть бути більш ефективними.

Крім того, реалізовано рекомендаційну систему з пошуків залізничних сполучень. Система базується на фільтрації на основі знань, що використовується для реалізації більш персоналізованих та досконалих ресурсів, де користувач активно задає пошукові запити власноруч. Ідея є авторською, бо мною було знайдено і переглянуто багато сайтів, які здійснюють пошук сполучень та, щонайбільше, дозволяють фільтрувати результати за якимось критерієм, але жоден не дозволяє отримувати рекомендації за кількома заданими критеріями.

Джерела

[1] Recommender Systems Handbook - Francesco Ricci, Lasse L. Eilersen,Dietmar Jannach,Kun-He Chen, 210, 845c.

[2] Manouselis, N., Costopoulou, C. Analysis and Classification of Multi-Criteria Recommender Systems. World Wide Web: Internet and Web Information Systems, 10(4):415–441, 2007.

[3] Matsatsinis, N.F., Samaras, A.P MCDA and preference disaggregation in group decision support. European Journal of Operational Research, 130(2):414–429, 2001.

[4] Рекомендаційні системи - що це [Електронний ресурс] – Режим доступу до ресурсу: https://jak.koshachek.com/articles/rekomendacijni-sistemi-shho-ce.html.

[5] Bobadilla J. Recommender systems survey / J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez // Knowledge-Based Systems. – Vol. 46. – 2013. – P. 109–132.

[6] Реалізація алгоритмів рекомендаційних систем Курсова робота/ Безштанько В. В.

[7] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An open architecture for collaborative filtering of netnews. In: CSCW ’94: Proc. of the 1994 ACM Conf. on Computer Supported Cooperative Work, pp. 175–186. ACM, New York, NY, USA (1994)

[8] Burke, R.: Hybrid web recommender systems. In: The Adaptive Web, pp. 377–408. Springer Berlin / Heidelberg.

[9] Коефіцієнт конверсії: визначення [Електронний ресурс] / Google – Режим доступу до ресурсу: https://support.google.com/google-ads/answer/2684489?hl=uk.

[10] Mladenic, D.: Text-learning and Related Intelligent Agents: A Survey. IEEE Intelligent Systems 14(4), 44–54 (1999)

[11] Rocchio, J.: Relevance Feedback Information Retrieval. In: G. Salton (ed.) The SMART retrieval system - experiments in automated document processing, pp. 313–323. PrenticeHall, Englewood Cliffs, NJ (1971)

[12] Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: The Adaptive Web, pp. 291–324. Springer Berlin / Heidelberg (2007)

[13]Косинус подібності [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D1%81%D0%B8%D0%BD%D.

[14] Mahmood, T., Ricci, F.: Towards learning user-adaptive state models in a conversational recommender system. In: A. Hinneburg (ed.) LWA 2007: Lernen - Wissen - Adaption, Halle, September 2007, Workshop Proceedings, pp. 373–378. Martin-Luther-University HalleWittenberg (2007)

[15] Felfernig, A., Burke, R.: Constraint-based recommender systems: technologies and research issues. In: Proceedings of the 10th International Conference on Electronic Commerce, ICEC'08, pp. 1–10. ACM, New York, NY, USA (2008)

[16] Arazy, O., Kumar, N., Shapira, B.: Improving social recommender systems. IT Professional 11(4), 38–44 (2009)

[17] Golbeck, J.: Generating predictive movie recommendations from trust in social networks. In: Trust Management, 4th International Conference, iTrust 2006, Pisa, Italy, May 16-19, 2006, Proceedings, pp. 93–104 (2006)

[18] Java [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Java.

[19] Recommender Systems: Use Cases, Choosing One For Your Business [Електронний ресурс] – Режим доступу до ресурсу: https://datarootlabs.com/blog/recommender-systems-use-cases-choosing-one-for-your-business.

[20] List of Recommender Systems [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/grahamjenson/list_of_recommender_systems.

[21] The Minkowski Bistance [Електронний ресурс] – Режим доступу до ресурсу: https://stats.stackexchange.com/questions/420520/what-do-p-and-q-refer-to-in-the-minkowski-distance.

[22] SCIENCE CHINA Information Sciences May 2014 Software-as-a-service (SaaS): perspectives and challenges TSAI WeiTek1,2 ∗, BAI XiaoYing2 & HUANG Yu1 https://www.researchgate.net/profile/Wei-Tek-Tsai/publication/271917699_Software-as-a-service_SaaS_Perspectives_and_challenges/links/5e69d699a6fdcc7595031c0a/Software-as-a-service-SaaS-Perspectives-and-challenges.pdf

[23CaseRecommender [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/caserec/CaseRecommender?utm_source=datarootlabs&utm_medium=blog.

[24] Content-Based Recommendation System using Word Embeddings [Електронний ресурс] – Режим доступу до ресурсу:

https://www.kdnuggets.com/2020/08/content-based-recommendation-system-word-embeddings.html.

[25] Word2Vec: як працювати з векторними представленнями слів [Електронний ресурс] – Режим доступу до ресурсу: https://neurohive.io/ru/osnovy-data-science/word2vec-vektornye-predstavlenija-slov-dlja-mashinnogo-obuchenija/.