

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)

Кафедра _____ Інформаційних управляючих систем _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

Дослідження методів побудови пояснень в рекомендаційній системі з
використанням темпоральних правил _____
(тема)

Виконав:

студент 2 курсу, групи ІУСТм-20-1

Літвінов О. О.

(прізвище, ініціали)

Спеціальність _____ 122 Комп'ютерні науки _____

(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна _____

(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Інформаційні управляючі
системи та технології _____

(повна назва освітньої програми)

Керівник _____ професор Чалий С. Ф. _____

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

_____ Петров К.Е. _____

(прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
Кафедра _____ Інформаційних управляючих систем _____
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)
Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)
Освітня програма _____ Інформаційні управляючі системи та технології _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Літвінову Олексію Олександровичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Дослідження методів побудови пояснень в рекомендаційній системі з використанням темпоральних правил _____

затверджена наказом університету від _____ 05 _____ листопада _____ 2021 р. № 1645 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії _____ 09 _____ грудня _____ 2021 р.

3. Вихідні дані до роботи _____ Науково-технічні публікації та інтернет джерела з тематики _____ атестаційної роботи _____

4. Перелік питань, що потрібно опрацювати в роботі _____ вступ, аналіз пояснень в рекомендаційних системах, аналіз особливостей пояснень в рекомендаційних системах, дослідження методів побудови пояснень в рекомендаційних системах, аналіз структури рекомендаційних систем для роботи в онлайн і офлайн режимах, дослідження моделей представлення темпоральних знань, постановка задачі кваліфікаційної роботи, удосконалення методу побудови пояснень з використанням темпоральних знань, формальне представлення темпоральних знань для побудови пояснень, удосконалений метод побудови пояснень з використанням темпоральних правил, технологія побудови пояснень в рекомендаційній системі з використанням темпоральних правил, практичне використання отриманих результатів, розробка програмного модулю формування пояснень в рекомендаційній системі, експериментальна перевірка удосконаленого методу формування пояснень, висновки _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз літератури та Інтернет-джерел	08.11.2021	
2	Постановка задачі	09.11.2021-11.11.2021	
3	Обробка матеріалу	12.11.2021-13.11.2021	
4	Дослідження методів побудови пояснень в	14.11.2021-15.11.2021	
5	Дослідження моделей представлення темпоральних	15.11.2021-16.11.2021	
6	Аналіз структури рекомендаційних систем для роботи в онлайн і офлайн режимах	17.11.2021-18.11.2021	
7	Апробація результатів дослідження на прикладі	19.11.2021-20.11.2021	
8	Написання пояснювальної записки	21.11.2021-28.11.2021	
9	Підготовка презентації	29.11.2021-30.12.2021	
10	Перевірка на плагіат	01.12.2021-02.12.2021	
11	Нормоконтроль	03.12.2021-08.12.2021	
12	Захист	10.12.2021	

Дата видачі завдання 08 листопада 2021 р.

Студент _____
(підпис)

Керівник роботи _____ професор Чалий С.Ф.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи містить: 84 с., 4 розділи, 11 рис., 7 табл., 38 джерел.

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ, ІНТЕРНЕТ-МАГАЗИН, МЕТОД УРАХУВАННЯ ТЕМПОРАЛЬНОЇ ДИНАМІКИ, РЕКОМЕНДАЦІЙНІ СИСТЕМИ, ФОРМУВАННЯ РЕКОМЕНДАЦІЙ, ШТУЧНИЙ ІНТЕЛЕКТ

У роботі проведено огляд методів формування пояснень у проектах рекомендаційних систем. Проаналізовано існуючі методи формування пояснень. На підставі проведеного аналізу запропоновано удосконалений метод побудови пояснень у рекомендаційних систем з використанням темпоральних правил шляхом видалення аномальних вхідних даних що до поведінки користувачів.

В ході дослідження отримані такі результати: визначені особливості пояснень в рекомендаційних системах; визначені існуючі методи побудови пояснень у рекомендаційних системах; визначено існуючі моделі представлення темпоральних знань; визначено формальне представлення темпоральних знань; виконана опис удосконаленого методу; проведено експериментальну перевірку удосконаленого методу.

ABSTRACT

Explanatory Note to certification work contains 84 pages, 4 sections, 11 pictures, 7 tables, 38 sources.

INTELLECTUAL SYSTEMS, ONLINE STORE, METHOD OF
TEMPORAL DYNAMICS, RECOMMENDATION SYSTEMS,
RECOMMENDATION FORMATION, RECOMMENDATION
RECOMMENDATIONS

The paper reviews the methods of forming explanations in the projects of recommendation systems. The existing methods of forming explanations are analyzed. Based on the analysis, an improved method for constructing explanations in recommendation systems using temporal rules by removing anomalous input data on user behavior is proposed.

The following results were obtained during the research: features of explanations in recommendation systems were determined; the existing methods of construction of explanations in recommendation systems are defined; the existing models of representation of temporal knowledge are defined; the formal representation of temporal knowledge is determined; the description of the improved method is executed; an experimental test of the improved method was performed.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Аналіз пояснень в рекомендаційних системах	10
1.1 Аналіз особливостей пояснень в рекомендаційних системах.....	10
1.2 Дослідження методів побудови пояснень в рекомендаційних системах	14
1.3 Аналіз структури рекомендаційних систем для роботи в онлайн і офлайн режимах	19
1.4 Дослідження моделей представлення темпоральних знань	26
1.5 Постановка задачі кваліфікаційної роботи.....	36
2 Удосконалення методу побудови пояснень з використанням темпоральних знань.....	38
2.1. Формальне представлення темпоральних знань для побудови пояснень	38
2.2 Удосконалений метод побудови пояснень з використанням темпоральних правил.....	43
3. Технологія побудови пояснень в рекомендаційній системі з використанням темпоральних правил.....	54
4. Практичне використання отриманих результатів.....	56
4.1 Розробка програмного модулю формування пояснень в рекомендаційній системі.....	56
4.2 Експериментальна перевірка удосконаленого методу формування пояснень	59
Висновки	63
Перелік джерел посилання	64
Додаток А Графічний матеріал.....	70

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

РС – рекомендаційна системи;

ШІ – штучний інтелект;

ALE – accumulated local effects;

ICE – individual conditional expectation;

MMD – maximum mean discrepancy;

PDP – partial dependency plots;

SLA – service level agreement;

STAG – sheffield temporal annotation guidelines.

ВСТУП

Бурхливе зростання обсягів даних призвело до появи безпрецедентної кількості даних та низки нових додатків для використання цих даних. Збільшення обсягу інформації надало світові широкий спектр продуктів та послуг на різних рівнях якості, що зробило процес прийняття рішень про те, який продукт купувати чи яку послугу використовувати є складним завданням для людей. У відповідь на цю проблему з'явилися рекомендаційні системи.

Рекомендаційні системи надають персоналізовані пропозиції товарів, послуг, на основі наявних даних про користувача. Вхідні дані генеруються на основі явного та неявного зворотного зв'язку від користувача. Тому рекомендаційні системи активно використовуються як додаткові модулі до систем електронної комерції, потокових сервісів при просуванні онлайн-заходів, бронюванні готелів.

Існуючі алгоритми побудови рекомендацій дозволяють ефективно прогнозувати інтереси користувачів за наявності даних про їх покупки та оцінки, виставлені користувачами. Однак у деяких ситуаціях інформація про користувачів може бути неточною або неповною, що призводить до викривлення рекомендацій. Отримані неправильні рекомендації не відповідають інтересам користувачів, що знижує довіру до рекомендаційної системи. Неточності та спотворення даних про користувацькі уподобання є результатом шилінгових атак. Такі атаки використовуються для зміни продажів цільових товарів у бажаному зловмисником напрямку.

Суть шилінгової атаки полягає у тому, щоб штучно змінити рейтинг цільових предметів. В результаті персональна рекомендація формується зі спотвореннями і може не враховувати реальні уподобання користувачів. Рекомендаційна система, використовуючи неправильний персональний список товарів та послуг, змушує користувача вибирати ті елементи, які цікаві зловмиснику.

Тому в даний час одним із напрямків розвитку рекомендаційних систем є доповнення рекомендацій поясненнями. Відповідно до цієї тенденції персоналізована інформація для користувача надається у двох аспектах: фактична рекомендація та її пояснення. У поясненні розкриваються причини для включення товару до рекомендованого списку товарів та послуг, щоб підвищити довіру користувача та згодом переконати його прийняти пропозиції, отримані цією системою. При побудові пояснень використовуються знання про взаємозв'язок між характеристиками товару та його розміщенням у рекомендованому персональному списку товарів. Зокрема, в поясненнях використовується інформація про популярність предмета, розподіл оцінок предмета у схожих користувачів, схожість тем, авторів, інтереси користувачів. Використання пояснень підвищує довіру користувача до списку пунктів.

Оскільки вимоги споживачів змінюються з часом, рекомендується враховувати темпоральну динаміку переваг користувачів під час побудови пояснень. Темпоральна динаміка відображається як впорядкована послідовність виборів користувача. Врахування темпоральної динаміки дозволяє швидко коригувати список рекомендацій та відповідних пояснень при зміні вимог користувачів з метою збільшення продажів пропонованих товарів та послуг.

Однак існуючі підходи не приділяють достатньою уваги до побудові персональних пояснень з урахуванням відмінності в поведінки користувачів. Отже задача побудови пояснень з урахуванням індивідуальної динаміки зміни потреб користувачів є актуальною. Тому вибір і зіставлення знань, що відображають зміни в уподобаннях користувачів з часом і використовуються для формування пояснень, є важливим завданням. Вирішення цієї проблеми створює умови для побудови та своєчасного уточнення пояснень з урахуванням темпоральної динаміки споживчих переваг в умовах неповних або невірних даних про таких користувачів.

1 АНАЛІЗ ПОЯСНЕНЬ В РЕКОМЕНДАЦІЙНИХ СИСТЕМАХ

1.1 Аналіз особливостей пояснень в рекомендаційних системах

Рекомендаційна системи (РС) — це програмні інструменти та методи, що надають пропозиції щодо елементів, які можуть бути корисними для користувача. Рекомендаційні системи стають все більш популярними в останні роки і використовуються в різних сферах, включаючи фільми, музику, новини, книги, науково-дослідницькі статті, пошукові запити, соціальні теги та продукти в цілому[2].

Рекомендаційні системи надають користувачам нові рекомендовані елементи, але іноді вони розглядаються як чорні ящики, де не надається пояснювальна інформація про них. Таким чином, ці рекомендації можуть супроводжуватися поясненнями, які описують, чому рекомендований конкретний пункт. Пояснення рекомендацій зазвичай полегшує користувачам прийняття рішень, підвищуючи коефіцієнти конверсії та збільшуючи задоволення та довіру до системи. Пояснення — це опис, який обґрунтовує рекомендації та допомагає користувачам краще зрозуміти, чи відповідає рекомендований елемент їхнім потребам чи ні. [3].

Надання пояснень разом із рекомендаціями приведе до кращого розуміння рекомендаційної системи. Протягом останнього десятиліття потреба в обґрунтуваннях і поясненнях стала привертати увагу, що нині є більш важливою через атаки шилінгу на користь певного предмета [4].

У дослідженнях рекомендаційних систем зростає усвідомлення необхідності зробити процес рекомендацій більш прозорим для користувачів [5]. В останні роки питання про те, як автоматично генерувати та представляти пояснення на стороні системи, привернуло все більший інтерес у дослідженнях. Сьогодні деякі основні засоби пояснення вже включені в веб-сайти електронної комерції, такі як Amazon.com [6].

Пояснення представлені через різні інтерфейси пояснень, які їх інтерпретують. Традиційні інтерфейси пояснення – це зазвичай текстові пояснення, пояснення на основі тегів, гістограми, радарні графіки, кругові діаграми, діаграми дерева та інші графічні зображення.

У спільноті РС все більше визнається, що такі показники точності, як середня помилка, точність і запам'ятовування, можуть лише частково оцінити рекомендаційну систему. Задоволення користувачів та довіра, все більше вважаються важливими.

Крім усього іншого, хороші детальні пояснення можуть допомогти викликати лояльність та довіру користувачів, підвищити задоволеність, допомогти користувачам швидше і легше знаходити те, що вони хочуть, і переконати користувачів купити або спробувати рекомендований товар. Таблиця 1.1 визначає сім можливих цілей які виконують пояснення в рекомендаційних системах.

Таблиця 1.1 – Цілі пояснень в рекомендаційних системах

Властивість	Визначення
Прозорість	Пояснює, як працює система
Перевірка	Дозволити користувачам повідомити системі, що це не так
Довіра	Підвищити довіру користувачів до системи
Переконливість	Переконувати користувачів спробувати або купити
Ефективність	Допомогти користувачам приймати правильні рішення
Оперативність	Допомогти користувачам швидше приймати рішення
Задоволення	Збільште простоту використання або задоволення

Прозорість – Пояснить, як працює система. Стаття в Wall Street Journal під назвою «If TiVo Thinks You Are Gay, Here's How to Set It Straight» описує розчарування користувачів нерелевантним вибором, зробленим відеоманітофоном, який записує програми, які, як припускає, сподобаються

його власнику, на основі переглядів глядача у минулому. Наприклад, один користувач, Іван, підозрював, що його TiVo вважав його геєм, оскільки відеомагнітофон незрозумілим чином записала програми на гей-теми. Цей користувач, безсумнівно, заслуговував на пояснення та можливість все прояснити.

Пояснення може пояснити, як була обрана рекомендація. В експертних системах, наприклад, у сфері прийняття медичних рішень, важливість прозорості вже давно визнана. Прозорість або евристика «Видимість статусу системи» також є встановленим принципом зручності використання, і його важливість також була підтверджена дослідженнями користувачів РС [7].

Перевірка – дати можливість користувачам повідомити системі, що вона робить неправильно. Пояснення можуть допомогти виділити та виправити помилкові припущення або кроки. Коли система збирає та інтерпретує інформацію у фоновому режимі, як у випадку з TiVo, стає все більш важливим зробити міркування доступними для користувача. Після прозорості, другий крок — дозволити користувачеві виправити міркування або зробити систему доступною для перегляду. Пояснення мають бути частиною циклу, де користувач розуміє, що відбувається в системі, і контролює тип рекомендацій, виправляючи системні припущення, де це необхідно [8]. Перевірка пов'язана з встановленим принципом юзабіліті-контролю користувача.

Довіра – підвищення довіри користувачів до системи. Довіра іноді пов'язана з прозорістю: попередні дослідження показують, що прозорість і можливість взаємодії з РС підвищує довіру користувачів [9]. Довіра до РС також може залежати від точності рекомендаційного алгоритму. Дослідження довіри користувачів (визначається як усвідомлена впевненість у компетенції системи рекомендацій) свідчить про те, що користувачі мають намір повернутися до РС, які вони вважають надійними.

Переконливість – переконує користувачів спробувати або купити. Пояснення можуть підвищити оцінку користувачами системи або надані рекомендації [10]. Це можна кваліфікувати як переконання, оскільки це спроба

отримати користь для системи, а не для користувача. Насправді було показано, що користувачами можна маніпулювати, щоб дати оцінку, ближчу до прогнозу системи, незалежно від того, є цей прогноз точним чи ні. У цьому дослідженні учасників переконали взяти фільм напрокат і, можливо, користувачі можуть менше вплинути на неправильні прогнози щодо предметів, які потребують більших інвестицій, наприклад, придбання персонального комп'ютера. Також важливо враховувати, що надмірне переконання може мати зворотний ефект, як тільки користувачі зрозуміють, що вони спробували або купили речі, які їм насправді не потрібні.

Ефективність – допомога користувачам приймати правильні рішення. Замість того, щоб просто переконати користувачів спробувати чи купити товар, пояснення також може допомогти користувачам прийняти кращі рішення. Ефективність за визначенням сильно залежить від точності рекомендаційного алгоритму. Ефективне пояснення допомогло б користувачеві оцінити якість запропонованих елементів відповідно до його власних уподобань. Це збільшило б ймовірність того, що користувач відкине нерелевантні параметри, допомагаючи їм розпізнати корисні. Наприклад, система рекомендацій книг з ефективними поясненнями допоможе користувачеві купувати книги, які їм дійсно подобаються. Білгік і Муні наголошують на важливості вимірювання здатності системи допомагати користувачеві приймати точні рішення щодо того, які рекомендації використовувати.

Оперативність – допомогти користувачам швидше приймати рішення. Пояснення можуть допомогти користувачам швидше вирішувати, який рекомендований елемент їм найкраще підходить. Оперативність — ще один встановлений принцип юзабіліті, тобто наскільки швидко можна виконати завдання. Цей критерій є одним із найбільш поширених у літературі про РС, враховуючи, що завдання РС полягає в пошуку голок у стогах сіна.

Оперативність можна підвищити, дозволивши користувачеві зрозуміти зв'язок між конкуруючими варіантами. У сфері цифрових камер конкуруючі

варіанти можна, наприклад, описати як «якщо менша роздільна здатність, то дешевше». Таким чином користувачі можуть швидко знайти щось дешевше, якщо вони готові задовольнитися меншою роздільною здатністю.

Задоволення – коли використання системи є веселим. Пояснення можуть підвищити задоволеність користувачів системою, хоча погані пояснення, ймовірно, знизять інтерес користувача або сприйняття системи. Було виявлено, що наявність довших описів окремих елементів позитивно корелює як з усвідомленою корисністю, так і з простотою використання рекомендаційної системи. Це можна розглядати як підвищення загальної задоволеності користувачів.

1.2 Дослідження методів побудови пояснень в рекомендаційних системах

Існує дві групи методів побудови пояснень в рекомендаційних системах:

- вибір ключових залежностей, що дають змогу узагальнити процес отримання рекомендацій;
- декомпозиція процесу отримання рекомендацій[11].

Перша група методів спрямована на побудову такого представлення процесу надання рекомендацій, що містить:

- зв'язок між ключовими змінними та отриманим результатом. Цей зв'язок визначає вплив ключових параметрів на отриману рекомендацію у зручній для користувача формі;
- причинний зв'язок між ключовими діями процесу формування пояснень.

Декомпозиція дає змогу виявити зв'язки між процедурами обробки вхідних даних та отриманим результатом. Такий підхід визначає причинно-наслідкові зв'язки між цими процедурами.

Ці посилання є основою для пояснення процесу формування рекомендації в рекомендаційній системі, що спрощує сприйняття і підвищує впевненість у цьому процесі.

Таким чином, перша група підходів реалізує принцип чорного ящика і передбачає побудову окремого пояснення після отримання результату в рекомендаційній системі.

Другий напрямок пояснень пов'язаний з прозорістю використовуваних алгоритмів. Тобто в рамках цього напрямку реалізується принцип білого ящика. Кожен прозорий алгоритм визначає причинно-наслідкові зв'язки, які призвели до відповідної рекомендації. Поясненням служить сам прозорий алгоритм. Такі результати інтерпретуються безпосередньо в процесі їх отримання.

Розрізняємо три типи підходів формування пояснень: пояснення на основі моделі, пояснення на основі атрибуції та пояснення на основі прикладів. В таблиці 1.2 показані основні особливості типи підходів формування пояснень.

Таблиця 1.2 – Головні особливості підходів формування пояснень

Пояснення	Головні особливості
На основі побудови моделі поведінки користувача	Пояснення враховує поведінку користувача.
На основі атрибуції	Пояснення відповідають питанням: Які домінуючі властивості вплинули в цій рекомендації? У поясненні кількісно оцінюється вплив кожної властивості на представлену рекомендацію.
На основі прикладів	Метод вибирає конкретні екземпляри з набору даних, щоб пояснити надану рекомендацію. Пояснення на основі прикладів доцільно використовувати, якщо рекомендаційна система може подати екземпляр даних зрозумілим для людини способом. Наприклад при представленні зображень, тому що користувач можемо переглядати їх безпосередньо.

Пояснення на основі моделі. Цей клас включає методи, які використовують модель для пояснення рекомендацій. Пояснення, засновані на моделі, підпадають під моделювання, а також до post-hoc пояснення. Або сама модель рекомендаційної системи використовується як пояснення (з'ясувальне моделювання), або для пояснення рекомендації створюється інша, більш інтерпретована модель. Зауважте, що модель рекомендаційної системи завжди може надати користувачеві додаткове пояснення, навіть якщо використовуються post-hoc пояснення, але цього пояснення може бути недостатньо, якщо модель рекомендаційної системи не піддається інтерпретації.

У моделюванні метою є розробка моделі рекомендаційної системи, яка сама по собі інтерпретується користувачем. Для досягнення цього можна вибрати клас моделі, який, як відомо, генерує інтерпретовані моделі для людей. Три класи моделей, які зазвичай вважаються інтерпретованими:

- це розріджені лінійні класифікатори (наприклад, лінійна/логістична регресія, загальні адитивні моделі):
- методи дискретизації (наприклад, учні на основі правил, дерева рішень);
- моделі на основі прикладів (наприклад, K-найближчі сусіди) [12].

Однак на інтерпретацію також впливають інші фактори, такі як кількість і зрозумілість вхідних функцій. Отже, хоча дерево рішень зазвичай вважається легшим для інтерпретації, ніж нейронна мережа, глибоке дерево рішень може бути менш інтерпретованим, ніж компактна нейронна мережа.

Інші способи отримання інтерпретованої моделі включають використання архітектурних модифікацій, розробку гібридних моделей або навчання моделі для надання пояснень. Архітектурні модифікації включають регуляризацію для встановлення обмежень форми [13], для стимулювання високих шарів нейронної мережі для представлення об'єкта [14] або для розробки нейронних мереж зі структурою моделі адитивного індексу [15]. Гібридні моделі поєднують інтерпретовані моделі та моделі чорного ящика.

Наприклад, Wang [16] використовує модель чорного ящика (тобто не інтерпретуючу) для підмножини даних, де інтерпретована модель має незадовільну продуктивність, а Papernot і McDaniel [17] використовують K-найближчих сусідів у кожному шарі глибокої нейронної мережі. Hind et al. [18] використовують навчальний набір, анотований поясненнями, щоб навчити модель завдання надавати пояснення.

Як альтернатива, ми можемо розробити сурогатну модель, яка наближається до моделі рекомендаційної системи, і використовувати цю модель у формі post-hoc пояснення. Lakkaraju et al. [19] представляють методологію, яка називається поясненнями чорного ящика через прозорі наближення (BETA), щоб отримати глобальну сурогатну модель. Їхній метод вивчає компактний набір правил прийняття рішень шляхом спільної оптимізації однозначності, вірності та інтерпретації. Bastani et al. [20] представити новий алгоритм для вивчення представлення у вигляді дерева рішень. Тан та ін. [21] отримати уявлення про моделі чорної скриньки, використовуючи модель дистиляції та порівнюючи з нещодавно навченою моделлю з тим же класом моделі. Однак пояснення на основі моделі завжди можна використовувати для створення інших типів пояснень, і оскільки обидва методи виводять локальні моделі, ми стверджуємо, що вони тут належать.

Пояснення на основі атрибуції. Методи атрибуції оцінюють або вимірюють пояснювальну силу входних ознак і використовують це для пояснення моделі рекомендаційної системи. Більшість доступних post-hoc методів пояснення належать до цього класу. Ці методи іноді також називають методами важливості/змінної ознаки, релевантності чи впливу. Ми можемо класифікувати методи атрибуції відповідно до механізму генерування пояснення на методи збурення та зворотного поширення [22]. Методи, засновані на зворотному поширенні, не залежать від моделі, оскільки вони часто розроблені для певного класу моделі або вимагають, щоб функція моделі була диференційованою.

Двома прикладами візуальних методів глобальної атрибуції є графіки часткової залежності (PDP) [23] і графіки накопичених локальних ефектів (ALE) [24], обидва показують середній вплив вхідних функцій на вихідні дані. Крім того, Feature Importance [23] вимірює збільшення помилки передбачення після перестановки ознаки, а Feature Interaction (або H-статистика) [25] обчислює величину варіації, що пояснюється умовами взаємодії. Існують інші методи для створення пояснень. Графіки індивідуальних умовних очікувань (ICE) показують очікуване передбачення моделі як функцію значень ознак для даного екземпляра [26]. Натхнені теорією ігор, значення Шеплі вказують, як справедливо розподілити важливість («виплату») між функціями («гравцями») шляхом усереднення граничного внеску для всіх можливих коаліцій. Кількісний вхідний вплив є одним наближенням вибірки для обчислення значень Шеплі [27]. Іншим способом обчислення цих значень є метод SHAP [28]. INVASE метод може визначити гнучку кількість релевантних функцій для кожного екземпляра, використовуючи модель, що складається з селектора, предиктора та базової мережі [29]. Нарешті, Lei et al. запровадив глобальну та локальну міру змінної важливості на основі висновку Leave-One-Covariate-Out.

Пояснення на основі прикладів. Цей клас методів пояснює модель завдання шляхом вибору екземплярів із набору даних або створення нових екземплярів, наприклад шляхом вибору прототипів (тобто екземплярів, які добре передбачені моделлю) та критики (тобто екземплярів, які погано передбачувані моделлю), визначення впливових екземплярів для параметрів або результату моделі або створення контрфактичного пояснення. Ці методи особливо інтуїтивно зрозумілі, якщо екземпляри можуть бути представлені зрозумілою людиною способом. У літературі цього класу доступно відносно небагато методів.

Kim et al. [30] розробив MMD-critic для вивчення прототипів і критики для даного набору даних, використовуючи максимальну середню розбіжність (MMD) як міру подібності. Метод обчислення впливових екземплярів був

розроблений Куком [31], який використовує діагностику видалення для вимірювання впливу видалення екземпляра на передбачення моделі. Впливові екземпляри можуть надати пояснення як на глобальному, так і на локальному рівні. Нарешті, контрфактичні пояснення вимагають гіпотетичного мислення у формі: «Якби X не відбулося, Y не відбулося б». Їх можна створити методом проб і помилок, але Wachter et al. [32] представляють більш складний підхід до створення безумовних контрфактичних пояснень.

Після огляду методів пояснень ми бачимо що вони не враховують зміни користувацьких вподобань з часом. Однак, враховуючи недостатньої кількості інформації про нових користувачів, а також перекручування рейтингів у результаті атак шилінгів, важливо сформулювати найбільш актуальні рекомендації. Для цього необхідно враховувати темпоральну динаміку вимог користувачів. Для вирішення цієї проблеми рекомендується використовувати темпоральні залежності, що відображають загальну зміну кількості проданих товарів або послуг протягом часом.

1.3 Аналіз структури рекомендаційних систем для роботи в онлайн і офлайн режимах

На рисунку 1.1 зображено узагальнені кроки роботи рекомендаційної системи.

На першому етапі проходить збір даних. Це перший і найважливіший крок створення механізму рекомендацій. Є два основних типи даних, які потрібно збирати. Неявні дані – сюди входять відомості, отримані в результаті таких дій, як історія веб-пошуку, кліки, події кошика, журнал пошуку та історія замовлень. Явні дані – це інформація, зібрана на основі відгуків клієнтів, таких як огляди та рейтинги, симпатії та антипатії, а також коментарі до продукту. Механізми рекомендацій також використовують дані атрибутів

клієнтів, такі як демографічні дані (вік, стать) та психографічні дані (інтереси, цінності), щоб ідентифікувати схожих клієнтів, а також дані характеристик (жанр, тип елемента) для визначення подібності продуктів.

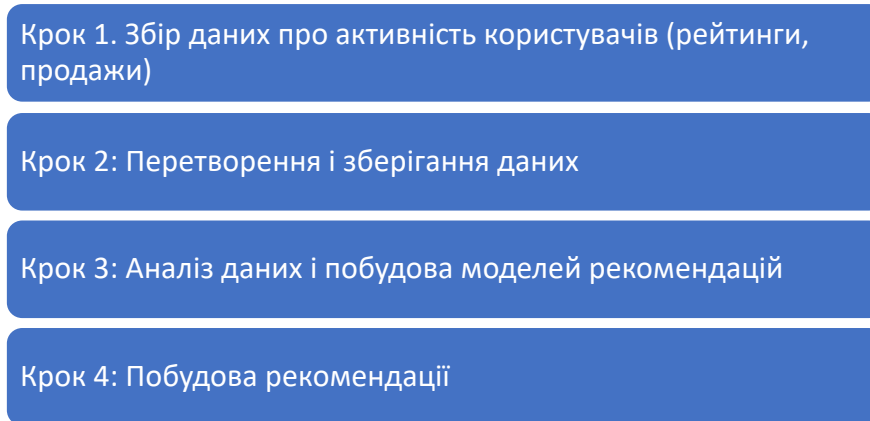


Рисунок 1.1 – Етапи роботи рекомендаційної системи.

На другому етапі відбувається зберігання даних. Згодом обсяг даних стане величезним. Це означає, що має бути доступне сховище, що може масштабуватися. Залежно від типу даних, що збираються, доступні різні типи сховищ.

Аналіз даних. Для використання дані повинні бути деталізовані і проаналізовані. Є кілька різних методів аналізу даних. Це включає:

- аналіз у реальному часі – дані обробляються в момент їх створення;
- пакетний аналіз – дані періодично обробляються;
- аналіз майже в реальному часі – дані обробляються за хвилини, а не за секунди, коли вони не потрібні негайно.

Останній крок – фільтрація. До даних застосовуються різні матриці або математичні правила та формули залежно від того, чи використовується фільтрація рекомендацій спільної, контентної або гібридної моделі. Результатом цього фільтрування є рекомендації.

Розглянемо загальну архітектуру рекомендаційної системи. Для початку розглянемо загальну системну схему для рекомендаційних систем на

наступному рисунку 1.2. Основні компоненти архітектури містять один або кілька алгоритмів машинного навчання.

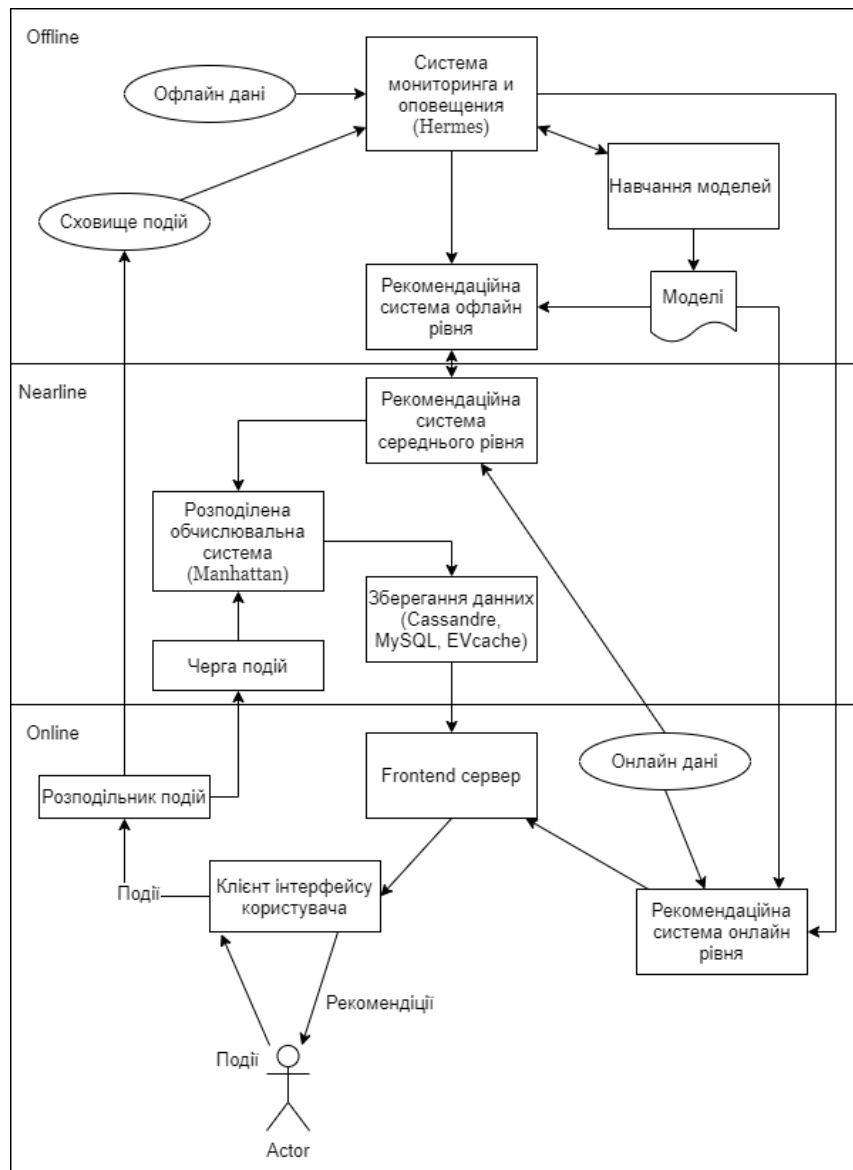


Рисунок 1.2 – Загальна схема рекомендаційної системи

Найпростіше, що можна зробити з даними – зберегти їх для подальшої автономної обробки, що призведе до створення частини архітектури для управління автономними завданнями. Однак обчислення можуть виконуватися в автономному режимі, онлайн або офлайн. Online-обчислення можуть оперативно реагувати на недавні події згенеровані користувачем, але

повинні реагувати на запити системи без затримок. Це може зменшити обчислювальну складність алгоритмів, що використовуються, а також обсяг даних, які можуть бути оброблені.

Offline розрахунки мають менше обмежень на обсяг вхідних даних для обробки та обчислювальну складність алгоритмів, оскільки розрахунки вони виконуються в пакетному режимі з менш суворими вимогами до часу. Однак рекомендація може легко застаріти між оновленнями, тому що найсвіжіші дані не були враховані. Одна з ключових проблем в архітектурі персональних рекомендацій полягає в тому, як безшовно поєднувати та керувати online і offline обчисленнями. Обчислення Nearline – це проміжний компроміс між цими двома режимами, в якому можливо виконувати обчислення, подібні до online, але не вимагаючи їх доступу в реальному часі.

Навчання моделі – це одна із форм розрахунків, яка використовує існуючі дані для створення моделі, яка буде використовуватися під час фактичного обчислення результатів. Інша частина архітектури описує, як різні типи подій та даних повинні оброблятися системою розподілу подій та даних. Пов'язане з цим питання полягає в тому, як об'єднати різні сигнали та моделі, які необхідні в offline, nearline та online -режимах. Нарешті, також необхідно з'ясувати, як поєднати проміжні результати рекомендацій зрозумілим для користувача чином.

Online-обчислення можуть швидко реагувати на події та використовувати найсвіжіші дані. Прикладом може бути складання галереї бойовиків, відсортованих для учасника з використанням поточного контексту. Для online-компонентів діють угоди про рівень обслуговування (SLA), які визначають максимальну затримку процесу при відповіді на запити від клієнтських додатків, поки наш користувач очікує на появу рекомендацій. Це може ускладнити виконання складних і довгих у обчислювальному відношенні алгоритмів для цього підходу. Крім того, чисто online-обчислення можуть не відповідати угоди про рівень обслуговування в деяких обставинах, тому завжди важливо думати про швидкий резервний механізм, такий як

повернення до попередньо обчисленого результату. Online обчислення також означають, що різні задіяні джерела даних також повинні бути доступні онлайн.

З іншого боку, offline обчислення надають більше варіантів алгоритмічного підходу, наприклад складні алгоритми які не обмежені на обсяг даних, що використовуються. Тривіальним прикладом може бути періодичне агрегування статистики мільйонів подій перегляду фільмів для складання базових показників популярності для рекомендацій. Offline системи також мають простіші інженерні вимоги. Наприклад, можуть бути легко виконані SLA зі скороченим часом відгуку, що накладаються клієнтами.

Однак, оскільки offline обробка не має жорстких вимог до затримки, вона не буде швидко реагувати на зміни в контексті або на нові дані. Зрештою, це може призвести до застарілості, що може погіршити досвід користувачів. Для offline обчислень також потрібна інфраструктура для зберігання, обчислення та доступу до великих наборів попередньо обчислених результатів.

Nearline обчислення можна розглядати як компроміс між двома попередніми режимами. У цьому випадку обчислення виконуються точно так само, як і в online випадку. Однак якщо скасовуємо вимогу до показу результатів, щойно вони обчислюються, і замість цього можемо зберігати їх, дозволяючи їм бути асинхронними. Nearline обчислення виконуються у відповідь на події користувача, щоб система могла краще реагувати на запити. Це відкриває двері для потенційно більш складної обробки кожної події. Прикладом є оновлення рекомендацій, щоб відображати, що фільм було переглянуто відразу після того, як учасник починає його дивитися. Результати можна зберігати в проміжному кешуванні або в системі зберігання. Nearline обчислення також є природним параметром для застосування алгоритмів інкрементального навчання.

Більшість обчислень, які необхідно виконати під час запуску алгоритмів машинного навчання персоналізації, може виконуватися у автономному

режимі. Це означає, що можна запланувати періодичне виконання завдань, і їх виконання не обов'язково має бути синхронним із запитом чи поданням результатів. У цю категорію потрапляють два основні види завдань: навчання моделі та пакетне обчислення проміжних чи остаточних результатів. У завданнях навчання моделі збираються відповідні існуючі дані і застосовуємо алгоритм машинного навчання, щоб отримати набір параметрів моделі. Ця модель зазвичай кодується та зберігається у файлі для подальшого використання.

Пакетне обчислення результатів – це описаний вище автономний процес обчислень, у якому використовується існуючі моделі та відповідні вхідні дані для обчислення результатів, які будуть використовуватися пізніше або для подальшої online-обробки, або для прямого подання користувачеві.

Обидві ці завдання вимагають обробки уточнених даних, які зазвичай створюються під час виконання запиту до бази даних. Оскільки ці запити обробляють великі обсяги даних, корисно запускати їх у розподіленому режимі. Netflix використовує внутрішній інструмент під назвою Hermes для опублікування отриманих даних.

Події та розповсюдження даних. Завдання перетворити дані про взаємодію учасників на аналітичні дані, які можна використовувати для покращення досвіду учасників. Для цього потрібно щоб різні програми не тільки забезпечували приємний інтерфейс користувача, але й збирали якомога більше користувальницьких подій. Ці дії можуть бути пов'язані з натисканнями, переглядом, переглядом або навіть вмістом області перегляду в будь-який час. Потім події можна агрегувати, щоб надати базові дані наших алгоритмів.

Події це невеликі одиниці чутливої на час інформації, які необхідно обробляти з мінімально можливою затримкою. Ці події направляються для запуску наступної дії або процесу, такого як оновлення набору результатів, близьких до лінії. З іншого боку, дані як про більш щільні інформаційні одиниці, які, можливо, потрібно обробити та зберегти для подальшого

використання. Тут не така важлива затримка, як якість і кількість інформації. Звичайно, є події користувача, які можна розглядати як події і дані і, отже, відправляти в обидва потоки.

У Netflix потік подій майже реальному часі управляється через внутрішню структуру під назвою Manhattan. Manhattan – це розподілена обчислювальна система, яка є центральною у нашій алгоритмічній архітектурі для рекомендацій.

Мета машинного навчання – скласти індивідуальні рекомендації. Ці результати рекомендацій можуть бути отримані безпосередньо зі списків, які були вираховані раніше, або вони можуть бути згенеровані одразу за допомогою online-алгоритмів. Звичайно, можемо подумати про використання комбінації того, де основна частина рекомендацій обчислюється в offline режимі і додаємо деяку свіжість шляхом подальшої обробки списків за допомогою online-алгоритмів, які використовують події в реальному часі.

У Netflix зберігаються offline та nearline результати в різних репозиторіях для подальшого використання під час запиту: основними сховищами даних є Cassandra, EVCache та MySQL. Кожне рішення має переваги та недоліки в порівнянні з іншими. MySQL дозволяє зберігати структуровані реляційні дані, які можуть бути потрібними для деяких майбутніх процесів за допомогою запитів загального призначення. Однак загальність досягається за рахунок проблем із масштабованістю у розподілених середовищах. І Cassandra, і EVCache пропонують переваги сховищ ключів. Cassandra – це добре відоме і стандартне рішення, коли потрібно розподілене та масштабоване сховище без SQL. Cassandra добре працює в деяких ситуаціях, однак у тих випадках, коли потрібні інтенсивні та постійні операції запису, використовується EVCache щоб бути кращим. Однак ключове питання полягає не так у тому, де їх зберігати, як у тому, як обробляти вимоги таким чином, щоб конфліктуючі цілі, такі як складність запиту, затримка читання/запису та узгодженість транзакцій, відповідали оптимальній точці для кожного варіанту використання.

1.4 Дослідження моделей представлення темпоральних знань

Визначення темпоральних відносин має вирішальне значення для розуміння темпоральних ситуації подій, описаних у дискурсі. Автоматичне виділення темпоральних зв'язків виявилось складним, хоча люди-читачі можуть це зробити дуже легко.

Типування темпоральних відношення складається з опису типу темпоральних впорядкування або відношення між парою темпоральних сутностей, які самі по собі є подіями або часом. Тобто вирішувати, в якому порядку відбуваються події відповідно до тексту.

Як тільки ми зможемо автоматично знайти й позначити темпоральних відносини між подіями та часом у дискурсі, стають доступними потужні методи для покращення взаємодії людини та комп'ютера та автоматичної обробки інформації, що зберігається природною мовою. Наприклад, системи можуть краще відповідати на запитання природною мовою; стає легше створювати кращі резюме дискурсу; робота над додатками судово-медичної експертизи – наприклад:

- створення графіків свідчень свідків у разі злочину або транспортної катастрофи може бути автоматизована;
- можна створювати історії, використовуючи потенційно неповні звіти з кількох джерел;
- і темпоральних вилучення інформації можна використовувати в повсякденному спілкуванні, для організації подій і автоматичного створення календарних зустрічей з особистого спілкування.

«Події» – це прикриваючий термін для ситуацій, які відбуваються або відбуваються. Події можуть бути пунктуальними або тривати протягом певного періоду часу. Події - це предикати, що описують стани або обставини, за яких щось досягається або зберігається.

Оскільки вони можуть описувати дію чи перехід, події часто виражаються дієсловами («Автобус раптово зупинився»). Номіналізована подія — це подія, яка представлена іменником. Події також можуть бути виражені іменниками, предикативами, прикметниками, прийменниковими фразами.

Часова інформація в тексті часто виражається за допомогою фрази, яка точно описує момент або тривалість. Іноді ці точки посилаються на абсолютно однозначний час (закріплений, наприклад, за допомогою календаря), що дуже допомагає при спробі відобразити події з дискурсу на часову шкалу. Також часто такі фрази явно вказують довжину інтервалу. Оскільки вони настільки відверті, ці фрази використовуються, коли тимчасовість є критичною. Таким чином, спроби витягти з дискурсу темпоральну інформацію повинні фіксувати та обробляти ці фрази.

Лінгвістична характеристика темпоральних виразів привела до дискусій і спостережень щодо їх використання та ситуації. Hitzeman виявив, що темпоральні вирази часто використовуються як маркери сегментації дискурсу, і підкреслює їх потенційну неоднозначність. Вони виявили, що інтерпретація даного часового виразу залежить від його синтаксичної позиції. Аналогічно, Bestgen і Vonk показують, що темпоральні вирази, які використовуються як прислівники, допомагають створити сцену для частини дискурсу, забезпечуючи контекст і часові рамки, а також є корисними маркерами сегментації дискурсу, покращуючи розуміння дискурсу. Cohen і Schwer виконують багатомовну характеристику темпоральних маркерів, описуючи такі вирази як такі, що складаються з трьох частин: розмір темпорального сегмента, відстань від темпорального центру (наприклад, опорної точки) і орієнтація, наприклад, майбутнє або минуле.

Для цієї тези «часовий вираз», або *Timex*, – це будь-який вираз, який позначає момент, інтервал або іншу часову область без необхідності покладатися на подію. Кожен інтервал складається з двох точок, між якими він отримує. Наприклад, «24 серпня 1997 року», «два тижні» і «зараз» – це

темпоральні вирази; «після грози» це не темпоральні вирази. Hobbs і Rap визначають «належний інтервал» як такий, коли початкова точка знаходиться перед кінцевою точкою. Відповідно до цього визначення, ця теза розглядає як інтервали лише «власний інтервал», тобто не розпізнається мінімальна атомна тривалість, і немає квантування часу в хрононі. Швидше, темпоральні сутності описуються нескінченно малими точками, які їх об'єднують.

Потрібно з'ясувати, де ці вирази зустрічаються в тексті, і зрозуміти дещо з їхньої семантики, перш ніж з'єднати їх за допомогою темпоральних відносин.

Види вираження часу:

– абсолютний – де в тексті явно вказано однозначний час. Залежно від деталізації інтервалу, текст містить достатньо інформації, щоб звузити точку або інтервал безпосередньо до одного окремого входження. Це на відміну від часу, який, хоча точний і, можливо, легко для людей закріпити на календарі, покладається на зовнішнє посилання. Наприклад, інтервал «четвер, 1 жовтня 2009 року», буде вважатися абсолютним, але «тиждень після» – ні, інформація не є явною або зберігається в одному місці. Останній вираз передбачає опору на деякий зовнішній опорний час. Абсолютні вирази іноді також відомі як повні вирази часу;

– дейктичний – випадки, коли за відомим часом висловлювання можна визначити період, про який йдеться. Ці вирази часу визначають тимчасову відстань і напрямок від часу висловлювання. Можна було побачити, як починається журнальний бюлетень «Два тижні тому ми ще були в Сайгоні», цей вираз залишає незрозумілий імпліцитний час мовлення, який можна було б сміливо вважати датою написання цієї роботи. Більш поширені приклади включають «завтра» і «вчора», обидва зміщені від часу мовлення; Щоб описати це дейктичні темпоральні вирази представляють ситуації, коли еталонний час і час мовлення однакові;

– анафоричний – коли мова та час опори не знаходяться в одній точці. Анафоричні темпоральні вирази складаються з трьох частин – темпоральна

відстань (наприклад, «4 дні»), часовий напрямок (минуле чи майбутнє) і прив'язка, з якої застосовуються відстань та напрямок. Якорем для анафоричних темпоральних виразів є поточний еталонний час відповідно до моделі Reichenbach's. Приклади фраз включають «наступний тиждень», «той вечір» або «кілька годин пізніше», жодна з яких не може бути прив'язана, навіть якщо час їхнього виступу відомий;

– тривалість – тривалість описує інтервал, обмежений початком і кінцем, де відстань між ними відома, але сам вираз не можна розмістити в жодній зовнішній системі часу (наприклад, календарі). Тривалість зазвичай включає одиницю часу як головний маркер; наприклад, «дев'яносто хвилин», це одна тривалість *time*. Цей тип темпоральних виразів легко сплутати з дейктичними виразами;

– набір – час, який регулярно повторюється, наприклад «кожне Різдво» або «кожен вівторок». Вони зазвичай мають регулярний інтервал між подіями і зберігаються протягом тривалого часу або описують точкову подію («кожен інший четвер о 16:30»).

Типи темпоральних відносин. Темпоральні алгебри та логіка дозволяють виводити зв'язки між подіями на основі їх зв'язку з іншими часами та подіями, використовуючи набір правил. Ці правила залежать від конкретного набору типів зв'язків подій і набору типів відносин. Інтервальна, точкова та напівінтервальна логіка доступна. Спираючись на STAG (Рекомендації щодо темпоральних анотацій Шеффілда), TimeML визначає власний набір інтервальних відношень на основі інтервальної алгебри Allen; точкова алгебра може бути корисною для швидкого міркування; напівінтервальне міркування полегшує тягар специфікації, необхідної, коли потрібно знайти обидві точки інтервалу, щоб уникнути надмірної специфікації під час роботи з подіями, описаними природною мовою.

Інтервальними алгебрами вважаються ті, що визначають типи відношень між інтервалами та набір аксіом для роботи з цими відношеннями; інтервал має початок і кінець. Деякі темпоральні логіки використовують точки

замість інтервалів. Для інтервальної логіки точкова подія може бути представлена інтервалом, початок і кінець якого відбуваються одночасно; правильний інтервал — це інтервал, коли кінець настає після початку (Hobbs і Pan).

Темпоральна логіка має справу з міркуваннями про відносини, які існують між інтервалами. Ранні приклади темпоральної логіки включають обчислення Пріора для логічного обчислення модального часу (Prior) і модель Bruce, яка також включає аксіоми для міркування подій у часовій системі.

На таблиці 1.3 представлено кілька темпоральних інтервальних алгебр, кожна з яких має певну мету, переваги та недоліки. Далі ми розглянемо більш подібна кожному з цих алгебр.

Застосування темпоральної логіки можна знайти в багатьох областях інформатики, включаючи перевірку та тестування чутливих до часу частин комп'ютерних програм, у наданні темпорального представлення даних для систем штучного інтелекту та для представлення темпоральної семантики в обробці природної мови.

Проста темпоральна логіка. Можна описати багато основних відношень між інтервалами, використовуючи лише три відношення — «раніше», «включає» і «одночасно». Якщо ми зустрінемо щось подібне до того, як «Я помився після чистки каналізації», якщо події позначено як E , ми можемо просто змінити порядок аргументів, щоб виконати $E_{\text{почистив}}$ раніше $E_{\text{помився}}$. У рамках більшого дослідження темпоральних міркувань щодо інформації, знайденої в дискурсі, Setzer et al. вводить мінімальну логіку, засновану на трьох простих відносинах, ніж вимагає лише десять правил для темпорального висновку.

Таблиця 1.3 – Моделі представлення темпоральних знань

Модель	Особливості	Переваги	Недоліки
Темпоральна логіка	Ключові відношення: «раніше», «включає» «одночасно»	Простота цієї системи робить її легко реалізованою і легкою для продумування.	Однак набір лише з трьох відношень невеликий, і темпоральні відношення, виражені природними мовами, не можуть бути представлені.
Логіка Allen	Описує набір темпоральних відносин, які можуть існувати між будь-якою парою подій.	Дає можливість описати більш складніші темпоральні відношення, виражені природними мовами	Не може описувати миттєві події
Логіка Freksa	Логіка Freksa доповнює логіку Allen. Вводить нове поняття – напівінтервал	Вирішує недоліки логіки Allen	Логіка складна для реалізації
Логіка на основі подій	Логіка працює лише з упорядкуванням окремих точок і не відповідає концепції інтервалу	Менш схильні до проблеми надмірної специфікації. Дуже швидка в обробці	Використовується в сфері повністю автоматизованого міркування

Простота цієї системи робить її легко реалізованою і легкою для продумування. Однак набір лише з трьох відношень невеликий, і темпоральні відношення, виражені природними мовами, можуть бути більш точно представлені за допомогою більш широкого набору типів темпоральних

відношень. Наприклад, якщо два інтервали перекриваються, але не мають спільних початкових або кінцевих точок не включає або одночасно не є достатньо точним, щоб описати їх тимчасове відношення.

Логіка часового інтервалу. Інтервальна логіка Allen описує набір темпоральних відносин, які можуть існувати між будь-якою парою подій. Allen вводить концепцію подій (представлених у вигляді інтервалів) як вузлів у графі, де ребра, що з'єднують вузли, представляють зв'язок між двома інтервалами.

Якщо неясно, що між парою подій має існувати один тип зв'язку, для позначення краю з'єднання використовується диз'юнкція всіх можливих типів зв'язку. Крім того, Allen надає алгоритм для виведення зв'язків між раніше не пов'язаними вузлами.

Відношення наведено в таблиці 1.4. Кожен з них дає певну конфігурацію початкових і кінцевих точок інтервалу. На основі цього надається таблиця транзитивності для висновку нових відносин між інтервалами, які містять загальні події.

Таблиця 1.4 – Перелік відношень в логіки Allen

before	де А закінчується перед початком В
after	де А починається після закінчення В
during	де А починається і закінчується, поки В триває
contains	обернено during
overlaps	де А починається перед В і закінчується під час В
overlapped-by	обернене overlaps
meets	де А закінчується в точці В починається
met-by	обернене meets
starts	де А і В мають однакову початкову точку, але А закінчується раніше, ніж В
started-by	як починається, але В закінчується першим
finishes	де А та В мають спільну кінцеву точку, але А починається пізніше (і, таким чином, коротше)
finished-by	як finishes, але В є коротшим/молодшим інтервалом
equal	де А і В починаються і закінчуються одночасно

Наприклад: речення «Червона Шапочка вирушила до міста. Перед від'їздом одягла накидку. Потім вона жила довго і щасливо.» може бути представлена 3-вузловим графом (події E1, E2 і E3) з двома позначеними ребрами (L1 і L2) (рисунок 1.3).

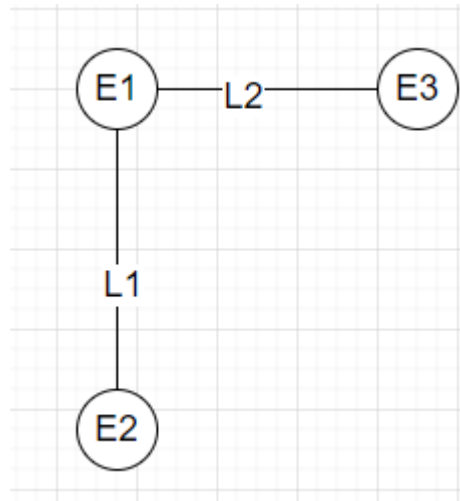


Рисунок 1.3 – Представлення темпоральних відношень у вигляді графу
де E1 – «вирушила»;
E2 – «жила довго і щасливо»;
E3 – «одягла накидку»;
L1 – E2 after E1;
L2 – E3 before E1.

Цей поточний графік залишає зв'язок між E3 та E2 невизначеним. Умови розповіді та людська інтуїція підказують нам, що ми повинні використовувати лінійну модель часу і припускають, що все, що станеться до того, як дівчина вирушить у дорогу, має статися до того, як вона «заживе довго і щасливо». У цьому випадку ми можемо формально описати це знання за допомогою правил:

$$\forall x, c : x \text{ after } c \rightarrow c \text{ before } x$$

$$\forall x, c, b : x \text{ before } c \& c \text{ before } b \rightarrow x \text{ before } b$$

Логіка Allen вважалася захоплюючою, оскільки вона була реалізована на той час, на відміну від інших темпоральних логік, а також була виразною. З тих пір вона була прийнята логіками, спільнотою верифікації та тестування та тими, хто цікавиться часом у мові. Для подальшого огляду логіки часових інтервалів слід дивитися Goranko et al та Galton.

Логіка часового інтервалу не ідеальна. Визначення послідовності в будь-якому, окрім найменших сценаріїв, швидко стає нерозв'язним і є NP-важким. Проблеми виникають при роботі з миттєвими подіями. Невідповідності з'являються, коли подіям дозволяється мати нульову тривалість, а система явно не структурована, щоб мати справу з цими Allen and Hayes.

Напівінтервали – це інтервали, де потрібно описати лише одну межу (наприклад, початкову або кінцеву точку). Стверджується, що такі розслаблені визначення в порівнянні з повністю описаними інтервалами можуть краще відображати відносини, виражені природною мовою. У цьому розділі ми обговорюємо недоліки алгебри часових інтервалів і вводимо систему міркувань з напівінтервалами.

Деякі звичайні завдання типізації відносин важко виконати з інтервальними відношеннями. Наприклад: статті новин зазвичай мають час створення документа або дату публікації, яка з'являється в метаданих документа та у вигляді часу в основній частині дискурсу. Вони часто містять принаймні кілька подій, початок яких описується в минулому часі. У цих випадках важко визначити, чи закінчується кінцева межа події на даті публікації чи продовжується після нього. Ця невизначеність ускладнює встановлення відношення з набору Allen для двох інтервалів.

З цією метою Freksa пропонує темпоральну алгебру, орієнтовану на тих, хто має справу з природною мовою. Він спирається на попередню фундаментальну роботу з логіки, яка справляється з невизначеністю часу, як описано мовою. Поки ми знаємо, що інтервали починаються, перш ніж закінчуються, ми можемо почати описувати відносини між напівінтервалами як диз'юнкції відносин Allen. Швидко помічено, що конкретні відносини Allen

виникають разом, коли мова йде про неповні знання про події. Freksa підсумовує ці поняття, визначаючи терміни для концептуальних сусідів: «два відносини між парами подій є концептуальними сусідами, якщо вони можуть бути безпосередньо перетворені один в одного шляхом безперервного деформування (тобто скорочення, подовження, переміщення) подій (у топологічному сенсі).

Система Freksa бореться з невизначеністю знань, які пов'язують дві події, і дозволяє нам отримувати інформацію з тексту, який може не описувати всі інтервали повністю. Використовуючи групи відносин, які зазвичай зустрічаються під час умовиводу, Freksa описує темпоральну алгебру, позначаючи певні групи відносин Allen як відношення самотійно. Алгебра визначає таблицю транзитивності. Таблиця 1.5 базується на групах зв'язків, які часто зустрічаються.

Таблиця 1.5 – Перелік відношень в логіки Freksa

before	A закінчується до початку B
after	A починається після закінчення B
includes	A починається до і закінчити після B
is_inciuded	A відбувається між початком і завершенням B
during	A відбувається протягом тривалості B
duringjnv	A триває, в якій виникає B
simultaneous	A і B відбуваються одночасно
iafter	A відбувається відразу після B
ibefore	A відбувається безпосередньо перед B
identity	A і B - це одна і та ж подія / час
begins	A починається в той же час, що і B, але закінчується першим
ends	A починається після B, але вони закінчуються одночасно
begun_by	A починається в той же час, що і B, але триває довше
ended_by	A починається до B, але вони закінчуються одночасно

Наприклад, з множини Freksa, відношення A старше B застосовується, коли початкова точка A відбувається перед початковою точкою B; на їхні кінцеві точки не приділяється увага, тому застосовується будь-який з A

[before, ibefore, ended_by, inciuded] В. З цього прикладу відразу стає зрозумілим принаймні один приклад англійською мовою, де логіка напівінтервалів була б корисною.

Темпоральну логіка на основі точок. Темпоральна логіка на основі точок працює лише з упорядкуванням окремих точок і не відповідає концепції інтервалу. Вони менш схильні до проблеми надмірної специфікації, яку мають повні інтервальні алгебри. Можна розкласти інтервали на їхню точку початку та кінця. Для опису бінарних відносин між цими точками потрібні лише оператори рівності та пріоритету. Точкові алгебри можуть бути дуже швидкими в обробці, функція, яку використовують такі інструменти, як SputLink та SAVaT. Вони також краще підходять для міркування на основі графіків про темпоральні структури в тексті. Однак людям складніше коментувати, використовуючи точки замість інтервалів, а семантику темпоральних відносин у тексті краще відображати за допомогою інтервальних або напівінтервальних міток. Через ці причини, а також через те, що темпоральна анотація вже є складним і виснажливим завданням для людей-анотаторів, точкове міркування та темпоральна логіка, як правило, обмежені сферою повністю автоматизованого міркування.

1.5 Постановка задачі кваліфікаційної роботи

Сучасні підходи до побудови рекомендацій не враховують зміни користувацьких вподобань з часом а також схильні перекручування рейтингів у результаті атак шилінгів. Доцільно враховувати темпоральну динаміку вимог користувачів і видалення аномальних вхідних даних що до поведінки користувачів.

Метою даної роботи є дослідження методів побудови пояснень в рекомендаційній системі з використанням темпоральних правил.

Об'єкт дослідження – процес побудови пояснень в рекомендаційній системі.

Предмет дослідження – методи побудови пояснень в рекомендаційних системах.

У даній кваліфікаційній роботі вирішуються наступні задачі:

- аналіз особливостей пояснень в рекомендаційних системах;
- дослідження методів побудови пояснень;
- дослідження моделей представлення темпоральних знань;
- удосконалення методу побудови пояснень з використанням темпоральних правил;
- розробка програмного модулю формування пояснень в рекомендаційній системі;
- експериментальна перевірка удосконаленого методу формування пояснень.

2 УДОСКОНАЛЕННЯ МЕТОДУ ПОБУДОВИ ПОЯСНЕНЬ З ВИКОРИСТАННЯМ ТЕМПОРАЛЬНИХ ЗНАНЬ

2.1. Формальне представлення темпоральних знань для побудови пояснень

Порівняльний аналіз основних підходів до побудови пояснень дозволяє зробити висновок, що описи повинні містити причинно-наслідкові зв'язки, які, швидше за все, відображають послідовність прийняття рішень в інтелектуальній системі. Такі залежності визначають як послідовність прийняття рішень у часі, так і безпосередній вплив параметрів цього процесу на отриманий результат з урахуванням явного або неявного представлення часу. Відповідно, темпоральне представлення причинно-наслідкових залежностей має враховувати:

- бінарні та ймовірнісні зв'язки між діями рішення;
- безпосередній вплив значень параметрів, що характеризують стан інтелектуальної системи, на отриманий результат.

Структурування узагальненого причинно-наслідкового зв'язку, що відповідає зазначеним вище вимогам, представлено в таблиці 2.1.

Інтеграція елементів, наведених у таблиці 2.1 на основі темпоральних підходів [33-35], передбачає розгляд причинно-наслідкового зв'язку в трьох аспектах:

- як причинний зв'язок між послідовними подіями;
- як імовірнісний зв'язок між подіями, станами, діями, що відбулися в різні моменти часу;
- як відображення інформаційного зв'язку між елементами інтелектуальної системи, що призводить до зміни її станів, такий зв'язок реалізується через передачу інформації між цими елементами.

Таблиця 2.1 – Елементи узагальненої причинно-наслідкової залежності

Елемент	Отримане значення	Ключові відмінності
Безпосередній причинний зв'язок між діями для отримання результатів в інтелектуальній системі	{true, false}	Залежність у поясненні має бути правдивою
Імовірнісний зв'язок між діями для отримання результату	Імовірнісністьне	Включення залежності в пояснення підвищує ймовірність істинності цього пояснення
Зв'язок між станом інтелектуальної системи та результатом	{true, false}	Залежність у поясненні визначає параметри стану інтелектуальної системи, які суттєво впливають на результат її роботи.

Перший компонент причинності $r_{i,j}^{(1)}$ виглядає так:

$$r_{i,j}^{(1)}: p_i \rightarrow p_j \mid t_j > t_i \quad (2.1)$$

де p_i, p_j – факти, між якими встановлено причинно-наслідковий зв'язок;

t_i, t_j – моменти часу виникнення фактів p_i, p_j відповідно.

Факти p_i, p_j для причинно-наслідкової залежності (2.1) є бінарними:

$$p_i = \text{true} \vee \text{false} \quad (2.2)$$

Причинно-наслідковий зв'язок для першого компонента визначається на основі опису причинно-наслідкового зв'язку, розглянутого в [36]:

$$\begin{aligned} \forall p_j \exists p_i : p_j = \text{true} \Leftarrow p_i = \text{true} \\ |\forall p_k \neq p_i p_j = \text{false} \Leftarrow p_k = \text{true} \end{aligned} \quad (2.3)$$

де p_i, p_j – факти, між якими встановлено причинно-наслідковий зв'язок.

Тобто існує причинно-наслідковий зв'язок між фактами p_i, p_j тільки при появі p_i призводить до виникнення p_j . Істинність інших фактів p_k не впливає на появу факту p_j . Додатковою умовою причинно-наслідкової залежності є те, що факти p_i, p_j повинні бути впорядковані в часі. Відповідно, залежність $r_{i,j}^{(1)}$ є істинним у випадку істинності фактів p_i, p_j та темпоральна послідовності цих фактів згідно (1):

$$r_{i,j}^{(1)} = \begin{cases} \text{true, if}(p_j = \text{true} \wedge p_i = \text{true}) \wedge t_j > t_i \\ \text{false, otherwise} \end{cases} \quad (2.4)$$

де p_i, p_j – факти, між якими встановлено причинно-наслідковий зв'язок.

Ключовою особливістю залежностей форми (1) є те, що вони визначають обмеження на можливі пояснення. Тобто причинно-наслідкові залежності $r_{i,j}^{(1)}$ повинні виконуватися для всіх можливих пояснень в інтелектуальній системі.

Другий компонент причинності:

$$r_{i,j}^{(2)} : p_i G p_j | t_j > t_i \quad (2.5)$$

де G – тип зв'язку між фактами p_i, p_j .

Імовірнісна залежність $r_{i,j}^{(2)}$ має дві ключові відмінності від залежності (1): задає різні види комунікації фактів з урахуванням темпорального аспекту;

має імовірнісний характер. Тип зв'язку G може визначити час, послідовність та умови істинності факту p_j після істинного факту p_i :

$$G \in \{T, O, U\} \quad (2.6)$$

де T – набір темпоральних залежностей з використанням абсолютних значень часу;

O – набори темпоральних залежностей, заснованих на апріорно заданому порядку фактів у часі;

U – набір темпоральних залежностей, що використовують умови для визначення порядку фактів у часі.

У першому випадку абсолютне значення часу або інтервалу часу, коли факт p_j стає істинним після встановлення факту p_i істинним.

Такий зв'язок дає змогу пояснити темпоральні властивості процесів у предметній області та, наприклад, врахувати технологічні обмеження за часом використання ресурсу, інтервалом обслуговування тощо.

У другому випадку час задається у відносному порядку виникнення фактів. Наприклад, факт p_j може стати істинним відразу після факту p_i , або через декілька проміжних фактів. Якщо факт p_j істинний відразу після факту p_i , цей зв'язок описує детальний ланцюг причинно-наслідкових зв'язків між діями інтелектуальної системи. Якщо необхідно виявити лише ключові залежності, які привели до результату, то проміжні факти не враховуватимуться.

Таким чином, з'єднання $r_{i,j}^{(2)}$ дає можливість побудувати спрощений і доступний як пояснення опис рекомендацій в інтелектуальній системі. Слід зазначити, що $r_{i,j}^{(2)}$ визначає причинно-наслідковий зв'язок лише для підмножини результатів інтелектуальної системи. Тому таке посилання дає правдоподібне пояснення отриманим рекомендаціям та висновкам. Узгоджена вага відображає ймовірнісний характер зв'язку в цій залежності:

$$r_{i,j}^{(2)} = \begin{cases} \omega_{i,j}^{(2)}, & \text{if}(p_j = \text{true} \wedge p_i = \text{true}) \wedge t_j G t_i \\ 0, & \text{otherwise} \end{cases} \quad (2.7)$$

де p_i, p_j – факти, між якими встановлено причинно-наслідковий зв'язок;

$\omega_{i,m}^{(t)}$ — вага, що характеризує ймовірність зв'язку між фактами;

Можливість узгодження шкал визначає необхідність їх використання замість значення ймовірності використання залежності $r_{i,j}^{(2)}$. Угода полягає у визначенні таких масштабів, які дають змогу отримати правильні пояснення відомих результатів інтелектуальної системи. Одна і та ж залежність може належати до кількох пояснень. Отже, відносна загальна вага правил повинна відповідати ймовірності використання кожного пояснення E_n для відомих результатів інтелектуальної системи:

$$\frac{\sum_{E_n} \omega_{i,j}^{(2)}}{\sum_E \omega_{i,j}^{(2)}} \rightarrow P(E_n) \quad (2.8)$$

де $E = \{E_n\}$ – набір пояснень, які були використані для опису результатів в інтелектуальній системі.

Відповідно до виразу (2.8) процедура визначення ваг правил є процедурою навчання з учителем. Результатом навчання повинні бути такі ваги правил, які встановлюють відому ймовірність використання кожного пояснення E_n .

Третій компонент репрезентації причинності $r_{i,j}^{(3)}$ визначає умови, за яких факт p_j стає істинним. Визначено властивості $r_{k,j}$ цього факту, що суттєво вплинуло на рішення, отримане в інтелектуальній системі. Кожна властивість $r_{k,j}$ стає істинною у разі отримання k – параметра заданого значення. У цьому випадку для істинності причинно-наслідкового зв'язку достатньо істинності певної підмножини властивостей:

$$r_{i,j}^{(3)} = \begin{cases} \text{true, if } (\forall p_k \in K \wedge p_{k,j} = \text{true}) \\ \text{false, otherwise} \end{cases} \quad (2.9)$$

де K – набір властивостей, які мають ключовий вплив на виникнення факту $p_{k,j}$.

Таким чином, залежності $r_{i,j}^{(1)}$ і $r_{i,j}^{(2)}$ призначені для пояснення послідовності дій для отримання результатів інтелектуальною системою. Залежність $r_{i,j}^{(3)}$ відображає значний вплив параметрів стану інтелектуальної системи на отриманий результат. Причинність у поясненні вичерпно представлена залежностями (2.4), (2.7), (2.9).

2.2 Удосконалений метод побудови пояснень з використанням темпоральних правил

У цій роботі ми використовуємо адаптацію темпорального правила на основі підходу, запропонованого в [37]. Покращений метод формування пояснень має додатковий крок для фільтрацій викидів.

Викиди – це екстремальні значення, які відхиляються з інших спостережень за даними, можуть вказувати на мінливість у вимірах, експериментальні помилки чи новизну.

Найчастіші причини викидів у наборі даних:

- помилки введення даних (людські помилки);
- помилки вимірювання (приладові помилки);
- експериментальні помилки (витяг даних або помилки планування/виконання експерименту);
- умисний (фіктивні викиди, зроблені для перевірки методів виявлення);

- помилки обробки даних (маніпуляції з даними або ненавмисні зміни набору даних);

- помилки вибірки (вилучення або змішування даних із неправильних або різних джерел);

- природний (не помилка, новинки у даних).

У сфері продаж виброси можуть бути спричинено:

- людські помилки, коли вводять неправильну кількість замовленого товару;

- завищення продаж, коли навмисно завищують популярність товарів;

- зменшення продаж із-за відсутності товару на складі.

Виброси у вхідних даних можуть негативно вплинути на якість побудови пояснень до рекомендацій і на довіру користувача до системи взагалі.

В удосконаленому методі побудови пояснень пропонується виконувати фільтрацію вибросів за допомогою методу Z-оцінка. В якості вхідних даних пропонується використовувати показник зміни кількості проданих товарів між послідовними фіксованими інтервалами часу. Цей показник може коректно показувати як плавне збільшення продаж товару (сезоні товари), так і товари зі стабільною кількістю продаж. В той же час дозволяє легко виявити періоди з випадками. На рисунку 2.1 показано обсяг продаж товару протягом часу.

Цей товар доволі популярне і користуються постійним попитом. При використанні графіку продажу не можливо виявити аномальне збільшення продаж у четвертому періоді часу. Але на другому графіку, де показано динаміку зміни продаж, дуже чітко видно аномальний виброс росту продаж.

Z-оцінка визначає положення вихідної оцінки з погляду її відстані від середнього значення при вимірі в одиницях стандартного відхилення. Z-оцінка позитивна, якщо значення вище за середнє, і негативна, якщо нижче за середнє.

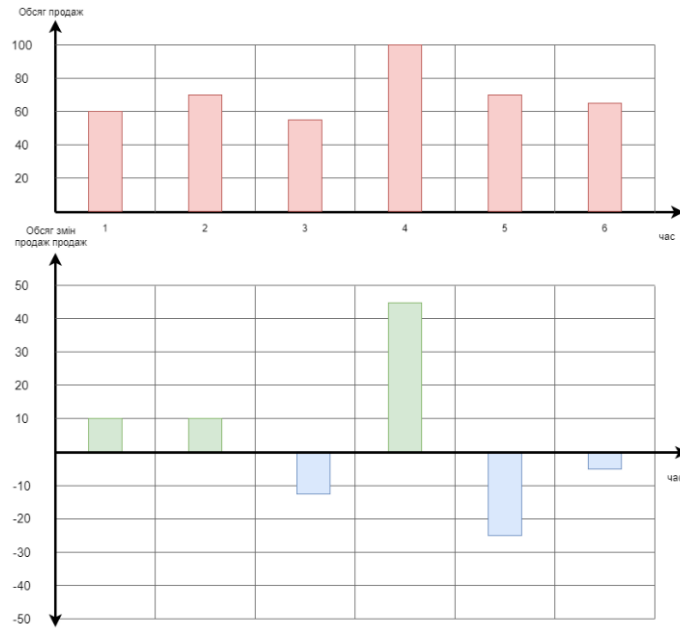


Рисунок 2.1 – Графік продажу товарів і динаміки зміни продаж протягом часу

Z-оцінка також відома як стандартна оцінка, оскільки дозволяє порівнювати оцінки різних типів змінних шляхом стандартизації розподілу. Стандартний нормальний розподіл - це розподіл нормальної форми із середнім значенням 0 та стандартним відхиленням.

Формула для розрахунку Z-показника:

$$Z = \frac{X - \mu}{\sigma} \quad (2.10)$$

де X – вихідний показник;

μ – середнє значення для сукупності;

σ – стандартне відхилення для сукупності.

Значення Z-показника показує, скільки стандартних відхилень ви відійшли від середнього. Якщо Z-оцінка дорівнює 0, це середнє значення.

Позитивний Z-бал вказує на те, що вихідний бал вище середнього. Наприклад, якщо z-оцінка дорівнює +1, це на 1 стандартне відхилення вище за

середнє. Негативний z-показник показує, що вихідний результат нижчий за середній. Наприклад, якщо z-оцінка дорівнює -2, це на 2 стандартні відхилення нижче середнього.

Темпоральні правила засновані на темпоральній модальній логіці, яка визначає істинність пропорційних змінних щодо часу. Правила відрізняються використанням ваг для оцінки порядку подій або станів у часі. Застосування вагових коефіцієнтів правила дозволяє перейти від двійкової до числової оцінки темпоральних змін.

Темпоральне правило являє собою зміну чисельних оцінок двох груп подібних подій $e_{i,j}$ з часом. Наприклад, правило може відображати зміни інтересу користувачів до конкретних предметів (товарів, послуг) протягом певного періоду часу ΔT . При явному зворотному зв'язку події відбуваються під час встановлення рейтингів. Зміни в уподобаннях представлені зміною рейтингів. У разі використання неявного зворотного зв'язку зміна інтересу користувача до товару описується зменшенням або збільшенням кількості покупок зазначеного товару за пару проміжків часу $(\Delta\tau_m, \Delta\tau_M) \in \Delta T$ такої ж тривалості. Другий інтервал пари відповідає моменту побудови пояснення.

Темпоральне правило $r_{i,m}$ для i -об'єкта пов'язує числові оцінки множин подій $E_i = \{e_{i,j}\}$ на інтервалах $\Delta\tau_m$ і $\Delta\tau_M$ і має такий вигляд:

$$r_{i,m} = q_{i,m} \omega_{i,m}^{(t)} q_{i,M}, m = \overline{1, M-1} \quad (2.11)$$

де $q_{i,m}$ і $q_{i,M}$ – числові оцінки множин подій на інтервалах $\Delta\tau_m$ і $\Delta\tau_M$ відповідно;

t – тип правила;

$\omega_{i,m}^{(t)}$ — вага правила.

Числова оцінка є функцією подій $e_{i,j}$:

$$q_{i,m} = f(E_i): \forall e_{i,j} \in E_i \exists \tau_i \in \Delta\tau_m \quad (2.12)$$

де τ_j – момент настання події $e_{i,j}$.

Для рекомендаційних систем числова оцінка визначає ступінь зацікавленості користувача в i -цільовий елементі. При використанні неявного зворотного зв'язку оцінка $q_{i,m}$ визначається як кількість покупок i -товарів за інтервал часу $\Delta\tau_m$. У разі явного зворотного зв'язку, чисельна оцінка встановлює середню оцінку i -позиції за вибраний інтервал часу.

У цьому методі використовуються тимчасові правила типу F (Future), оскільки тимчасовий оператор типу F дозволяє пов'язати пару довільних моментів в один потік часу. Поєднання цих правил дозволяє описати зміну продажів для всіх моментів, що належать до заданого інтервалу часу.

Загальний вигляд правила r типу F з вагою ω між моментами покупок t_m і t_s часу має вигляд:

$$r = n_m F^{(\omega)} n_s \quad (2.13)$$

де n_m і n_s - кількість покупок у момент часу t_m і t_s відповідно.

Використовуючи тимчасові правила, метод описує динаміку зміни споживчого попиту за заданий інтервал часу ΔT . Кінцем цього інтервалу є момент часу t_{now} . Тривалість ΔT залежить від типу товарів і особливостей предметної області споживання і повинна враховувати цикли зміну попиту споживачів. Для товарів, які продаються циклічно, 3-4 рази на рік, інтервал розрахунку повинен становити кілька місяців. Прикладами такої продукції є туристичні тури, одяг.

Період ΔT ділиться на рівні послідовність інтервалів $\Delta\tau_m$. Кожен інтервал $\Delta\tau_m$ відображають рівень деталізації часу (години, доби тощо). Для кожного часового інтервалу $\Delta\tau_m$. Проводиться розрахунок кількості продаж для кожного товару. На базі цих розрахунків формується набір F - тимчасових правил $r_{j,m}$, які з'єднують кожен інтервал $\Delta\tau_m$ зі скінченним інтервалом часу $\Delta\tau_m$. Вага кожного сформованого правила залежить від величини зміни

покупок між інтервалами: чим більше різниця у кількість проданого товару, тим більше вага правила.

Приклад розглянутих тимчасових правил для одного елемента i_j наведено на рисунку 2.2.

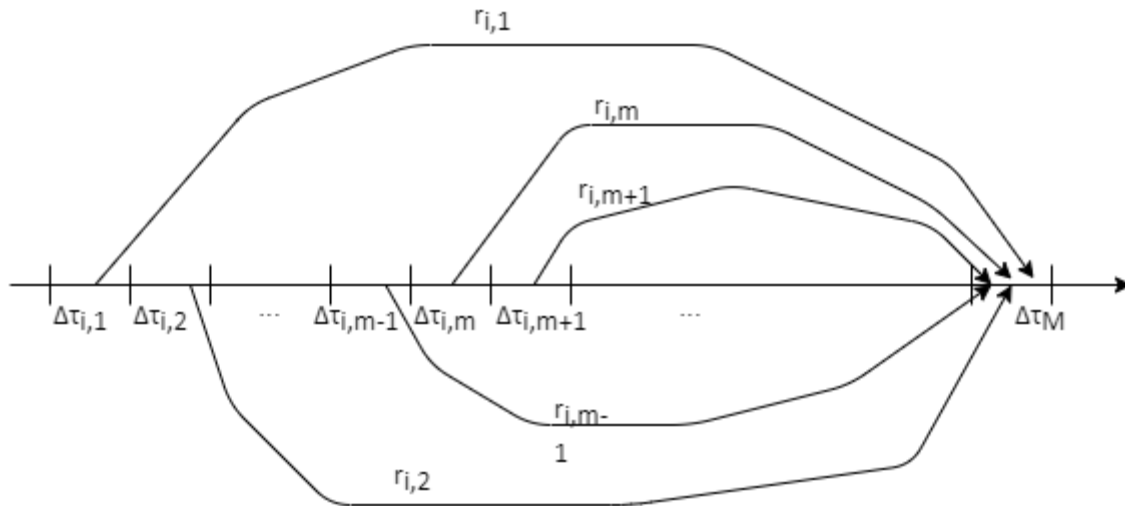


Рисунок – 2.2 Часові правила r_{jm} щодо продажу товару i_j за період часу ΔT

Кожне правило r_{jm} описує зміну продажів товару i_j за пару інтервалів часу ($\Delta\tau_m, \Delta\tau_M$). Увесь набір сформованих правил $R_j = \{r_{jm}\}$ описує вид зміни обсягів продажів товарів за певний проміжок часу ΔT .

Метод як вхідні дані використовує: журнал продажів L ; період часу для розрахунку пояснень ΔT ; час грануляції рівень $\Delta\tau$; підмножина пунктів I рекомендації, для яких генеруються пояснення.

Кожен запис u із журналу продажів U містить чотири параметри (користувач u_k , позиція i_j , кількість придбаних позицій n_j , мітка часу $\Delta\tau_n$). Період часу ΔT залежить від категорії елементів, для яких генеруються рекомендації.

Метод включає наступні етапи.

Етап 1. Вибір підмножини $U_{\Delta T}$ записів з журналу продажів.

Етап 2. Фільтрація журналу продаж від викидів.

Крок 2.1 Розрахунок динаміки зміни продаж товарів для кожного інтервалу часу.

Крок 2.2 Розрахунок середнього і середньоквадратичного значення динаміки зміни продаж.

Крок 2.3 Розрахунок Z-показника для кожного інтервалу часу.

Крок 2.4 Виключення інтервалів з викидами.

Етап 3. Перетворення журналу для заданого рівня грануляції Δt .

Крок 3.1 Нормалізація кількості продаж товарів.

Крок 3.2. Розбиття періоду часу на інтервали.

Крок 3.3. Підсумування покупок для кожного товару.

Етап 4. Побудова набору R_j незважених тимчасових правил.

Етап 5. Розрахунок набору ваг W .

Етап 6. Розрахунок темпоральної динаміки за період часу ΔT .

Етап 7. Формування пояснень

Етап 1. Вибір підмножини $L_{\Delta T}$ записів $l=(u_k, i_j, n_j, \tau_n)$ журналу продажів L для розрахунку тимчасової динаміки з підмножини товарів I за заданий період часу ΔT . Отримана підмножина:

$$U_{\Delta T} = \{(u_k, i_j, n_j, \tau_n): \forall n \tau_n \in \Delta T, \forall j i_j \in I\} \quad (2.14)$$

де u_k – данні користувача;

n_j – кількість придбаних позицій;

$\Delta \tau_n$ – мітка часу $\Delta \tau_n$.

Етап 2. Фільтрація журналу продаж від викидів.

Крок 2.1 Розрахунок динаміки зміни продаж товарів для кожного інтервалу часу.

$$d_{i,i+1} = n_{i+1} - n_i \quad (2.15)$$

де n_{i+1}, n_i – кількість проданих товарів за інтервал часу $\Delta\tau_{i+1}, \Delta\tau_i$ відповідно

Крок 2.2 Розрахунок середнього і середньоквадратичного значення динаміки зміни продаж.

$$\bar{d} = \frac{1}{n} \sum_{i=1}^n d_{i,i+1} \quad (2.16)$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (d_{i,i+1} - \bar{d})^2} \quad (2.17)$$

де \bar{d} – середнє значення динаміки зміни продаж;

σ – середньоквадратичного значення динаміки зміни продаж.

Крок 2.3 Розрахунок Z-показника для кожного інтервалу часу.

$$Z_i = \left| \frac{d_{i,i+1} - \bar{d}}{\sigma} \right| \quad (2.18)$$

Крок 2.4 Виключення інтервалів з викидами. Виключаймо інтервали часу які мають Z показник більше ніж ϵ .

$$U = \{n_i : Z_i < \epsilon\} \quad (2.19)$$

де U – фільтровані данні.

Етап 3. Перетворення журналу для заданого рівня грануляції $\Delta\tau$.

Етап 3.1 Нормалізація кількості покупок n_j для кожної запису в журналі продажів $U_{\Delta\tau}$. Нормування проводиться з метою порівняння часової динаміки для різних об'єктів ij . Результатом цього кроку є набір U_{norm} :

$$U_{\text{norm}} = \left\{ (u_k, i_j, n_j, \tau_n) : n_j = \frac{n_j}{\max(n_j)} \right\} \quad (2.20)$$

Етап 3.2. Розбиття початкового періоду часу ΔT на впорядковану послідовність інтервалів часу ΔT_m , як показано на рис 1.

$$\Delta T = \Delta \tau_1 + \Delta \tau_2 + \dots + \Delta \tau_m + \dots + \Delta \tau_M, \forall m \Delta \tau_n = \Delta \tau \quad (2.21)$$

де $\Delta \tau_1$ – інтервал часу.

Етап 3.3. Підсумування покупок для кожного товару $i_j \in I$ для заданого рівня грануляції $\Delta \tau$:

$$n_{jm} = \sum_j n_j | \tau_n \in \Delta \tau_m \quad (2.22)$$

де n_i – кількість проданих товарів за інтервал часу $\Delta \tau_i$.

Змінна n_{jm} може містити покупки за годину, день, тиждень тощо.

Результатом виконання кроку є набір елементів Q , який містить список елементів, і кожен елемент містить кількість покупок товарів i_j для інтервалу $\Delta \tau_m$:

$$Q = \{(i_j, n_{jm}, \Delta \tau_n) : \forall \tau_n \in \Delta T, i_j \in I\} \quad (2.23)$$

Етап 4. Побудова набору R_j незважених тимчасових правил для кожного елемента i_j .

На цьому етапі формуються правила $r_{m,j}$ типу «Future», які відображають зміну кількості продажів позиції i_j між інтервалами $\Delta \tau_m$ і $\Delta \tau_M$:

$$R_j = \{r_{m,j} : \forall j r_{m,j} = (n_{jm} F n_{jM})\} \quad (2.24)$$

де $r_{m,j}$ – правила темпоральних відносин типу «Future».

Етап 5. Розрахунок набору ваг $W = \{W_j\}$ тимчасових правил для позицій з урахуванням популярності об'єктів та різниці продажів між інтервалами

$$\Delta\tau_m \text{ і } \Delta\tau_M.$$

Популярність позицій враховується завдяки нормалізації ваг на кроці 2.1. Отже, для обчислення ваги $\omega_{m,j}$ достатньо правил $r_{m,j}$, щоб знайти різницю продажів на інтервалах $\Delta\tau_m$ і $\Delta\tau_M$:

$$W_j = \{\omega_{m,j}: \forall m \omega_{m,j} = n_{jM} - n_{jm}\} \quad (2.25)$$

де W_j – загальна вага правил.

Етап 6. Розрахунок темпоральної динаміки за період часу ΔT для кожного продукту i_j від початкового рекомендованого набору. Часова динаміка для кожної позиції i_j задається елементом g_j , що дорівнює загальній зміні нормованої кількості проданих товарів за цей період часу:

$$G = \{g_j: \forall j g_j = \sum_m \omega_{m,j}\} \quad (2.26)$$

де g_j – загальна зміна нормованої кількості проданих товарів.

Етап 7. Формування пояснень до рекомендованого товару у вигляді переліку товарів, упорядкованих за часовою динамікою за умови позитивної динаміки:

$$E = \{(i_j, g_j): \forall j g_j > 0\} \quad (2.27)$$

де g_j – загальна зміна нормованої кількості проданих товарів;

де E – пояснень до рекомендованого товару.

Отриманий на цьому етапі список пар (i_j, g_j) показує, наскільки зріс інтерес користувачів до продукту i_j за певний період часу ΔT , а отже є

поясненням рекомендацій цього пункту, який задовольняє критерію довіри [38].

3. ТЕХНОЛОГІЯ ПОБУДОВИ ПОЯСНЕНЬ В РЕКОМЕНДАЦІЙНІЙ СИСТЕМІ З ВИКОРИСТАННЯМ ТЕМПОРАЛЬНИХ ПРАВИЛ

Технологія використання удосконаленого методу формування пояснень складається з наступних етапів (рисунок 3.1).

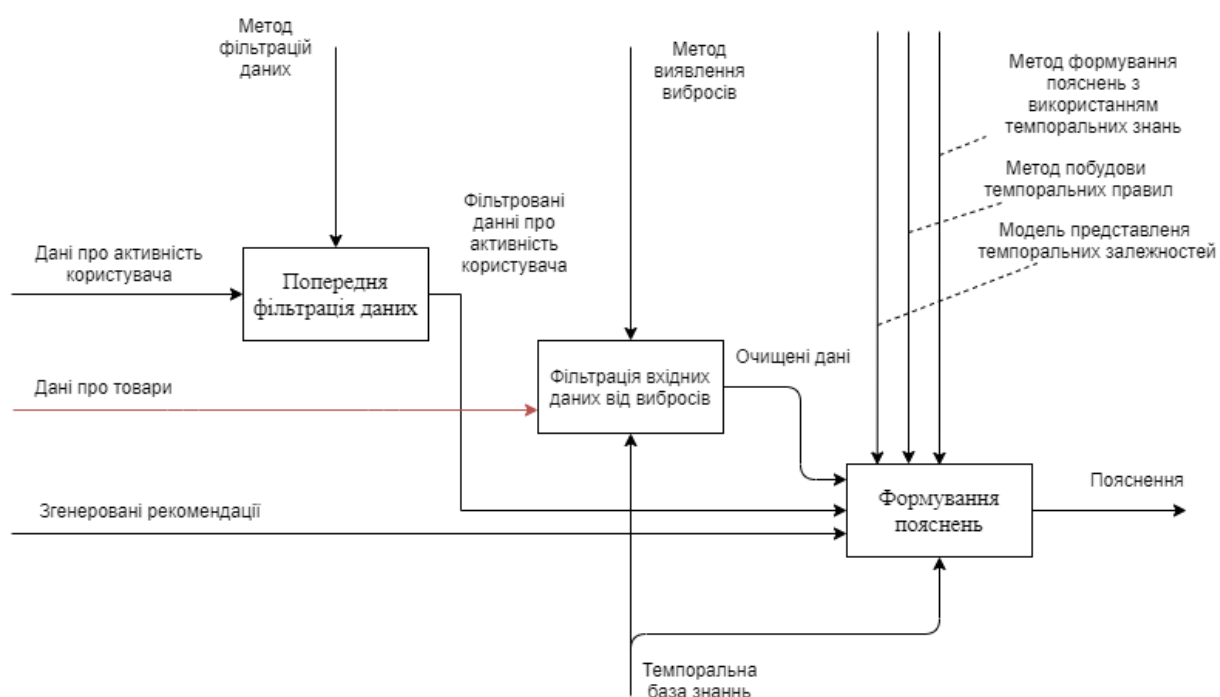


Рисунок 3.1 – Технологія формування пояснень з використанням удосконаленого методу

Вхідними даними для всього процесу формування пояснень є данні про активність користувача, а саме історія покупок і переглядів товарів, залишені відгуки і оцінки під товарами. На першому етапі ці данні повинні бути профільтровані. При фільтрації проводиться вибірка даних з встановленого часового інтервалу, наприклад 3 місяці, на базі якого будуть зформовані пояснення.

На другому етапі виконується фільтрація даних про продажі товарів від вибросів. Метод фільтрації проводить аналіз динаміки зміни продаж між інтервалами часу. Це дозволяє виявити аномальні підвищення та пониження показників продаж товару і виключити ці аномальні інтервали часу.

Після етапу фільтрації вибросів система передає очищені дані методам формування пояснень з використанням темпоральних правил. На цьому етапі проводиться розбиття очищених даних на фіксовані інтервали часу, це може бути дні, тижні, місяці. Для кожного інтервалу часу проводиться нормалізація кількості продаж товарів. Розрахунок набору ваг тимчасових правил для товару з враховує популярності та різниці продажів між інтервалами часу. Популярність товару враховується завдяки нормалізації кількості продаж. Виконується розрахунок темпоральної динаміки за період часу вхідних даних для кожного продукту від початкового рекомендованого набору.

В кінці ми отримуймо пояснення які показують, наскільки зріс інтерес користувачів до товару за певний період часу.

4. ПРАКТИЧНЕ ВИКОРИСТАННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Розробка програмного модулю формування пояснень в рекомендаційній системі

Розроблений програмний додаток являє собою модуль на мові OutSystems до системи інтернет-магазину, який розроблено на платформі OutSystems. Модуль аналізує данні користувача і від рекомендаційної системи та видає пояснення.

OutSystems – це гнучка low-code платформа для розробки корпоративних мобільних та веб-додатків, які розгортаються у локальному чи гібридному середовищі.

Low-code платформа LCDP (low-code development platform) – це програмне забезпечення, яке надає програмістам середовище для створення прикладних програм через графічний інтерфейс та різні конфігурації замість традиційного програмування. LCDP скорочують обсяг традиційного ручного коду, забезпечуючи прискорену розробку, встановлення та розгортання додатків. Загальною перевагою є те, що LCDP мають більш м'яку криву навчання, ніж більшість мов програмування і, відповідно, ширше коло людей може брати участь у розробці.

Особливості OutSystems

- моніторинг та звітність у режимі реального часу про використання та продуктивність усіх додатків;
- повністю масштабований через API;
- повністю захищена функціональність;
- швидке розгортання та моніторинг мобільного додатка;
- повна автоматизація DevOps;
- контроль доступу та керування дозволами;
- вбудована підтримка гнучкої розробки;
- вбудований механізм налагодження;

- підтримка багатомовних інтерфейсів;
- повна підтримка пакетних процесів та планування;
- вбудований механізм налагодження;
- підтримка всього життєвого циклу розробки додатків.

OutSystems пропонує такі переваги:

- OutSystems забезпечує високопродуктивну візуальну розробку та надає потужні інструменти для розгортання та керування вашими додатками;
- OutSystems дозволяє без проблем інтегрувати ваші програми у будь-яку систему;
- за допомогою OutSystems розробники можуть створювати адаптивні веб-додатки та залучати власні мобільні можливості. Панелі моніторингу продуктивності в реальному часі також забезпечують відмінний інтерфейс користувача;
- OutSystems дає вам можливість розширювати свої програми за допомогою власного внутрішнього та зовнішнього коду;
- OutSystems забезпечує велику масштабованість додатків та підвищує їх продуктивність незалежно від складності, обсягу даних та кількості користувачів;
- OutSystems має новітні функції безпеки, які забезпечують безпеку ваших додатків на всіх етапах - від проектування до розгортання.

Відмінна вертикальна та горизонтальна масштабованість. Легке додавання додаткової обчислювальної потужності та інтерфейсних серверів у будь-яке виробниче середовище, що допомагає досягти високої доступності при збільшенні навантаження користувача, складності бізнес-логіки, обсягу пакетної обробки або кількості транзакцій.

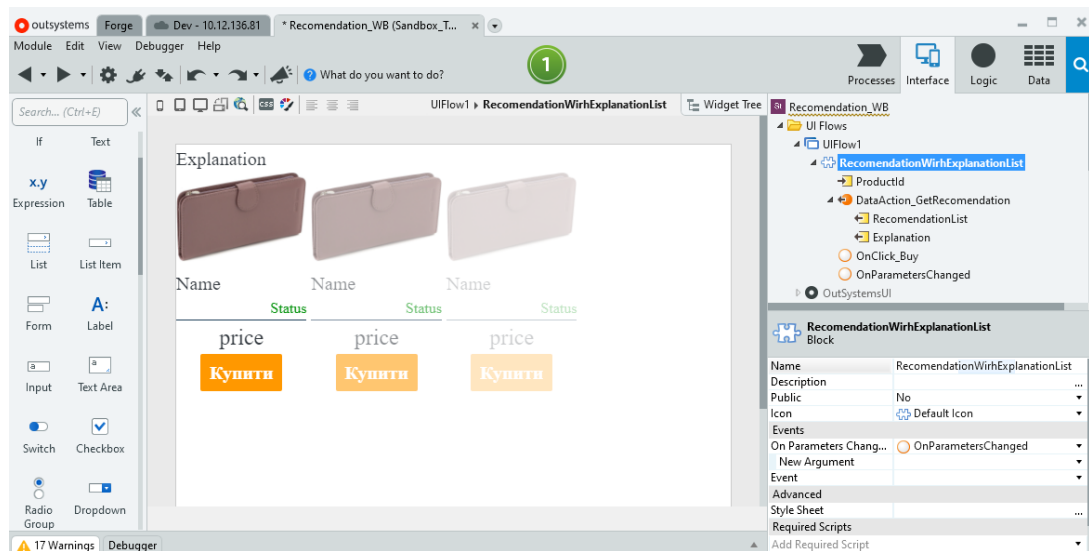
В ході розробки було створено 3 модуля:

- Recommendation_Explanation_WB – містить візуальні блоки для відображення рекомендацій і пояснень до них;

– Recommendation_Explanation_CS – містить сховище даних діяльності користувачів, також має інтеграції з модулем Users і Product, для отримання більш детальної інформації;

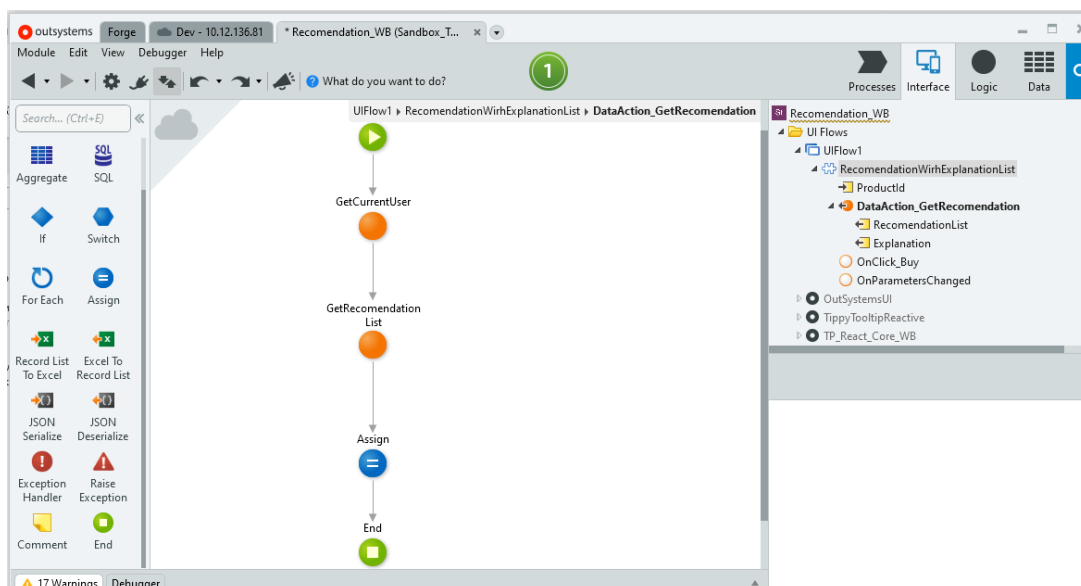
– Recommendation_lib – модуль інтеграції з бібліотекою обчислення рекомендацій і пояснень до них з метою підвищення швидкості роботи модуля.

На рисунку 4.1 зображено веб блок RecommendationWirhExplanationList, який використовується на сторінці з товаром і відображає рекомендації з поясненнями. Веб блок викликає функцію GetRecommendationList з модуля Recommendation_Explanation_CS, яка повертає рекомендації і пояснення (рисунок 4.2).



Рисунку 4.1 – Структура веб блока RecommendationWirhExplanationList

В модулі Recommendation_Explanation_CS було створена функція AddUserActivity, яка викликається при кожній дії користувача: перегляд товару, купівля товару, написання відгука, помітка товару, використання пошуку, перегляд списку товарів в категорії. Також була створення функція GetRecommendationList яка робить запит до рекомендаційної системи і доповнює їх поясненнями, базуючись на інформації про користувача.



Рисунку 4.2 – Код отримання даних веб блока RecommendationWirhExplanationList

В модулі Recommendation_lib проведена інтеграція з бібліотекою PredictionIO. Бібліотека виконує як синхронні обчислення (при запиті згенерувати рекомендації), так і асинхронні де проводиться обробка усіх дій користувача.

4.2 Експериментальна перевірка удосконаленого методу формування пояснень

Експеримент був проведено в інтернет магазині в категорії «Жіночі гаманці». Показана на екран рекомендація супроводжується згенерованим поясненням: «Попит на ці товари з цієї категорії підвищився на 35% за тиждень». У результаті, користувач системи при пошуку гаманця може перевести увагу на позиції які користуються попитом та отримали найкращі оцінки від інших користувачів.

Використовувались наступні вхідні данні:

- категорія товару;
 - записи журналу продажу про кількість продаж товару за проміжок часу;
 - записи про докладну інформацію про користувачів ресурсу;
 - записи про історію переглянутих товарів користувачами;
 - записи про вподобання користувачів ресурсу;
- В таблиці 4.1 наведено приклад журналу продаж товарів.

Таблиці 4.1 – Приклад журналу продаж

InvoiceNo	ProductCode	ProductName	Quantity	InvoiceDate	CustomerID
58567	031-950	Crocodile leather	1	01.12.2021 9:26	8268
58568	ST006	ST Leather pink	1	01.12.2021 9:48	9647
58569	st.kup.knopka	ST Leather	1	01.12.2021 10:20	1247

На рисунку 4.3 зображено приклад екрану пояснень з використанням темпоральною залежністю.

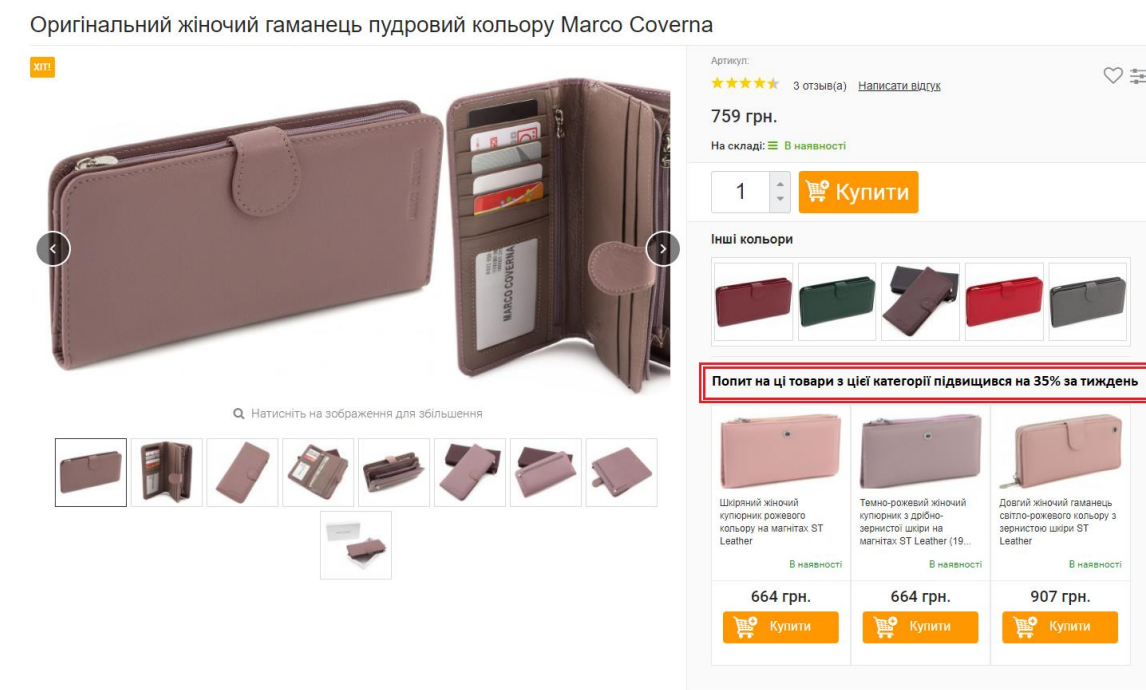
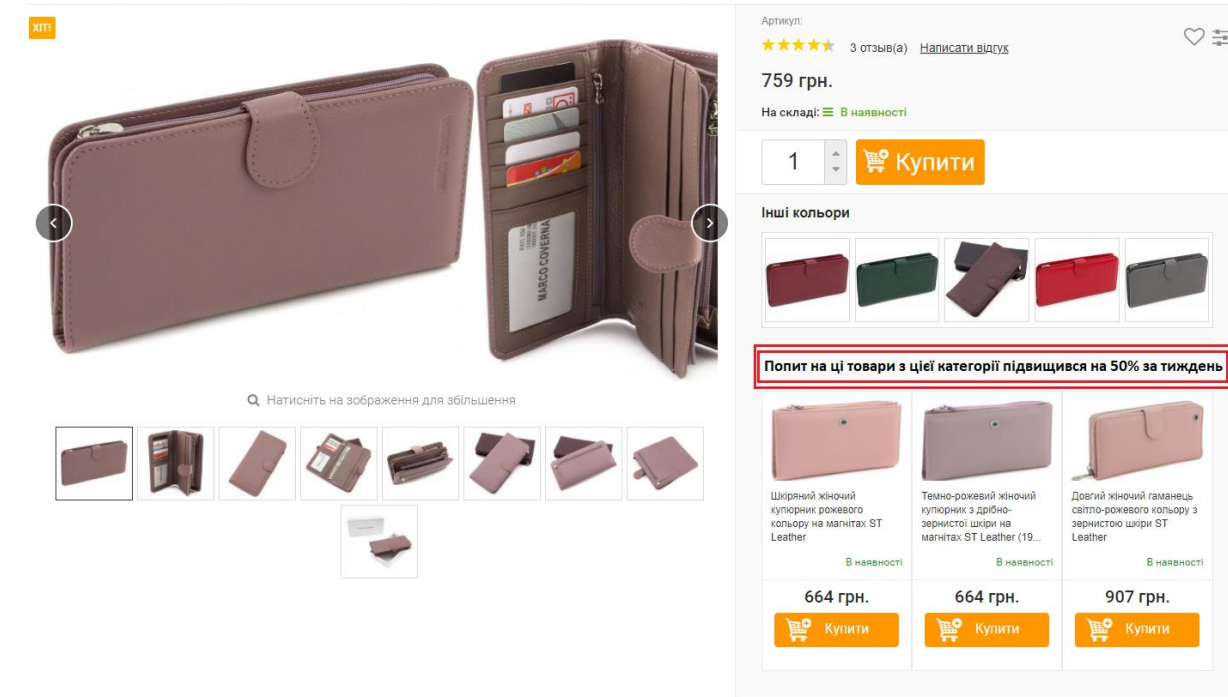


Рисунок 4.3 – Приклад екрану пояснення з використанням темпоральною залежністю

В другому прикладі ми отримали вхідні данні продаж товарів із категорії «Жіночі гаманці» що був розрахований за іншій період часу. На рисунку 4.4 зображено пояснення до рекомендації яке інформує що за попередній тиждень попит і рейтинги товарів зросли на 50%. Це може бути зв'язано сезонністю формування пояснень, наприклад через циклічні події або свята як Новий Рік або 8 березня. Таким чином користувач бачить велике збільшення попиту на данні товари. Як результат, надання пояснення з урахуванням темпоральної динаміки, може змінити вибір кінцевого користувача, а уточнююче пояснення до рекомендації збільшує довіру клієнта та підвищує шанси на покупку рекомендованих товарів.

Оригінальний жіночий гаманець пудровий кольору Marco Coverna



Артикул: ★★★★★ 3 отзв(а) [Написати відгук](#)

759 грн.

На складі: В наявності

1 [Купити](#)

Інші кольори

Попит на ці товари з цієї категорії підвищився на 50% за тиждень




 Шкіряний жіночий купорник рожевого кольору на магнітах ST Leather В наявності 664 грн. Купити	 Темно-рожевий жіночий купорник з дрібно-зернистої шкіри на магнітах ST Leather (19... В наявності 664 грн. Купити	 Довгий жіночий гаманець світло-рожевого кольору з зернистою шкіри ST Leather В наявності 907 грн. Купити
--	---	--

Рисунок 4.4 – Приклад екрану пояснення за умови зміни попиту

На рисунку 4.5 надається зручний вивід інформації про продажі і активність користувачів в категорії «Жіночі гаманці» в вересні і жовтні. На даному графіку показано кількість переглядів товарів з категорії «Жіночі

гаманці», графік переходів користувачів до рекомендованих товарів, середній час перебування і конверсія.

З рисунка 4.5 видно що після використання допрацьованого метода формування пояснень збільшилося кількість замовлень.

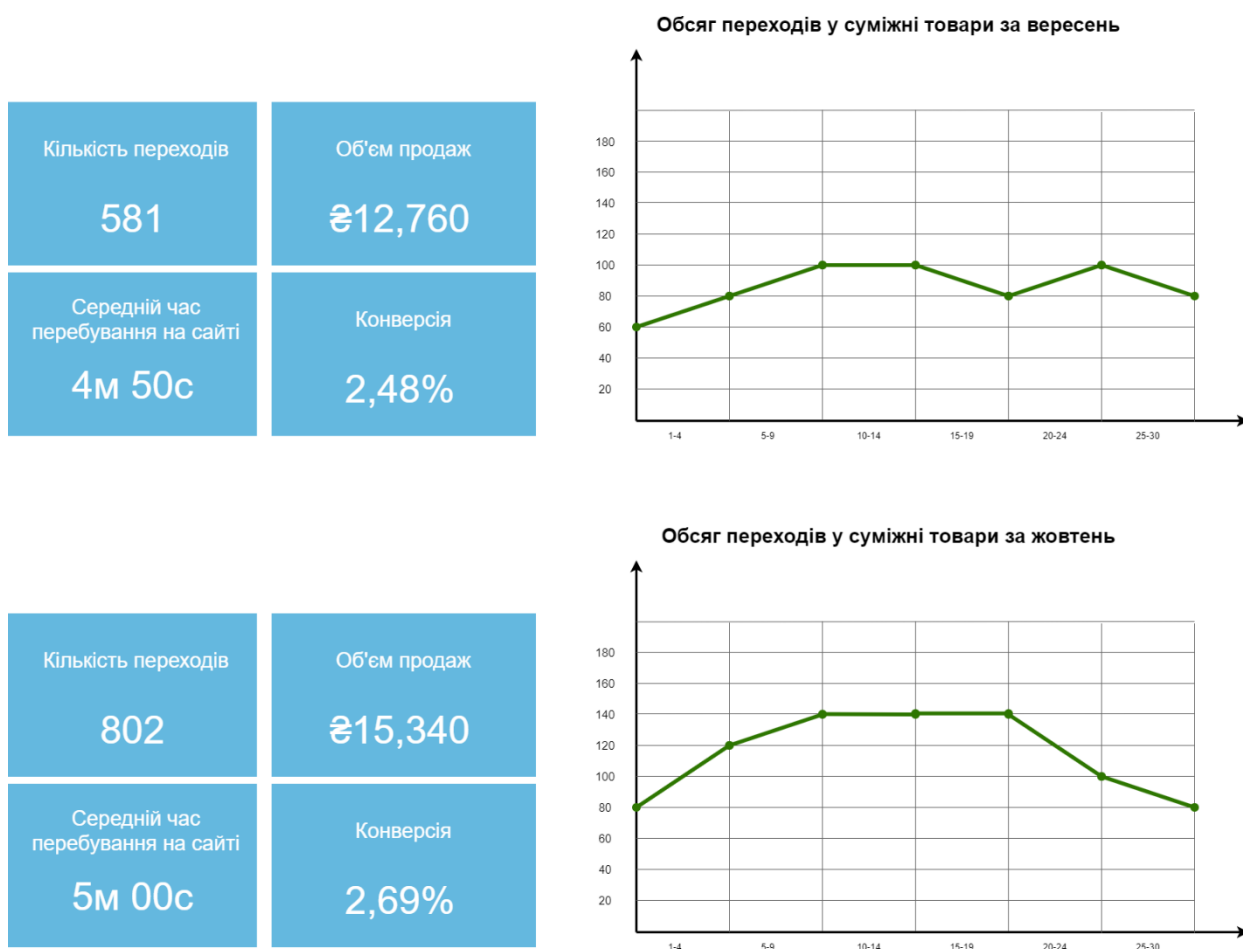


Рисунок 4.5– Інформації про продажі і активність користувачів

Користувач став отримувати більш детальні і доцільно сформовані пояснення до рекомендацій. На графіку також зображено збільшення кількості переходів по рекомендованим товарів, що каже про підвищення якості рекомендації. По результатам експерименту можна зробити висновок що дороблений метод формування пояснення до рекомендації спрацювало та привело до збільшення обсягу продаж і позитивного результату для користувача.

ВИСНОВКИ

Метою даної кваліфікаційної роботи є дослідження методів побудови пояснень в рекомендаційній системі з використанням темпоральних правил.

В кваліфікаційній роботі при дослідженні методів формування пояснень і було визначено що більшість методів не враховують зміни вподобань користувача з часом. Отже задача побудови пояснень з урахуванням індивідуальної динаміки зміни потреб користувачів є актуальною. Тому вибір і зіставлення знань, що відображають зміни в уподобаннях користувачів з часом і використовуються для формування пояснень, є важливим завданням.

В результаті виконання кваліфікаційної роботи було проведено аналіз особливостей пояснень в рекомендаційних системах. Було описано цілі пояснень в РС. Досліджено методів побудови пояснень в рекомендаційних системах. Дослідження моделей представлення темпоральних знань.

Удосконалено метод побудови пояснень в рекомендаційній системі з використанням темпоральних правил шляхом видалення аномальних вхідних даних щодо поведінки користувачів, що дає можливість підвищити довіру користувачів до отриманих рекомендацій і, на цій основі, збільшити продажі в рекомендаційній системі.

Виконано розробку додаткових модулів на платформі Outsystems для формування і відображення пояснень до рекомендацій з використанням удосконаленого методу.

Експериментальна перевірка удосконаленого методу побудови пояснень показала збільшення конверсії з 2,48% до 2,69%.

Результати кваліфікаційної роботи представлені у матеріалах 25-го міжнародного молодіжного форуму «Радіоелектроніка та молодь у ХХІ столітті» 2021р. опубліковані тези доповіді «Розробка метода побудови пояснень в рекомендаційній системі туристичного оператора»

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи (для студентів усіх форм навчання другого (магістерського) рівня вищої освіти спеціальності 122 Комп'ютерні науки освітньо-професійної програми «Інформаційні управляючі системи та технології») / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2021. – 30 с.
2. Prem Melville and Vikas Sindhwani. Recommender systems. In Encyclopedia of machine learning, pages 829–838. Springer, 2011
3. Nava Tintarev and Judith Masthoff. Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction*, 22(4):399–439, 2012.
4. Sergio Cleger-Tamayo, Juan M Fernandez-Luna, and Juan F Huete. Explaining neighborhood-based recommendations. In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, pages 1063– 1064. ACM, 2012.
5. David McSherry. Explanation in recommender systems. *Artificial Intelligence Review*, 24(2):179–197, 2005.
6. Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. How should i explain? A comparison of different explanation types for recommender systems. *International Journal of HumanComputer Studies*, 72(4):367–382, 2014.
7. J. V. Barneveld and M. V. Setten. Personalized digital television, chapter 10, pages 259–285. Kluwer Academic Publishers, 2004
8. B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Transactions on Graphics*, 21(4):833–854., 2002

9. M. Bilgic and R. J. Mooney. Explaining recommendations: Satisfaction vs. promotion. In Beyond Personalization Workshop, IUI, 2005.
10. D. Billsus and M. J. Pazzani. A personal news agent that talks, learns, and explains. In Proceedings of the Third International Conference on Autonomous Agents, 1999.
11. Lou, Y., Caruana, R. and Gehrke, J. (2012), “Intelligible models for classification and regression”, Proc. of the 18th ACM SIGKDD int. conf. on Knowledge discovery and data mining, pp. 150–158. DOI: <https://doi.org/10.1145/2339530.2339556>.
12. Kim B. Interactive and interpretable machine learning models for human machine collaboration [Doctoral dissertation]: Massachusetts Institute of Technology, 2015.
13. Wu M, Hughes MC, Parbhoo S, Zazzi M, Roth V, Doshi-Velez F. Beyond sparsity: Tree regularization of deep models for interpretability, Thirty-Second AAAI Conference on Artificial Intelligence. (2018).
14. Zhang Q, Nian Wu Y, Zhu S-C. Interpretable convolutional neural networks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) pp. 8827-8836. <https://doi.org/10.1109/CVPR.2018.00920>.
15. Vaughan J, Sudjianto A, Brahim E, Chen J, Nair VN. Explainable neural networks based on additive index models [Preprint]. June 5, 2018. <https://arxiv.org/pdf/1806.01933.pdf>
16. Wang T. Hybrid decision making: When interpretable models collaborate with blackbox models, J. Mach. Learn. Res. (2019) pp. 1-31.
17. Papernot N, McDaniel P. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning [Preprint]. March 18, 2018. <https://arxiv.org/pdf/1803.04765.pdf>
18. Hind M, Wei D, Campbell M, Codella NC, Dhurandhar A, Mojsilović A, et al. Ted: Teaching AI to explain its decisions, Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. (2019) pp. 123-129. <https://doi.org/10.1145/3306618.3314273>.

19. Lakkaraju H, Kamar E, Caruana R, Leskovec J. Interpretable & explorable approximations of black box models [Preprint]. July 4, 2017. <https://arxiv.org/pdf/1707.01154.pdf>
20. Bastani O, Kim C, Bastani H. Interpreting blackbox models via model extraction [Preprint]. January 24, 2017. <https://arxiv.org/abs/1904.11829>
21. Tan S, Caruana R, Hooker G, Lou Y. Distill-and-compare: Auditing black-box models using transparent model distillation, Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. (2018) pp. 303-310.
22. Ancona M, Ceolini E, Öztireli C, Gross M. Towards better understanding of gradientbased attribution methods for deep neural networks [Preprint]. March 7, 2018. <https://arxiv.org/abs/1711.06104>
23. Fisher A, Rudin C, Dominici F. All models are wrong but many are useful: Variable importance for black-box, proprietary, or misspecified prediction models, using model class reliance [Preprint]. December 23, 2018. <https://arxiv.org/pdf/1801.01489.pdf>
24. Apley DW, Zhu J. Visualizing the effects of predictor variables in black box supervised learning models [Preprint]. August 18, 2019. <https://arxiv.org/pdf/1612.08468.pdf>
25. Friedman JH, Popescu BE. Predictive learning via rule ensembles, Ann. Appl. Stat. 2(2008) pp. 916-954. <https://doi.org/10.1214/07-AOAS148>.
26. Goldstein A, Kapelner A, Bleich J, Pitkin E. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation, J. Comput. Graph. Stat. 24(2015) pp. 44-65. <https://doi.org/10.1080/10618600.2014.907095>.
27. Datta A, Sen S, Zick Y. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems, 2016 IEEE Symposium on Security and Privacy. (2016) pp. 598-617. https://doi.org/10.1007/978-3-319-54024-5_4.

28. Lundberg SM, Lee S-I. A unified approach to interpreting model predictions, *Advances in Neural Information Processing Systems* 30. (2017) pp. 4765-4774.
29. Yoon J, Jordon J, van der Schaar M. Invase: Instance-wise variable selection using neural networks, *International Conference of Learning Representations*. (2018) pp. 1-24.
30. Kim B, Khanna R, Koyejo OO. Examples are not enough, learn to criticize! Criticism for interpretability, *Advances in Neural Information Processing Systems* 29. (2016) pp. 2280-2288.
31. Cook RD. Detection of influential observation in linear regression, *Technometrics*. 19(1977) pp. 15-18.
<https://doi.org/10.1080/00401706.1977.10489493>
32. Wachter S, Mittelstadt B, Russell C. Counterfactual explanations without opening the black box: Automated decisions and the gpdr, *Harv. J. L. & Tech*. 31(2017) pp. 841- 887. <https://doi.org/10.2139/ssrn.3063289>.
33. Chalyi, S., Leshchynskyi, V. and Leshchynska, I. (2019), “Modeling explanations for the recommended list of items based on the temporal dimension of user choice”, *Control, navigation and communication systems*, Vol. 6 (58), pp. 97-101. DOI: <https://doi.org/10.26906/SUNZ.2019.6.097>.
34. . Levykin, V. and Chala, O. (2018), “Development of a method for the probabilistic inference of sequences of a business process activities to support the business process management”, *Eastern-European Journal of Eenterprise Technologies*, Vol. 5/3(95), pp. 16-24. DOI: <https://doi.org/10.15587/1729-4061.2018.142664>.
35. Chalyi, S. and Pribylnova, I. (2019), “The method of constructing recommendations online on the temporal dynamics of user interests using multilayer graph”, *EUREKA: Physics and Engineering*, 2019, Vol. 3, pp. 13-19.
36. Fair, D. (1979), “Causation and the flow of energy”, *Erkenntnis*, Vol. 14, pp. 219–250. DOI: <https://doi.org/10.1007/BF00174894>.

37. Chalyi, S., Leshchynskyi, V. (2020). Method of constructing explanations for recommender systems based on the temporal dynamics of user preferences. *EUREKA: Physics and Engineering*, 3, 43–50. doi: <https://doi.org/10.21303/2461-4262.2020.001228>
38. Tintarev, N., Masthoff, J. (2007). A Survey of Explanations in Recommender Systems. 2007 IEEE 23rd International Conference on Data Engineering Workshop. doi: <https://doi.org/10.1109/icdew.2007.4401070>
39. Chalyi S., Pribylnova I. The method of constructing recommendations online on the temporal dynamics of user interests using multilayer graph // *EUREKA: Physics and Engineering*. – 2019. – vol. 3. – P. 13-19. doi: 10.21303/2461-4262.2019.00894
40. Chalyi S., Leshchynskyi V., Leshchynska, I. method of forming recommendations using temporal constraints in a situation of cyclic cold start of the recommender system // *EUREKA: Physics and Engineering*. – 2019. – vol. 4. – P. 34–40
41. Chalyi S., Leshchynskyi V., Leshchynska I. Detailing explanations in the recommender system based on matching temporal knowledge // *Eastern-European Journal of Enterprise Technologies*. - 2020. - vol 4. -No 2 (106). - P. 6-13.
42. Chalyi S., Leshchynskyi V. method of constructing explanations for recommender systems based on the temporal dynamics of user preferences // *EUREKA: Physics and Engineering*. – 2020. – vol. 3. – P. 43-50.
43. Chala O., Novikov, L., Chernyshova, L. method for detecting shilling attacks in recommender systems using objective feedback // *EUREKA: Physics and Engineering*. - 2020. - vol. 5. - P. 21-30. doi: 10.21303/2461-4262.2020.001394
44. Chala O., Novikov, L., Chernyshova, L. method for detecting shilling attacks in e-commerce systems using weighted temporal rules // *EUREKA: Physics and Engineering*. – 2019. – vol. 5. – P. 29-36. doi: 10.21303/2461-4262.2019.00983

45. Чалий С.Ф., Лещинський В.О., Лещинська І.О. (2018). Моделювання контексту в рекомендаційних системах. Проблеми інформаційних технологій, 1(023), 21-26.

46. Levykin V., Chala O. (2018). Development of a method for the probabilistic inference of sequences of a business process activities to support the business process management. Eastern-European Journal of Enterprise Technologies, 5/3(95), 16-24. DOI: 0.15587/1729-4061.2018.142664.

47. Чалий С. Ф. Декларативно-темпоральний підхід до побудови пояснень в інтелектуальних інформаційних системах / С. Ф. Чалий, В. О. Лещинський, І. О. Лещинська // Вісник Національного технічного університету "ХПІ". Сер. : Системний аналіз, управління та інформаційні технології = Bulletin of the National Technical University "KhPI". Ser. : System analysis, control and information technology : зб. наук. пр. – Харків : НТУ "ХПІ", 2020. – № 2 (4). – С. 51-56.