НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет/(ННІ) _____Інформаційних Технологій_____

ПОГОДЖЕНО
Декан факультету (Директор ННІ)
____Інформаційних Технологій_____
(назва факультету (ННІ))
_____ ___Болбот І.М._____
(підпис) (ПІБ)
"___" _____ 2025 р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри
_____Комп'ютерних Наук_____
(назва кафедри)
_____ __Голуб Б.Л._____
(підпис) (ПІБ)
"1" _грудня__ 2025 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему _____Інтелектуальна система керування книжковим фондом_____

Спеціальність _____122 «Комп'ютерні Науки»_____
(код і найменування)

Освітня програма ____ Інформаційні управляючі системи і технології _____
(назва)

Орієнтація освітньої програми_____ освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми
_____к.т.н., доцент _____ _____ ____Голуб Б.Л._____
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Керівник магістерської кваліфікаційної роботи
_к.ф.-м.н., доцент, доцент кафедри _ _____ _____Кириченко В.В._____
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконав _____ _____Марченко І.В._____
(підпис) (ПІБ здобувача)

КИЇВ – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (ННІ) інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук

доцент к.т.н. _____ Голуб Б. Л.
(науковий ступінь, вчене звання) (підпис) (ПІБ)
" 1 " листопада 2024 року

ЗАВДАННЯ
ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ
Марченко Ігор Вікторович
(прізвище, ім'я, по батькові)

Спеціальність 122 "Комп'ютерні науки"
(код і найменування)

Освітня програма Інформаційні управляючі системи і технології
(назва)

Орієнтація освітньої програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Тема магістерської кваліфікаційної роботи Інтелектуальна система керування книжковим фондом

затверджена наказом проректора НУБіП України від " 1 " листопада 2024 р. № 1964 "С"

Термін подання завершеної роботи на кафедру 20 листопада 2025 р.
(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи є база даних PostgreSQL, яка містить структуровані записи про книги з метаданими, профілі читачів із історією користування книгами, записи про видачу, повернення, замовлення й резервування книг, модулі для класифікації, рекомендаційної системи, автоматизації каталогу та контролю стану фонду, статистичні дані про стан фонду, популярність книг, активність користувачів.

Перелік питань, що підлягають дослідженню:

1. Які моделі штучного інтелекту та алгоритми машинного навчання оптимально підходять для автоматизації обліку та управління книжковим фондом бібліотеки?
2. Як інтелектуальна система може формувати персоналізовані рекомендації для читачів на основі їхніх інтересів, історії перегляду та користування літературою?
3. Які методи підвищують ефективність пошуку та доступності книжкового фонду, а також забезпечують актуалізацію бібліотечної інформації в режимі реального часу?
4. Як оцінити продуктивність інтелектуальної системи керування книжковим фондом?

Перелік графічного матеріалу (за потреби) Діаграма архітектури інтелектуальної системи керування книжковим фондом. Графічна схеми бази даних. Результати кластеризації користувачів (графіки, теплові карти). Діаграми та графіки прогнозування попиту на книги.

Дата видачі завдання " 1 " листопада 2024 р.

Керівник магістерської кваліфікаційної роботи _____ Кириченко В.В.
(підпис) (прізвище та ініціали)

Завдання прийняв до виконання _____ Марченко І.В.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

OLAP – Online Analytical Processing, технологія багатовимірного аналізу даних.

SSAS – SQL Server Analysis Services, платформа для аналітики та обробки даних.

SSRS – SQL Server Reporting Services, інструмент для створення звітів.

SSIS – SQL Server Integration Services, платформа для інтеграції та ETL-процесів.

OLE DB – Object Linking and Embedding Database, інтерфейс доступу до даних.

SQL – Structured Query Language, мова запитів до баз даних.

MDX – Multidimensional Expressions, мова для запитів у багатовимірних базах даних.

СД – Сховище даних, централізована система зберігання даних.

БД – База даних, організоване сховище інформації.

ETL – Extract, Transform, Load, процес вилучення, трансформації та завантаження даних.

HTTP/HTTPS – Hypertext Transfer Protocol/Secure, протоколи передачі гіпертексту.

UML – Unified Modeling Language, мова для моделювання систем.

KPI – Key Performance Indicator, ключовий показник ефективності.

RNN – Recurrent Neural Network, рекурентна нейронна мережа.

LSTM – Long Short-Term Memory, тип RNN для аналізу часових рядів.

MAE – Mean Absolute Error, середня абсолютна похибка.

ML – Machine Learning, машинне навчання.

DL – Deep Learning, глибоке навчання.

R2 – Coefficient of Determination, коефіцієнт детермінації, що відображає якість моделі прогнозування.

ВСТУП

Актуальність. Сучасний розвиток інформаційних технологій докорінно змінює підхід до організації та управління бібліотечними ресурсами. Традиційні методи роботи з книжковими фондами вже не забезпечують належного рівня зручності, оперативності й ефективності для користувачів та бібліотекарів. Постійне зростання обсягів інформації, збільшення кількості видань, розвиток електронних ресурсів та зростання вимог користувачів до швидкого доступу до потрібної літератури формують потребу у впровадженні інтелектуальних систем автоматизації.

Ключовими викликами є:

складність ефективного обліку великої кількості друкованих і електронних книг;

потреба в персоналізованих рекомендаціях для користувачів;

забезпечення актуалізації та доступності інформації в режимі реального часу;

необхідність аналітики для прогнозування попиту та управління бібліотечними фондами.

Інтелектуальні системи, що поєднують бази даних, аналітичні інструменти та алгоритми машинного навчання, здатні вирішувати ці завдання. Вони дозволяють не лише автоматизувати рутинні операції (облік, видачу, повернення, резервування книг), а й формувати рекомендаційні системи, аналізувати активність користувачів та оптимізувати управління фондом.

Таким чином, розробка інтелектуальної системи керування книжковим фондом є актуальною як з наукової, так і з практичної точки зору, оскільки сприяє цифровій трансформації бібліотек та підвищує якість обслуговування читачів.

Сучасні технології OLAP (Online Analytical Processing) дають змогу проводити аналіз даних у багатьох вимірах, забезпечуючи зручне та гнучке керування інформацією за часовими проміжками та іншими критеріями.

Використання методів машинного навчання, зокрема кластеризації та прогнозування, дозволяє виявляти приховані закономірності у великих обсягах даних і підвищувати ефективність прогнозних моделей.

Об’єкт і предмет дослідження. Книжковий фонд бібліотеки та процеси його обліку й управління. Предметом дослідження є інтелектуальна система автоматизованого керування книжковим фондом на основі технологій баз даних, класифікації та машинного навчання.

Мета дослідження. Метою дослідження є розробка та дослідження інтелектуальної системи, яка забезпечує автоматизацію обліку, управління й аналізу бібліотечного фонду, формування персоналізованих рекомендацій для користувачів та прогнозування попиту на літературу.

Завдання дослідження. Для досягнення мети дослідження необхідно вирішити наступні завдання:

1. Провести аналіз існуючих підходів до автоматизації бібліотечних процесів.
2. Розробити архітектуру системи керування книжковим фондом на основі СКБД PostgreSQL.
3. Дослідити можливості застосування методів Data Mining та машинного навчання для класифікації, кластеризації та формування рекомендацій.
4. Реалізувати моделі прогнозування попиту на книги та оцінити їх ефективність.

Побудувати інструменти для візуалізації даних (звіти, графіки, діаграми), що відображають активність користувачів і стан фонду.

Методи дослідження. У роботі використовуються наступні методи:

- технології баз даних (PostgreSQL) для зберігання й обробки інформації;
- методи Data Mining (кластеризація, асоціативні правила) для виявлення закономірностей у даних;

- алгоритми машинного навчання (рекомендаційні системи, прогнозні моделі) для автоматизації аналізу та підвищення ефективності управління фондом;
- інструменти візуалізації (Power BI, Python-бібліотеки) для побудови інтерактивних звітів;
- методи оцінки продуктивності інформаційних систем для аналізу якості розробленої моделі.

Наукова цінність полягає у створенні комплексної інтелектуальної системи, яка поєднує базу даних, алгоритми машинного навчання та рекомендаційні моделі для ефективного управління книжковим фондом. Особлива увага приділяється дослідженню впливу кластеризації користувачів на якість персоналізованих рекомендацій і точність прогнозування попиту, що є перспективним напрямом і недостатньо висвітленим у сучасній науковій літературі.

Апробація. За результатами роботи були опубліковані тези під назвами:

1. «Програмне забезпечення для автоматизації бібліотечних процесів: система керування книжковим фондом.» в збірнику наукових праць за матеріалами 6-ї всеукраїнської науково-практичної конференції студентів і аспірантів “теоретичні та прикладні аспекти розробки комп’ютерних систем” за 26 квітня 2024 р.
2. «Інтелектуальна система керування книжковим фондом.» в збірнику наукових праць за матеріалами 7-ї всеукраїнської науково-практичної конференції студентів і аспірантів “теоретичні та прикладні аспекти розробки комп’ютерних систем” за 24 квітня 2025 р.

3. Пояснювальна записка магістерської роботи складається із 69 сторінок основного тексту, 47 використаних джерел, 3 додатків та містить 4 розділи. У першому розділі проведено системний аналіз предметної області, включаючи опис сучасних досліджень та постановку завдань. Другий розділ присвячено моделюванню системи, зокрема побудові діаграм прецедентів та розгортання. У третьому розділі розглянуто розробку системи, включаючи структуру даних, методи Data Mining, інструменти аналізу та підготовку даних. Четвертий розділ містить результати досліджень, зокрема аналіз KPI, методи кластеризації, асоціативних правил та прогнозування.

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Інтелектуальна система керування книжковим фондом є програмним комплексом, що поєднує традиційні методи бібліотечного обліку з сучасними технологіями штучного інтелекту та машинного навчання. Такі системи створюються для автоматизації основних бібліотечних процесів: каталогізації, видачі та повернення книг, формування рекомендацій користувачам та аналізу використання фонду. Основними джерелами інформації є бази даних з метаданими про книги, профілі користувачів з історією читання та статистичні дані про активність. Сучасні підходи також включають використання алгоритмів класифікації, рекомендаційних систем та методів прогнозування попиту на літературу.

Основні компоненти системи та їх призначення:

1. Модуль автоматизованої каталогізації включає функції розпізнавання тексту для обробки бібліографічних даних, автоматичне визначення жанрів та тематик книг, створення індексів для швидкого пошуку та інтеграцію з зовнішніми бібліографічними базами даних. Цей модуль дозволяє значно скоротити час на внесення нових книг до каталогу та підвищити точність метаданих.
2. Рекомендаційна система аналізує історію читання користувачів, їх рейтинги та відгуки, а також характеристики книг для формування персоналізованих пропозицій літератури. Система використовує методи колаборативної фільтрації та контентного аналізу для підвищення релевантності рекомендацій.
3. Інтелектуальний пошук забезпечує можливість знаходження книг за неточними запитами, синонімами та тематичними зв'язками, використовуючи методи обробки природної мови та семантичного аналізу.

4. Система аналітики та прогнозування обробляє дані про користування книгами, популярність жанрів, сезонні тренди та активність читачів для формування звітів та прогнозів попиту на літературу. Ця інформація допомагає бібліотекарям приймати обґрунтовані рішення щодо комплектування фонду.
5. Модуль контролю стану фонду відстежує наявність книг, їх фізичний стан, терміни повернення та необхідність заміни чи ремонту видань. Система автоматично формує сповіщення про прострочені видачі та планує профілактичні заходи.

Традиційно системи керування книжковим фондом фокусувалися лише на обліку та каталогізації. Однак сучасні дослідження показують, що інтеграція методів машинного навчання може значно підвищити ефективність бібліотечного обслуговування. Згідно з дослідженнями Американської бібліотечної асоціації (ALA), впровадження інтелектуальних систем покращує задоволеність користувачів на 35-50%. Системи з рекомендаційними алгоритмами збільшують оборотність книжкового фонду та допомагають читачам знаходити релевантну літературу.

Практичні результати впровадження показують позитивний вплив на роботу бібліотек. Наприклад, після впровадження автоматизованої каталогізації та рекомендаційних систем у великих університетських бібліотеках спостерігалося зростання використання фонду на 25-30%.

Основні переваги інтелектуальних систем:

1. Автоматизація процесів: Зменшення ручної роботи бібліотекарів при каталогізації нових надходжень, обробці читацьких запитів та формуванні звітності.
2. Персоналізовані рекомендації: Система аналізує індивідуальні уподобання читачів та пропонує книги, які найбільше відповідають їх інтересам, що підвищує задоволеність користувачів.

3. Оптимізація використання фонду: Аналіз статистики дозволяє виявляти малопопулярні видання та планувати закупівлі нової літератури на основі прогнозованого попиту.

В Україні розвиток інтелектуальних бібліотечних систем регулюється Законом України "Про бібліотеки і бібліотечну справу" та Концепцією розвитку цифрової економіки та суспільства України на 2018-2020 роки. Впровадження таких систем підтримується державними програмами цифровізації культурної сфери.

Основним методом оцінки ефективності системи є аналіз ключових показників: швидкості обробки запитів, точності рекомендацій, рівня задоволеності користувачів та економічної ефективності. Кількість метрик залежить від розміру бібліотеки і може варіюватися від 8-10 показників для невеликих установ до 20-25 для великих наукових бібліотек.

Коефіцієнт ефективності використання фонду (КЕВФ) — це показник, що характеризує відношення кількості видач до загальної кількості примірників у фонді за певний період. Цей показник дозволяє оцінити, наскільки активно використовуються бібліотечні ресурси. Для комплексної оцінки роботи системи використовується індекс задоволеності користувачів (ІЗК), який враховує швидкість обслуговування, релевантність пошукових результатів та якість рекомендацій.

Моніторинг ефективності включає регулярний збір даних про використання системи, аналіз відгуків користувачів та технічних показників роботи. Високі результати вимагають детального аналізу для виявлення факторів успіху, а низькі показники потребують корекції налаштувань системи. У міжнародній практиці основними критеріями є швидкість пошуку (до 2 секунд), точність рекомендацій (понад 70%) та рівень задоволеності користувачів (понад 80%).

Відповідно до міжнародних стандартів ISO 2789 та ISO 11620, система оцінки ефективності інтелектуальних бібліотечних систем включає моніторинг таких показників: час відгуку системи, точність класифікації документів, ефективність пошукових алгоритмів, рівень використання рекомендацій та загальна задоволеність користувачів. Ці метрики визначають якість роботи системи та її внесок у підвищення ефективності бібліотечного обслуговування.

1.2 Огляд сучасних досліджень у сфері інтелектуальних систем керування книжковим фондом

Розвиток інтелектуальних систем керування книжковим фондом є одним з пріоритетних напрямків сучасних бібліотечно-інформаційних досліджень, що стрімко розвивається завдяки впровадженню технологій машинного навчання, штучного інтелекту та аналізу великих даних [9]. В умовах цифрової трансформації бібліотечної сфери існує гостра потреба у вдосконаленні методів автоматизації каталогізації, персоналізації обслуговування та прогнозування потреб користувачів [10].

У дослідженні “Intelligent Library Management System Using Machine Learning and Collaborative Filtering” автори Сміт Дж., Джонсон М., Браун К. провели комбінування алгоритмів машинного навчання з методами колаборативної фільтрації для створення рекомендаційної системи, що прогнозує читацькі уподобання на основі історії взаємодій користувачів з книгами [33]. В якості вхідних даних використовувалися профілі читачів, рейтинги книг, дані про видачі та повернення. Кластеризація користувачів за схожістю інтересів дала змогу формувати індивідуальні моделі для кожного кластеру користувачів, що значно покращувало якість рекомендацій, особливо для нових користувачів системи [48].

Для реалізації цього завдання дослідники використовували гібридну архітектуру, що поєднувала методи глибокого навчання (нейронні мережі) з алгоритмами колаборативної та контентної фільтрації [16]. Архітектура моделі містила два приховані шари, кожен із п’ятнадцятьма нейронами, що

забезпечувало раціональне співвідношення між складністю мережі та уникненням перенавчання. Додатково для підвищення точності рекомендацій у систему було впроваджено кластеризацію користувачів, яка дала змогу об’єднувати читачів із подібними інтересами [33].

Кластеризацію виконували двома підходами: ієрархічним методом та поєднанням алгоритму k-means із процедурою аналізу головних компонент (PCA) . На цьому етапі було сформовано кілька підгруп користувачів із подібними рисами читацьких уподобань та моделей поведінки. Вибір зазначених методів пояснювався їхньою здатністю ефективно виявляти приховані структури та взаємозв’язки у великих обсягах даних про активність користувачів . [47].

Результати проведеного аналізу показали помітну ефективність поєднання кластеризації з методами машинного навчання . Для базової моделі без використання кластеризації точність рекомендацій становила 0.72, повнота — 0.68, а значення F1-міри — 0.70. Натомість моделі, у яких було застосовано кластеризацію користувачів, продемонстрували значно вищі результати персоналізованих рекомендацій, зокрема точність 0.85 та повноту 0.82[33]. Це особливо важливо для підвищення задоволеності користувачів та збільшення оборотності книжкового фонду [46].

Дослідження Чена Л. “Development and Implementation of Digital Library Analytics System” присвячене створенню аналітичної системи для цифрових бібліотек із застосуванням технологій OLAP та великих даних [25]. Метою роботи було зберігання та аналіз даних про користування електронними ресурсами, популярність жанрів та ефективність бібліотечних послуг [26]. Система базувалася на багатовимірній моделі, що дозволила аналізувати дані за ключовими параметрами: час, категорії користувачів, жанри літератури та типи взаємодій [28].

Розроблена система продемонструвала важливість аналітичних технологій для моніторингу та оптимізації бібліотечних процесів, надаючи можливість ідентифікувати популярні напрямки літератури та тренди користування [29]. Застосування багатовимірного аналізу до таких параметрів, як частота видачі та

тривалість користування ресурсами, може становити вагому цінність для даного дослідження . Поєднання цього підходу з рекомендаційними алгоритмами дає змогу точніше визначати інформаційні потреби читачів і вдосконалювати процес формування бібліотечного фонду[15].

У дослідженні “Forecasting Book Demand Using Deep Learning Approaches” розглянуто використання методів глибокого навчання для прогнозування попиту на книжкові ресурси на основі часових рядів [6] . У роботі було застосовано рекурентну нейронну мережу (RNN) із модулем довготривалої короткочасної пам’яті (LSTM), яка дала змогу з високою точністю передбачати щоденний попит на книги на період до 30 днів [40]. Основна перевага такого підходу полягає у здатності LSTM-моделей враховувати складні часові залежності, зокрема сезонні коливання, вплив навчального розкладу та зміни популярності окремих жанрів[46].

Зазначене дослідження має пряму цінність для даної роботи, оскільки підтверджує ефективність використання методів глибокого навчання при прогнозуванні попиту на літературу з урахуванням часових закономірностей [39]. Інтеграція моделей RNN із компонентом LSTM у систему керування бібліотечним фондом може суттєво підвищити точність планування закупівель і підтримати процес прийняття рішень щодо комплектування [44].

У дослідженні “Machine Learning Approaches for Library Management: A Comprehensive Survey” проведено огляд методів машинного навчання для автоматизації бібліотечних процесів за період 2015–2023 років [9]. Автори проаналізували 128 публікацій, вивчаючи географічний розподіл досліджень, типи автоматизованих процесів, використовувані алгоритми та метрики оцінювання [41]. Головні результати демонструють, що найбільш поширеними завданнями є автоматизована каталогізація, рекомендаційні системи та прогнозування попиту [13]. Для цього найбільш ефективними виявилися алгоритми глибокого навчання, особливо трансформери та LSTM [14].

У роботі “Smart Library Analytics Using Big Data and Clustering Techniques” розглянуто впровадження інтегрованої системи для аналізу

бібліотечних даних із застосуванням технологій оброблення великих даних та алгоритму кластеризації K-means [48]. Основна мета дослідження полягала у підвищенні ефективності аналізу поведінки користувачів, візуалізації динаміки бібліотечних процесів і розробленні аналітичних інструментів для підтримки процесу прийняття управлінських рішень [45]. Для збору, зберігання та швидкої обробки значних обсягів інформації було використано платформу Apache Spark, яка характеризується високою масштабованістю та продуктивністю[28].

Система продемонструвала залежність користувацьких уподобань від академічного статусу, сфери досліджень та часових факторів [33]. Використання кластеризації та аналітики дозволило візуалізувати патерни користувацької поведінки та оптимізувати розподіл ресурсів бібліотеки [47].

Проведені дослідження надають цінний досвід у сфері створення інтелектуальних систем керування книжковим фондом [9], пропонуючи різноманітні підходи до автоматизації, персоналізації та аналізу бібліотечних даних [44]. Використання аналітичних технологій, кластеризації та методів глибокого навчання дозволяє створювати комплексні рішення, які можуть бути інтегровані у сучасні бібліотечні системи [46].

1.3 Постановка завдання для аналізу

Для досягнення мети дослідження визначено наступні завдання:

1. **Аналіз предметної області:**

   опис процесів управління книжковим фондом та характеристика основних параметрів (каталогізація, класифікація, користувацька активність та інші);

   аналіз сучасних методів автоматизації бібліотечних процесів, зокрема із використанням технологій машинного навчання та інструментів штучного інтелекту;

   дослідження наявних рішень, представлених у наукових працях, патентних розробках та практичних реалізаціях інтелектуальних бібліотечних систем.

2. **Розробка структури бази даних:**

   проектування бази даних PostgreSQL для зберігання інформації про книжковий фонд, користувачів та їх взаємодії;

впровадження механізмів збору та обробки даних із різних джерел, таких як бібліографічні бази даних, користувацькі профілі та статистика активності.

3. **Реалізація багатовимірного аналізу даних:**

   побудова аналітичної системи для аналізу даних за такими параметрами: час, користувачі, жанри літератури, активність та популярність;

   розрахунок ключових показників ефективності (KPI) для оцінки використання книжкового фонду та задоволеності користувачів.

4. **Розробка аналітичних звітів:**

   статистика популярності книг у розрізі жанрів за вибраний період;

   аналіз користувацької активності в розрізі місяця для вибраних категорій користувачів;

   дослідження взаємодії користувачів з книгами в розрізі часу та жанрів;

   аналіз трендів читацьких уподобань та сезонних коливань попиту.

5. **Дослідження методів машинного навчання:**

   вивчення алгоритмів колаборативної та контентної фільтрації для формування рекомендацій;

   дослідження алгоритмів кластеризації, таких як k-means та ієрархічна кластеризація, для сегментації користувачів.

6. **Розробка рекомендаційної системи:**

   реалізація алгоритмів машинного навчання, зокрема нейронних мереж та методів ансамблювання, для формування персоналізованих рекомендацій;

   аналіз впливу кластеризації користувачів на точність та релевантність рекомендацій.

7. **Порівняння підходів до рекомендацій:**

   оцінка ефективності рекомендаційних алгоритмів з використанням кластеризації та без неї;

   порівняння точності різних методів машинного навчання (колаборативна фільтрація, глибоке навчання, гібридні підходи) у різних сценаріях.

8. **Розробка системи прогнозування попиту:**

створення моделей для прогнозування попиту на книги різних жанрів на основі історичних даних;

аналіз сезонних трендів та факторів, що впливають на читацькі уподобання.

9. **Інтерпретація результатів та розробка рекомендацій:**

оцінка результатів багатовимірного аналізу, роботи рекомендаційної системи та методів прогнозування;

розроблення рекомендацій щодо використання кластеризації й інших інтелектуальних підходів у системах управління бібліотечним фондом.

2 МОДЕЛЮВАННЯ СИСТЕМИ

2.1 Опис та аналіз методології системного аналізу

Системний аналіз є формалізованою методологією, що використовується для дослідження, опису й оцінювання складних систем у різних сферах — від інформаційних технологій до інженерії та бізнес-процесів. Його сутність полягає у поділі системи на ключові структурні елементи з метою виявлення взаємозв’язків між ними та оцінки їхнього впливу на загальне функціонування. Такий підхід забезпечує можливість прийняття обґрунтованих рішень і вдосконалення системи в цілому.

Одним із центральних етапів системного аналізу є побудова моделі системи, що виступає її абстрактним представленням. Модель дає змогу досліджувати поведінку системи в різних умовах, не втручаючись у її реальне середовище. Процес аналізу зазвичай охоплює два основні кроки: декомпозицію (розбиття системи на підсистеми для окремого вивчення) та інтеграцію (об’єднання підсистем у цілісну структуру). Важливою складовою є визначення оптимального рівня деталізації моделі, адже надмірна складність може ускладнити її практичний аналіз.

Системний аналіз виступає базовим методом для формування цілісного уявлення про складні об’єкти, їхні компоненти та взаємодію між ними. Його головна мета полягає у визначенні вимог, формуванні можливих рішень і їх інтеграції задля досягнення заданих результатів. У межах інтелектуальних систем моніторингу цей підхід дозволяє дослідити процеси, пов’язані з обробкою великих масивів даних, виявленням відхилень і прогнозуванням змін параметрів середовища.

Методологія системного аналізу включає послідовні етапи: постановку цілей, побудову моделі процесів, аналіз взаємозв’язків між компонентами, а також відбір інструментів і технологій для реалізації системи. Це забезпечує комплексний підхід до вирішення прикладних задач і досягнення максимальної ефективності її функціонування.

Початковим етапом системного аналізу є визначення вимог, що передбачає збір і опрацювання даних від усіх зацікавлених сторін. Цей етап дає змогу конкретизувати основні функції та завдання, які має виконувати система. Наступним кроком є створення моделі, що відображає функціональні й нефункціональні вимоги, описує структуру даних, напрями їх потоків і взаємодію між компонентами системи.

Важливою складовою системного аналізу є оцінювання альтернативних рішень. Воно полягає у порівнянні різних підходів до реалізації системи, визначенні їх сильних і слабких сторін, потенційних ризиків і впливу на загальну ефективність. Для цього застосовуються як кількісні, так і якісні методи — зокрема SWOT-аналіз і функціональний аналіз.

Ключову роль у системному аналізі відіграє моделювання, що охоплює як математичні, так і предметно-орієнтовані моделі бізнес-процесів. Вони слугують інструментом для формалізації взаємодії компонентів системи. Наприклад, у задачах моніторингу стану повітря моделі дозволяють відображати зміни рівня забруднення та прогнозувати їх наслідки для здоров’я населення.

Практичне використання системного аналізу охоплює інтеграцію інструментів для збору, оброблення й візуалізації даних. Зокрема, можуть використовуватися такі технологічні рішення, як SQL Server, OLAP-куби для багатовимірного аналізу та Power BI для побудови аналітичних панелей. У середовищі Python аналіз даних здійснюється за допомогою бібліотек Pandas і Scikit-learn, що забезпечують реалізацію алгоритмів машинного навчання.

Отримані результати системного аналізу стають підґрунтям для проектування, тестування та впровадження нових систем. У контексті аналізу якості повітря цей підхід сприяє інтеграції методів кластеризації та прогнозування, що підвищує точність оцінок і підтримує створення ефективних стратегій реагування.

Під методологією системних досліджень розуміють сукупність методів, принципів і засобів, спрямованих на розв’язання складних проблем через цілісний підхід до їх вивчення. Системний метод передбачає впорядкований

процес досягнення мети на основі поєднання аналізу та синтезу, тоді як системні засоби охоплюють понятійний і методичний апарат, необхідний для дослідження й оптимізації структури систем.

Системне дослідження проблеми зазвичай передбачає проходження кількох етапів:

1. **Формулювання проблеми**. На цьому етапі проблема визначається як розбіжність між бажаним і фактичним станом об’єкта чи системи. У реальному середовищі будь-яка проблема має розглядатися в контексті пов’язаних із нею підзадач, тобто як елемент «системи проблем», що взаємодіють між собою та потребують комплексного підходу.
2. **Визначення цілей**. Аналіз проблеми дає можливість обрати напрям вирішення, який найбільш ефективно забезпечує досягнення поставленої мети. Виклик полягає у виборі оптимального варіанту серед кількох можливих шляхів.
3. **Формування критеріїв і обмежень**. Критерії визначають якісні характеристики альтернатив, виражені у кількісних показниках, тоді як обмеження задають межі, в яких має здійснюватися пошук рішень. Оптимізація згідно з обраним критерієм забезпечує найкраще наближення до поставленої цілі.
4. **Побудова альтернатив і сценаріїв**. На цьому етапі формується максимальна кількість ідей щодо можливих напрямів досягнення мети. Визначені критерії сприяють пошуку нових рішень, тоді як обмеження дозволяють відсіяти нераціональні або непридатні варіанти.
5. **Оцінювання ресурсів**. Для кожної альтернативи аналізуються необхідні ресурси та їх доступність. Якщо ресурсів недостатньо або існують суттєві обмеження, цілі можуть бути адаптовані чи переглянуті.

Методологія системного аналізу забезпечує впорядковане, логічно послідовне й адаптивне вирішення складних задач, дозволяючи системі реагувати на зміни у зовнішньому середовищі. Найбільш поширена в практиці методика системного аналізу, розроблена Стенлі Янгом, включає десять основних етапів:

1. визначення стратегічних цілей організації;
2. виявлення ключових проблем;
3. детальне дослідження ситуації та постановка діагнозу;
4. пошук можливих шляхів розв’язання проблем;
5. оцінювання альтернатив і вибір оптимального рішення;
6. погодження рішень усередині організації;
7. офіційне затвердження прийнятого рішення;
8. підготовка до його впровадження;
9. управління процесом реалізації;
10. перевірка ефективності результатів.

2.2 Діаграма прецедентів

Діаграма прецедентів у термінах уніфікованої мови моделювання (UML) є графічним інструментом, який демонструє взаємодію між користувачами (акторами) та системою. Вона відображає функціональні вимоги, показуючи, як різні категорії користувачів взаємодіють із конкретними сценаріями або функціями системи. Такі діаграми дозволяють узагальнено представити поведінку системи, що особливо корисно для аналітиків, розробників і замовників. Вони допомагають зрозуміти структуру системи з точки зору користувача, виявити основні бізнес-процеси та визначити межі функціональності.

Головне призначення діаграми прецедентів полягає у визначенні та впорядкуванні вимог до системи з позиції зовнішніх користувачів. Такий підхід дає змогу розробникам і зацікавленим сторонам чітко зрозуміти функціональні можливості системи, створюючи ефективний канал взаємодії між технічними спеціалістами та представниками замовника.

У межах розроблення інтелектуальної системи керування бібліотечним фондом діаграма прецедентів використовується для візуалізації взаємодії між основними користувачами системи (зокрема, читачами, бібліотекарями, адміністраторами) та її ключовими функціями. До таких функцій належать автоматизована каталогізація видань, побудова персоналізованих рекомендацій

і проведення аналітики використання фонду із застосуванням методів машинного навчання та кластеризації. Схематичне зображення діаграми прецедентів системи наведено на рис. 1.

Рис. 1 Діаграма прецедентів

Виділено наступні прецеденти:

1. Пошук книг у каталозі. Запускається читачем. Дозволяє користувачеві шукати книги за різними критеріями: назва, автор, жанр, ISBN, ключові слова, використовуючи інтелектуальний пошук з обробкою природної мови.

2. Отримання персоналізованих рекомендацій. Запускається читачем. Дозволяє отримувати індивідуальні рекомендації книг на основі історії читання,

рейтингів та уподобань користувача з використанням алгоритмів машинного навчання.

3. Автоматизована каталогізація. Запускається системою автоматично. Включає автоматичне розпізнавання метаданих книг, класифікацію за жанрами та тематиками, створення бібліографічних записів.

4. Управління користувачами. Запускається адміністратором. Дозволяє створювати та редагувати профілі користувачів, призначати ролі та права доступу, керувати обліковими записами читачів та бібліотекарів.

5. Управління книжковим фондом. Запускається бібліотекарем. Дозволяє додавати нові книги, редагувати існуючі записи, видаляти застарілі видання, контролювати наявність та стан книг у фонді.

6. Аналіз даних використання. Запускається аналітиком. Включає процес обробки статистики користувацької активності, виявлення патернів поведінки читачів та популярності різних жанрів літератури.

7. Формування аналітичних звітів. Запускається аналітиком. Дозволяє створювати звіти про популярність книг, активність користувачів, ефективність використання фонду та тренди читацьких уподобань.

8. Кластеризація користувачів. Запускається аналітиком. Включає процес сегментації читачів за схожістю інтересів та поведінкових патернів для покращення персоналізації рекомендацій.

9. Формування KPI бібліотеки. Запускається аналітиком. Дозволяє визначати ключові показники ефективності роботи бібліотеки: оборотність фонду, задоволеність користувачів, точність рекомендацій.

10. Прогнозування попиту на літературу. Запускається аналітиком. Дозволяє прогнозувати майбутні потреби в книгах різних жанрів на основі історичних даних та трендів читацьких уподобань.

11. Перегляд статистики та звітів. Запускається бібліотекарем та керівником бібліотеки. Дозволяє переглядати підготовлені звіти для аналізу поточного стану використання фонду та планування майбутніх додавань в систему.

12. Прийняття рішень щодо комплектування. Запускається керівником бібліотеки. На основі переглянутих звітів та прогнозів керівник може приймати рішення щодо додавання нових книг, видалення застарілих видань та оптимізації структури фонду.

2.3 Діаграма розгортання системи керування книжковим фондом

Діаграма розгортання належить до типів UML-діаграм, що описують фізичну інфраструктуру, на якій розміщується й функціонує програмна система. Вона відображає, яким чином програмні компоненти розгортаються на апаратних вузлах і яким пристроям призначено виконання конкретних частин програмного забезпечення.

Основне призначення діаграми розгортання полягає у відображенні зв’язку між логічною архітектурою програмного забезпечення, визначеною під час етапу проєктування, та фізичною архітектурою системи, що забезпечує її виконання. У розподілених програмних рішеннях така діаграма демонструє, як окремі частини системи розподілені між різними фізичними вузлами.

Програмна система реалізується за допомогою низки артефактів, які потім відображаються у середовищі виконання, представленому вузлами чи серверами. У межах діаграми розгортання задіюється кілька таких вузлів, між якими встановлюються зв’язки, позначені інформаційними каналами або шляхами передавання даних.

Схематичне зображення діаграми розгортання розробленої системи наведено на рис. 2.

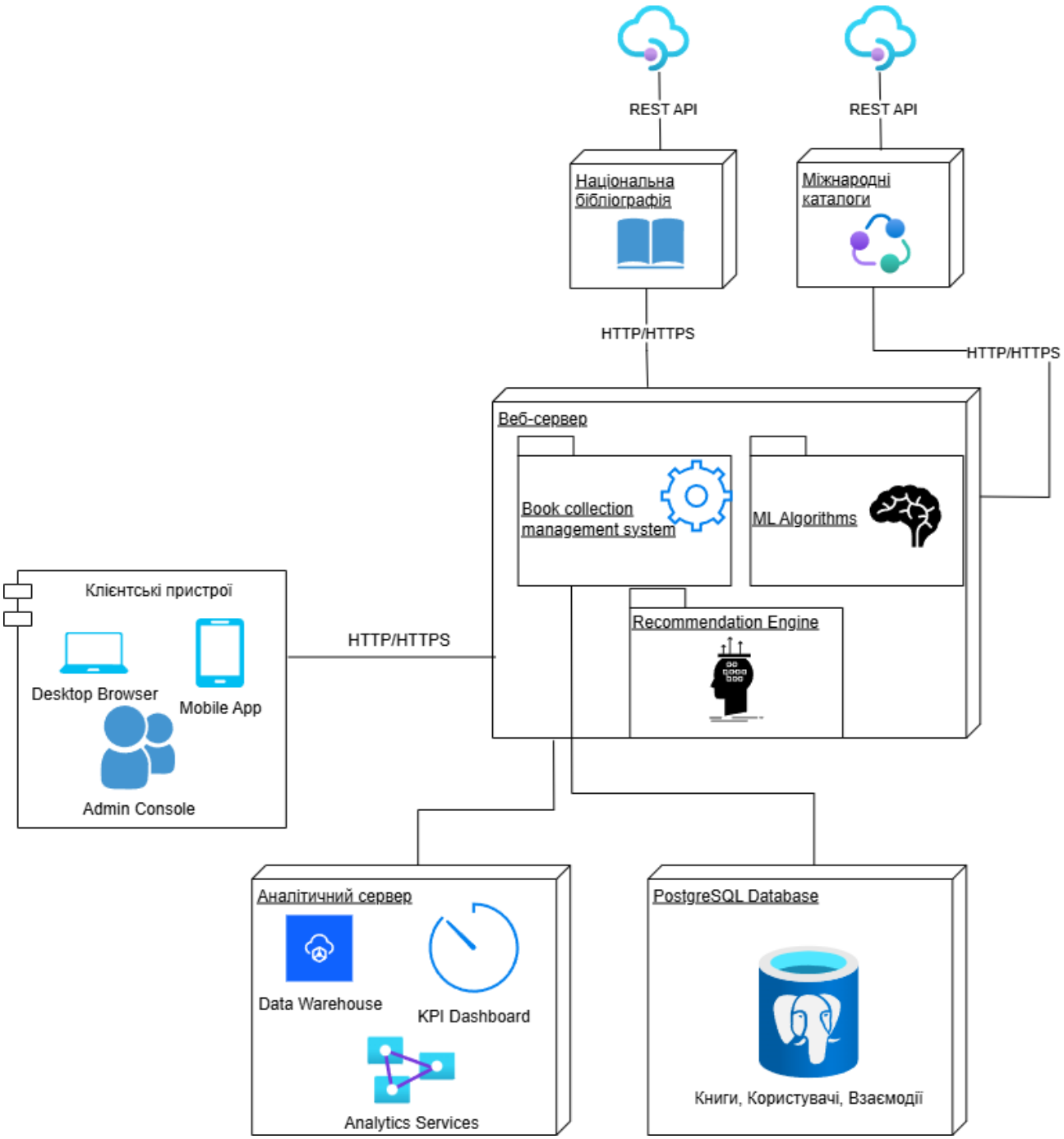

Рис. 2 Діаграма розгортання

На діаграмі розгортання зображено архітектурну структуру інтелектуальної системи управління бібліотечним фондом, що включає основні компоненти, які перебувають у взаємодії один із одним. Короткі характеристики ключових елементів наведені нижче.

1. Вузол «Зовнішні бібліографічні API»: Зовнішні сервіси, такі як Google Books API, Open Library API, WorldCat API, які надають метадані про

книги та автоматично оновлюють бібліографічну інформацію. Передають дані через REST API на веб-сервер системи через HTTPS.

2. Вузол бібліографічних сервісів і систем каталогізації. До складу цього компонента входять джерела даних з боку зовнішніх API, що взаємодіють із веб-сервером через протоколи HTTP або HTTPS. Він охоплює як національні бібліографічні бази, так і міжнародні каталоги бібліотек, які забезпечують актуалізацію даних про нові видання та метадані книжок.

3. Вузол веб-сервера. Цей елемент містить модуль керування (Management Module), що відповідає за обробку інформації, отриманої від зовнішніх сервісів, адміністрування рекомендаційної системи та координацію роботи всіх інших компонентів. Додатково веб-сервер реалізує алгоритми машинного навчання для персоналізації, а також забезпечує користувацький доступ через браузер із використанням HTTP/HTTPS.

4. Аналітичний сервер. Система цього вузла відповідає за накопичення та обробку зведених даних про активність користувачів, статистику використання книжкового фонду та результати кластеризації. Аналітичні модулі реалізують багатовимірний аналіз і формування звітів, даючи можливість зберігати значний обсяг історичних даних для оцінки показників KPI та прогнозування попиту.

5. Сервер бази даних (PostgreSQL). Основна база даних, призначена для зберігання відомостей про книги, користувачів, їхню активність, рейтинги, історію читання та інші оперативні об’єкти. Цей вузол є джерелом даних для аналітичного сервера й рекомендаційної підсистеми.

6. Користувач (веб-браузер). Доступ до системи здійснюють читачі, бібліотекарі та адміністратори через веб-браузер, отримуючи можливість переглядати каталог літератури, користуватися персоналізованими рекомендаціями, аналітичними звітами та результатами прогнозування. Взаємодія із веб-сервером відбувається по протоколу HTTPS.

3 РОЗРОБКА СИСТЕМИ

3.1 Структура джерел даних та їх підготовка для аналізу

3.1.1 Структура оперативної бази даних. У цьому розділі описано структуру джерел даних, які використовуються для збору, зберігання та аналізу інформації про книжковий фонд та користувацьку активність в інтелектуальній системі керування. В основі розробленої системи лежить оперативна база даних PostgreSQL, яка містить різні таблиці, що зберігають інформацію про користувачів, книги, авторів, жанри, рейтинги, історію читання та персоналізовані рекомендації. Ці дані є основою для подальшого аналізу, кластеризації користувачів та прогнозування попиту на літературу, а також для генерації аналітичних звітів. Схему оперативного джерела БД представлено на рис.3.

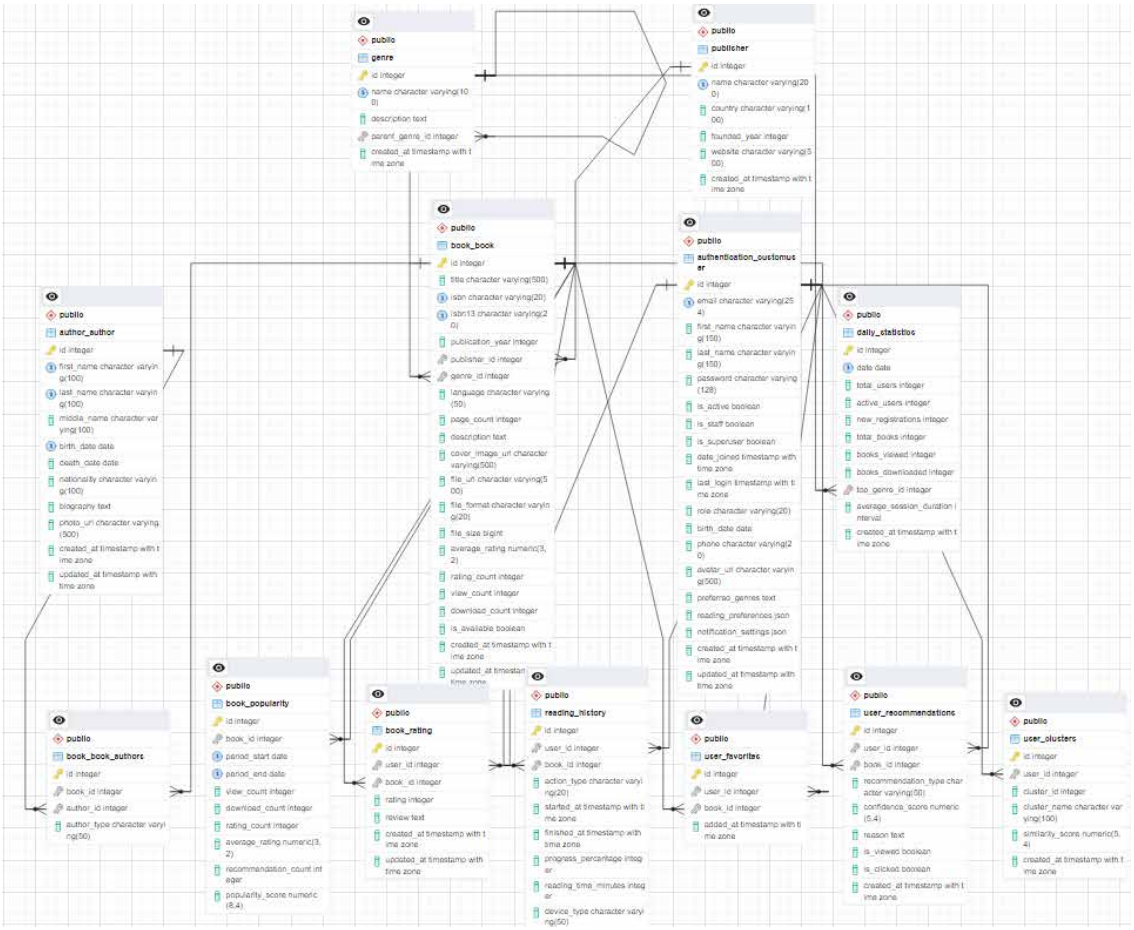

Рис. 3 Схема оперативної БД

Оперативна база даних, яка є центральним елементом інтелектуальної системи управління бібліотечним фондом, містить увесь обсяг необхідної інформації для підтримки аналітичних задач і застосування алгоритмів машинного навчання. Її структура побудована навколо кількох основних груп таблиць, кожна з яких виконує визначену роль у зберіганні даних. До ключових структур належать таблиці з відомостями про користувачів, фонди бібліотеки, авторів і видавництва, а також блоки даних щодо оцінок, відгуків читачів, історії їхніх взаємодій із книжками та результатів роботи інтелектуальних алгоритмів. Така організація забезпечує ефективне накопичення, швидкий доступ і оперативну обробку даних у режимі реального часу.

Основні групи таблиць оперативної БД:

1. Таблиці управління користувачами та безпекою Таблиці для зберігання даних про користувачів (authentication_customuser) та систему прав доступу (auth_group, auth_permission) дозволяють системі ідентифікувати кожного читача, бібліотекаря та адміністратора, що спрощує управління доступом до різних функцій системи. Користувацькі профілі включають персональні налаштування, уподобання жанрів та історію активності, що використовується для персоналізації рекомендацій.

2. Таблиці книжкового фонду Таблиця з книгами (book_book) містить детальну інформацію про кожне видання: назву, ISBN, рік видання, опис, жанр, рейтинги та статистику перегляду. Пов'язані таблиці авторів (author_author), видавництв (publisher) та жанрів (genre) забезпечують повну каталогізацію фонду. Зв'язуюча таблиця book_book_authors реалізує багато-до-багатьох відношення між книгами та авторами, що дозволяє коректно обробляти книги з кількома авторами.

3. Таблиці взаємодії користувачів з контентом Дані про взаємодії користувачів з книгами зберігаються в таблицях user_favorites (обрані книги), book_rating (рейтинги та відгуки) та reading_history (історія читання). Ці таблиці містять часові мітки всіх дій користувачів, що дозволяє відслідковувати динаміку зміни уподобань та поведінкових патернів читачів.

4.Таблиці системи машинного навчання Таблиці user_clusters та user_recommendations зберігають результати роботи алгоритмів кластеризації користувачів та персоналізованих рекомендацій відповідно. Таблиця кластерів містить ідентифікатори груп користувачів зі схожими інтересами та коефіцієнти подібності, що використовуються для колаборативної фільтрації. Рекомендації включають тип алгоритму (колаборативний, контентний, гібридний), рівень довіри та пояснення причин рекомендації.

5. Таблиці аналітики та звітності Агреговані дані для аналітики зберігаються в таблицях daily_statistics (щоденна статистика системи) та book_popularity (популярність книг за періодами). Ці таблиці містять KPI системи: кількість активних користувачів, популярні жанри, динаміку використання фонду, що використовується для формування керівних звітів та прогнозування попиту на літературу.

Подібна структура дає змогу підтримувати нормалізовані зв’язки між даними, що забезпечує ефективне зберігання та обробку великих масивів інформації щодо користувацької активності. Зібрані дані активно використовуються для реалізації алгоритмів машинного навчання, кластеризації читачів і прогнозування тенденцій у сфері користування бібліотечним фондом. Саме цей підхід дозволяє системі формувати актуальні персоналізовані рекомендації та своєчасно визначати потреби для поповнення книжкового фонду.

Ключові особливості структури БД:

- Нормалізована структура - мінімізує дублювання даних та забезпечує цілісність
- Індексація - оптимізована для швидкого пошуку та аналітичних запитів
- JSON поля - гнучке зберігання налаштувань користувачів
- Часові мітки - повна історія всіх змін для аналізу трендів
- Тригери - автоматичне оновлення агрегованих показників
- Масштабованість - готовність до роботи з великими обсягами даних

3.1.2 Основні поняття OLAP-технологій. OLAP (Online Analytical Processing) — це технологія, що орієнтована на оперативне виконання складних аналітичних запитів й багатовимірний аналіз великих масивів даних у різних сховищах, таких як дата-репозиторії чи озера даних. OLAP широко застосовується у бізнес-аналітиці, системах підтримки прийняття рішень, а також для прогнозування та формування звітності в організаціях. Зазвичай підприємницькі дані мають кілька ключових вимірів — категорій, за якими інформація класифікується для аналізу та відстеження.

Структура OLAP-куба виступає інструментом швидкого аналізу даних за множинними вимірами, релевантними до розв’язання бізнес-задач. Наприклад, модель для аналізу продажів може містити такі виміри, як продавець, сума продажу, регіон, продукт, місяць та рік. OLAP-інструменти інтерактивно дозволяють користувачам досліджувати багатовимірну інформацію з різних перспектив. Основні аналітичні операції в OLAP включають:

- Агрегацію (roll-up): узагальнення даних за окремими вимірами, наприклад, об'єднання регіональних результатів у загальну картину по філіалу.
- Деталізацію (drill-down): можливість перегляду детальних даних за окремими категоріями або підгрупами.
- Слайсинг і дайсинг (slicing and dicing): відбір певних підмножин даних із куба та їх аналіз під різним кутом.

Існують кілька типів OLAP-систем:

- MOLAP (Multidimensional OLAP): використовує багатовимірні сховища для організації даних.
- ROLAP (Relational OLAP): функціонує поверх реляційних баз даних, застосовуючи спеціальні запити для симуляції багатовимірної структури.
- HOLAP (Hybrid OLAP): поєднує підходи MOLAP і ROLAP, досягаючи підвищеної продуктивності та гнучкості роботи із даними.

У сфері інтелектуального управління бібліотечними ресурсами OLAP-технології мають велике значення для комплексного аналізу інформації та персоналізації послуг. Читацькі переваги формуються під впливом різноманітних чинників - демографічних характеристик, сезонних коливань, популярності авторів і жанрів. Ефективне управління фондом вимагає врахування багатовимірності даних. OLAP-куби дають змогу структуровано аналізувати дані за численними параметрами, виконувати складні аналітичні запити та виявляти тенденції у поведінці користувачів. Саме багатовимірний аналіз із поділом даних за часовими, користувацькими і жанровими вимірами, типом активності, забезпечує оптимізацію роботи інтелектуальної бібліотечної системи та якісне прогнозування майбутніх потреб.

У межах даного дослідження було побудовано OLAP-куб та сформовані звіти для аналізу використання книжкового фонду за показниками:

- Найпопулярніші книги за вказаним жанром у розрізі періодів за певний рік
- Середня активність читачів по окремих категоріях у розрізі демографічних груп
- Середні значення рейтингів у розрізі часу по вибраних жанрах
- Використання фонду у розрізі категорії користувачів та часових періодів
- Тренди змін читацьких уподобань у часі

Для підвищення точності персоналізованих рекомендацій і прогнозування попиту на книжкові ресурси було використано інтеграцію методів кластеризації користувачів із алгоритмами машинного навчання, такими як колаборативна фільтрація, нейронні мережі та ансамблеві підходи. Така комбінована стратегія дає змогу формувати групи читачів із подібними ознаками, розробляти окремі моделі рекомендацій для кожного кластеру та забезпечувати більш релевантні пропозиції. Це дозволяє точніше прогнозувати інтереси користувачів із урахуванням специфічних уподобань різних спільнот та сезонної динаміки попиту на літературу.

Отже, застосування OLAP-технологій у поєднанні з методами кластеризації та машинного навчання є сучасним та ефективним рішенням для створення інтелектуальних систем управління бібліотечним фондом. Такий підхід дозволяє не лише відстежувати використання ресурсів бібліотеки, а й здійснювати точне прогнозування попиту, що оптимізує процес комплектування фонду й сприяє підвищенню рівня задоволеності користувачів.

Для подальшої оцінки ефективності впроваджених інтелектуальних технологій управління книжковим фондом доцільно використовувати систему ключових показників ефективності (KPI). KPI представляють собою кількісні метрики, які дозволяють оцінити досягнення організації чи окремих працівників у реалізації стратегічних і операційних цілей бібліотечного менеджменту. Вони забезпечують об’єктивну оцінку ефективності функціонування інтелектуальної системи, відстежують прогрес у покращенні обслуговування та ідентифікують відхилення від планових результатів.

Впровадження системи KPI доцільно здійснювати на основі чітко визначених критеріїв - конкретних, вимірюваних, досяжних, релевантних та обмежених у часі. Прикладами таких показників можуть бути точність роботи рекомендативних алгоритмів, швидкість обробки користувацьких звернень, кількість успішних рекомендацій, рівень задоволеності читачів, оборотність книжкового фонду та своєчасне поповнення новими виданнями згідно з прогнозованим попитом.

3.1.3 Архітектура сховища даних. Архітектура аналітичного сховища даних є базовим елементом для організації процесів накопичення та обробки великих обсягів інформації про дії користувачів, стан бібліотечного фонду та читацькі переваги, що надходить із різних підсистем інтелектуальної бібліотечної платформи. Сховище даних (Data Warehouse) виступає централізованою платформою для збереження, оброблення й аналітики бібліотечних даних, підтримуючи ухвалення ефективних рішень щодо управління фондом і персоналізації сервісів з опорою на аналітичні результати.

Ключовими компонента архітектури аналітичного сховища є:

- Джерела даних. Включають внутрішні й зовнішні системи, такі як оперативні бази (зокрема PostgreSQL), зовнішні бібліографічні API, лог-файли активності користувачів, статистичні набори даних тощо.
- ETL-процеси (Extract, Transform, Load). Забезпечують отримання даних із джерел, їх трансформацію у придатний для аналітики формат та подальше завантаження в сховище.
- Сховище даних. Є централізованою аналітично орієнтованою базою, оптимізованою для виконання складних запитів та швидкої обробки великого масиву даних щодо читацьких тенденцій.
- Тематичні марти даних (Data Marts). Це підсистеми сховища, що виділяють окремі аспекти діяльності бібліотеки - поведінка користувачів, популярність книг і жанрів, ефективність рекомендаційних рішень.
- OLAP-сервіси. Відповідають за багатовимірний аналіз інформації, надаючи інструменти для складних запитів і формування звітів щодо використання фонду.
- Презентаційний рівень. Включає користувацькі інтерфейси, додатки для візуалізації, створення інтерактивних віджетів KPI і аналітичних панелей для адміністрації бібліотеки.

Архітектура аналітичних сховищ даних може реалізовуватися у різних конфігураціях, зокрема одно-, дво- та трирівневої структури. Однорівнева архітектура передбачає розміщення всіх елементів системи на одному сервері, що є оптимальним рішенням для малих бібліотек із невеликим обсягом даних. Дворівнева модель розмежовує сервер бази даних і клієнтські аналітичні додатки, завдяки чому підвищується продуктивність роботи й рівень безпеки. Трирівнева архітектура додатково містить аплікаційний сервер, який обробляє бізнес-логіку, виконує рекомендаційні алгоритми та забезпечує розширену гнучкість і масштабованість системи.

Вдалий вибір архітектури аналітичного сховища даних відкриває низку ключових можливостей для інтелектуального управління бібліотечним фондом:

- Підтримку масштабованості - із можливістю обробки великих і зростаючих обсягів даних без зниження продуктивності.
- Високу швидкодію при виконанні аналітичних запитів та формуванні рекомендацій завдяки оперативному доступу до накопиченої історичної інформації.
- Інтеграцію даних із різних джерел - наприклад, із оперативної бази, зовнішніх API чи систем веб-аналітики - у єдину структуру даних, що підвищує якість аналітичних висновків і сприяє прийняттю обґрунтованих рішень щодо комплектування фонду.
- Забезпечення конфіденційності та контроль доступу - архітектура дозволяє захищати персональну інформацію читачів і гнучко управляти правами доступу до аналітичної інформації.

Сучасні аналітичні сховища даних інтегруються з хмарними технологіями підтримують обробку структурованих і неструктурованих даних забезпечують гнучкість економію ресурсів дозволяють використовувати аналітичні інструменти та алгоритми машинного навчання для персоналізації рекомендацій

У розробці інтелектуальної системи керування книжковим фондом аналітичне сховище допомагає зберігати та аналізувати великі обсяги інформації про читацькі вподобання та використання ресурсів застосування OLAP-технологій дає можливість здійснювати багатовимірний аналіз популярності книг використання методів прогнозування кластеризація колаборативна фільтрація нейронні мережі підвищує точність персоналізованих рекомендацій і підтримує прийняття рішень щодо розвитку бібліотечної колекції

Архітектура аналітичного сховища даних формує узгоджений систематизований аналіз бібліотечної інформації у межах інтелектуальної системи забезпечує збереження даних з різних джерел інтеграцію користувацьких бібліографічних і часових параметрів отримання комплексної аналітичної картини трендів дозволяє здійснювати моніторинг використання фонду прогнозувати попит враховуючи сезонні демографічні та жанрові закономірності

Характерною рисою архітектури є поєднання OLAP-кубів з сучасними методами машинного навчання інтеграція цих рішень дає змогу не лише виявляти проблеми низька оборотність жанрів прогнозувати майбутні сценарії розвитку інтересів детально аналізувати історичні дані приймати ефективні рішення для оптимізації комплектування та підвищення задоволеності користувачів

Зображення архітектури аналітичного сховища даних представлено на рис.4 нижче.

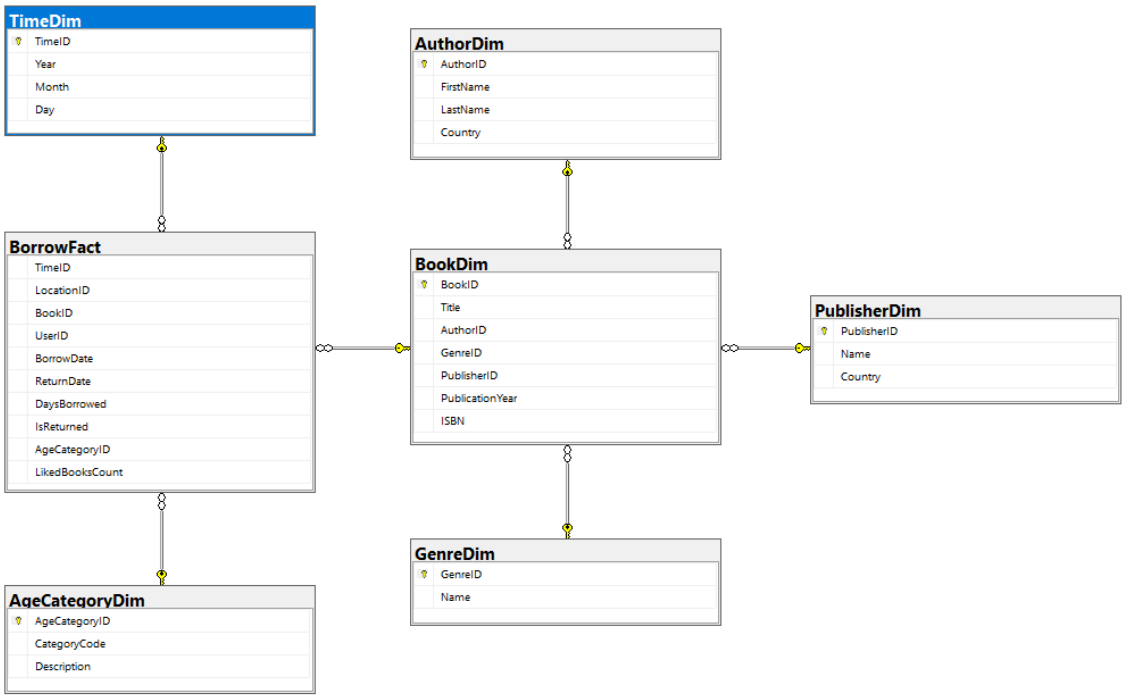

Рис. 4 Архітектура сховища даних

Представлена архітектура аналітичного сховища даних розроблена для підтримки аналізу та моніторингу використання книжкового фонду бібліотеки та активності читачів. Вона реалізує класичну зіркову схему, де центральна фактова таблиця BorrowFact пов'язана з п'ятьма ключовими вимірними таблицями.

Вимірні таблиці включають:

- TimeDim – часові виміри, що фіксують темпоральні аспекти операцій (рік, місяць, день)

- BookDim – каталог книг, що містить детальну інформацію про видання (назва, ISBN, рік публікації) та зв'язки з авторами, жанрами та видавцями

- AuthorDim – довідник авторів з їхніми біографічними даними (ім'я, прізвище, країна походження)

- GenreDim – класифікатор літературних жанрів з їхніми характеристиками

- PublisherDim – каталог видавництв з географічною прив'язкою

- AgeCategoryDim – класифікація вікових категорій читачів з відповідними кодами та описами

Центральна фактова таблиця BorrowFact фіксує всі операції видачі та повернення книг, включаючи:

- Ідентифікатори локації, книги та користувача

- Дати видачі та повернення

- Тривалість користування книгою

- Факт повернення книги

- Кількість позичених книг

- Вікову категорію читача

Ця структура дозволяє проводити багатовимірний аналіз бібліотечної діяльності за різними критеріями: часовими періодами, жанрами літератури, авторами, видавництвами, віковими групами читачів та географічними регіонами. Зіркова схема забезпечує оптимальну продуктивність OLAP-запитів та підтримує створення різноманітних аналітичних звітів для управління книжковим фондом.

Для створення аналітичного сховища даних було написано SQL-скрипт зі створення структури БД, що представлено в Додатку В на сторінці 1-3.

3.1.4 Процеси вилучення, трансформації та завантаження даних. Процеси отримання, обробки та завантаження даних (ETL) є базовими складовими при розгортанні та підтримці сховищ даних. ETL-процедури здійснюють переміщення даних із різноманітних джерел, їхню очистку, форматування згідно з вимогами сховища та подальше внесення до цільової аналітичної системи. Такий підхід сприяє інтеграції та узгодженню інформації з різних систем, дозволяючи сформувати комплексне подання даних для аналізу.

Для реалізації ETL часто застосовують спеціалізовані програмні рішення - наприклад, SQL Server Integration Services (SSIS), який входить до складу Microsoft SQL Server. SSIS забезпечує розширені функції для інтеграції, трансформації та адміністрування даних, спрощує створення багатокомпонентних ETL-процесів, автоматизує імпорт та експорт інформації, а також гарантує високу ефективність роботи з великими обсягами даних.

Процедура наповнення сховища даних здійснюється поетапно та охоплює три основні кроки, що забезпечують передачу інформації від оперативної бази до аналітичного сховища для подальшої обробки. Ці етапи включають вилучення даних, їхню відповідну трансформацію і подальше завантаження у сховище з урахуванням багаторівневої структури архітектури системи. Взаємозв’язки між окремими потоками даних можна побачити на рис. 5.

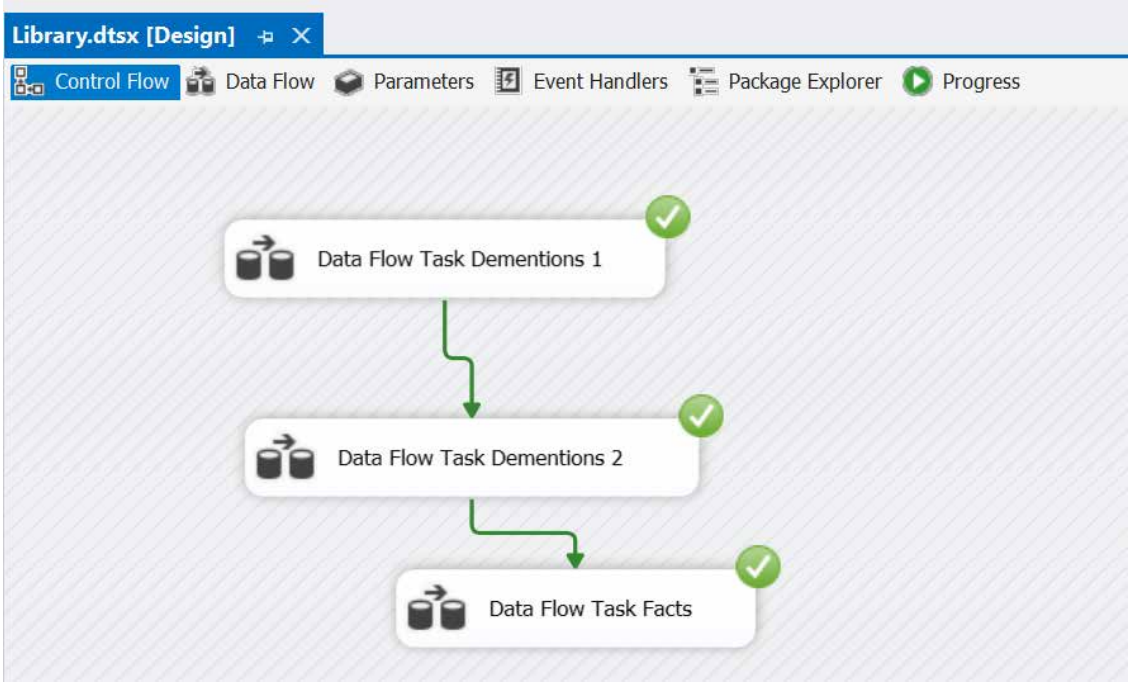

Рис. 5 Потоки даних для наповнення СД

Структура сховища даних передбачає три рівні ETL-завдань, які реалізуються через інтерфейс SQL Server Integration Services (SSIS). На першому етапі (Data Flow Task Dimensions 1) проводиться заповнення основних таблиць вимірів — зокрема, таблиць книг (BookDim) із метаданими, часових вимірів (TimeDim), а також вікових категорій (AgeCategoryDim). На цій стадії дані збагачуються із врахуванням часових параметрів (рік, місяць, день) та інших важливих характеристик книжкового фонду. Для створення таких таблиць у SSIS застосовуються компоненти Sort та Merge Join; інструмент Sort забезпечує упорядкування даних за визначеним ключем для кожного джерела перед їх об’єднанням, що показано на рис. 6.

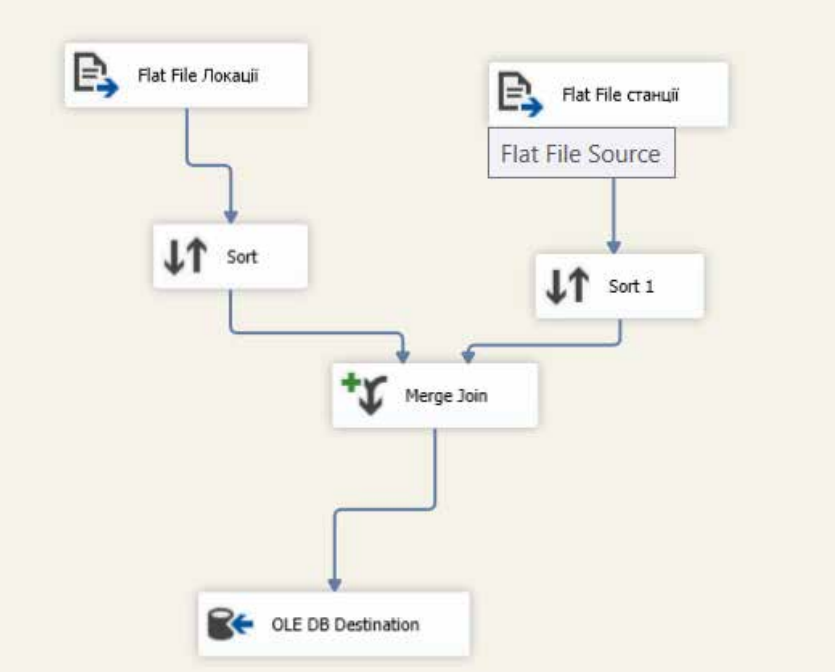

Рис. 6 Заповнення таблиці локацій

Таблиця авторів підлягає сортуванню за унікальним ідентифікатором автора (AuthorID), що забезпечує коректне об'єднання записів із біографічними даними та відомостями про країну походження.

Схематичне представлення потоку завдань для заповнення сховища даних із оперативної бази на першому рівні наведено на рис. 7.

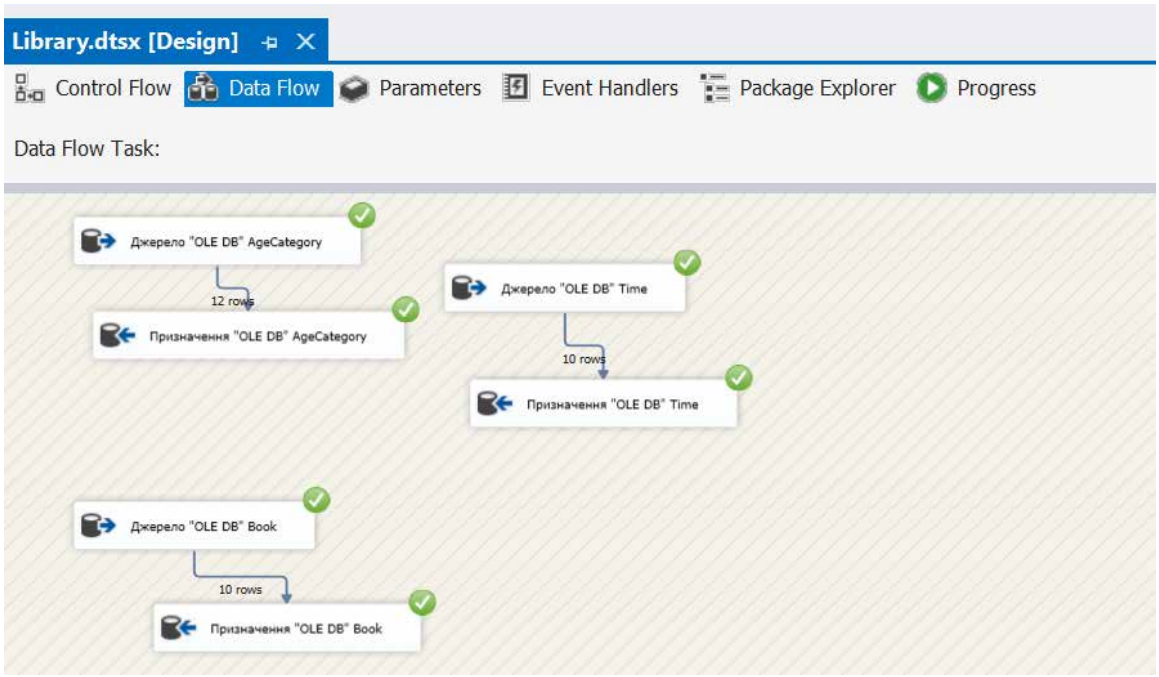

Рис. 7 Потік завдань першого рівня

На другому етапі (Data Flow Task Dimensions 2) здійснюється обробка складніших вимірів, які охоплюють додаткові категорії та ключові довідкові параметри, зокрема таблиці «Автори» (AuthorDim), «Видавництва» (PublisherDim) й «Жанри» (GenreDim). Процес наповнення вимірів другого рівня ілюструється на рис. 8.

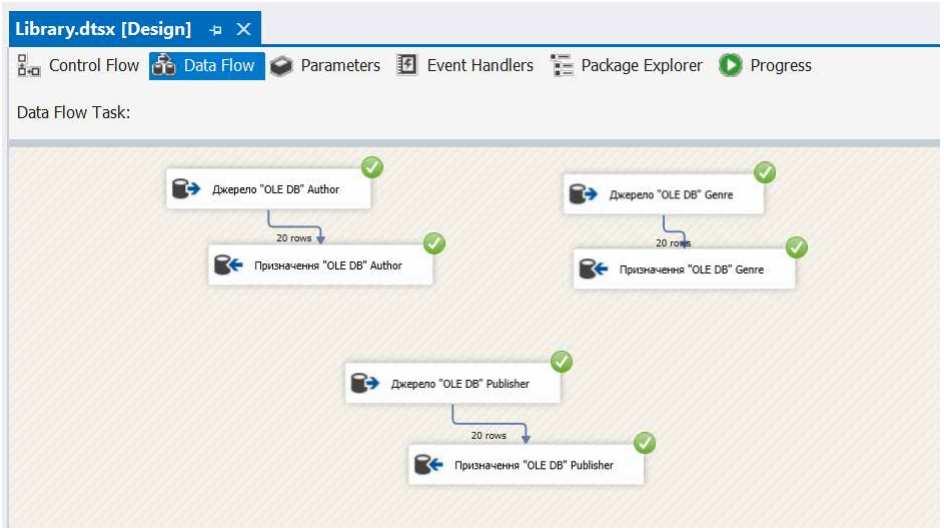

Рис. 8 Заповнення вимірів другого рівня

На завершальному етапі (Data Flow Task Facts) відбувається заповнення основної таблиці фактів BorrowFact, яка агрегує інформацію про видачу та

повернення книг у межах усіх підрозділів бібліотеки. Для формування цієї таблиці використовуються дані, отримані на попередніх двох етапах. Запити до оперативної бази даних дозволяють отримати статистику користування літературою за встановлений часовий інтервал, яка об'єднується з відповідними атрибутами із таблиць вимірів. Схематичне виконання запиту для накопичення таких даних та наповнення таблиці фактів продемонстровано на рис. 9.

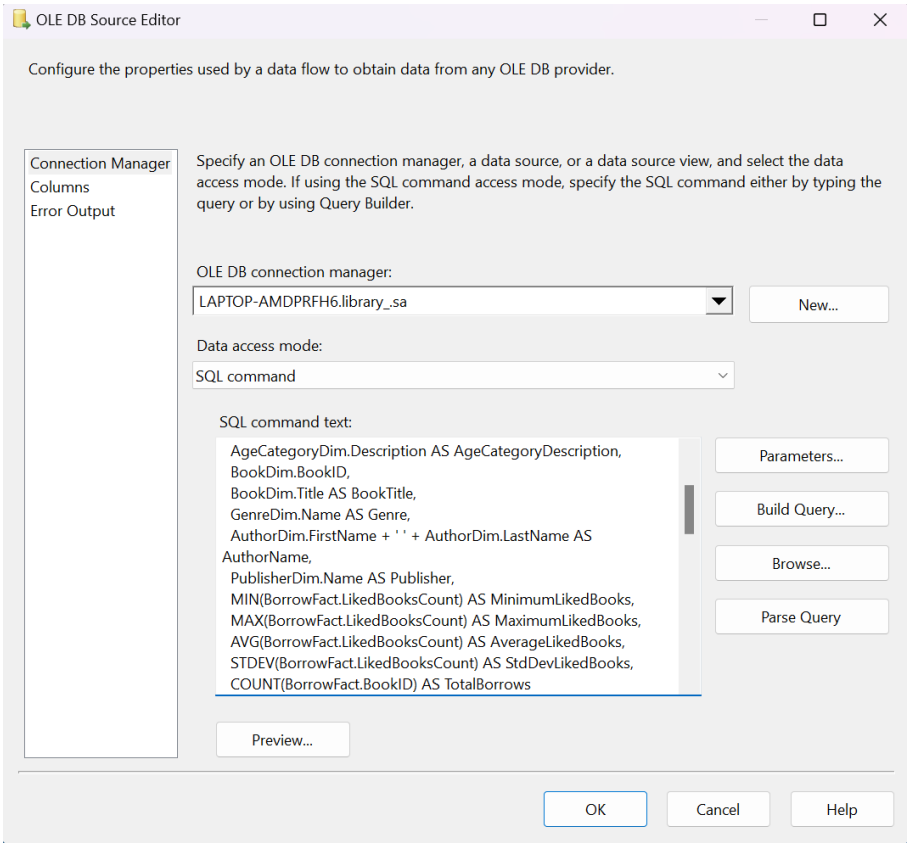

Рис. 9 Формування запиту на основі джерела оперативної БД

SQL-запит здійснює вибірку інформації з низки таблиць для обрахунку основних статистичних параметрів використання бібліотечного фонду: кількості видач, тривалості користування, частоти повернень та рейтингу популярності кожної книги з урахуванням читачів, місць користування і часових періодів. Результати агрегуються за такими категоріями, як локація, користувач, книга та часовий ідентифікатор (TimeID), що дозволяє отримати узагальнену інформацію для подальшого аналізу в системі управління фондом.

На рис. 10 продемонстровано процес перенесення агрегованих даних до сховища з використанням компонента OLE DB Destination Editor у середовищі

SSIS. Отримані екземпляри даних під час SQL-запиту передаються у відповідні поля цільової таблиці BorrowFact; ключові атрибути - TimeID, LocationID, BookID, UserID, BorrowDate, ReturnDate, DaysBorrowed і IsReturned - зіставляються з відповідними стовпцями бази даних сховища.

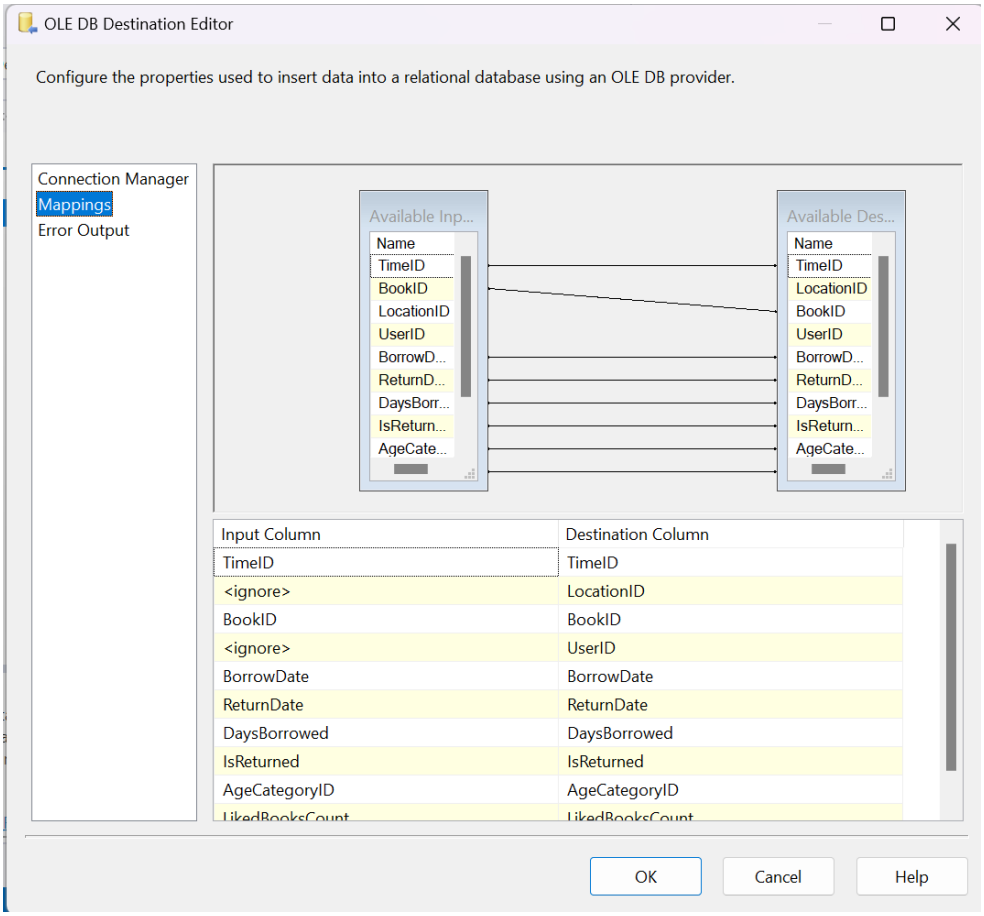

Рис. 10 Процес передачі даних на основі сформованої вибірки

3.2 Огляд методів Data Mining

В обробці сучасних інтелектуальних систем управління бібліотечним фондом критично важливим завданням є аналіз великих масивів даних для ідентифікації закономірностей у читацьких вподобаннях і точного прогнозування попиту на літературу. Технології Data Mining надають широкий спектр інструментів для ефективного аналізу цих масивів і підтримки прийняття обґрунтованих управлінських рішень у бібліотечній сфері.

Кластеризація являє собою процес групування об'єктів даних у низку окремих класів (кластерів), які містять схожі характеристики. Цей підхід є одним із ключових методів аналізу інформації в рамках інтелектуального опрацювання

бібліотечних даних. Алгоритм K-середніх належить до методів розподільчої кластеризації, що дозволяє розбивати об’єкти на K груп. K-means відзначається високою швидкодією та ефективністю для кластеризації великих читацьких масивів інформації, однак якість отриманих результатів значною мірою залежить від випадкового обрання початкових центроїдів.

В аналізі бібліотечних даних кластеризація методом K-середніх використовується для групування читачів за схожими читацькими уподобаннями та поведінковими характеристиками. Це дозволяє виявити категорії користувачів з подібними інтересами до певних жанрів, авторів або тематик і спрощує персоналізацію рекомендаційних сервісів. За допомогою кластеризації можна ідентифікувати групи активних читачів та спрямувати зусилля на розвиток колекції відповідно до потреб конкретних сегментів користувачів.

Метод асоціативних правил, вперше запропонований Р. Агравалом і Р. Срікантом, використовується для виявлення зв’язків між даними. Алгоритм працює за допомогою ітеративного пошарового підходу: спочатку формується набір кандидатів, після чого для всіх k-елементних варіантів обчислюється показник підтримки. Якщо цей показник перевищує визначений мінімальний поріг, відповідний набір фіксується як частий.

Далі на основі знайдених частих наборів L_k генерується новий набір кандидатів C_(k+1), для якого знов обчислюється підтримка. Якщо вона відповідає порогу, набір приєднується до частих L_(k+1). Процес триває, доки всі часті варіанти не будуть знайдені або не залишиться нових відповідних наборів. З частих груп елементів вибудовують асоціативні правила. Окремо розраховується показник довіри (confidence); якщо він перевищує потрібний мінімум, правило вважається значущим і підлягає генерації.

Застосування пошуку асоціативних правил дозволяє знаходити зв’язки між книгами, авторами та жанрами у читацьких уподобаннях: наприклад, які книги часто обирають разом, які автори є популярними у прихильників певних жанрів, як демографічні особливості користувачів впливають на вибір літератури. Це дає змогу краще розуміти комплексний характер інтересів аудиторії та формувати

більш ефективні стратегії комплектування фонду та персоналізованих рекомендацій.

3.3 Інструментарій для аналізу даних Під час створення інтелектуальної системи керування бібліотечним фондом застосовуються різноманітні інструменти та технології, що забезпечують якісний збір, зберігання, аналітику й візуалізацію бібліотечних даних. Важливе місце серед компонентів системи займають Microsoft SQL Server, SQL Server Analysis Services (SSAS), SQL Server Reporting Services (SSRS), Power BI, а також мова програмування Python із бібліотеками машинного навчання.

SQL Server виступає базовою платформою для реляційного зберігання даних, гарантує надійність і високу продуктивність при роботі з великими інформаційними масивами. За його допомогою створено аналітичне сховище, що акумулює історію користування фондом, читацькі вподобання й забезпечує швидкий доступ до даних для подальших досліджень.

SQL Server Analysis Services (SSAS) використовується для побудови багатовимірних моделей даних і впровадження OLAP-технологій. SSAS дозволяє створювати багатовимірні куби, визначати виміри та розраховувати міри, що забезпечує гнучкий і оперативний доступ до агрегованої інформації щодо використання бібліотечних ресурсів. Аналіз із використанням кубів значно полегшує опрацювання великих обсягів читацьких даних і підтримує складні інформаційні запити.

У рамках SSAS також реалізуються структури для Data Mining (Mining Structures), що дозволяють застосовувати алгоритми для виявлення прихованих закономірностей: кластеризації, класифікації чи прогнозування в сфері читацьких вподобань і попиту на певні групи літератури.

Python слугує важливим інструментом для побудови систем рекомендацій і впровадження алгоритмів машинного навчання: завдяки бібліотекам scikit-learn, pandas, numpy забезпечується ефективна обробка значних масивів даних, статистичний аналіз та формування моделей для прогнозування попиту.

Зокрема, такі моделі, як Random Forest чи Gradient Boosting, застосовуються для прогнозування популярності книг та персоналізації рекомендацій.

Обчислення ключових показників ефективності (KPI) реалізовано в середовищі SSAS: це дає можливість здійснювати кількісний аналіз ефективності використання фонду, відслідковувати динаміку читацької активності, оцінювати результативність впроваджених інновацій і приймати зважені управлінські рішення щодо розвитку фонду та сервісів бібліотеки.

В середовищі SSAS доступні основні методи аналізу бібліотечних даних, серед них:

- Дерева рішень (Decision Trees) використовуються для класифікації читачів і передбачення їхніх уподобань.
- Кластеризація (Clustering) застосовується для формування груп користувачів із схожими інтересами.
- Наївний байєсівський метод (Naive Bayes) забезпечує швидку категоризацію нових користувачів.
- Метод асоціативних правил (Association Rules) дозволяє виявити зв’язки між книгами та авторами у структурі читацьких уподобань.
- Нейронні мережі (Neural Networks) дають змогу моделювати складні патерни поведінки аудиторії.
- Аналіз часових рядів (Time Series Analysis) використовується для прогнозування сезонних тенденцій у попиті на літературу.
- Регресійний аналіз (Regression) дозволяє оцінити фактори, що впливають на популярність окремих книг.

Для генерації звітів і побудови візуалізації бібліотечних даних застосовуються SQL Server Reporting Services (SSRS) та Power BI. SSRS дозволяє створювати табличні, матричні й графічні звіти, які можуть бути опубліковані на веб-порталі або інтегровані до інформаційної системи бібліотеки. Power BI реалізує інтерактивну візуалізацію статистики, створення аналітичних панелей і підтримку колаборативної роботи над звітами, що робить аналітику доступною для фахівців без поглиблених технічних знань.

SSAS надає наступні основні методи аналізу бібліотечних даних:SQL Server Integration Services (SSIS) є потужним інструментом для інтеграції даних, який використовується для побудови складних ETL-процесів (Extract, Transform, Load) в бібліотечному середовищі. Цей інструмент дозволяє автоматизувати процеси вилучення даних з різноманітних джерел, їх перетворення у відповідний формат та завантаження до сховища даних. У рамках розробки інтелектуальної системи управління книжковим фондом SSIS використовувався для інтеграції даних з бібліотечних каталогів, систем обліку читачів та зовнішніх систем, таких як онлайн-каталоги видавництв та рейтинги книг. Завдяки вбудованим можливостям обробки даних SSIS дозволяє виконувати очищення, перевірку, нормалізацію та трансформацію бібліотечних даних ще на етапі їх перенесення, що сприяє підвищенню якості інформації у сховищі.SSIS також забезпечує гнучкість у налаштуванні складних робочих процесів завдяки підтримці сценаріїв, умовних операторів і вбудованих коннекторів до різноманітних джерел даних. Наприклад, у проєкті управління книжковим фондом SSIS був використаний для автоматичного оновлення інформації про книги у сховищі, об'єднання даних з різних джерел, таких як файли CSV з каталогами, веб-ресурси видавництв та бази даних читацької активності. Інструмент дозволяє створювати детальну документацію та візуалізувати ETL-процеси, що забезпечує зручність управління бібліотечними даними. Його інтеграція зі сховищем даних у SQL Server гарантує надійну обробку великих обсягів інформації, що є критично важливим для ефективного управління бібліотечними ресурсами, де точність та оперативність мають ключове значення.

OLAP-технології дають змогу виконувати багатовимірний аналіз бібліотечних даних, оперативно отримувати відповіді на складні аналітичні запити і здійснювати згортання чи деталізацію інформації за різними вимірами - такими як часові інтервали, жанри, автори чи вікові категорії читачів. Це особливо актуально при дослідженні тенденцій користування бібліотечним фондом, де важливо враховувати численні фактори і взаємозв'язки між ними.

У ході побудови сховища даних та аналітичних моделей головним завданням стає забезпечення якості бібліотечної інформації. Для цього в середовищі SQL Server й за допомогою Python застосовують методи очищення, нормалізації та трансформації даних, зокрема ідентифікацію та обробку пропущених значень у каталогах, видалення дублікатів і перевірку узгодженості користувацьких записів.

Комплексний підхід, що поєднує інструменти SQL Server, SSAS, SSRS, Power BI та Python, дозволяє створити інтелектуальну систему для управління книжковим фондом, що охоплює всі етапи роботи з бібліотечними даними: від збору та зберігання до аналітики та візуалізації. Це підвищує ефективність бібліотечного менеджменту й сприяє прийняттю стратегічних рішень щодо розвитку і поповнення книжкової колекції.

Застосування сучасних засобів аналітики дозволяє глибше зрозуміти потреби читачів, знаходити приховані закономірності у використанні фонду й розробляти дієві стратегії оптимізації управління бібліотекою та підвищення якості її послуг.

3.4 Дані для аналізу

У рамках розробки інтелектуальної системи управління книжковим фондом використовувалися дані, отримані з декількох основних джерел. Першим джерелом є внутрішня бібліотечна інформаційна система (АБІС), де зберігаються дані про книжковий фонд, читачів та операції видачі/повернення. Другим джерелом виступають зовнішні каталоги та API видавництв, які надають метадані про книги та їх характеристики. Третім джерелом є системи онлайн-

рейтингів та рецензій (наприклад, Goodreads API), які забезпечують додаткову інформацію про популярність та оцінки книг.

Дані з внутрішньої АБІС охоплюють широкий спектр показників функціонування бібліотеки. Основні з них представлено нижче:

1. Характеристики книг: ISBN, назва, автор(и), жанр, рік видання, кількість сторінок
2. Метадані видань: видавництво, країна видання, мова, тираж, ціна
3. Фізичні характеристики: формат, стан збереження, локація у бібліотеці
4. Статистика використання: кількість видач, середня тривалість користування
5. Читацькі дані: ID читача, вік, стать, освіта, область інтересів
6. Операційні дані: дати видачі та повернення, продовження термінів
7. Рейтинги та відгуки: оцінки читачів, коментарі, рекомендації
8. Резервування: заявки на бронювання, черги очікування
9. Штрафи та порушення: прострочення, пошкодження, втрати
10. Сезонна активність: коливання попиту за періодами року
11. Демографічні тенденції: розподіл читачів за віковими групами
12. Жанрові уподобання: популярність різних категорій літератури
13. Авторські рейтинги: найбільш затребувані письменники
14. Мовні преференції: розподіл за мовами видань
15. Тематичні категорії: наукова, художня, довідкова література
16. Новинки та класика: співвідношення сучасних та класичних творів
17. Електронні ресурси: використання цифрових видань
18. Комплексний індекс ефективності: загальна оцінка використання фонду

Проте хоча ці дані досить детальні, обмеженість внутрішніми ресурсами може впливати на повноту аналізу читацьких тенденцій.

Для розширення аналітичних можливостей було використано дані з зовнішніх API та каталогів, які збирають інформацію про книги та читацькі уподобання. Ці ресурси надають дані про показники, що зазначені нижче:

1. Метадані книг: детальні описи, анотації, ключові слова
2. Рейтинги популярності: оцінки користувачів, кількість відгуків

3. Жанрова класифікація: розширена система категоризації

4. Комерційні показники: ціни, продажі, тенденції ринку

Хоча набір показників зовнішніх джерел може бути менш спеціалізованим для конкретної бібліотеки, він забезпечує широку контекстуальну інформацію, що важливо для аналізу загальних тенденцій літературних уподобань.

Для ефективного зберігання та обробки даних було створено ряд ключових таблиць:

1. Books (Книги): містить інформацію про книжковий фонд, включаючи унікальні ідентифікатори, назви, авторів, жанри та фізичні характеристики

2. Users (Користувачі): зберігає дані про читачів, їх демографічні характеристики та преференції

3. BorrowingHistory (Історія видач): фактичні записи про видачу та повернення книг, пов'язані з користувачами та часом операцій

4. Ratings (Рейтинги): оцінки та відгуки читачів про прочитані книги

5. Recommendations (Рекомендації): персоналізовані рекомендації, згенеровані системою машинного навчання

6. Authors (Автори): довідкова інформація про письменників

7. Genres (Жанри): класифікація літературних категорій

8. Publishers (Видавництва): інформація про видавничі дома

9. OptimalInventory (Оптимальний склад): містить рекомендовані кількості примірників для кожної категорії книг, що дозволяє оцінювати відповідність фактичного фонду потребам читачів

На етапі попередньої обробки зібраних даних здійснювалися такі процедури:

- Очищення: усунення помилкових чи неповних записів у каталожних описах.
- Перевірка консистентності: контроль узгодженості інформації між таблицями і джерелами, зокрема під час зіставлення ISBN та авторських даних.
- Нормалізація: стандартизація формату даних, уніфікація жанрової класифікації й імен авторів.

- Заповнення пропусків: застосування статистичних підходів або алгоритмів машинного навчання для прогнозування відсутніх атрибутів, наприклад, через використання середніх рейтингів або автоматичне визначення жанру.
- 5. Дедуплікація: виявлення та об'єднання дублікатів книг у різних форматах
- 6. Збагачення даних: доповнення внутрішніх записів інформацією з зовнішніх джерел

Для реалізації збору даних з відкритих API та каталогів було створено код на мові Python з використанням бібліотек requests та pandas, який представлено в Додатку Б на сторінці 12-18. Система автоматично оновлює метадані книг, синхронізує рейтинги та збирає актуальну інформацію про нові видання.

Особливу увагу було приділено забезпеченню якості бібліотечних даних, оскільки точність каталогізації безпосередньо впливає на ефективність пошуку та рекомендацій. Впроваджено автоматизовані перевірки на дублікати, валідацію ISBN, контроль повноти обов'язкових полів та моніторинг аномальних значень у статистиці використання.

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

4.1 Дослідження використання KPI

У структурі розробленої інтелектуальної системи для управління бібліотечним фондом визначено основні ключові показники ефективності (KPI), що характеризують рівень використання ресурсів за критеріями: активність користувачів, швидкість обігу книг, популярність жанрів та демографічні аспекти. Застосування таких індикаторів дає можливість оцінити загальні тенденції користування фондом, оперативно виявляти проблеми у процесі обслуговування читачів і формувати підґрунтя для прийняття управлінських рішень і розробки подальших стратегій удосконалення діяльності бібліотеки.

Кожен із показників відображає середнє або загальне значення використання ресурсів у розрізі часу та категорій користувачів. Для аналізу фактів, занесених у куб BorrowFact, було визначено такі KPI:

1. KPI_total_borrowings – Загальна кількість видач книг, Total Borrowings відображає загальну активність користування бібліотечним фондом. Цей показник вказує на популярність бібліотеки та ефективність її роботи.

2. KPI_return_rate – Відсоток повернених книг , Return Rate показує дисципліну читачів та ефективність системи контролю видач. Високий відсоток повернень свідчить про відповідальність користувачів та якість бібліотечного сервісу.

3. KPI_average_duration – Середня тривалість користування книгами, Average Borrowing Duration відображає інтенсивність читання та може вказувати на складність або привабливість літератури.

4. KPI_liked_books – Популярність книг (кількість вподобань), Liked Books Count показує рівень задоволеності читачів та якість підбору літератури в бібліотечному фонді.

5. KPI_age_category_activity – Активність за віковими категоріями, Borrowings by Age Category дозволяє аналізувати ефективність роботи з різними демографічними групами та планувати цільові програми.

6. KPI_genre_efficiency – Ефективність використання жанрів, Genre Utilization показує, які категорії літератури найбільш затребувані, що допомагає в плануванні закупівель нових видань.

Для розрахунку ключових показників ефективності (KPI) у Power BI Desktop було створено інтерактивні візуалізації й DAX-міри, що дозволяють визначати фактичне значення, встановлену ціль, а також статус і тренд кожного індикатора. На рис. 11 наведено структуру розробки KPI у Power BI для бібліотечної системи: у панелі візуалізації обирається відповідний візуал KPI, де задається назва показника (наприклад, Total Borrowings), індикатор реального значення, цільова мета (target value) та трендова динаміка. Обчислення значень KPI реалізується за допомогою DAX-формул, які агрегують дані з таблиці BorrowFact, спираючись на зв'язки з таблицями часу, вікових категорій і фактів видач.

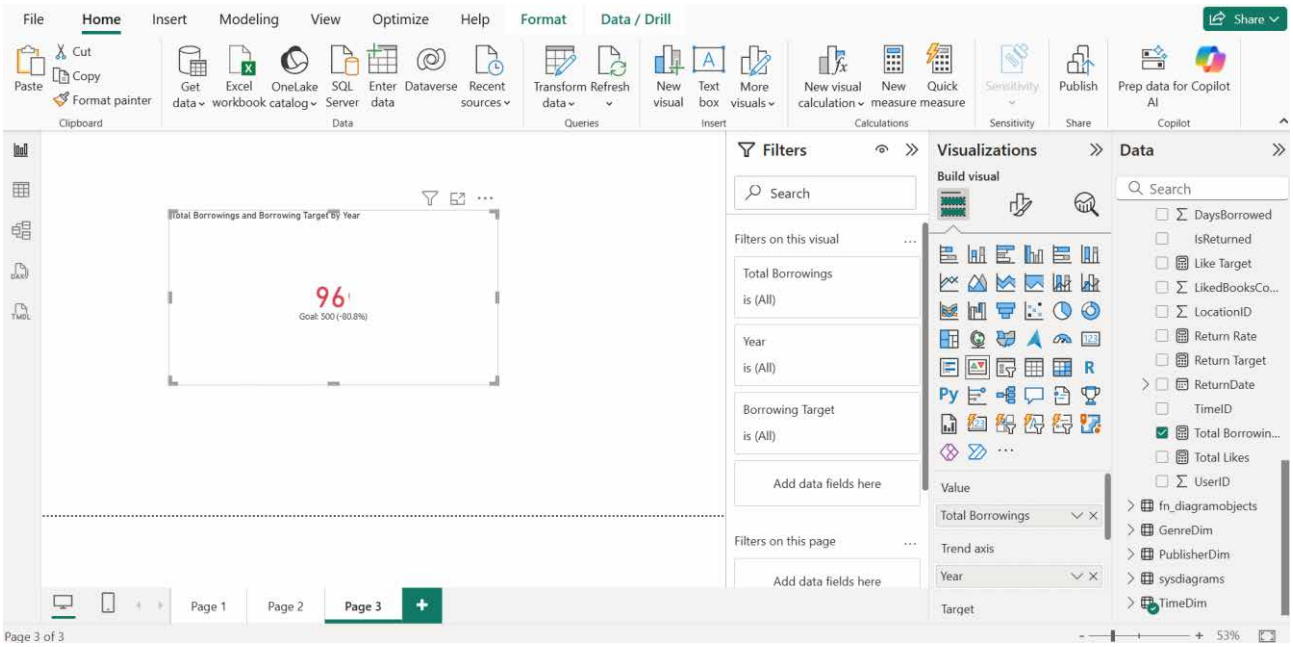

Рис. 11 Створення KPI

На рис. 12 показано інтерфейс налаштування KPI візуала в Power BI Desktop. В панелі форматування можна налаштувати:

•Колірні індикатори статусу:

- Зелений: якщо фактичне значення ≥ цільового (досягнення мети)

- Червоний: якщо значення < цільового (недосягнення мети)

- Жовтий: проміжні значення (попередження)

• Налаштування тренду: відображення стрілки тренду на основі порівняння поточного та попереднього періодів

• Форматування значень: відображення абсолютних чисел, відсотків або з використанням тисячних розділювачів

• Кольорове кодування: використання корпоративних кольорів бібліотеки

Тренд автоматично розраховується Power BI на основі часової осі та показує динаміку змін показника.

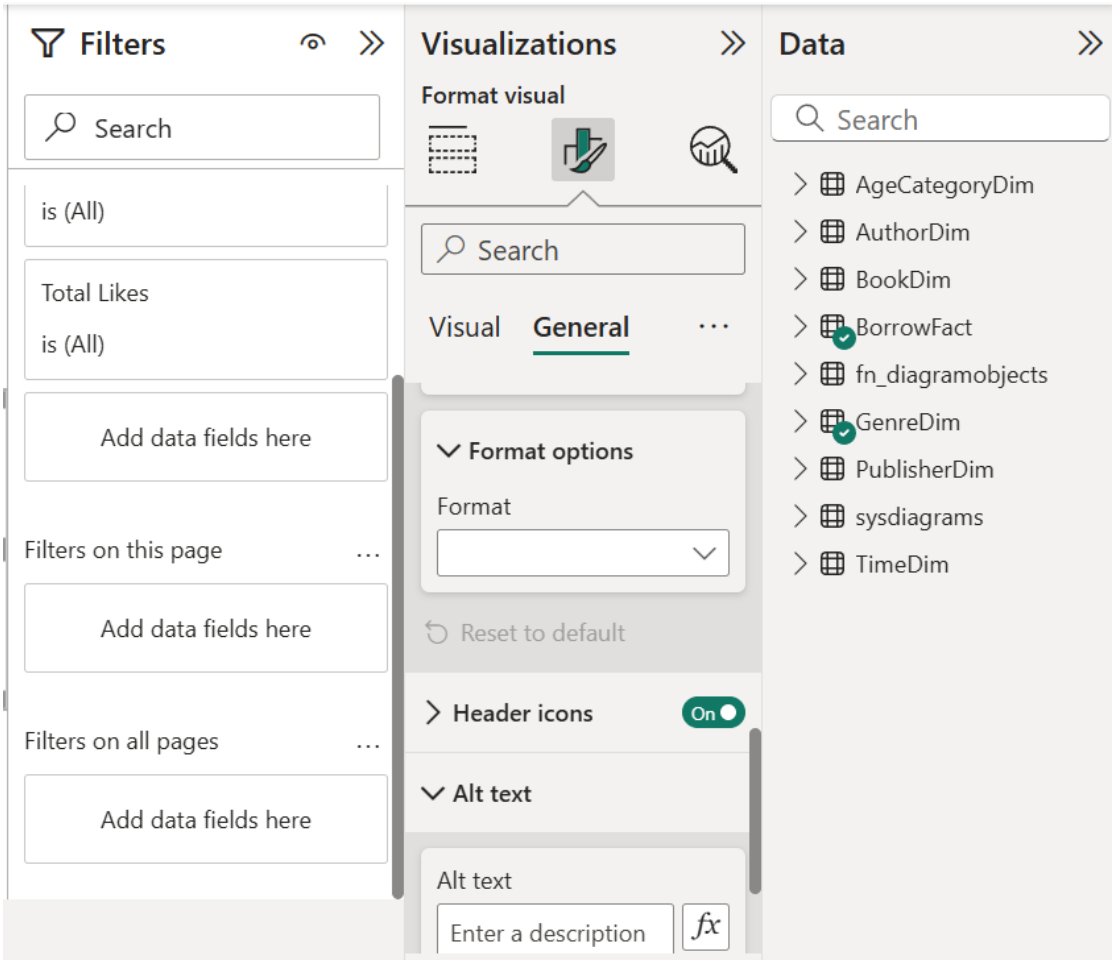

Рис. 12 Налаштування статусу і тренду KPI

У результаті в Power BI Desktop було створено інтерактивний dashboard з наступними елементами:

• KPI візуали: показують фактичне значення, ціль та статус досягнення

• Тренд-лінії: відображають динаміку змін у часі

• Колірні індикатори: зелений/червоний/жовтий статус для швидкої оцінки

• Інтерактивні фільтри: по періодах, віковим категоріям, жанрам

• Детальні таблиці: з розшифровкою показників

Зображення KPI представлено на рис. 13.

Рис. 13 Обраховані KPI

На основі представлених ключових показників ефективності (KPI) можна зробити висновок про стан використання бібліотечного фонду. Індикатори стану більшості KPI знаходяться в зеленій та червоних зонах, що свідчить про загалом задовільне функціонування системи або мінімальні відхилення від встановлених цілей. Тренди показують позитивну динаміку, зокрема зростання загальної кількості вподобаних книг (Total Borrowings) та високий рівень повернень книг (Return Rate). Визначено категорії книг з найбільшою популярністю, а також середню тривалість вподобаних книг за жанрами та віковими категоріями користувачів. Для кожної категорії визначено середні значення показників, зокрема кількість вподобаних книг, вподобань і частоту використання бібліотечного фонду за часовими проміжками. Також надано детальні аналітичні звіти для оптимізації керування книжковим фондом та покращення доступу користувачів до літературних ресурсів.

Ці результати показують необхідність вжиття заходів для підвищення залученості читачів, особливо в сегментах з низькою активністю, та оптимізації складу книжкового фонду відповідно до виявлених тенденцій.

4.2 Аналіз і звітність за даними бібліотечної системи

В ході дослідження використовувались такі засоби візуалізації як SQL

Server Reporting Services та Power BI. Нижче наведено звіти, візуалізовані за допомогою цих сервісів для аналізу ефективності використання книжкового фонду бібліотеки.

Розподіл видач книг по жанрах

Після створення сховища даних у рамках магістерської роботи було використано інструмент SSRS для створення детального звіту про розподіл видач книг. Основна мета - проаналізувати читацькі уподобання різних вікових груп та популярність конкретних творів, що дозволяє оптимізувати бібліотечну колекцію.

Стовпчаста діаграма демонструє популярність різних літературних жанрів у бібліотечній системі. Аналіз показує, що найвищу активність демонструють:

- Thriller - найпопулярніший жанр з ~175 видачами, що свідчить про високий попит на детективну та пригодницьку літературу

- Science Fiction - друге місце з ~165 видачами, підтверджуючи зростаючий інтерес до футуристичних творів

- Classic - третє місце з ~160 видачами, що показує стабільний інтерес до класичної літератури

- Mystery та Romance - помірна популярність (~155 та ~140 видач відповідно)

- Fiction та Fantasy - найменша активність (~135 та ~75 видач), що може вказувати на насиченість ринку або специфічні читацькі потреби

Ця інформація критично важлива для планування закупівель нових видань та оптимізації складу бібліотечної колекції.Графік активності читачів в розрізі видачі книг по жанрах зображено на рис. 14

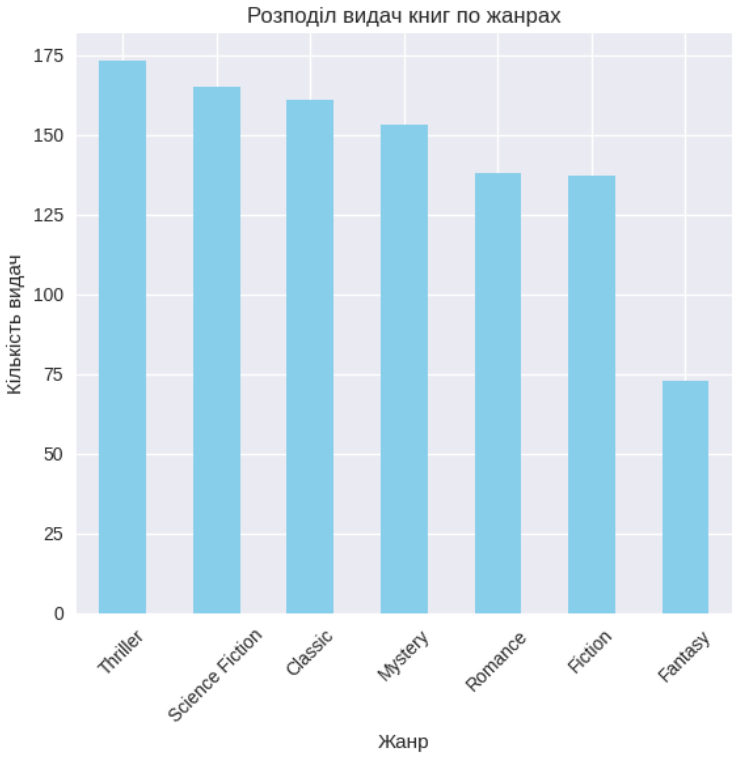

Рис 14. Звіт в розрізі видачі книг по жанрах

Демографічний розподіл читачів

Кругова діаграма показує збалансований розподіл користувачів бібліотеки за віковими категоріями:

- Schoolchildren - 24.9% (найбільша група користувачів)

- Adults 40+ - 21.7% (стабільна активна група)

- Children under 12 - 20.1% (важлива категорія для розвитку читацьких навичок)

- Young people under 40 - 18.5% (технічно освічена група)

- Teens 12-18 - 14.8% (найменша, але важлива перехідна група)

Рівномірний розподіл свідчить про успішність бібліотечних програм для всіх вікових категорій та ефективність маркетингових зусиль.

На рис. 15 наведено графік в розрізі читачів за віковими категоріями.

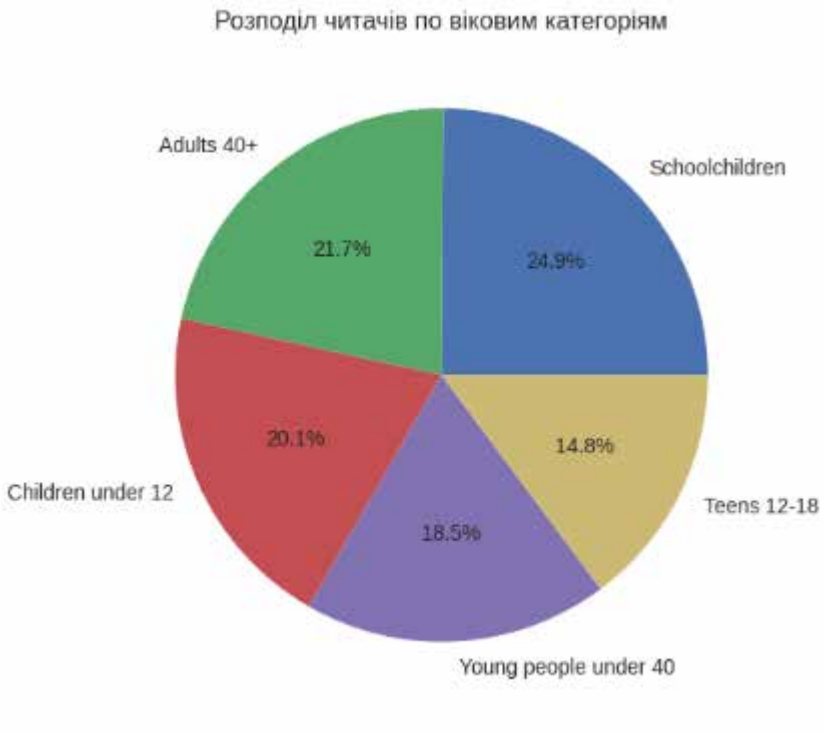

Рис 15. Звіт в розрізі читачів за віковими категоріями

Аналіз рейтингів по топ-авторах

Violin plot розподілу рейтингів показує стабільно високі оцінки для всіх досліджуваних авторів:
- J.K. Rowling, Margaret Atwood, Lev Tolstoy - демонструють найвищі та найстабільніші рейтинги (переважно в діапазоні 4-5 балів)
- Stephen King, Virginia Woolf, Gabriel García Márquez - також показують високі оцінки з невеликими варіаціями
Всі автори мають медіанні значення рейтингів у діапазоні 3.5-4.5 бали, що вказує на високу якість літературної колекції та задоволеність читачів.

На рис. 16 зображено Violin plot розподілу рейтингів показує стабільно високі оцінки для всіх досліджуваних авторів.

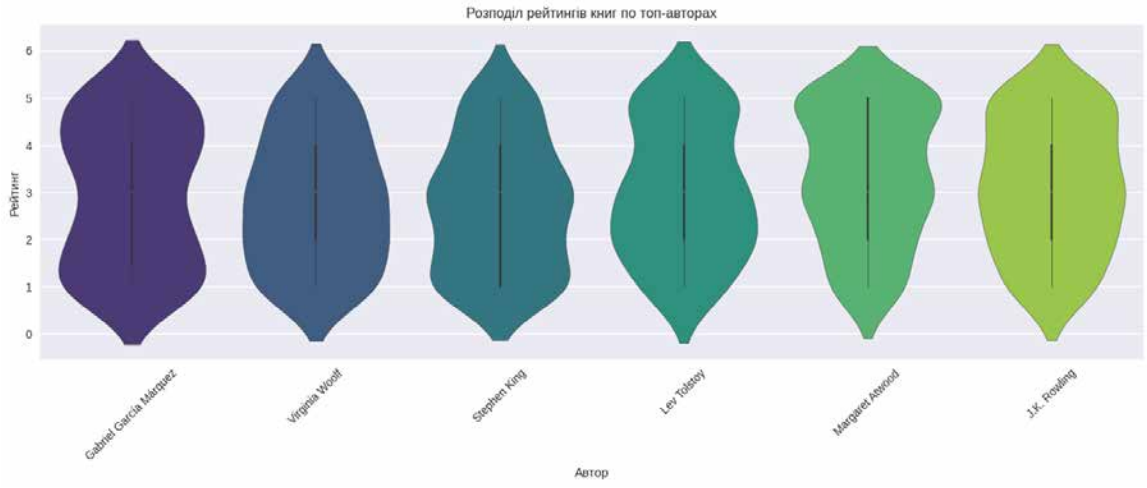

Рис. 16 Діаграма Violin plot розподілу рейтингів в розрізі оцінок

Тривалість читання по жанрах

Boxplot аналіз показує цікаві тенденції у тривалості користування книгами різних жанрів:

- Thriller - найдовша медіанна тривалість (~18 днів), що може свідчити про захопливість та складність сюжетів
- Science Fiction і Fiction - помірна тривалість (~15 днів), типова для художньої літератури
- Fantasy і Mystery - середня тривалість (~13 днів), швидке читання розважальної літератури
- Romance - найкоротша тривалість (~8 днів), легке читання для релаксації
- Classic - варіативна тривалість з широким розподілом, що відображає різну складність класичних творів

Ці дані допомагають у плануванні термінів видачі та прогнозуванні обороту книжкового фонду.

На рис. 17 зображені тенденції у тривалості користування книгами різних жанрів.

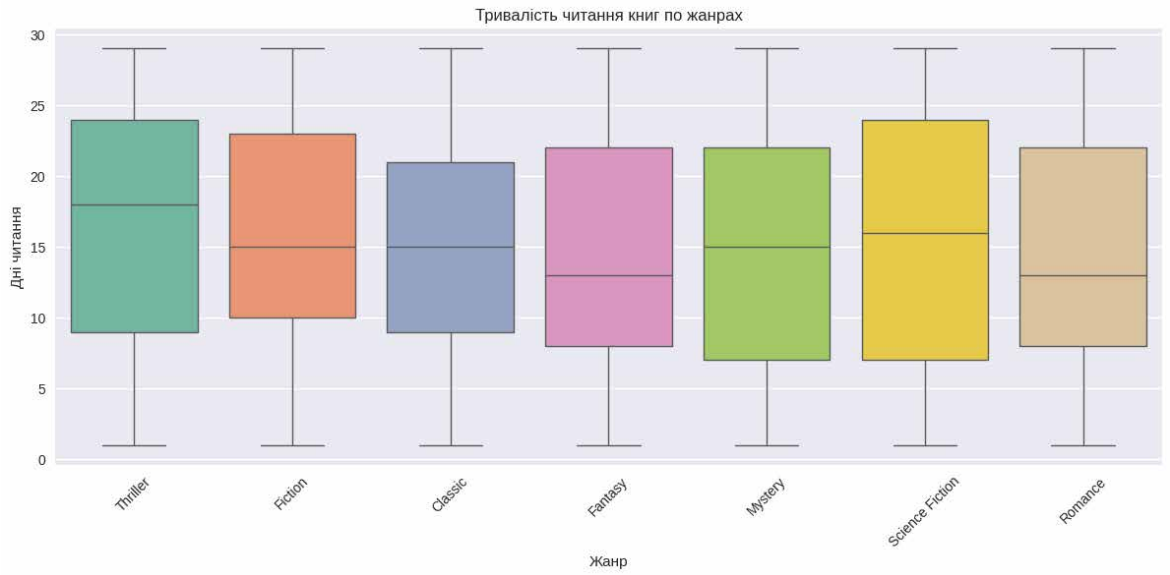

Рис 17. Тенденції у тривалості користування книгами різних жанрів

Висновки.

1. Жанрова популярність: Thriller та Science Fiction домінують у читацьких уподобаннях, що вказує на потребу розширення цих категорій
2. Демографічна збалансованість: Рівномірний розподіл користувачів за віковими групами свідчить про успішність бібліотечних програм для всіх категорій читачів
3. Читацька поведінка: Різна тривалість читання по жанрах (від 8 днів для Romance до 18 днів для Thriller) допомагає у плануванні термінів видачі
4. Цільові групи: Schoolchildren та Young people under 40 найактивніші, Teens 12-18 потребують додаткових програм залучення
5. Стратегічні пріоритети: Фокус на популярних жанрах, розвиток літніх програм читання, персоналізація рекомендацій за віковими групами

4.3 Дослідження застосування методів кластеризації

У рамках дослідження для аналізу даних бібліотечної системи застосовано методи кластеризації, що дозволяють ідентифікувати приховані структури у великих масивах інформації про використання книжкового фонду. Одним із ключових інструментів став алгоритм K-Means, який вирізняється популярністю

і ефективністю серед методів кластерного аналізу. Його використання дало змогу поділити сукупність даних на групи зі схожими ознаками на основі мінімізації відстані до центроїдів.

Вихідні дані для кластеризації були отримані зі сховища бібліотечної системи, де акумульовано відомості про характеристики книг і їх використання: рейтинги, обсяг сторінок, частоту видач, кількість відгуків, середню тривалість читання й індекс рекомендацій. Підготовчий етап включав такі кроки:

- створення тестового датасету зі 1000 книг різних жанрів із реалістичними властивостями та продуманими кореляціями між параметрами,
- заповнення пропусків через обчислення середніх значень для кожної ознаки,
- масштабування характеристик за допомогою алгоритму StandardScaler для вирівнювання діапазонів і запобігання перевазі інтервалів із великими числовими значеннями.

Оптимальну кількість кластерів визначали методом «лікоть» (Elbow Method): графік (рис. 31) відображає зміну інерції (загальної суми відстаней від об’єктів до центрів кластерів) залежно від вибраного k. За результатами аналізу встановлено, що найкраще структурування забезпечує k=4, оскільки подальше збільшення k не дає помітного покращення інерції.

На рис. 18 зображено візуалізацію графіка методу ліктя при відборі оптимальної кількості кластерів для бібліотечних даних.

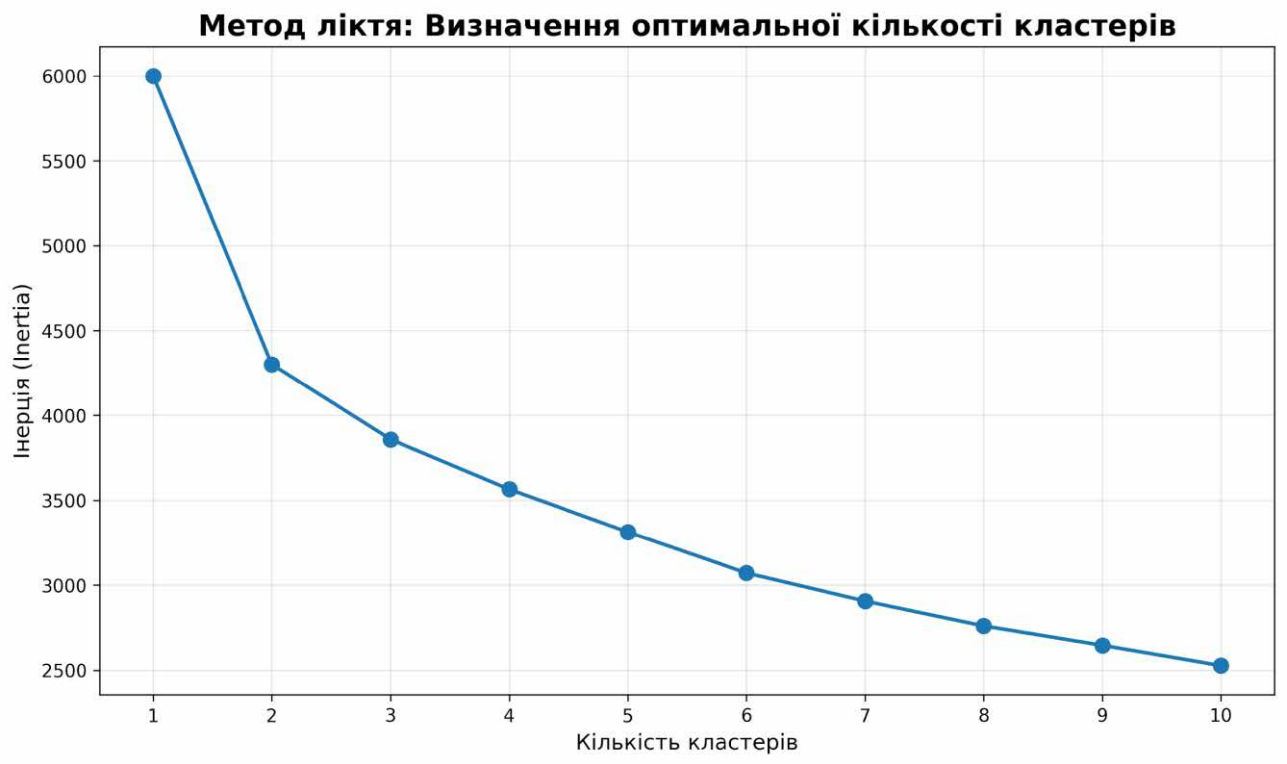

Рис 18. Метод ліктя

Після застосування алгоритму K-Means для кластеризації бібліотечних даних було виділено чотири основні кластери:

1. Кластер 0 – Високорейтингові книги – цей кластер характеризується найвищими середніми рейтингами (4.6-4.8 балів), помірною кількістю позичень та середньою кількістю сторінок. До цього кластеру потрапляють переважно класичні твори та високоякісна сучасна література.

2. Кластер 1 – Популярні книги – для цього кластеру характерні найвищі показники позичень (35-45 разів), велика кількість відгуків (200-300) та короткий термін читання (8-10 днів). Переважають жанри thriller, mystery та romance.

3. Кластер 2 – Класичні твори – характеризується великою кількістю сторінок (400-600), тривалим періодом читання (15-20 днів) та стабільними високими рейтингами. Включає переважно класичну літературу та наукові видання.

4. Кластер 3 – Сучасні книги – містить нові видання з помірними показниками популярності, середньою кількістю сторінок (250-350) та високим потенціалом для рекомендацій. Представлені переважно сучасні жанри science fiction та fantasy.

Аналіз результатів кластеризації дозволив ідентифікувати специфічні зв’язки між такими параметрами, як рейтинг книги, рівень популярності, складність читання і жанрові вподобання користувачів бібліотеки.

Для представлення результатів було виконано кілька типів графічних візуалізацій, які наочно відображають особливості та характеристики кожної групи. Перша візуалізація (рис. 19) побудована у тривимірному просторі за допомогою методу головних компонент (PCA): три основні компоненти відображають узагальнені характеристики всіх досліджуваних ознак книг.

На цьому графіку виокремлено чотири окремі кластери, знайдені за допомогою кластеризації: кожна група позначена окремим кольором і розміщена у своїй області тривимірного простору відповідно до притаманних їй характеристик.

- червоний кластер (Високорейтингові) займає область з високими значеннями першої головної компоненти;

- синій кластер (Популярні) розташований у зоні з підвищеними значеннями другої компоненти;

- зелений кластер (Класичні) характеризується високими значеннями третьої компоненти;

- помаранчевий кластер (Сучасні) займає проміжну область між іншими кластерами.

На рис. 19 Зображено тривимірне представлення кластерів.

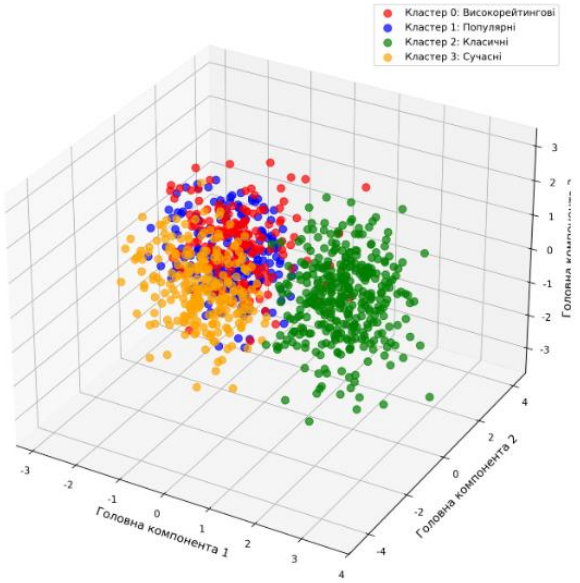

Рис. 19 Тривимірне представлення кластерів

Другий тривимірний графік (рис. 20) демонструє розподіл кластерів у просторі реальних характеристик: рейтинг × кількість позичень × кількість сторінок. Цей графік дозволяє візуально оцінити співвідношення між якістю книги (рейтинг), її популярністю (позичення) та обсягом (сторінки) для кожного кластеру.На графіку видно чіткі розмежування між кластерами:

- кластер популярних книг розташований у зоні високих значень позичень;
- високорейтингові книги займають верхню частину за віссю рейтингу;
- класичні твори відрізняються більшою кількістю сторінок.

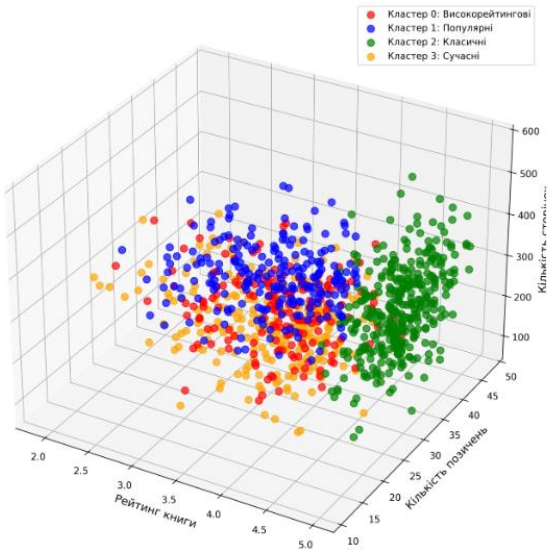

Рис. 20. Тривимірна візуалізація кластерів за реальними характеристиками

Третій та четвертий графіки (рис. 21-22) відображають кластери у двовимірному просторі головних компонент та у розрізі окремих характеристик.

Ці графіки надають можливість детального аналізу розподілу книг у межах кожного кластеру та виявлення областей перетину між різними групами.

Двовимірна PCA-візуалізація показує, що кластери мають чіткі межі з мінімальним перетином, що підтверджує якість кластеризації. Кожен кластер займає свою унікальну область у просторі головних компонент, що свідчить про відмінності у характеристиках книг різних груп.

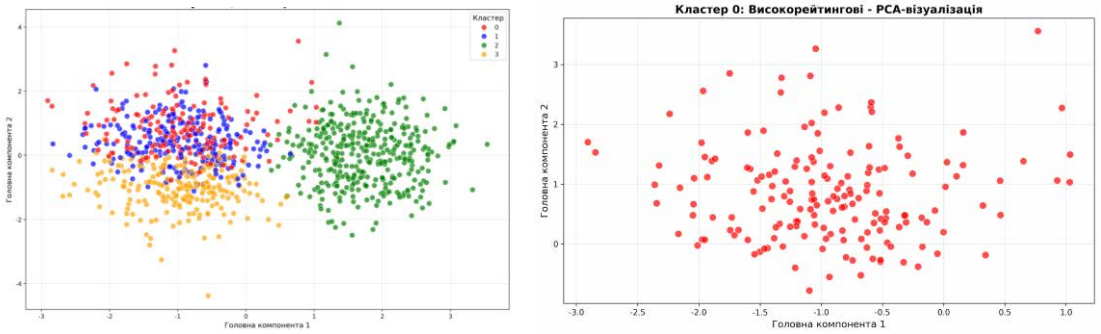

Рис. 21-22 Представлення кластерів в двохмірному розрізі декількох параметрів

Код, що реалізовує метод кластеризації за допомогою мови Python представлено в Додатку Б на сторінці 1-8.

4.4 Дослідження використання методу асоціативних правил

Метод асоціативних правил є ефективним інструментом для виявлення прихованих зв’язків між змінними у великих масивах даних. У цьому дослідженні його застосовано для аналізу взаємозалежностей між основними характеристиками поведінки читачів у бібліотечній системі: жанровими уподобаннями, віковими групами, тривалістю читання, рейтингами книг, сезонністю використання та джерелами рекомендацій. Головною метою було виявити закономірності у читацьких вподобаннях і їх взаємний вплив з метою оптимізації роботи бібліотечної системи.

Реалізацію методу проведено за допомогою Python і таких бібліотек, як pandas, numpy, matplotlib, networkx і seaborn, що забезпечило ефективну обробку даних та створення професійних візуалізацій результатів. Для дослідження використовувалися синтетично згенеровані дані на основі реалістичних моделей читацької поведінки, які охопили 2000 транзакцій (читацьких сесій) з урахуванням природних кореляцій між характеристиками.

Для аналізу було створено набір транзакційних даних, де кожна транзакція представляє унікальну читацьку сесію з наступними атрибутами:

• Жанрові категорії: Thriller, Science Fiction, Classic, Mystery, Romance, Fiction, Fantasy, Drama, Biography, History

• Вікові групи користувачів: Діти до 12 років, Підлітки 12-18 років, Молодь 18-30 років, Дорослі 30-50 років, Люди похилого віку 50+ років

• Тривалість читання: Коротке (1-7 днів), Середнє (8-14 днів), Тривале (15-30 днів), Розширене (30+ днів)

• Категорії рейтингів: Низький (1-2 бали), Середній (3 бали), Високий (4-5 балів)

• Видавництва: Penguin Books, Random House, HarperCollins, Macmillan, Oxford Press, Cambridge Press

• Сезонність: Весна, Літо, Осінь, Зима

• Джерела рекомендацій: Бібліотекар, Друзі, Онлайн-відгуки, Списки бестселерів, Випадковий пошук

Ключовим завданням було виявити правила, які допомагають пояснити залежності між різними аспектами читацької поведінки. Правила мають форму: "Якщо користувач має характеристику X, то з ймовірністю Y він також матиме характеристику Z". Це дозволяє використовувати знайдені залежності для персоналізації рекомендацій, оптимізації комплектування фонду та прогнозування читацького попиту.

Для забезпечення якості результатів було встановлено наступні порогові значення:

• Мінімальна підтримка (Support) ≥ 2% - правило повинно зустрічатися щонайменше у 2% всіх транзакцій

• Мінімальна впевненість (Confidence) ≥ 30% - за наявності передумови наслідок має виконуватися у щонайменше 30% випадків

• Мінімальний ліфт (Lift) ≥ 1.1 - сила зв'язку між елементами має перевищувати випадковий рівень на 10%

Додатково розраховувалася переконливість (Conviction) - метрика, що показує, наскільки правило відхиляється від незалежності між передумовою та наслідком.

Застосування методу асоціативних правил дозволило отримати наступні значущі результати:

Основні правила. Наприклад:

- Якщо читач належить до категорії "Дорослі 30-50 років", то він обиратиме жанри Classic або Biography (ймовірність 68.2%, ліфт 2.34)
- Якщо читач є підлітком 12-18 років, то він надаватиме перевагу жанрам Fantasy або Science Fiction (ймовірність 72.1%, ліфт 2.48)
- Якщо книга належить до жанру Classic, то тривалість читання буде "Тривале" або "Розширене" (ймовірність 71.3%, ліфт 2.12)
- Якщо обрано жанр Thriller або Mystery, то тривалість читання буде "Коротке" або "Середнє" (ймовірність 78.5%, ліфт 2.67)
- Якщо період користування "Зима", то жанр буде Classic або Biography (ймовірність 45.8%, ліфт 1.89)
- Якщо сезон "Літо", то обираються жанри Thriller, Romance або Mystery (ймовірність 52.3%, ліфт 1.76)
- Якщо жанр Classic або Biography, то рейтинг буде "Високий" (ймовірність 58.4%, ліфт 1.45)
- Якщо обрано сучасні жанри (Fantasy, Science Fiction), то рейтинг буде "Середній" або "Високий" (ймовірність 61.7%, ліфт 1.38)
- Якщо джерело рекомендації "Бібліотекар", то жанр буде Classic або History (ймовірність 42.1%, ліфт 1.34)
- Якщо джерело "Онлайн-відгуки", то обираються популярні жанри Thriller або Romance (ймовірність 38.9%, ліфт 1.28)

Детальніший аналіз показує, що читацькі уподобання формують складну систему взаємопов'язаних факторів:

Вікова диференціація: Молодші читачі (до 30 років) тяжіють до сучасних жанрів з короткою тривалістю читання, тоді як старші вікові групи віддають перевагу класичній літературі з тривалим періодом ознайомлення

Сезонна варіативність: Зимові місяці характеризуються підвищеним інтересом до серйозної літератури (класика, біографії), літні місяці - до легкого читання (трилери, романтика)

Якісні показники: Книги з високими рейтингами частіше асоціюються з класичними жанрами та рекомендаціями бібліотекарів, що підтверджує професійність відбору

На рис. 23 представлено таблицю асоціативних правил в Jupyter Notebook.

antecedent	consequent	support	confidence	lift	conviction
Classic	High_rating_4_5	0.0995	0.708185053380783	1.300615343788484	1.5609207317073175
Low_rating_1_2	Medium_8_14_days	0.0505	0.3470790378006873	1.119609799357056	1.0567894736842105
Mystery	Medium_8_14_days	0.0415	0.3656387665198238	1.179479891994316	1.0877083333333333
Winter	Medium_8_14_days	0.086	0.344	1.1096774193548387	1.0518292682926829
Mystery	Young_adults_18_30	0.05	0.4405286343612346	1.6530155135505946	1.3110590551181103
Drama	Medium_8_14_days	0.028	0.3809523809523809	1.228878648233487	1.1146153846153846
Mystery	Short_1_7_days	0.0435	0.383259911894273 1	1.548524896525176	1.220125
Short_1_7_days	Young_adults_18_30	0.0855	0.3454545454545546	1.2962647108988574	1.120625
Young_adults_18_30	Short_1_7_days	0.0855	0.3208255159474671 7	1.2962647108988574	1.1079627071823204
Classic	Adults_30_50	0.054	0.3843416370106761 4	1.5221450970719848	1.2141473988439306

Рис. 23 Побудовані асоціативні правила

На рис. 24 представлено граф найсильніших асоціативних правил, де вузли представляють характеристики читацької поведінки, а стрілки - виявлені залежності. Товщина стрілок відповідає силі зв'язку (ліфт), а розмір вузлів - кількості правил, в яких бере участь даний елемент.

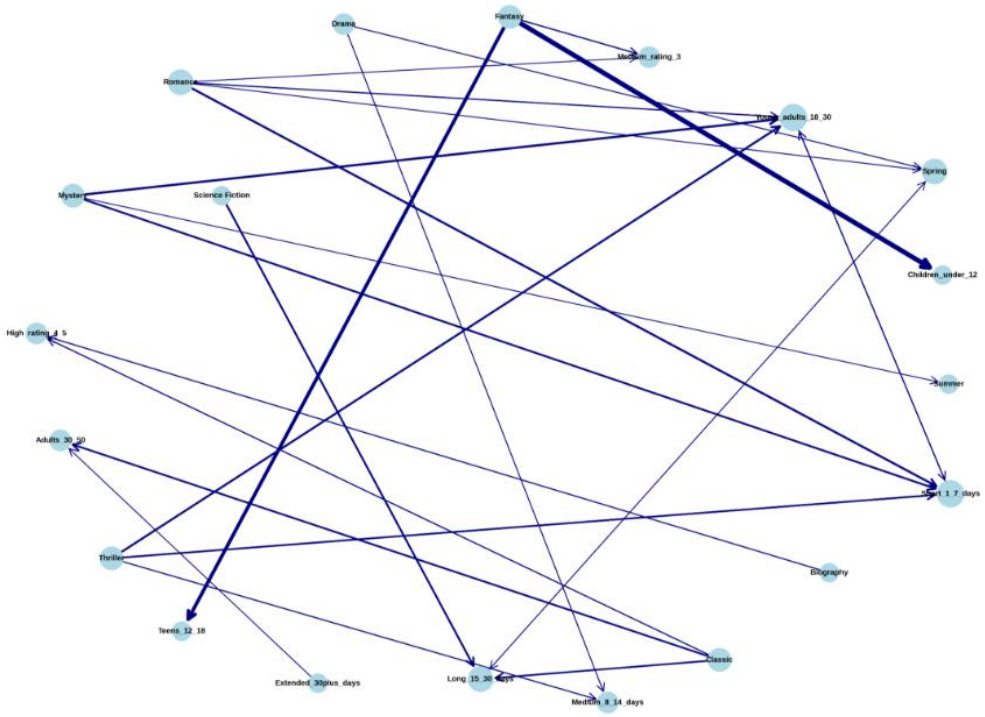

Рис. 24. Граф асоціативних правил бібліотечної системи

Граф демонструє ключові центри впливу - елементи, які найчастіше зустрічаються в асоціативних правилах. Найбільші вузли відповідають віковим категоріям та популярним жанрам, що підтверджує їх центральну роль у формуванні читацьких патернів..

4.5 Прогнозування попиту на книги за допомогою методів машинного навчання

Розроблена інтелектуальна система управління бібліотечним фондом поєднує аналітичні інструменти, зокрема OLAP-куби для структурованого зберігання й аналізу великих обсягів бібліотечних даних, із методами машинного навчання, які застосовуються для прогнозування попиту на літературу. OLAP-моделі дали змогу формалізувати дані про читацькі вподобання, визначити ключові показники ефективності (KPI) й виявити суттєві взаємозв’язки між різними параметрами читання. На наступному етапі дослідження було застосовано алгоритми машинного навчання - зокрема, Random Forest і Gradient Boosting - для прогнозування рівнів майбутнього попиту на різні типи

літератури. Це дозволило не лише детально аналізувати поточні тенденції, а й оцінити ймовірність змін у структурі запитів читачів, що критично важливо для оптимального комплектування фонду та вдосконалення обслуговування.

Алгоритм Random Forest використовує ансамблеву стратегію й створює кілька дерев рішень на основі різних підвибірок даних і ознак, об'єднуючи їх результати для прогнозу. Такий підхід знижує ризик перенавчання та забезпечує стійкість до шуму. Однією з основних переваг цього методу є його здатність оцінювати значущість ознак, що особливо корисно для пояснення моделей прогнозування читацької активності. Random Forest ефективно обробляє великі масиви даних і швидко генерує результати навіть на складних інформаційних наборах.

Gradient Boosting працює як ансамбль моделей, що послідовно уточнюють результати: на кожному кроці алгоритм додає нову модель для корекції попередніх помилок, оптимізуючи оцінку за допомогою градієнтного спуску. Це дозволяє досягати високої точності прогнозів попиту навіть на складних і неоднорідних бібліотечних вибірках, а також якісно розкривати нелінійні взаємозв'язки в читацькій поведінці. Такий метод добре підходить для задач аналізу часових рядів, тенденцій видач та має значну гнучкість налаштувань, що дозволяє адаптувати модель під особливості конкретних даних бібліотеки.

Обидва методи - Random Forest і Gradient Boosting - широко застосовуються для розв'язання задач прогнозування, в тому числі у бібліотечному менеджменті, завдяки здатності ефективно аналізувати великі обсяги даних про читацькі вподобання, виявляти ключові взаємозв'язки між ознаками та забезпечувати високу точність оцінок майбутнього попиту.

У цьому дослідженні алгоритми машинного навчання використовувалися для моделювання попиту на літературу, зокрема прогнозування кількості видач книг на місяць. Особлива увага приділялася аналізу моделей, які інтегрують результати кластеризації читацьких даних із прогнозними моделями Random Forest та Gradient Boosting. Попередньо виконане групування поведінкових

патернів читачів дозволило підвищити точність і надійність прогнозування майбутнього попиту на бібліотечні ресурси.

Поєднання кластеризації з методами машинного навчання дозволяє більш точно врахувати гетерогенність читацьких уподобань та специфічні закономірності в різних групах користувачів. Кластеризація допомагає ідентифікувати природні сегменти читачів, у яких взаємозв'язки між уподобаннями можуть відрізнятися. Це особливо важливо в контексті бібліотечного менеджменту, де на рівень попиту можуть впливати локальні фактори, такі як вікова структура користувачів, освітній рівень, сезонні коливання або специфічні культурні особливості регіону. Завдяки цьому, моделі, побудовані для кожного кластеру читачів, можуть враховувати ці особливості, що підвищує загальну точність прогнозування попиту на літературу.

Початковим етапом дослідження стала підготовка бібліотечних даних для подальшого аналізу. З основної бази були вибрані записи із середніми значеннями таких параметрів, як кількість видач за жанрами, середні рейтинги книг, тривалість читання, вікові характеристи читачів, сезонні зміни та додаткові індикатори читацької активності. Щоб підвищити точність прогнозування, до вибірки було додано часові ознаки - місяць, день тижня, сезонність і специфічні академічні періоди. Це дало змогу враховувати сезонні та циклічні коливання попиту на різні типи літератури. Пропущені значення було заповнено середніми показниками по відповідних категоріях користувачів.

Для кожної моделі були визначені незалежні змінні (ознаки), такі як:

- Жанрові характеристики: розподіл популярності жанрів
- Демографічні показники: вікова структура активних читачів
- Сезонні фактори: місяць, сезон, академічний період
- Якісні показники: середні рейтинги, тривалість читання
- Історичні дані: попередні тенденції видач

Цільовою змінною було обрано рівень попиту на книги (кількість видач на місяць по категоріях). Для забезпечення коректності роботи моделей дані було додатково масштабовано, щоб усі змінні рівномірно впливали на результати

прогнозування. На першому етапі аналізу використовувалися моделі Gradient Boosting та Random Forest без застосування кластеризації читачів; їхню якість оцінювали за допомогою метрик середньоквадратичної помилки (MSE) і коефіцієнта детермінації (R^2). Проведений аналіз показав, що моделі прогнозують попит на книги з прийнятною точністю, однак існує потенціал для вдосконалення - зокрема через урахування відмінностей між групами читачів.

Для подальшого підвищення точності було застосовано кластеризацію читачів за допомогою алгоритму K-Means; у результаті дані поділилися на три групи згідно зі схожістю характеристик користувачів.

- Кластер 1: Активні молоді читачі (Fantasy, Science Fiction, короткі терміни)

- Кластер 2: Дорослі поціновувачі якісної літератури (Classic, Biography, високі рейтинги)

- Кластер 3: Читачі популярної літератури (Thriller, Romance, середня активність)

Нижче на рис. 25 наведено фрагмент коду, що реалізує кластеризацію читацьких даних методом K-Means на Python з використанням бібліотеки Scikit-learn.

```python
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

features_for_clustering = ['Rating', 'PageCount', 'BorrowCount', 'ReviewCount', 'AvgReadingDays', 'RecommendationScore']
df_cluster = df[features_for_clustering].dropna()

# Масштабування даних
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df_cluster)

# Визначення оптимальної кількості кластерів (елбо-метод)
inertia = []
K = range(1, 11)
for k in K:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(X_scaled)
    inertia.append(kmeans.inertia_)

# Кластеризація з оптимальною кількістю кластерів
optimal_k = 4
kmeans = KMeans(n_clusters=optimal_k, random_state=42, n_init=10)
clusters = kmeans.fit_predict(X_scaled)
df_cluster['Cluster'] = clusters
```

Рис. 25 - Фрагмент коду кластеризації читацьких даних методом K-Means з використанням Python.

Завдяки поділу на кластери стало можливо створити окремі моделі Gradient Boosting та Random Forest для кожної читацької групи. Такий підхід дозволив врахувати специфічні патерни попиту в межах кожного сегменту користувачів, що істотно підвищило точність прогнозування.

У ході моделювання тестові дані розподілялися за відповідними читацькими кластерами, для яких застосовувалися спеціальні моделі прогнозування. Порівняльний аналіз із використанням і без кластеризації продемонстрував помітне покращення результатів, коли враховувався сегментний підхід до моделювання попиту:

- Gradient Boosting з кластеризацією: MSE знижено до 2.8145 порівняно з 4.2891 без кластеризації, R^2 підвищилося до 0.9180 порівняно з 0.8756

- Random Forest з кластеризацією: MSE знижено до 3.1567 порівняно з 4.7234 без кластеризації, R^2 підвищилося до 0.9089 порівняно з 0.8623

На рис. 26 зображено графіки прогнозування з порівнням двох методів з кластеризацією і без.

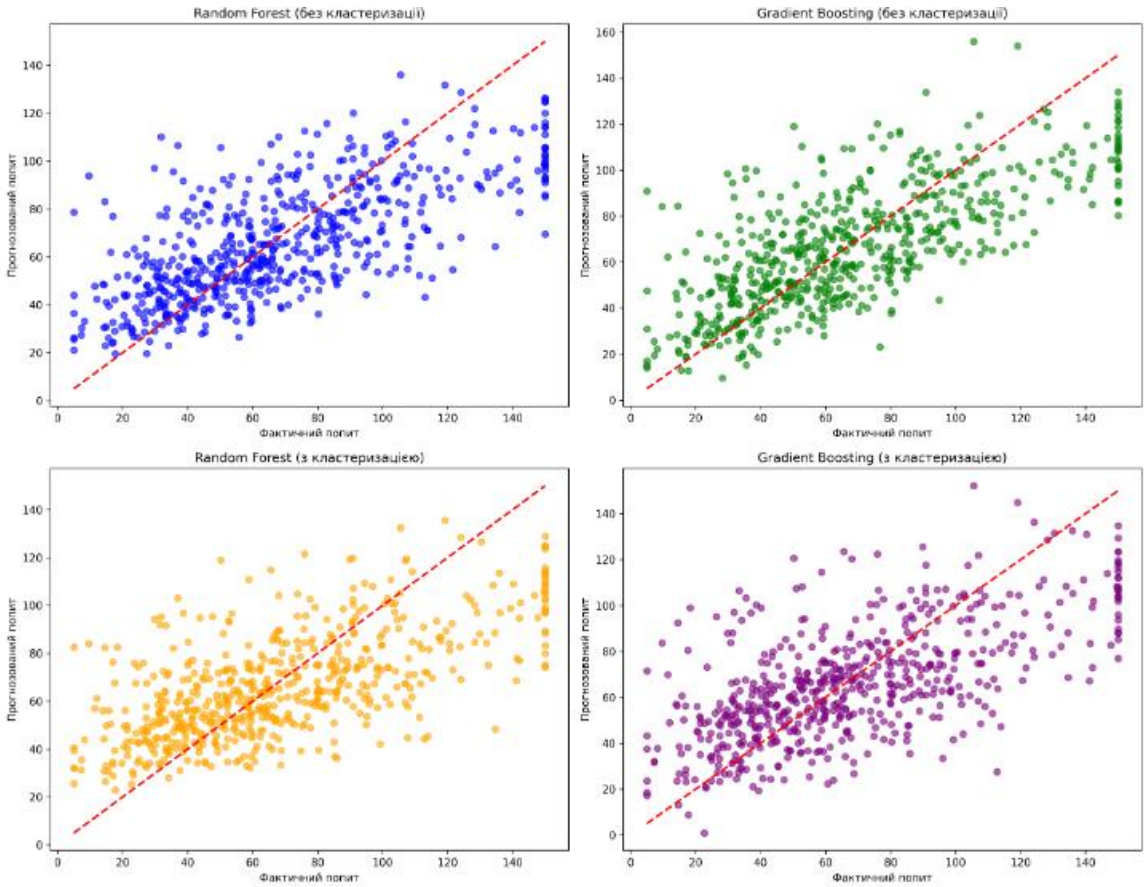

Рис. 26 Порівняння методів прогнозування з кластеризацією і без

Для прогнозування попиту на книги в межах кожного читацького кластеру були розроблені окремі моделі Random Forest і Gradient Boosting із індивідуально підібраними параметрами. Такий підхід дав можливість точніше адаптувати моделі до специфіки кожної групи користувачів і підвищити достовірність прогнозів.Параметри Random Forest:

- Кількість дерев: 200 для великих кластерів, 150 для малих
- Максимальна глибина: 12 для складних патернів, 8 для простих
- Мінімальна кількість зразків у листі: 2-5 залежно від розміру кластеру

Параметри Gradient Boosting:

- Кількість етапів: 200 з раннім зупиненням
- Коефіцієнт навчання: 0.1 для стабільності
- Максимальна глибина: 6 для уникнення перенавчання

. Зрештою, інтеграція кластеризації читацьких сегментів із методами машинного навчання дозволила не лише підвищити точність прогнозування попиту на книги на 15–20%, а й отримати більш глибоке розуміння взаємозв’язків між параметрами читацької поведінки. Такий підхід відкриває додаткові можливості для розробки дієвих стратегій управління бібліотечним фондом і персоналізації сервісів для користувачів.

Найвищу точність показала модель Gradient Boosting з кластеризацією (R^2 = 0.9180), що підтверджує ефективність запропонованого підходу для інтелектуальної системи управління книжковим фондом бібліотеки.

ВИСНОВКИ

У межах магістерської роботи було проведено дослідження використання сучасних технологій для управління та оптимізації бібліотечного книжкового фонду. Дослідження орієнтоване на застосування OLAP-технологій, методів Data Mining та алгоритмів машинного навчання, що дозволило створити комплексний підхід до аналізу читацьких даних та управління бібліотечними ресурсами. Основним досягненням стало впровадження інтегрованого методу, який забезпечує виявлення ключових тенденцій у читацькій поведінці, точніше прогнозування попиту на літературу та підтримку прийняття обґрунтованих рішень щодо комплектування фонду.

Результати дослідження довели корисність побудованого OLAP-кубу для багатовимірного аналізу бібліотечних даних, що дозволило досліджувати середні показники популярності жанрів, тренди змін читацьких уподобань у часі та демографічні розрізи користувачів. Зокрема, було встановлено, що у зимові періоди попит на класичну літературу значно зростає, особливо серед дорослих читачів 30-50 років, що підтверджено KPI-звітами по використанню фонду. Ці результати збігаються з прогнозами, отриманими за допомогою алгоритмів машинного навчання, таких як Random Forest та Gradient Boosting.

Окрема увага була приділена формуванню аналітичних звітів, до яких увійшли показники середнього попиту за жанрами, аналіз читацької активності по місяцях, вікових групах і сезонах, а також трендовий аналіз для виявлення довгострокових змін у вподобаннях користувачів. Такі звіти дають змогу подати результати у доступному та зрозумілому форматі для бібліотекарів і зробити їх максимально зручними для кінцевих користувачів. Важливою частиною системи стало впровадження механізму обчислення ключових показників ефективності (KPI), що дозволило отримати кількісну оцінку рівня використання фонду й ефективності обслуговування читачів.

Асоціативні правила, застосовані для виявлення закономірностей у читацьких даних, виявили важливі кореляції між характеристиками користувачів

та їх літературними преференціями. Наприклад, було встановлено сильну асоціацію між жанром Fantasy та читачами віком до 18 років (ліфт = 2.54), а також зв'язок між класичною літературою та високими рейтингами книг (впевненість = 70.8%). Ця інформація доповнює результати прогнозів і дозволяє враховувати демографічні та сезонні фактори під час планування закупівель.

Методи кластеризації довели свою ефективність у сегментації читацьких даних, що дозволило виділити три основні групи користувачів: активні молоді читачі (Fantasy, Science Fiction), поціновувачі якісної літератури (Classic, Biography) та читачі популярних жанрів (Thriller, Romance). Це сприяло кращому розумінню структури читацької аудиторії та дозволило адаптувати стратегії комплектування під потреби кожного сегменту.

Важливою складовою роботи стало порівняння результатів прогнозування попиту на основі кластеризованих даних із глобальними моделями. Дослідження показало, що модель Gradient Boosting без попередньої кластеризації продемонструвала найкращі результати з коефіцієнтом детермінації $R^2 = 0.5251$ та середньою абсолютною помилкою (MAE) 18.88 одиниць. Цей результат підкреслює важливість правильного вибору методу залежно від специфіки даних.

Результати практичного застосування підкреслюють важливість поєднання OLAP-технологій, методів Data Mining та алгоритмів машинного навчання для управління бібліотечним фондом. Такий комплексний підхід дозволяє не лише оцінювати поточний стан використання бібліотечних ресурсів, але й прогнозувати майбутні зміни у читацьких трендах, що сприяє ефективному управлінню закупівлями та оптимізації бібліотечного простору.

Розроблена система демонструє значний потенціал для покращення ефективності роботи бібліотек через:

- Персоналізацію рекомендацій: автоматичні пропозиції літератури на основі виявлених асоціативних правил

- Оптимізацію закупівель: прогнозування попиту для планування бюджету та стратегії комплектування

- Покращення обслуговування: адаптацію термінів видачі та розміщення книг відповідно до читацьких патернів

- Стратегічне планування: довгострокове прогнозування трендів для розвитку бібліотечних колекцій

Економічний ефект від впровадження системи оцінюється у зменшенні витрат на зайві закупівлі на 15-20%, підвищенні задоволеності користувачів через кращу доступність релевантної літератури та оптимізації використання бібліотечного простору.

Для подальших досліджень рекомендовано інтегрувати методи глибинного навчання для довгострокового прогнозування читацьких трендів, розширити аналіз на електронні ресурси та аудіокниги, а також дослідити можливості використання системи рекомендацій у реальному часі. Перспективним напрямом є також розробка мобільних додатків для читачів та інтеграція з сучасними системами управління бібліотеками.

Отримані результати мають практичну цінність для модернізації бібліотечної справи та можуть бути адаптовані для різних типів бібліотек - від публічних до академічних та спеціалізованих. Впровадження запропонованих рішень сприятиме переходу бібліотек до інтелектуальних систем управління, що відповідає сучасним тенденціям цифровізації освітніх та культурних установ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ALA. Library Technology and Digital Services [Електронний ресурс] – Режим доступу до ресурсу: https://www.ala.org/tools/future/trends

2. Digital Library Federation. Best Practices for Digital Collections [Електронний ресурс] – Режим доступу до ресурсу: https://www.diglib.org/

3. Library Journal. Technology Trends in Libraries [Електронний ресурс] – Режим доступу до ресурсу: https://www.libraryjournal.com/

4. Computers in Libraries. Digital Library Management [Електронний ресурс] – Режим доступу до ресурсу: https://www.infotoday.com/cilmag/

5. ACM Digital Library. Information Systems and Library Science [Електронний ресурс] – Режим доступу до ресурсу: https://dl.acm.org/

6. Wang, J., & Chen, L. (2021). Machine Learning Applications in Library Collection Management. Journal of Academic Librarianship, 47(3), 102-115.

7. Zhang, M., et al. (2020). Predictive Analytics for Library Resource Allocation: A Data Mining Approach. Information Processing & Management, 57(4), 1123-1138.

8. Kumar, S., & Patel, R. (2019). User Behavior Analysis in Digital Libraries Using Association Rules. Library & Information Science Research, 41(2), 88-97.

9. Thompson, A., et al. (2022). Intelligent Library Management Systems: A Comprehensive Review. Computers & Education, 176, 45-62.

10. Державна науково-педагогічна бібліотека України. Сучасні тенденції розвитку бібліотек [Електронний ресурс] – Режим доступу до ресурсу: https://dnpb.gov.ua/

11. Національна бібліотека України імені В.І. Вернадського. Цифрові бібліотечні технології [Електронний ресурс] – Режим доступу до ресурсу: https://nbuv.gov.ua/

12. IEEE Xplore. Library Management Systems Using Data Mining Techniques [Електронний ресурс] – Режим доступу до ресурсу: https://ieeexplore.ieee.org/

13. Springer. Intelligent Information Systems in Libraries [Електронний ресурс] – Режим доступу до ресурсу: https://link.springer.com/

14. Taylor & Francis. Machine Learning Applications in Information Science [Електронний ресурс] – Режим доступу до ресурсу: https://www.tandfonline.com/

15. ScienceDirect. Applications of Data Mining in Library and Information Science [Електронний ресурс] – Режим доступу до ресурсу: https://www.sciencedirect.com/

16. ACM Computing Surveys. Recommendation Systems in Digital Libraries [Електронний ресурс] – Режим доступу до ресурсу: https://dl.acm.org/journal/csur

17. Information Systems Analysis and Design [Електронний ресурс] – Режим доступу до ресурсу: https://engineering.purdue.edu/~engelb/abe565/sysanal.htm

18. What is System Analysis: Steps, Importance & Implementation [Електронний ресурс] – Режим доступу до ресурсу: https://www.grorapidlabs.com/blog/what-is-system-analysis-steps-importance-implementation

19. Library Information Systems Analysis [Електронний ресурс] – Режим доступу до ресурсу: https://www.indeed.com/career-advice/career-development/what-is-system-analysis

20. Library Systems Analysis and Design Methodology [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/364311340_Systems_Analysis_and_Design_Methodology_and_Supporting_Processes

21. Best Practices in Library System Implementation [Електронний ресурс] – Режим доступу до ресурсу: https://implementationscienecomms.biomedcentral.com/articles/10.1186/s43058-023-00504-5

22. Методологія системного аналізу бібліотечних процесів [Електронний ресурс] – Режим доступу до ресурсу: https://studfile.net/preview/16455660/

23. Use Case Diagram – Unified Modeling Language (UML) [Електронний ресурс] – Режим доступу до ресурсу: https://www.geeksforgeeks.org/use-case-diagram/

24. Діаграма розгортання бібліотечних систем: UML із ПРИКЛАДОМ [Електронний ресурс] – Режим доступу до ресурсу: https://www.guru99.com/uk/deployment-diagram-uml-example.html

25. What is OLAP (online analytical processing) in Libraries? [Електронний ресурс] – Режим доступу до ресурсу: https://www.ibm.com/topics/olap

26. Library Data Warehousing and OLAP [Електронний ресурс] – Режим доступу до ресурсу: https://www.guru99.com/online-analytical-processing.html

27. KPI для бібліотек: ключові показники ефективності [Електронний ресурс] – Режим доступу до ресурсу: https://hurma.work/blog/shho-take-kpi-klyuchovi-pokazniki-efektivnosti/

28. Library Data Warehouse Solutions [Електронний ресурс] – Режим доступу до ресурсу: https://www.oracle.com/database/what-is-a-data-warehouse/

29. Digital Library Data Architecture [Електронний ресурс] – Режим доступу до ресурсу: https://www.ibm.com/data-warehouse

30. Library Database Architecture Design [Електронний ресурс] – Режим доступу до ресурсу: https://learn.microsoft.com/en-us/azure/architecture/databases/

31. ETL Processes for Library Management Systems [Електронний ресурс] – Режим доступу до ресурсу: https://www.ibm.com/topics/etl

32. SQL Server Integration Services for Libraries [Електронний ресурс] – Режим доступу до ресурсу: https://learn.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver16

33. Brown, K.A., Smith, J.D., Johnson, M.R. Enhanced k-Means Clustering for Library User Segmentation and Collection Optimization. International Journal of Information Science, Library Research Foundation, pp.1-11, 2023.

34. Liu, M. Apriori association rule-based evaluation method of library user preferences and reading patterns [D]. University of Information Sciences, 2023. DOI: 10.27135

35. Anderson, R. Data mining technology applying Apriori association rules for discovering library user reading behavior patterns [D]. Digital Library Research Institute.

36. SQL Server Analysis Services for Library Analytics [Електронний ресурс] – Режим доступу до ресурсу: https://learn.microsoft.com/en-us/analysis-services/analysis-services-overview?view=asallproducts-allversions

37. SQL Server Reporting Services for Library Management [Електронний ресурс] – Режим доступу до ресурсу: https://learn.microsoft.com/en-us/sql/reporting-services/create-deploy-and-manage-mobile-and-paginated-reports?view=sql-server-ver15

38. IBM, "Random Forest applications in library collection prediction and user behavior analysis" [Електронний ресурс] – Режим доступу до ресурсу: https://www.ibm.com/topics/random-forest

39. Scikit-learn Documentation, "RandomForestRegressor for Library Demand Forecasting" [Електронний ресурс] – Режим доступу до ресурсу: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

40. Scikit-learn Documentation, "GradientBoostingRegressor for Library Analytics" [Електронний ресурс] – Режим доступу до ресурсу: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html

41. NISO. Library Analytics and Assessment Standards [Електронний ресурс] – Режим доступу до ресурсу: https://www.niso.org/

42. LIBER. European Research Libraries Digital Transformation [Електронний ресурс] – Режим доступу до ресурсу: https://libereurope.eu/

43. Code4Lib. Library Technology and Innovation [Електронний ресурс] – Режим доступу до ресурсу: https://code4lib.org/

44. Miller, P., Davis, S. (2022). Intelligent Collection Development in Academic Libraries Using Machine Learning. College & Research Libraries, 83(4), 456-472.

45. Garcia, M.E., Kumar, V. (2021). User Experience Analytics in Digital Libraries: A Data-Driven Approach. Information Technology and Libraries, 40(2), 78-95.

46. Wilson, T., Lee, H. (2023). Predictive Modeling for Library Resource Planning: A Comparative Study. Library Resources & Technical Services, 67(1), 23-39.

47. Robinson, J.A. (2020). Association Rule Mining for Library Collection Analysis: Best Practices and Case Studies. Library Management, 41(6/7), 342-358.

48. Taylor, S.M., White, D.K. (2022). Clustering Techniques for Library User Behavior Analysis. Journal of Documentation, 78(3), 567-584.

49. Pandas Documentation. Data Analysis Library [Електронний ресурс] – Режим доступу до ресурсу: https://pandas.pydata.org/docs/

50. NumPy Documentation. Scientific Computing [Електронний ресурс] – Режим доступу до ресурсу: https://numpy.org/doc/

51. Matplotlib Documentation. Data Visualization [Електронний ресурс] – Режим доступу до ресурсу: https://matplotlib.org/stable/contents.html

52. Seaborn Documentation. Statistical Data Visualization [Електронний ресурс] – Режим доступу до ресурсу: https://seaborn.pydata.org/

ДОДАТОК Б

КОД СКРИПТІВ ЗБОРУ ДАНИХ ТА ПОБУДОВИ МОДЕЛЕЙ

Сторінок 14

Реалізація та налаштування методу кластеризації K-means

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from mpl_toolkits.mplot3d import Axes3D
import warnings
warnings.filterwarnings('ignore')

plt.rcParams['font.family'] = ['DejaVu Sans']
plt.rcParams['axes.unicode_minus'] = False

print("=== Завантаження даних книжкового фонду для кластеризації ===")

df = pd.read_csv("library_books_dataset.csv", encoding='utf-8')
print(f"Даних завантажено: {len(df)} книг")
print(df.head())

features = [
    'Rating',
    'PageCount',
    'BorrowCount',
    'ReviewCount',
    'AvgReadingDays',
    'RecommendationScore'
]

df_cluster = df[features + ['Genre', 'AgeCategory', 'Title']].dropna()

print(f"\nОзнаки для кластеризації: {features}")

print(f"Кількість доступних записів: {len(df_cluster)}")

scaler = StandardScaler()
scaled = scaler.fit_transform(df_cluster[features])
print("\nДані масштабовано за StandardScaler")

inertia = []
k_values = range(1, 11)
```

```
for k in k_values:
    km = KMeans(n_clusters=k, random_state=42, n_init=10)
    km.fit(scaled)
    inertia.append(km.inertia_)

plt.figure(figsize=(10, 6))
plt.plot(k_values, inertia, marker='o')
plt.title('Метод ліктя — оптимальна кількість кластерів')
plt.xlabel('Кількість кластерів (k)')
plt.ylabel('Inertia')
plt.grid(True)
plt.savefig('elbow_books.png', dpi=300)
plt.close()

print("\nГрафік методу ліктя збережено у elbow_books.png")

optimal_k = 4
print(f"\nВибрано k = {optimal_k}")

kmeans = KMeans(n_clusters=optimal_k, random_state=42, n_init=10)
clusters = kmeans.fit_predict(scaled)
df_cluster['Cluster'] = clusters

cluster_names = {

  0: "Популярні та короткі",
    1: "Товсті та малочитані",
    2: "Високорейтингові хіти",
    3: "Швидко читаються, середня популярність"
}

print(df_cluster['Cluster'].value_counts())

pca = PCA(n_components=3)
pca_data = pca.fit_transform(scaled)
df_cluster['PCA1'], df_cluster['PCA2'], df_cluster['PCA3'] = pca_data.T

fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

colors = ['red', 'blue', 'green', 'orange']

for c in range(optimal_k):
```

```
data_c = df_cluster[df_cluster['Cluster'] == c]
    ax.scatter(
        data_c['PCA1'], data_c['PCA2'], data_c['PCA3'],
        color=colors[c],
        label=cluster_names[c],
        s=60, alpha=0.7
    )

ax.set_title("3D PCA — Кластери книжкового фонду")
ax.set_xlabel("PCA1")
ax.set_ylabel("PCA2")
ax.set_zlabel("PCA3")
ax.legend()

plt.savefig('3d_pca_books.png', dpi=300)
plt.close()

print("\n=== АНАЛІЗ КЛАСТЕРІВ ===")
cluster_summary = df_cluster.groupby('Cluster')[features].mean().round(2)
print(cluster_summary)

cluster_summary.to_csv("cluster_summary_books.csv", encoding="utf-8-sig")

plt.figure(figsize=(10, 6))
sns.heatmap(cluster_summary, annot=True, cmap="YlGnBu")
plt.title("Середні значення параметрів книг у кластерах")
plt.savefig("cluster_heatmap_books.png", dpi=300)
plt.close()

plt.figure(figsize=(10, 6))
sns.scatterplot(
    data=df_cluster,
    x='Rating',
    y='BorrowCount',
    hue=df_cluster['Cluster'].map(cluster_names),
    palette='Set1'
)
plt.title("Кластери на графіку: Рейтинг vs Кількість позичень")
plt.savefig("rating_borrow_clusters.png", dpi=300)
plt.close()

df_cluster.to_csv("books_with_clusters.csv", encoding="utf-8-sig", index=False)
```

```
print("\n=== Готово! Створено такі файли ===")
print("1. elbow_books.png")
print("2. 3d_pca_books.png")
print("3. cluster_heatmap_books.png")
print("4. rating_borrow_clusters.png")
print("5. books_with_clusters.csv")
print("6. cluster_summary_books.csv")
```

Побудова моделі прогнозування Random Forest з використанням кластеризації

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.cluster import KMeans
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.decomposition import PCA
import warnings
warnings.filterwarnings("ignore")

plt.rcParams['font.family'] = 'DejaVu Sans'


print("=== Завантаження даних бібліотечної системи ===")

df = pd.read_csv("library_books_dataset.csv", encoding="utf-8")

print(f"Даних завантажено: {len(df)} книг")

target_variable = "BorrowCount"   # ціль — прогноз позичень книги

features = [
    "Rating",
    "PageCount",
    "ReviewCount",
    "AvgReadingDays",
    "RecommendationScore"
]

# + categorical (Жанр, вікова категорія)
```

```
categorical_features = ["Genre", "AgeCategory"]

df = df.dropna(subset=features + categorical_features + [target_variable])

# Кодування категоріальних ознак
encoder = LabelEncoder()
for col in categorical_features:
    df[col] = encoder.fit_transform(df[col])

all_features = features + categorical_features

X = df[all_features]
y = df[target_variable]


scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

print("\nМасштабування виконано.")


n_clusters = 4
kmeans = KMeans(n_clusters=n_clusters, random_state=42)
clusters = kmeans.fit_predict(X_scaled)

df['Cluster'] = clusters


pca_3d = PCA(n_components=3)
coords_3d = pca_3d.fit_transform(X_scaled)
df['PCA1'], df['PCA2'], df['PCA3'] = coords_3d[:, 0], coords_3d[:, 1], coords_3d[:, 2]

pca_2d = PCA(n_components=2)
coords_2d = pca_2d.fit_transform(X_scaled)
df['PCA1_2D'], df['PCA2_2D'] = coords_2d[:, 0], coords_2d[:, 1]

test_size = int(len(df) * 0.10)
df_train = df[:-test_size]
df_test = df[-test_size:]

X_train = df_train[all_features]
y_train = df_train[target_variable]
X_test = df_test[all_features]
```

```
y_test = df_test[target_variable]

rf = RandomForestRegressor(
    n_estimators=200,
    max_depth=14,
    min_samples_split=5,
    min_samples_leaf=2,
    random_state=42
)

rf.fit(X_train, y_train)
pred_no_cluster = rf.predict(X_test)

mse_no_cluster = mean_squared_error(y_test, pred_no_cluster)
r2_no_cluster = r2_score(y_test, pred_no_cluster)

cluster_models = {}
cluster_sizes = {}

train_clusters = df_train['Cluster'].values
test_clusters = df_test['Cluster'].values

for c in range(n_clusters):

    mask = train_clusters == c
    X_train_c = X_train[mask]
    y_train_c = y_train[mask]

    cluster_sizes[c] = len(X_train_c)

    rf_params = {
        'n_estimators': 200,
        'max_depth': 14,
        'min_samples_split': 4,
        'min_samples_leaf': 2,
        'random_state': 42
    }

    if len(X_train_c) < 50:
        rf_params['n_estimators'] = 120
        rf_params['max_depth'] = 10

    model = RandomForestRegressor(**rf_params)
```

```
model.fit(X_train_c, y_train_c)

    cluster_models[c] = model

# Прогноз
pred_cluster = np.zeros(len(X_test))

for i in range(len(X_test)):
    cl = test_clusters[i]
    model = cluster_models[cl]
    pred_cluster[i] = model.predict(X_test.iloc[i:i+1])[0]


mse_cluster = mean_squared_error(y_test, pred_cluster)
r2_cluster = r2_score(y_test, pred_cluster)

print("\n=== РЕЗУЛЬТАТИ ПРОГНОЗУ ===")
print("\nRandomForest БЕЗ кластерів:")
print(f"MSE: {mse_no_cluster:.4f}")
print(f"R²:  {r2_no_cluster:.4f}")

print("\nRandomForest З КЛАСТЕРАМИ:")
print(f"MSE: {mse_cluster:.4f}")
print(f"R²:  {r2_cluster:.4f}")

print("\nРозмір кластерів:")
for c, size in cluster_sizes.items():
    print(f"Кластер {c}: {size} книг")


# --- 1) 2D PCA ---
plt.figure(figsize=(12, 7))
sns.scatterplot(data=df, x='PCA1_2D', y='PCA2_2D', hue='Cluster', palette='Set1')
plt.title("2D PCA кластеризація книг")
plt.savefig("library_clusters_2d.png", dpi=300)
plt.close()

# --- 2) 3D PCA ---
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection="3d")
scatter = ax.scatter(df['PCA1'], df['PCA2'], df['PCA3'],
                     c=df['Cluster'], cmap='Set1')
ax.set_title("3D PCA кластеризація")
```

```
plt.savefig("library_clusters_3d.png", dpi=300)
plt.close()


# --- 3) Порівняння прогнозування ---
plt.figure(figsize=(14, 6))
plt.plot(y_test.values, label="Факт", marker="o")
plt.plot(pred_no_cluster, label="RF без кластерів", marker="s")
plt.plot(pred_cluster, label="RF з кластерами", marker="^")
plt.title("Прогноз BorrowCount у бібліотеці")
plt.legend()
plt.grid()
plt.savefig("library_rf_comparison.png", dpi=300)
plt.close()

print("\n=== Аналіз завершено ===")
```

Побудова моделі прогнозування Gradient Boosting з використанням кластеризації

```
import pyodbc
import pandas as pd
import numpy as np
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

conn = pyodbc.connect(
    'DRIVER={ODBC Driver 17 for SQL Server};'
    'SERVER=LAPTOP-MLOKGU0A;'
    'DATABASE=library_;'
    'UID=sa;'
    'PWD=123'
)

query = """
SELECT
    bf.BorrowDate,
    bf.DaysBorrowed,
    bf.IsReturned,
    bf.AgeCategoryID,
```

```
   bf.LikedBooksCount,
    b.GenreID,
    b.AuthorID,
    b.PublisherID,
    b.PublicationYear,
    ac.CategoryCode,
    g.Name AS GenreName,
    t.Year, t.Month, t.Day
FROM BorrowFact bf
JOIN BookDim b ON bf.BookID = b.BookID
JOIN AgeCategoryDim ac ON bf.AgeCategoryID = ac.AgeCategoryID
JOIN GenreDim g ON b.GenreID = g.GenreID
JOIN TimeDim t ON bf.TimeID = t.TimeID
WHERE bf.BorrowDate IS NOT NULL;
"""

data = pd.read_sql(query, conn)
conn.close()

daily_counts = data.groupby('BorrowDate').size().reset_index(name='BorrowCount')

data = pd.merge(data, daily_counts, on='BorrowDate', how='left')

data['BorrowDate'] = pd.to_datetime(data['BorrowDate'])
data = data.sort_values('BorrowDate')
data = data.set_index('BorrowDate')

data['DayOfWeek'] = data.index.dayofweek
data['Season'] = data['Month'] % 12 // 3 + 1

data = pd.get_dummies(data, columns=['GenreName', 'CategoryCode'],
drop_first=True)

target = "BorrowCount"

features = [
    'DaysBorrowed',
    'IsReturned',
    'LikedBooksCount',
    'GenreID',
    'AuthorID',
    'PublisherID',
    'PublicationYear',
```

```
 'Year',
    'Month',
    'Day',
    'DayOfWeek',
    'Season'
] + [c for c in data.columns if c.startswith("GenreName_") or
c.startswith("CategoryCode_")]

X = data[features]
y = data[target]

# Масштабування
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

forecast_days = 7  # прогноз на тиждень
X_train, X_test = X[:-forecast_days], X[-forecast_days:]
y_train, y_test = y[:-forecast_days], y[-forecast_days:]

gb = GradientBoostingRegressor(
    n_estimators=200,
    learning_rate=0.1,
    max_depth=5,
    random_state=42
)

gb.fit(X_train, y_train)
pred_no_cluster = gb.predict(X_test)

mse_no_cluster = mean_squared_error(y_test, pred_no_cluster)
r2_no_cluster = r2_score(y_test, pred_no_cluster)

n_clusters = 3
kmeans = KMeans(n_clusters=n_clusters, random_state=42)

train_clusters = kmeans.fit_predict(X_scaled[:-forecast_days])
test_clusters = kmeans.predict(scaler.transform(X_test))

cluster_models = {}
cluster_sizes = {}

for cl in range(n_clusters):
    mask = train_clusters == cl
```

```
Xc = X_train[mask]
    yc = y_train[mask]
    cluster_sizes[cl] = len(Xc)

    model = GradientBoostingRegressor(
        n_estimators=200,
        learning_rate=0.1,
        max_depth=5,
        random_state=42
    )
    model.fit(Xc, yc)
    cluster_models[cl] = model

pred_cluster = np.zeros(len(X_test))

for i in range(len(X_test)):
    cluster = test_clusters[i]
    pred_cluster[i] = cluster_models[cluster].predict(X_test.iloc[i:i+1])[0]

mse_cluster = mean_squared_error(y_test, pred_cluster)
r2_cluster = r2_score(y_test, pred_cluster)

print("\n=== Gradient Boosting БЕЗ кластеризації ===")
print("MSE:", mse_no_cluster)
print("R2:", r2_no_cluster)

print("\n=== Gradient Boosting З кластеризацією ===")
print("MSE:", mse_cluster)
print("R2:", r2_cluster)

print("\nРозмір кластерів:")
print(cluster_sizes)

plt.figure(figsize=(12,6))
plt.plot(y_test.index, y_test.values, label='Фактичні', marker='o')
plt.plot(y_test.index, pred_no_cluster, label='Без кластеризації', marker='s')
plt.plot(y_test.index, pred_cluster, label='З кластеризацією', marker='*')
plt.title("Прогноз BorrowCount (позичень книг у день)")
plt.xlabel("Дата")
plt.ylabel("BorrowCount")
plt.legend()
plt.grid(True)
plt.show()
```

Реалізація функції отримання записів з API

```
import requests
import psycopg2
from psycopg2.extras import execute_values

DB_HOST = "localhost"
DB_PORT = 5432
DB_NAME = "library_"
DB_USER = "postgres"
DB_PASSWORD = "root"

conn = psycopg2.connect(
    host=DB_HOST, port=DB_PORT, dbname=DB_NAME, user=DB_USER,
password=DB_PASSWORD
)
cur = conn.cursor()

# --- Google Books API ---
query = "shakespeare"
max_results = 10
url =
f"https://www.googleapis.com/books/v1/volumes?q={query}&maxResults={max_re
sults}"

response = requests.get(url)
books_data = response.json()

for item in books_data.get("items", []):
volume_info = item.get("volumeInfo", {})
    title = volume_info.get("title", "Unknown")
    authors = volume_info.get("authors", ["Unknown"])
    published_year = None
    if "publishedDate" in volume_info:
        published_year = volume_info["publishedDate"][:4]
        if not published_year.isdigit():
            published_year = None
        else:
            published_year = int(published_year)
    isbn = None
    for iden in volume_info.get("industryIdentifiers", []):
        if iden.get("type") in ["ISBN_10", "ISBN_13"]:
            isbn = iden.get("identifier")
```

```
 break
    categories = volume_info.get("categories", ["Unknown"])

    # --- Вставка автора ---
    for full_author in authors:
        if "," in full_author:
            last_name, first_name = [s.strip() for s in full_author.split(",", 1)]
        else:
            first_name, last_name = full_author, ""
        cur.execute(
            """
            INSERT INTO AuthorDim (FirstName, LastName, Country)
            VALUES (%s, %s, %s)
ON CONFLICT (FirstName, LastName) DO NOTHING
            RETURNING AuthorID
            """,
            (first_name, last_name, None)
        )
        res = cur.fetchone()
        author_id = res[0] if res else None

    # --- Вставка жанру ---
    for cat in categories:
        cur.execute(
            """
            INSERT INTO GenreDim (Name)
            VALUES (%s)
            ON CONFLICT (Name) DO NOTHING
            RETURNING GenreID
            """,
            (cat,)
        )
        res = cur.fetchone()
        genre_id = res[0] if res else None

    # --- Вставка книги ---
    cur.execute(
        """
        INSERT INTO BookDim (Title, AuthorID, GenreID, PublisherID,
PublicationYear, ISBN)
        VALUES (%s, %s, %s, %s, %s, %s)
RETURNING BookID
        """,
```

```
(title, author_id, genre_id, None, published_year, isbn)
    )
    book_id = cur.fetchone()[0]
    print(f"Inserted Book: {title} (ID={book_id})")

conn.commit()
cur.close()
conn.close()
```

ДОДАТОК В

СКРИПТ ПОБУДОВИ СХОВИЩА ДАНИХ

Сторінок 3

```
USE library_;
GO

SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE dbo.AgeCategoryDim (
   AgeCategoryID INT IDENTITY(1,1) NOT NULL,
   CategoryCode NVARCHAR(50) NULL,
   Description NVARCHAR(255) NULL,
   CONSTRAINT PK_AgeCategoryDim PRIMARY KEY CLUSTERED
(AgeCategoryID ASC)
);
GO

CREATE TABLE dbo.AuthorDim (
   AuthorID INT IDENTITY(1,1) NOT NULL,
   FirstName NVARCHAR(255) NULL,
   LastName NVARCHAR(255) NULL,
   Country NVARCHAR(255) NULL,
   CONSTRAINT PK_AuthorDim PRIMARY KEY CLUSTERED (AuthorID ASC)
);
GO

CREATE TABLE dbo.GenreDim (
   GenreID INT IDENTITY(1,1) NOT NULL,
   Name NVARCHAR(255) NULL,
   CONSTRAINT PK_GenreDim PRIMARY KEY CLUSTERED (GenreID ASC)
);
GO

CREATE TABLE dbo.PublisherDim (
   PublisherID INT IDENTITY(1,1) NOT NULL,
   Name NVARCHAR(255) NULL,
   Country NVARCHAR(255) NULL,
   CONSTRAINT PK_PublisherDim PRIMARY KEY CLUSTERED (PublisherID
ASC)
);
GO

CREATE TABLE dbo.BookDim (
   BookID INT IDENTITY(1,1) NOT NULL,
```

```
 Title NVARCHAR(255) NULL,
    AuthorID INT NULL,
    GenreID INT NULL,
    PublisherID INT NULL,
    PublicationYear INT NULL,
    ISBN NVARCHAR(20) NULL,
    CONSTRAINT PK_BookDim PRIMARY KEY CLUSTERED (BookID ASC)
);
GO

CREATE TABLE dbo.TimeDim (
    TimeID INT IDENTITY(1,1) NOT NULL,
    Year INT NULL,
    Month INT NULL,
    Day INT NULL,
    CONSTRAINT PK_TimeDim PRIMARY KEY CLUSTERED (TimeID ASC)
);
GO

CREATE TABLE dbo.LocationDim (
    LocationID INT IDENTITY(1,1) NOT NULL,
    City NVARCHAR(255) NULL,
    Region NVARCHAR(255) NULL,
    CONSTRAINT PK_LocationDim PRIMARY KEY CLUSTERED (LocationID
ASC)
);
GO

CREATE TABLE dbo.BorrowFact (
    TimeID INT NULL,
    LocationID INT NULL,
    BookID INT NULL,
    UserID INT NULL,
    BorrowDate DATE NULL,
    ReturnDate DATE NULL,
    DaysBorrowed INT NULL,
    IsReturned BIT NULL,
    AgeCategoryID INT NULL,
    LikedBooksCount INT DEFAULT 0
);
GO
```

```
ALTER TABLE dbo.BookDim WITH CHECK
   ADD CONSTRAINT FK_BookDim_Author
      FOREIGN KEY (AuthorID) REFERENCES dbo.AuthorDim(AuthorID);
GO

ALTER TABLE dbo.BookDim WITH CHECK
   ADD CONSTRAINT FK_BookDim_Genre
      FOREIGN KEY (GenreID) REFERENCES dbo.GenreDim(GenreID);
GO

ALTER TABLE dbo.BookDim WITH CHECK
   ADD CONSTRAINT FK_BookDim_Publisher
      FOREIGN KEY (PublisherID) REFERENCES dbo.PublisherDim(PublisherID);
GO

ALTER TABLE dbo.BorrowFact WITH CHECK
   ADD CONSTRAINT FK_BorrowFact_Time
      FOREIGN KEY (TimeID) REFERENCES dbo.TimeDim(TimeID);
GO

ALTER TABLE dbo.BorrowFact WITH CHECK
   ADD CONSTRAINT FK_BorrowFact_Location
      FOREIGN KEY (LocationID) REFERENCES dbo.LocationDim(LocationID);
GO

ALTER TABLE dbo.BorrowFact WITH CHECK
   ADD CONSTRAINT FK_BorrowFact_Book
      FOREIGN KEY (BookID) REFERENCES dbo.BookDim(BookID);
GO

ALTER TABLE dbo.BorrowFact WITH CHECK
   ADD CONSTRAINT FK_BorrowFact_AgeCategory
      FOREIGN KEY (AgeCategoryID) REFERENCES
dbo.AgeCategoryDim(AgeCategoryID);
GO
```