Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Факультет інформаційних технологій
(факультет)

Кафедра системного аналізу та управління
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня магістра

Студента Сербіна Дмитра Володимировича
академічної групи 124м-22-1
спеціальності 124 Системний аналіз
на тему: «Розробка рекомендаційної системи для торгової мережі з використанням методів Data Science»

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		Рейтинговою	**Інституційною**	
кваліфікаційної роботи	*к.ф.-м.н., доц. Коряшкіна Л.С.*			
розділів:				
Інформаційно-аналітичний	*к.ф.-м.н., доц. Коряшкіна Л.С.*			
Спеціальний розділ	*к.ф.-м.н., доц. Коряшкіна Л.С.*			
Рецензент				
Нормоконтролер	*к.ф.-м.н., доц. Хом'як Т.В.*			

Дніпро
2023

ЗАТВЕРДЖЕНО:

завідувач кафедри

Системного аналізу та управління
(повна назва)

_____ *к.т.н., доц. Желдак Т.А.*
(підпис) (прізвище,ініціали)

«____» _____20___року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня *магістра*

студенту *Сербіну Д.В.* **академічної групи** *124м-22-1*
спеціальності: *124 Системний аналіз*
на тему «*Розробка рекомендаційної системи для торгової мережі з використанням методів Data Science»*
затверджену наказом ректора НТУ «Дніпровська політехніка»
від *№ 1227-с від 09.10.2023 р.*

Розділ	Зміст	Терміни виконання
1. Інформаційно-аналітичний розділ	*Дослідити моделі та методи розробки сучасних рекомендаційних систем. Визначити основні задачі побудови рекомендаційних систем.*	09.10.2023 – 09.11.2023
2.Спеціальний розділ	*Вирішити задачу розрахунку рекомендацій для торгової мережі. Підготувати дані, обрати та реалізувати методи для побудови рекомендаційної системи.*	10.11.2023– 05.12.2023

Завдання видано _____ *доц. Коряшкіна Л.С.*
(підпис) (прізвище, ініціали)

Дата видачі: *09.10.2023 р.*

Дата подання до екзаменаційної комісії: _____

Прийнято до виконання _____ Сербін Д.В.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 74 с., 6 рис., 15 табл., 3 додатка, 13 джерел.

Об’єктом дослідження в роботі є розробка рекомендаційної системи для торгової мережі.

Предметом дослідження алгоритми та методи, що застосовуються для створення рекомендаційних систем.

Метою даної кваліфікаційної роботи є розробка рекомендаційної системи, що підвищить користувальницький досвід за допомогою методів інтелектуального аналізу даних.

Методи дослідження: RFM–аналіз з використанням K-means кластеризації, колаборативна фільтрація з розглядом підходів: Клієнт-до-Клієнта та Продукт-до-Продукту.

В інформаційно–аналітичному розділі наведено відомості про об’єкт дослідження, поставлені задачі дослідження та обрані методи їх розв’язання.

У спеціальному розділі виконано розв'язання задачі розробки рекомендаційної системи сформованими методами.

Практична цінність отриманих результатів полягає у можливості масштабувати та поширювати отримані результати для інших задач схожих, за формулюванням та структурою даних.

Ключові слова: РЕКОМЕНДАЦІЙНА СИСТЕМА, RFM–АНАЛІЗ, K-MEANS, КЛАСТЕРИЗАЦІЯ, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ.

ABSTRACT

Explanatory note: 74 p., 6 pictures, 15 tables, 3 annexs, 13 sources.

The object of research in the work is the development of a recommendation system for a trade network.

The subject of research is algorithms and methods used to create recommender systems.

The goal of this qualification work is to develop a recommender system that will improve the user experience using methods of intelligent data analysis.

Research methods: RFM-analysis using K-means clustering, collaborative filtering with consideration of approaches: User-to-User and Item-to-Item.

The informational and analytical section provides information about the object of the study, the set research tasks and the chosen methods of solving them.

In a special section, the solution to the task of developing a recommendation system using established methods is performed.

The practical value of the obtained results lies in the ability to scale and distribute the obtained results for other problems similar in formulation and data structure.

Key words: RECOMMENDATION SYSTEM, RFM-ANALYSIS, K-MEANS, CLUSTERIZATION, COLLABORATIVE FILTERING.

ЗМІСТ

ВСТУП

Сучасний ринок роздрібної торгівлі, особливо супермаркети та гіпермаркети, зазнає значних трансформацій у зв'язку з ростом конкуренції та змінами у споживчому підході. Для супермаркетів стає важливим завжди знаходити способи покращити обслуговування клієнтів, зробити шопінг більш комфортним та персоналізованим. Рекомендаційні системи відіграють ключову роль у вирішенні цих завдань. Давайте розглянемо, чому супермаркетам важлива рекомендаційна система.

Рекомендаційні системи допомагають стимулювати продажі, пропонуючи клієнтам товари, які вони можуть бути зацікавлені в придбанні. Це дозволяє підвищити середній чек та об'єм продажів. Коли користувач отримує персоналізовані рекомендації, ймовірність того, що він здійснить покупку, зростає.

Супермаркети бажають забезпечити клієнтам зручний та приємний досвід під час шопінгу. Рекомендаційні системи допомагають робити покупки більш зручними, пропонуючи користувачам товари, які відповідають їхнім потребам і вподобанням. Це сприяє задоволенню клієнтів та збільшує їхню лояльність.

Рекомендаційні системи допомагають супермаркетам зрозуміти попит клієнтів на різні товари. Аналіз покупок та взаємодії з товарами дозволяє оптимізувати запаси та управляти асортиментом більш ефективно. Також можна вчасно виявляти товари, які вимагають додаткової реклами або акцій.

Рекомендаційні системи дозволяють створювати персоналізовані пропозиції та акції для клієнтів. Це дозволяє привертати увагу клієнтів до товарів, які їх цікавлять, та мотивувати їх до покупок.

Рекомендаційні системи збирають велику кількість даних про покупки та взаємодію клієнтів з товарами. Це дозволяє проводити аналіз та прогнозувати поведінку клієнтів, виявляти тенденції та попит на конкретні товари.

Рекомендаційні системи можуть бути потужним інструментом для залучення нових клієнтів. Персоналізовані рекомендації та акції можуть привертати увагу нових користувачів та стимулювати їх до першого візиту у супермаркет.

В сучасному ринку супермаркетів конкуренція дуже висока. Супермаркети, які використовують передові технології, включаючи рекомендаційні системи, мають перевагу перед конкурентами. Вони можуть пропонувати клієнтам більш персоналізований та зручний сервіс, що збільшує їхню привабливість для споживачів.

Рекомендаційні системи стають невід'ємною частиною успішної діяльності супермаркетів та гіпермаркетів. Вони допомагають підвищити продажі, покращити користувальницький досвід, оптимізувати управління асортиментом та залучити нових клієнтів. Рекомендаційні системи не лише полегшують життя клієнтів, але й забезпечують конкурентну перевагу на ринку роздрібної торгівлі.

1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ РОЗДІЛ

1.1 Історичний контекст та бізнес у розрізі рекомендаційних систем

1.1.1 Історія рекомендаційних систем

Історія рекомендаційних систем є цікавою та багатогранною. Ці системи, які сьогодні є невід'ємною частиною багатьох онлайн-платформ, пройшли тривалий шлях розвитку та еволюції. Вони постійно адаптуються до змін у технологіях, поведінці споживачів та ділових потребах. Нижче наведено докладний огляд історії рекомендаційних систем.

Початкові етапи (1980-ті - середина 1990-х років)

Ранні рекомендаційні системи з'явилися у 1980-х роках, коли основною метою було допомогти користувачам знаходити релевантну інформацію в середовищі з обмеженою кількістю контенту. Вони в основному базувалися на простих фільтрах, які використовували критерії на кшталт жанру або популярності.

Розвиток колаборативного фільтрування (середина 1990-х)

Ключовим моментом у розвитку рекомендаційних систем стало поява колаборативного фільтрування в середині 1990-х років. Ця технологія використовує дані про поведінку користувачів (наприклад, оцінки або історію покупок) для виявлення схожості між користувачами та рекомендації продуктів. GroupLens і Tapestry були одними з перших систем, які застосували цей підхід.

Розквіт Інтернет-комерції (кінець 1990-х - 2000-ті роки)

З розвитком Інтернету та електронної комерції у кінці 1990-х та на початку 2000-х років рекомендаційні системи стали важливою складовою онлайн-магазинів. Amazon і Netflix стали піонерами у використанні персоналізованих рекомендацій для підвищення продажів та задоволеності клієнтів. Amazon використовував алгоритми, які рекомендували продукти на основі покупок і переглядів інших користувачів, тоді як Netflix запровадив просунуті алгоритми для рекомендації фільмів.

1.1.2 Розвиток технологій та алгоритмів (2010-ті роки)

У 2010-ті роки рекомендаційні системи стали ще більш розвинутими, з використанням складних алгоритмів машинного навчання та штучного інтелекту. Алгоритми глибокого навчання, такі як нейронні мережі, почали використовуватися для обробки великих обсягів даних та створення більш точних та персоналізованих рекомендацій. Це також був період, коли з'явилися системи рекомендацій у соціальних мережах, таких як Facebook та Twitter, які використовували дані про соціальні зв'язки для покращення рекомендацій.[1]

1.1.3 Використання великих даних та штучного інтелекту (сучасність)

У наш час рекомендаційні системи стали ще більш інтегрованими з використанням великих даних, штучного інтелекту та технологій обробки природної мови. Вони не тільки рекомендують продукти або контент, але й адаптуються до індивідуальних переваг користувачів, їхньої поведінки та навіть емоційного стану. Рекомендаційні системи також стали більш прозорими та етичними, зосереджуючись на приватності та безпеці даних користувачів.

1.1.4 Майбутнє рекомендаційних систем

Майбутнє рекомендаційних систем обіцяє ще більше інновацій та розвитку. Очікується, що вони стануть більш контекстно-свідомими, здатними адаптуватися не тільки до індивідуальних переваг, але й до конкретних ситуацій та контекстів, у яких знаходиться користувач. Також можливе розширення застосування штучного інтелекту для більш глибокого розуміння людської поведінки та психології.

Рекомендаційні системи продовжують еволюціонувати, пристосовуючись до змін у технологіях, бізнес-моделях та потребах користувачів. Вони стають все більш витонченими та інтелектуальними, пропонуючи високий рівень персоналізації та значно покращуючи користувацький досвід.

1.2 Визначення бізнес-цілей для рекомендаційної системи

Визначення бізнес-цілей для рекомендаційної системи є ключовим етапом в процесі її розробки та імплементації. Бізнес-цілі визначають стратегічний напрямок для системи та служать основою для оцінки її успішності. Нижче наведено детальний огляд цього процесу.[2]

Розуміння бізнес-контексту

Перш ніж формулювати конкретні бізнес-цілі, важливо розуміти загальний бізнес-контекст, в якому функціонує компанія. Це включає аналіз ринку, визначення цільової аудиторії, розуміння конкурентних переваг та виявлення ключових викликів, з якими стикається бізнес. Такий аналіз допоможе визначити, у яких аспектах рекомендаційна система може найефективніше сприяти досягненню бізнес-цілей.

Визначення стратегічних бізнес-цілей

На цьому етапі ключово визначити, які стратегічні цілі компанії можуть бути підтримані за допомогою рекомендаційної системи. Ці цілі можуть включати:

Збільшення продажів: Використання рекомендаційної системи для пропонування продуктів, які можуть зацікавити користувачів, сприяючи таким чином збільшенню середнього чеку та загального обсягу продажів.

Підвищення лояльності клієнтів: Через персоналізовані рекомендації можна підвищити задоволеність клієнтів, що сприятиме зростанню їхньої лояльності та частоти покупок.

Оптимізація запасів: Рекомендаційна система може допомогти в аналізі тенденцій попиту, що дозволить краще планувати запаси та уникнути надмірних запасів або дефіциту.

Поліпшення маркетингових стратегій: Аналіз даних про поведінку клієнтів може допомогти у формуванні ефективніших маркетингових кампаній, націлених на конкретні сегменти користувачів.

1.2.1 Формулювання Ключових Показників Ефективності (KPI)

Після визначення бізнес-цілей необхідно розробити систему KPI для оцінки ефективності рекомендаційної системи. Ці показники можуть включати:

Коефіцієнт конверсії: Відсоток відвідувачів, які здійснили покупку після перегляду рекомендацій.

Середній чек: Середня сума, витрачена клієнтами при кожній покупці.

Частота повторних покупок: Частота, з якою клієнти повертаються для здійснення додаткових покупок.[1]

Рівень утримання клієнтів: Відсоток клієнтів, які продовжують користуватися послугами компанії протягом певного періоду.

Аналіз і Постійне Вдосконалення

Важливо регулярно аналізувати ефективність рекомендаційної системи та проводити її оптимізацію. Це включає аналіз KPI, збір зворотного зв'язку від користувачів та адаптацію системи до змін у споживацьких тенденціях та поведінці.

1.2.2 Врахування Етичних та Юридичних Аспектів

При розробці та впровадженні рекомендаційної системи також важливо враховувати етичні та юридичні аспекти, особливо щодо збору та обробки даних користувачів. Забезпечення приватності даних та дотримання регулятивних норм є ключовими для підтримки довіри клієнтів.

1.3 Вимоги до рекомендаційної системи

Вимоги до рекомендаційної системи є фундаментальною складовою її розробки та впровадження. Ці вимоги визначають, як система має функціонувати, які дані вона буде обробляти, які технології будуть використані, та як вона буде інтегруватися з іншими системами компанії. Нижче наведено детальний огляд ключових аспектів, які слід враховувати при формулюванні вимог до рекомендаційної системи.

1.3.1 Технічні вимоги до рекомендаційної системи

Збір та обробка даних: Система повинна ефективно збирати та обробляти великі обсяги даних, включаючи дані про поведінку користувачів, історію покупок, перегляди продуктів тощо.

Інтеграція з існуючими системами: Рекомендаційна система має бути сумісною та інтегрованою з іншими бізнес-системами, наприклад, з системами управління запасами, CRM-системами та електронною комерцією.

Масштабованість: Система має бути масштабованою, щоб витримувати зростання обсягів даних та користувацької активності.

Висока продуктивність та надійність: Важливо забезпечити високу швидкість роботи та надійність системи, щоб гарантувати постійну доступність рекомендацій для користувачів.

1.3.2 Функціональні Вимоги

Персоналізація: Система повинна враховувати індивідуальні переваги та інтереси користувачів для створення персоналізованих рекомендацій.

Алгоритми рекомендацій: Розробка та впровадження ефективних алгоритмів для генерації релевантних та точних рекомендацій на основі різноманітних джерел даних.

Інтерактивність та відгуки користувачів: Система має дозволяти користувачам взаємодіяти з рекомендаціями та надавати зворотний зв'язок для покращення точності рекомендацій.

Візуалізація та інтерфейс: Розробка інтуїтивно зрозумілого та зручного інтерфейсу для представлення рекомендацій.

1.3.3 Етичні та Юридичні Вимоги

Дотримання конфіденційності та приватності даних: Забезпечення безпеки даних користувачів та дотримання всіх законодавчих та регулятивних норм у сфері обробки персональних даних.

Прозорість алгоритмів: Рекомендаційна система має бути прозорою у своїх методах роботи, дозволяючи користувачам розуміти, як формуються рекомендації.

1.3.4 Аналітичні Вимоги

Моніторинг та аналіз: Система повинна включати інструменти для моніторингу та аналізу ефективності рекомендацій, включаючи збір метрик, таких як коефіцієнт кліків, конверсія, взаємодія користувачів тощо.

Постійне вдосконалення: Можливість постійного вдосконалення та оптимізації алгоритмів на основі зібраних даних та зворотного зв'язку від користувачів.

1.4 План розробки рекомендаційної системи для мережі продуктових супермаркетів

1. Визначення цілей і вимог.
2. Інтеграція даних.
3. Збір даних про поведінку клієнтів.
4. Оцінка якості даних.
5. Підготовка даних.
6. Вибір моделі та алгоритмів.
7. Навчання і тестування моделі.

Також, блок-схему плану зображено на рис. 1.1

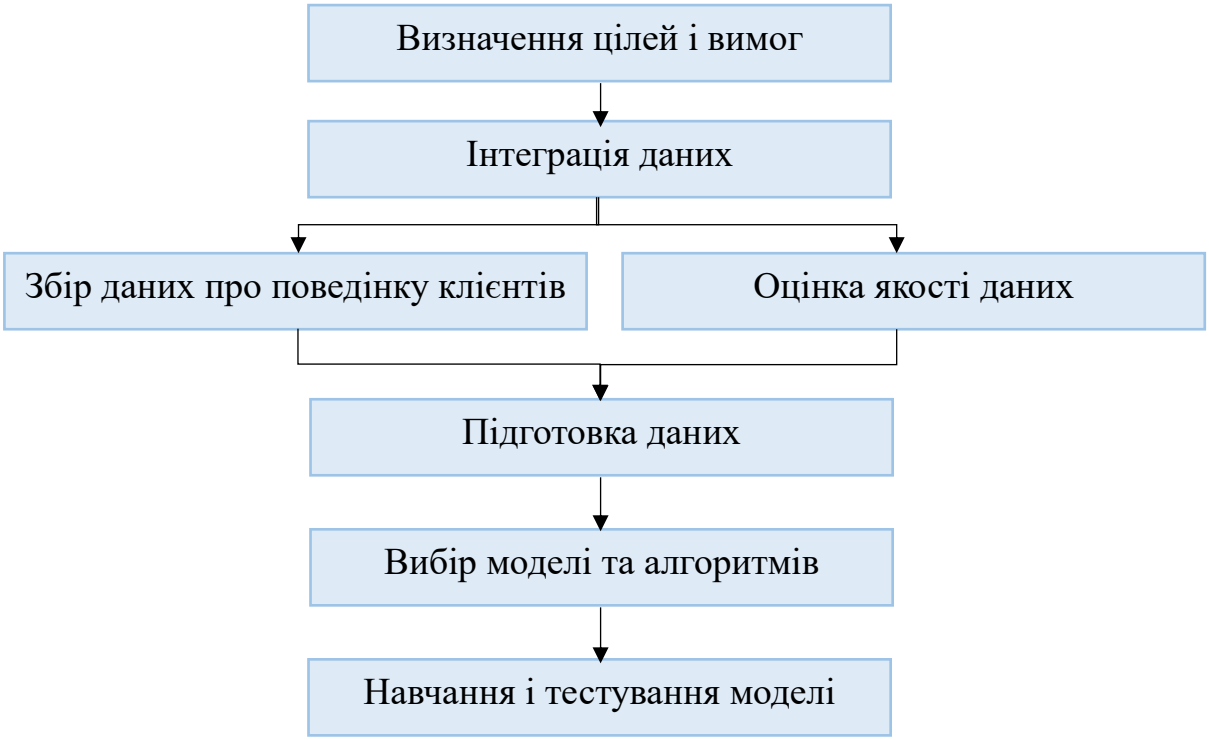

Рис. 1.1 Блок-схема розробки рекомендаційної системи

1.4.1 Визначення обсягу і типу даних

Один із ключових етапів при розробці рекомендаційної системи для мережі супермаркетів полягає у визначенні обсягу і типу даних, які будуть збиратися та використовуватися для побудови цієї системи. Цей етап є важливим, оскільки від нього залежать подальші кроки збору, обробки та аналізу даних. Давайте розглянемо цей процес докладніше.

Обсяг даних визначається об'ємом і розмахом інформації, яку необхідно зібрати та обробити для рекомендаційної системи. Він включає в себе кількість записів, а також кількість ознак або характеристик кожного запису. Наприклад, це можуть бути дані про тисячі або мільйони покупок клієнтів у різних магазинах, інформація про сотні тисяч товарів, а також демографічні дані та активність користувачів в онлайн-середовищі. Обсяг даних потрібно визначити на етапі планування проекту, оскільки це впливає на вимоги до інфраструктури для зберігання і обробки даних.

Типи даних визначають, яку саме інформацію необхідно зібрати. Для супермаркетів типи даних можуть бути різноманітними, включаючи:

1. Дані про покупки клієнтів: Це може бути інформація про товари, які купують клієнти, дату покупки, кількість і вартість товарів.
2. Демографічні дані клієнтів: Інформація про вік, стать, освіту, сімейний стан тощо.
3. Інформація про продукти: Включає назву товару, категорію, опис, ціну, характеристики.
4. Акції та знижки: Дані про акції та знижки, які проводяться в магазинах.

Важливо визначити, які саме дані є необхідними для побудови рекомендацій, оскільки це допомагає зрозуміти, які інформаційні ресурси необхідно зібрати та які типи аналізу будуть застосовуватися для надання рекомендацій клієнтам супермаркету.[1]

Таким чином, визначення обсягу і типу даних є важливим кроком у плануванні та розробці рекомендаційної системи для мережі супермаркетів, оскільки від цього залежать подальші кроки збору та обробки даних, а також успішність системи в наданні рекомендацій користувачам.

1.4.2 Джерела даних

При розробці рекомендаційної системи для мережі супермаркетів важливо визначити джерела даних, з яких буде здійснюватися збір інформації. Вибір джерел даних має вирішальне значення, оскільки від цього залежить якість та обсяг доступної інформації. Давайте детальніше розглянемо цей процес.

Бази даних магазинів: Одним з основних джерел даних можуть бути внутрішні бази даних магазинів мережі. Ці дані можуть включати інформацію про покупки клієнтів, наявність товарів, акції та знижки. Для отримання доступу до цих даних необхідно встановити зв'язок з відповідними департаментами мережі та забезпечити згоду на використання даних.

Онлайн-платформи: Іншим джерелом можуть бути онлайн-платформи, де клієнти роблять покупки або здійснюють інтеракцію з продуктами. Наприклад, сайт мережі супермаркетів або мобільний додаток. Для збору даних з цих джерел, може бути використано API (інтерфейси програмування додатків) або інші методи звернення до даних.

Анкети та опитування: Для збору демографічних даних та вподобань клієнтів можна проводити анкетування або опитування. Це джерело даних може бути корисним для розуміння потреб і смаків клієнтів.

Зовнішні джерела: Додаткові дані можна отримати з зовнішніх джерел, таких як бази даних виробників товарів, статистичні дані про споживання продуктів, географічні дані тощо.

Соціальні медіа: Інформація про активність клієнтів у соціальних мережах може бути використана для розуміння їхніх інтересів і вподобань. Аналіз публікацій та взаємодії може надати цінну інформацію.

Системи відстеження клієнтів в магазинах: Деякі магазини використовують системи відстеження клієнтів, такі як RFID-технології, для визначення маршрутів покупців та їхньої активності у магазині. Ці дані можуть бути використані для аналізу поведінки клієнтів.

Дані з бонусних програм та карток лояльності: Інформація з бонусних програм і карток лояльності магазинів містить важливі дані про покупки і вподобання клієнтів.

В якості основного джерела даних у Кваліфікаційній роботі буде обрана база POS системи магазина.[4]

1.4.3 Інтеграція даних з використанням MySQL

Інтеграція даних - це процес об'єднання даних з різних джерел в єдину структуру для подальшого аналізу, обробки та використання. MySQL є однією з найпопулярніших систем керування базами даних (СКБД), і вона широко використовується для зберігання та управління даними. Інтеграція даних з використанням MySQL включає в себе наступні кроки:

Збір даних з різних джерел: Перший крок у процесі інтеграції даних - це збір даних з різних джерел, таких як інші бази даних, текстові файли, API, інші додатки тощо. Ці джерела даних можуть бути розташовані на різних серверах або в різних форматах.

Створення структури бази даних MySQL: Після збору даних необхідно створити структуру бази даних MySQL, яка відповідає потребам проекту. Це включає в себе створення таблиць, визначення полів та їх типів даних, індексів тощо.

Завантаження даних в MySQL: Після створення структури бази даних необхідно завантажити дані з різних джерел в MySQL. Це може бути зроблено за допомогою інструментів для імпорту даних, таких як LOAD DATA INFILE, INSERT INTO, або використовуючи програмні бібліотеки або скрипти.

Трансформація даних: Після завантаження даних може виникнути потреба в їхній трансформації. Це включає в себе операції перетворення,

об'єднання, відфільтрування та інші маніпуляції з даними, щоб вони відповідали вимогам проекту.

Автоматизація процесу інтеграції: Для ефективної інтеграції даних можна використовувати автоматизовані інструменти, які допомагають у виконанні рутинних операцій, таких як імпорт та трансформація даних.

Оновлення та синхронізація: Для забезпечення актуальності даних може бути потрібно регулярно оновлювати та синхронізувати інформацію з джерелами.

Забезпечення безпеки та доступу: Важливо враховувати питання безпеки та доступу до інтегрованих даних, зокрема встановити обмеження доступу до бази даних і шифрування даних при необхідності.

Інтеграція даних з використанням MySQL дозволяє створити єдиний та цілісний набір даних, який може бути використаний для подальшого аналізу та розробки рекомендаційних систем, забезпечуючи доступність та надійність даних.

1.4.4 Інтеграція даних з використанням Google Cloud Dataflow

Google Cloud Dataflow - це хмарний сервіс для обробки даних в режимі реального часу та по партіях. Він надає високорівневий API для створення та запуску потокової та пакетної обробки даних. Давайте розглянемо процес інтеграції даних на прикладі Google Cloud Dataflow:

1. Підготовка даних:

- Вихідні дані можуть знаходитися в різних джерелах, таких як Google Cloud Storage, BigQuery, Cloud Pub/Sub або зовнішні джерела даних.
- Дані повинні бути попередньо оброблені та підготовлені для обробки в Google Cloud Dataflow.

2. Написання програми на Apache Beam:

- Google Cloud Dataflow базується на Apache Beam, відкритому коді для написання універсальних пайплайнів обробки даних.

- Ви створюєте програму на мові програмування Apache Beam (Java, Python), щоб описати свій пайплайн обробки даних.

3. Визначення пайплайну:

- Ваша програма на Apache Beam визначає структуру пайплайну, включаючи джерела даних, перетворення та кінцеві призначення.

- Програма описує, як дані мають бути оброблені, трансформовані та куди направляти результати.

4. Запуск пайплайну:

- Ви передаєте вашу програму на виконання в Google Cloud Dataflow.

- Сервіс автоматично керує виділенням ресурсів та розподілом завдань для обробки даних в хмарі.

5. Використання перетворень Apache Beam:

- Apache Beam надає різні перетворення для обробки даних, такі як фільтрація, групування, об'єднання та інші.

- Ви можете використовувати ці перетворення для маніпулювання та обробки даних у вашому пайплайні.

6. Моніторинг та відладка:

- Під час виконання пайплайну ви можете використовувати інструменти моніторингу Google Cloud Platform для відстеження продуктивності та відладки.

- Ви можете моніторити прогрес виконання, переглядати журнали та аналізувати продуктивність вашого пайплайну.

7. Збереження результатів:

- Результати обробки можуть бути збережені в цільові сховища даних, такі як BigQuery, Google Cloud Storage або інші сховища даних у хмарі.

8. Автомасштабування**:**

- Google Cloud Dataflow автоматично масштабується залежно від обсягу оброблюваних даних, забезпечуючи ефективне використання ресурсів.

9. Зупинка та масштабування:

- Після завершення обробки даних ви можете зупинити виконання пайплайну та масштабувати ресурси відповідно до вимог вашого додатка.

10. Моніторинг та оптимізація:

- Після завершення пайплайну ви можете проаналізувати моніторингові дані, щоб оптимізувати продуктивність та витрати ресурсів у майбутньому.

Google Cloud Dataflow забезпечує зручні засоби для створення, масштабування та моніторингу вашого пайплайну обробки даних в хмарі, роблячи процес інтеграції даних більш керованим та ефективним.

1.4.5 Business Intelligence (BI) системи для бізнес аналізу даних

Системи BI є програмними та апаратними засобами, призначеними для збору, аналізу, перетворення та відображення даних з метою підтримки прийняття рішень в організації. Ці системи об'єднують різноманітні технології, методи та інструменти для роботи з даними та перетворення їх у цінну інформацію для бізнесу.

Системи BI інтегруються з різними джерелами даних, такими як бази даних, електронні таблиці, текстові документи, зовнішні API та інше. Це може включати як внутрішні дані організації, так і зовнішні дані про ринок та конкурентів.

Системи BI надають інструменти для проведення аналізу даних. Це включає створення звітів, дашбордів, а також використання методів статистики та машинного навчання для виявлення закономірностей та тенденцій.

Системи BI надають засоби візуалізації даних, такі як графіки, діаграми, карти і т.д. Це допомагає легше інтерпретувати інформацію та виявляти важливі тенденції.

Системи BI дозволяють створювати звіти та дашборди для розподілу інформації співробітникам на всіх рівнях управління. Це забезпечує єдине розуміння бізнес-процесів та допомагає приймати обґрунтовані рішення.

В кінцевому підсумку мета систем BI - надати керівникам та співробітникам необхідну інформацію для прийняття обґрунтованих рішень, заснованих на даних і фактах.

Системи BI виявляються корисними для бізнесу з багатьох причин, таких як підвищення ефективності операцій, виявлення нових можливостей для зростання, зниження ризиків, а також покращення загальної прозорості та розуміння бізнес-процесів. В результаті вони є важливим інструментом для сучасних організацій, які прагнуть залишатися конкурентоспроможними та успішними на ринку.

1.4.6 Приклади BI систем

1. Tableau:

- Інтерактивні візуалізації: Tableau відомий своєю потужною системою візуалізації даних. Користувачі можуть легко створювати інтерактивні графіки, діаграми, карти та інші візуальні елементи.
- Підтримка багатьох джерел даних: Tableau інтегрується з різними джерелами даних, включаючи бази даних, хмари та файлові системи.
- Розширені можливості аналізу: Користувачі можуть проводити складний аналіз даних за допомогою функцій, таких як параметри, розрахункові поля і т.д.
- Tableau Server та Tableau Online: Tableau надає засоби для розміщення та обміну візуалізаціями через Tableau Server або в хмарі за допомогою Tableau Online.

2. Microsoft Power BI:

- Інтеграція з екосистемою Microsoft: Power BI тісно інтегрований з продуктами Microsoft, такими як Excel, Azure, SharePoint. Це забезпечує зручність використання для користувачів, знайомих з Microsoft.

- Можливості самообслуговування: Power BI надає можливості самообслуговування, що дозволяє кінцевим користувачам створювати звіти та дашборди без глибоких навичок програмування.

- Потужні аналітичні інструменти: Поміж візуалізації Power BI надає потужні інструменти аналітики, включаючи можливості машинного навчання та аналізу великих даних.

3. Looker:

- Централізоване управління даними: Looker надає єдиний джерело правди для даних, що полегшує централізоване управління і поліпшує консистентність.

- Асоціативна модель даних: Looker використовує асоціативну модель даних, що дозволяє користувачам досліджувати дані та відносини між ними більш гнучко.

- LookML: Мова LookML дозволяє визначати та налаштовувати дані, метадані та візуалізації, забезпечуючи гнучкість в налаштуванні.

4. IBM Cognos Analytics:

- Широкий спектр можливостей: Cognos Analytics надає багато функцій, включаючи створення звітів, дашбордів, а також можливості аналізу та прогнозування.

- Спільна робота та розподіл: Cognos дозволяє користувачам спільно працювати над проектами та розповсюджувати результати аналізу через різні канали.

- Інтеграція з різними джерелами: Cognos може інтегруватися з різними джерелами даних, включаючи великі дані, що забезпечує комплексний аналіз.

1.4.7 Підготовка даних до аналізу

Підготовка даних до аналізу - це важливий етап у роботі з даними, який передує їхньому аналізу та використанню. Цей етап включає в себе ряд операцій та методів для очищення, трансформації та підготовки даних для подальшого використання у різних аналітичних задачах. Ось деякі з основних методів підготовки даних:

Очищення даних:

1. Видалення дублікатів: Видалення повторних записів з набору даних.

Обробка відсутніх значень:

1. Вирішення проблеми пропущених значень шляхом їх заповнення або видалення.

Виявлення та виправлення помилок:

1. Виявлення та корекція помилкових даних.

Нормалізація:

1. Приведення даних до стандартного формату або діапазону значень.
2. Кодування категоріальних даних:
3. Перетворення категоріальних даних у числовий формат.

Вибір ознак:

1. Вибір та розгляд лише значущих ознак для аналізу.
2. Інтеграція даних:
3. Об'єднання даних з різних джерел в єдиний набір.
4. З'єднання таблиць та даних для аналізу.

Розбиття даних:

1. Розділення набору даних на тренувальну та тестову вибірки для машинного навчання.
2. Розбиття на групи для агрегації та аналізу окремих підгруп.
3. Видалення аномалій:

4. Виявлення та обробка викидів у даних, що можуть впливати на результати аналізу.

Інженерія ознак:

1. Створення нових ознак на основі існуючих для покращення якості моделі або аналізу.

2. Витягнення важливих характеристик з даних, таких як екстракція текстової інформації або обробка зображень.

3. Масштабування даних:

4. Нормалізація масштабу даних для забезпечення рівності ваги різних ознак.

Візуалізація даних:

1. Використання графіків та візуалізації для розуміння розподілу даних та виявлення закономірностей.

2. Забезпечення безпеки даних.

3. Захист конфіденційної інформації.

4. Контроль доступу до даних.

Документація:

Збереження документації про операції та методи, застосовані під час підготовки даних.

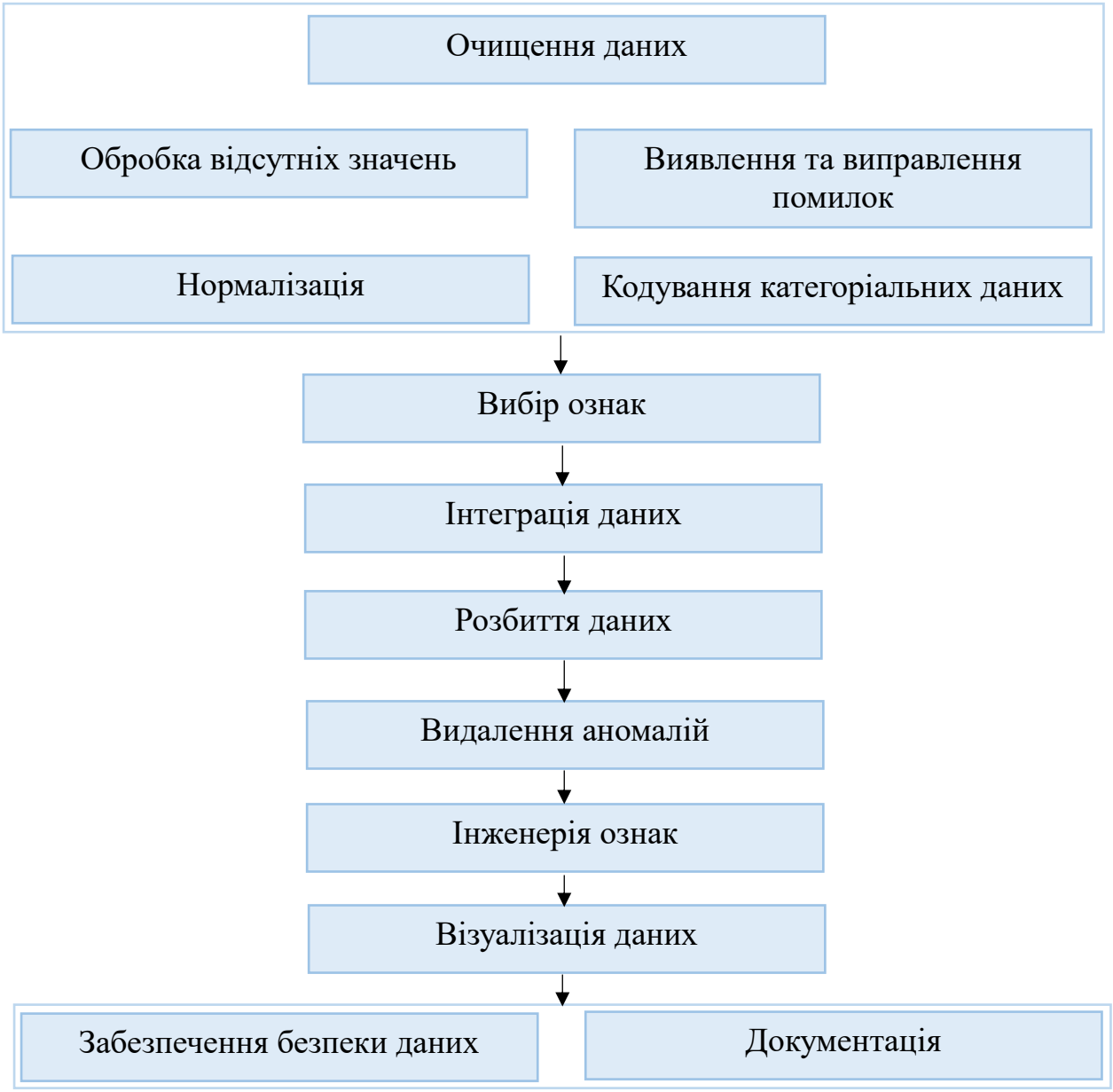

Рис. 1.2 Блок-схема кроків підготовки даних

Підготовка даних є важливим кроком у аналізі даних та розробці моделей машинного навчання, оскільки від цього залежить точність та надійність результатів. Добре підготовлені дані допомагають виявити закономірності, отримати цінну інформацію та роблять можливим побудову дієвих моделей та аналітичних звітів.[4]

1.4.8 Вибір моделі та алгоритмів для рекомендаційної системи

Колаборативний метод (алгоритм ближніх сусідів). Для рекомендацій на основі спільних користувачів або товарів можна використовувати алгоритм ближніх сусідів. Даний метод аналізує схожість між користувачами або товарами на основі їхньої історії покупок або оцінок. Це дозволяє рекомендувати товари, які сподобалися іншим користувачам зі схожими смаками або інтересами.

"**item-to-item**" рекомендації - рекомендуються схожі товари на основі історії покупок користувача. Цей метод аналізує спільність покупок користувача та інших користувачів і рекомендує товари, які схожі на ті, які вже були придбані користувачем.

Матрична розкладання (SVD) - може бути використана для скорочення розмірності та виявлення патернів у великих наборах даних. Це може допомогти зменшити обсяг обчислень та покращити ефективність рекомендаційної системи.

Комбінування методів в рекомендаційних системах використовують комбінацію різних методів для покращення якості рекомендацій. Наприклад, можна об'єднати колаборативний метод з "item-to-item" підходом або додати SVD для матричного розкладання.

Оптимізація та налаштування. Параметри моделей можуть бути оптимізовані для покращення результатів. Забезпечення швидкості та масштабованості системи для великої кількості користувачів та товарів.

1.4.9 Навчання моделі в рекомендаційній системі

Розділення даних на тренувальний і тестовий набори - для оцінки якості моделі розділіть дані на тренувальний і тестовий набори. Зазвичай використовуються відсотки, наприклад, 80% даних для навчання і 20% для тестування.

Навчання моделі. Тренувальний набір даних використовується для навчання моделі. Під час навчання модель вивчає закономірності в даних і створює прогнози для рекомендацій.

Оцінка моделі. Після навчання моделі тестовий набір використовується для оцінки її якості. Метрики оцінки: середньоквадратична помилка (RMSE), точність, відгук.

Оптимізація моделі. Вносити зміни у її параметри або вибір альтернативних алгоритмів, щоб покращити її результати.[3]

1.5 Нові обчислюванні технології. Хмарні сервиси

Хмарні сервіси відіграють важливу роль у розвитку та використанні рекомендаційних систем. Завдяки хмарним технологіям, рекомендаційні системи стають більш доступними, гнучкими та ефективними. Нижче наведено докладний огляд того, як хмарні сервіси впливають на рекомендаційні системи.

Легкість впровадження та масштабування

Хмарні сервіси дозволяють швидко розгортати рекомендаційні системи без необхідності великих капіталовкладень у власну інфраструктуру. Це забезпечує високу гнучкість у масштабуванні ресурсів відповідно до потреб бізнесу, що особливо важливо для обробки пікових навантажень або швидкого зростання обсягів даних.

Обробка великих даних

Рекомендаційні системи часто вимагають обробки великих обсягів даних для генерації точних рекомендацій. Хмарні сервіси надають необхідні засоби для ефективного зберігання та обробки великих даних, включаючи машинне навчання та штучний інтелект.

Гнучкість та інтеграція

Хмарні рішення часто пропонують різноманітні інструменти та сервіси, які можуть бути легко інтегровані з рекомендаційними системами. Це включає інтеграцію з різними джерелами даних, аналітичними інструментами, CRM-системами та іншими бізнес-застосуваннями.

Доступність та надійність

Хмарні сервіси забезпечують високу доступність та надійність інфраструктури, що є критично важливим для рекомендаційних систем, які повинні бути доступні 24/7. Хмарні провайдери зазвичай пропонують гарантії рівня сервісу, які забезпечують стабільність та надійність системи.

Безпека даних

Хмарні платформи надають розширені можливості забезпечення безпеки даних, включаючи шифрування даних, управління доступом, резервне копіювання та відновлення даних. Це особливо важливо для рекомендаційних систем, які обробляють великі обсяги персональної інформації.

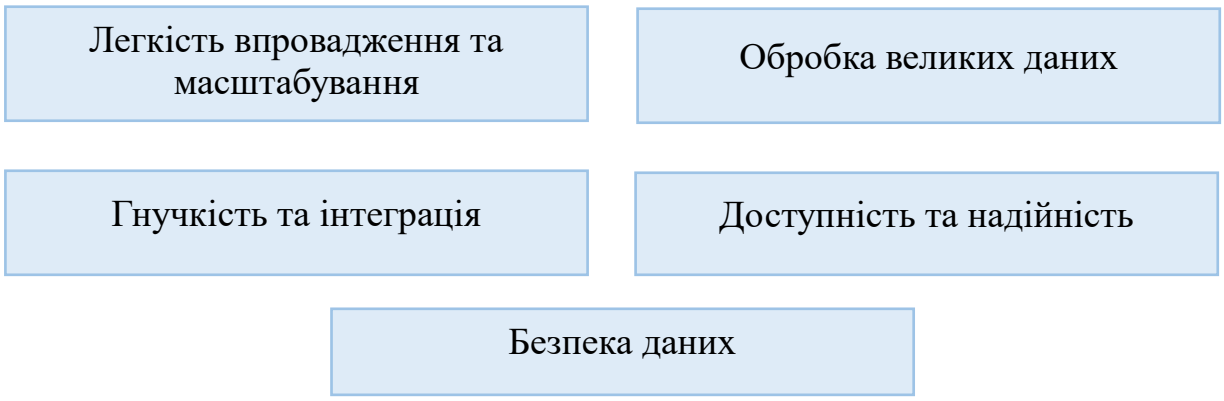

Рис. 1.3 Переваги хмарних сервісів

1.5.1 Приклади Хмарних сервісів

Хмарні сервіси пропонують широкий спектр рішень, які можуть бути використані для різних бізнес-потреб, включаючи розробку та впровадження рекомендаційних систем. Ось кілька прикладів хмарних сервісів, які часто використовуються в цій сфері

1. Amazon Web Services (AWS)

Опис: AWS є одним із провідних хмарних сервісів, який пропонує широкий спектр інструментів та послуг для розробки, розгортання та управління програмами та інфраструктурою.

Приклади застосування:

Amazon S3: Послуга зберігання даних, ідеально підходить для масштабованих рішень зберігання.

Amazon EC2: Віртуальні сервери, які дозволяють масштабувати обчислювальні ресурси.

AWS Lambda: Послуга безсерверного виконання коду, дозволяє запускати код відповідно до подій, таких як оновлення баз даних або зміни у веб-застосунках.

Amazon Machine Learning: Набір інструментів і сервісів для розробки моделей машинного навчання.

2. Microsoft Azure

Опис: Azure від Microsoft надає комплексний набір хмарних послуг, що включають інструменти для обчислень, аналітики, зберігання та мережі.

Приклади застосування:

Azure Virtual Machines: Віртуальні машини для розгортання та управління застосунками.

Azure SQL Database: Хмарна база даних, що підтримує різноманітні мови запитів.

Azure Machine Learning Studio: Інтерактивне середовище для розробки, тестування та розгортання моделей машинного навчання.

Azure Blob Storage: Об'єктне зберігання для зберігання великих обсягів неструктурованих даних.

3. Google Cloud Platform (GCP)

Опис: GCP пропонує набір хмарних послуг, які дозволяють розгортати, будувати та оптимізувати застосунки та послуги.

Приклади застосування:

Google Compute Engine: Сервіс віртуальних машин з гнучкими налаштуваннями.

Google Cloud Storage: Ефективне зберігання даних для різноманітних потреб.

BigQuery: Сервіс для обробки та аналізу великих обсягів даних.

Google Cloud ML Engine: Платформа для побудови та тренування моделей машинного навчання.

4. IBM Cloud

Опис: IBM Cloud включає IaaS, PaaS, та SaaS пропозиції, орієнтовані на бізнес-рішення, зокрема для розробки інтелектуальних систем.

1.5.2 Приклади застосування хмарних сервісів

IBM Watson: Штучний інтелект для бізнесу, що допомагає у аналізі даних та підвищенні ефективності.

IBM Cloud Databases: Різноманітні опції для баз даних, включаючи PostgreSQL, MongoDB та інші.

Переваги використання хмарних сервісів

Гнучкість та масштабованість: Можливість швидко масштабувати ресурси відповідно до змінних потреб бізнесу.

Зниження витрат: Зменшення витрат на інфраструктуру та обслуговування завдяки моделям оплати за фактичне використання.

Швидкість інновацій: Хмарні сервіси дозволяють швидко впроваджувати нові технології та інновації.

Фокус на основному бізнесі: Звільнення ресурсів та уваги для зосередження на ключових бізнес-задачах, а не на управлінні IT-інфраструктурою.

Використання хмарних сервісів для рекомендаційних систем дозволяє компаніям бути більш гнучкими, інноваційними та ефективними у своїх підходах до персоналізації взаємодії з клієнтами.

1.6 Рекомендаційні моделі в сучасному рітейлі

1.6.1 Колаборативний метод

Колаборативний метод в рекомендательних системах - це підхід, що базується на використанні інформації про вподобання або поведінку користувачів для створення персоналізованих рекомендацій. Основна ідея полягає в тому, що якщо користувач А схожий на користувача В у своїх вподобаннях або діях, то має сенс запропонувати користувачеві А те, що сподобалося користувачеві В.

Існують два основних види колаборативних методів:

1. User-Based Collaborative Filtering (UBCF): У цьому методі рекомендації створюються на основі схожості між користувачами. Якщо користувач А і користувач В мають схожі вподобання або історії покупок, система припускає, що їм сподобаються схожі елементи. Для обчислення схожості часто використовуються метрики, такі як косинусна схожість або кореляція.

Кроки реалізації:

Побудова матриці уподобань:

- Створення матриці, де рядки представляють користувачів, а стовпці – елементи.

- Заповнення матриці значеннями, що відображають уподобання користувачів до елементів.

Вимірювання схожості:

- Обчислення схожості між користувачами за допомогою різних метрик (косинусна схожість)

- Чим вище схожість між користувачами, тим ближчі їхні уподобання.

Розрахунок рекомендацій:

- Для конкретного користувача знаходяться найбільш схожі користувачі.

- Елементи, які сподобалися цим схожим користувачам, можуть бути рекомендовані поточному користувачеві.

2. Item-Based Collaborative Filtering (IBCF): Цей метод використовує інформацію про схожість самостійно елементів (товарів). Якщо користувачу A сподобався елемент X, то може бути запропоновано користувачеві A елемент Y, схожий з X. Для вимірювання схожості елементів також використовуються косинусна схожість, кореляція та інші методи.

Кроки реалізації:

Побудова матриці уподобань:

- Як і в UBCF, будується матриця уподобань користувачів і елементів.

Вимірювання схожості елементів:

- Обчислення схожості між елементами (товарами).

- Чим ближче елементи, тим більше ймовірність того, що, якщо користувач позитивно відреагував на один елемент, йому сподобається і інший.

Розрахунок рекомендацій:

- Для конкретного елемента знаходяться найбільш схожі елементи.

- Елементи, схожі на ті, що користувачу вже сподобались, можуть бути рекомендовані йому.

Переваги та недоліки колаборативного методу:

Переваги:

- Персоналізація: Забезпечують персоналізовані рекомендації на основі уподобань користувача.
- Виявлення патернів: Можуть виявляти приховані патерни у поведінці користувачів.

Недоліки:

- Проблема холодного старту: Важко надати рекомендації для нових користувачів, у яких немає історії взаємодії.
- Проблема розрідженості даних: Матриця уподобань може бути дуже розрідженою, особливо якщо множина користувачів і елементів велика.

1.6.2 ALS метод

Постановка задачі:

- Ми маємо матрицю вподобань R, де рядки представляють користувачів, а стовпці - елементи (зазвичай товари).
- Матриця R має багато пропущених значень, оскільки кожен користувач не взаємодіє з кожним елементом.

Модель ALS:

- ALS намагається розкласти матрицю R на добуток двох менших матриць: матриці користувачів U і матриці елементів V.
- При цьому $R \approx \mathrm{U} \cdot V^T$

Оптимізація:

- Метод використовує ітеративний процес оптимізації, чергуючи оптимізацію U і V.

- На кожному кроці вирішується оптимізаційна задача для U або V, у якій мінімізується відхилення між R і $\mathrm{U} \cdot V^T$

- Процес триває до зближення до оптимальних значень U і V.

Рекомендації:

- Після навчання матриць U і V можна використовувати їх для прогнозування пропущених значень в матриці R.

- Рекомендації генеруються на основі оцінок відповідних відносин між користувачами і елементами.

1.6.3 Метод однорукого бандита

Метод однорукого бандита (multi-armed bandit) - це алгоритм прийняття рішень, який виникає в контексті теорії ігор та теорії прийняття рішень. Його назва вказує на аналогію з ігровим автоматом (бандитом), у якого є багато рукояток (ручок або дій), кожна з яких може викликати певний результат. Гравець повинен визначити оптимальну стратегію вибору рукояток, щоб максимізувати свій загальний виграш.

Задача:

- Є кілька альтернатив (дій або рукояток), і кожна альтернатива може призводити до різних виграшів або втрат.

- Мета - визначити таку стратегію вибору альтернатив, яка максимізує очікуваний виграш протягом деякого часу.

Експлорація та експлуатація:

- Експлорація: Спроби нових альтернатив для отримання більше інформації про їхній потенційний виграш.

- Експлуатація: Вибір альтернативи на основі поточних знань для максимізації виграшу.

Ймовірність вибору:

- Кожна альтернатива має ймовірність вибору, яку визначає алгоритм на основі знань, накопичених під час експлорації.

Оновлення знань:

- Після кожного вибору альтернативи оновлюються знання про її виграш або втрати.

- Методи оновлення можуть включати в себе експоненційне згорткове вивчення, інформаційний критерій Акаіке та інші.

Типи методів однорукого бандита:

1. Жадібні методи:

- Вибір альтернативи, яка має найвищий поточний виграш.

- Може призводити до сильної експлуатації та ігнорування експлорації.

2. Експлораційно-експлуатаційні методи:

- Забезпечують баланс між експлорацією та експлуатацією.

- Наприклад, "євристичне згорткове вивчення" може випадковим чином вибирати альтернативи з ймовірностями, пропорційними їхнім очікуваним виграшам.

3. Байєсівські методи:

- Використовують байєсівське вивчення для оцінки ймовірностей виграшу альтернатив.

- Ефективно враховують нестаціонарність у часі.

Переваги методу однорукого бандита:

1. Простота і розуміння: Метод є досить простим у використанні та розумінні, що полегшує його впровадження в практиці.

2. Адаптивність: Метод може швидко адаптуватися до змінних умов і невизначеності, забезпечуючи гнучкість в прийнятті рішень.

3. Відсутність потреби у великій кількості даних: Для початкових рішень не потрібно мати обширну інформацію або велику історію взаємодії. Алгоритм може почати експериментувати з різними альтернативами відразу.

4. Можливість збільшення виграшу: Якщо система добре збалансована між експлорацією і експлуатацією, метод може привести до максимізації виграшу в умовах невизначеності.

Недоліки методу однорукого бандита:

1. Ризик сильної експлуатації: Якщо не забезпечити достатньої експлорації, може виникнути ризик сильної експлуатації певних альтернатив, що може призвести до упущених можливостей в оптимізації.

2. Неефективність у стаціонарних умовах: В умовах, коли умови майже не змінюються, інші більш складні стратегії можуть працювати ефективніше, а метод однорукого бандита може бути менш оптимальним.

3. Не враховує контекст: Метод не завжди враховує контекст або залежність між діями, що може призвести до менш точних рішень.

4. Потреба в налагодженні параметрів: В деяких випадках може виникнути необхідність в налагодженні параметрів методу для досягнення оптимальної продуктивності.

1.6.4 RFM – аналіз

RFM аналіз (Recency, Frequency, Monetary) є метод сегментації клієнтів на основі їх поведінки в трьох ключових аспектах:

Recency (Останній раз): Оцінює час від останньої взаємодії клієнта з продуктом або послугою. Цей параметр передбачає, що більш свіжі дані більш актуальні та точно відображають поточні уподобання та потреби клієнта.

Frequency (Частота): Визначає, наскільки часто клієнт взаємодіє з продуктом або здійснює покупки. Часті взаємодії можуть свідчити про більший рівень задоволеності продуктом або те, що клієнт є постійним користувачем.

Monetary (Грошові кошти): Оцінює загальний обсяг коштів, які клієнт витратив на продукти чи послуги. Цей аспект враховує фінансовий внесок клієнта в бізнес.

Застосування RFM аналізу в системі рекомендацій включає наступні кроки:

1. Збір даних: Спочатку необхідно зібрати дані про взаємодію клієнтів із продуктом чи послугою. Це можуть бути дані про покупки, візити на сайт, час останньої взаємодії тощо.

2. Ранжування по RFM: Кожен клієнт ранжується за трьома аспектами RFM аналізу, присвоюючи їм відповідні оцінки чи категорії. Наприклад, клієнти можуть бути розділені на категорії від "1" до "5" за кожним із трьох параметрів.

3. Сегментація клієнтів: Клієнти групуються в різні сегменти на основі їх комбінованих RFM значень. Наприклад, одні клієнти можуть потрапити до сегменту "Кращий" з високими значеннями за всіма трьома

параметрами, тоді як інші можуть бути в сегменті "сплячих" клієнтів з низьким значенням по якомусь із параметрів.

4. Рекомендації для сегментів: Для кожного сегмента можуть бути розроблені стратегії рекомендацій, що індивідуалізуються. Наприклад, для сегмента з високим Monetary, але низьким Frequency можна запропонувати рекомендації, спрямовані на збільшення частоти покупок.

5. Персоналізація рекомендацій: На основі результатів аналізу клієнтам пропонуються персоналізовані рекомендації, які відповідають їх RFM сегменту.

Припустимо, що значення Recency, Frequency та Monetary оцінюються від 1 до 5, де 5 – найвищий рівень, а 1 – найменший рівень. Кожному клієнту надається оцінка за кожним із трьох параметрів.

Таблиця 1.1

Приклад RFM - сегментації

Клієнт	Recency	Frequency	Monetary	Сегмент
-1-	**-2-**	**-3-**	**-4-**	**-5-**
1442	4	5	4	Кращий
5768	2	3	2	Сплячий
0953	5	2	1	Поточний
9576	3	4	3	зацікавлений
3765	1	1	5	Перспективный

Ці сегменти можуть використовуватися для розробки персоналізованих стратегій, таких як рекомендації продуктів, програм лояльності чи маркетингові кампанії, спрямовані на конкретні групи клієнтів.

RFM аналіз дозволяє більш ефективно адаптувати маркетингові стратегії та рекомендації під індивідуальні потреби та характеристики клієнтів, що в результаті сприяє покращенню користувальницького досвіду та збільшенню ефективності продажів.

1.7 Висновок

У інформаційно-аналітичному розділі мною були розглянуті задачі, які вирішують за допомогою рекомендаційних систем. Представлені, технології та методи, які використовують при розроблені рекомендаційних систем. Були розглянуті, шляхи обчислення даних для рекомендаційних систем, так як, для кращого аналізу потрібно багато даних, які стає все складніше обчислювати на локальних машинах.

2 СПЕЦІАЛЬНИЙ РОЗДІЛ

2.1 Опис даних для рекамендоційної системи

2.1.1 Опис контексту даних

Цей набір даних Online Retail містить усі транзакції, здійснені в магазині роздрібної торгівлі, що базується в Великобританії. Датасет містить дані за період з 01-12-2020 по 09-12-2022. Компанія в основному продає унікальні подарункові вироби на всі випадки життя. Багато клієнтів компанії - оптовики.

2.1.2 Інформація про атрибути даних

InvoiceNo - Номер накладної, Іменний, 6-значний інтегральний номер, який унікально присвоюється кожній транзакції.

StockCode - Код товару (позиції), Іменний, 5-значний інтегральний номер, унікально призначений кожному окремому продукту.

Description - Назва товару (позиції), Іменний.

Quantity - Кількість кожного продукту, для кожної транзакції у замовлені, Числовий.

InvoiceDate - Дата та час реєстрації замовлення. День і час створення транзакції.

UntilPrice - Ціна за одиницю, Числовий. Ціна продукту за одиницю в грошовому еквіваленті.

CustomerID – Унікальній номер клієнта, Числовий, 5-значний номер, який унікально присвоюється кожному клієнту.

Country - Назва країни. Іменний. Назва країни, де проживає клієнт.

2.2 Вирішенні задачі методом RFM – аналізу

2.2.1 Імпорт необхідних бібліотек

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
import random
from sklearn.cluster import KMeans
from sklearn.metrics.pairwise import cosine_similarity
```

2.2.2 Завантаження даних

```
data = pd.read_csv("Online_Retail.csv")
data.head(10)
```

Таблиця 2.1

Огляд даних магазину

	Invoice No	Stock Code	Description	Quantity	Invoice Date	Unit Price	CustomerID	Country
-1-	**-2-**	**-3-**	**-4-**	**-5-**	**-6-**	**-7-**	**-8-**	**-9-**
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2021 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2021 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2021 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2021 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2021 8:26	3.39	17850.0	United Kingdom

2.2.3 Огляд статистики даних

data.info()

Таблиця 2.2

Статистики за даними

Номер строки	Column	Non-Null Count	Dtype
-1-	**-2-**	**-3-**	**-4-**
0	InvoiceNo	541909	object
1	StockCode	541909	object
2	Description	540455	object
3	Quantity	541909	int64
4	InvoiceDate	541909	object
5	UnitPrice	541909	float64
6	CustomerID	406829	float64
7	Country	541909	object
dtypes: float64(2), int64(1), object(5)			

data.describe()

Таблиця 2.3

Базова статистика за числовими даними

Статистика	Quantity	UnitPrice	CustomerID
-1-	-2-	-3-	-4-
count	541909.000000	541909.000000	406829.000000
mean	9.552250	4.611114	15287.690570
std	218.081158	96.759853	1713.600303
min	-80995.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16791.000000
max	80995.000000	38970.000000	18287.000000

2.2.4 Видалення пропусків з стовбця 'CustomerID'

```
data = data[~data['CustomerID'].isna()]
data.info()
```

Таблиця 2.4

Статистики за даними

Номер строки	Column	Non-Null Count	Dtype
-1-	**-2-**	**-3-**	**-4-**
0	InvoiceNo	406829	object
1	StockCode	406829	object
2	Description	406829	object
3	Quantity	406829	int64
4	InvoiceDate	406829	object
5	UnitPrice	406829	float64
6	CustomerID	406829	float64
7	Country	406829	object
dtypes: float64(2), int64(1), object(5)			

2.2.5 Робота с даними формату дати

```
#Перетворення даних про дату у формат дати:
data['InvoiceDate'] = pd.to_datetime(data['InvoiceDate'])
#Перетворення у подрібний нам формат:
data['InvoiceDay'] = data['InvoiceDate'] \
.apply(lambda x: dt.datetime(x.year, x.month, x.day))
```

Таблиця 2.5

Попередній перегляд дат

	InvoiceDay
-1-	-2-
0	2021-12-01
1	2021-12-01
2	2021-12-01
3	2021-12-01
4	2021-12-01

2.2.6 Розраховуемо суму замовленя в розрізі конкретного замовлення

```
data['TotalSum'] = data.Quantity * data.UnitPrice.
```

2.2.7 Розробка RFM класифікації для клієнтів

```
last_date = max(data.InvoiceDay) + dt.timedelta(1)
rfm = data.groupby(by = ['CustomerID'] ).agg({
    'InvoiceDay': lambda x: (last_date - x.max()).days,
    'InvoiceNo': 'count',
    'TotalSum': 'sum'
})
```

2.2.8 Підготовка даних до кластеризації

```
r_labels = range(4, 0, -1) #[4, 3, 2, 1]
r_groups = pd.qcut(rfm['Recency'], q=4, labels=r_labels)
f_labels = range(1, 5) # [1, 2, 3, 4]
f_groups = pd.qcut(rfm['Frequency'], q=4, labels=f_labels)
m_labels = range(1, 5)
m_groups = pd.qcut(rfm['Monetary'], q=4, labels=m_labels)
rfm['R'] = r_groups.values
rfm['F'] = f_groups.values
rfm['M'] = m_groups.values
```

2.2.9 Кластерізація k-means

```
X = rfm[['R', 'F', 'M']]
kmeans = KMeans(n_clusters=10, init='k-means++', max_iter=300)
kmeans.fit(X)
rfm['kmeans_cluster'] = kmeans.labels_
rfm[rfm['kmeans_cluster'] == 0]
```

2.2.10 Візуалізація кластерів

```
# Кількість кластерів
num_clusters = 10
fig, axes = plt.subplots(num_clusters // 2, 2, figsize=(12, 20))
axes = axes.ravel()

# Цикс для побудови візуалізації кластерів
for cluster_id in range(num_clusters):
    cluster_data = rfm[rfm,kmeans_cluster == cluster_id]

    sns.scatterplot(data=cluster_data, x='Recency', y='Frequency',
hue='Monetary', palette='viridis', ax=axes[cluster_id])

    # Задання назви кластерів
    axes[cluster_id].set_title(f'Cluster {cluster_id}')
# Задання легенди
handles, labels = axes[0].get_legend_handles_labels()
fig.legend(handles, labels, loc='center right')

plt.tight_layout()
plt.show()
```

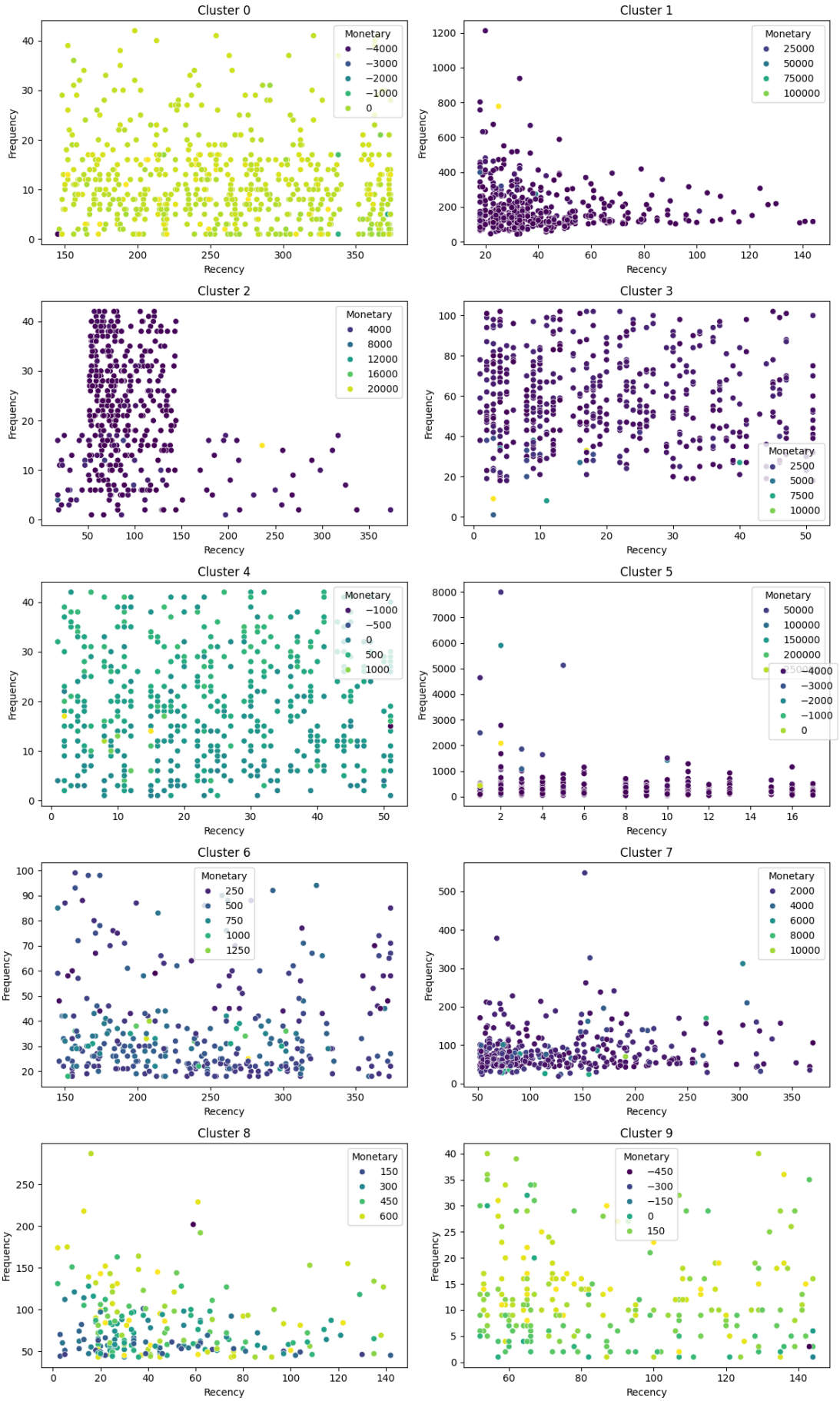

Рис. 2.1 Візулізація кластерів за покзником Recency від Fryquency

```
# Побудова гістограм для Recency
plt.figure(figsize=(12, 6))
for cluster_id in range(num_clusters):
    plt.subplot(2, 5, cluster_id + 1)
    sns.histplot(rfm[rfm['kmeans_cluster'] == cluster_id]['Recency'], bins=20, kde=True)
    plt.title(f'Cluster {cluster_id}')
    plt.xlabel('Recency')
    plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```

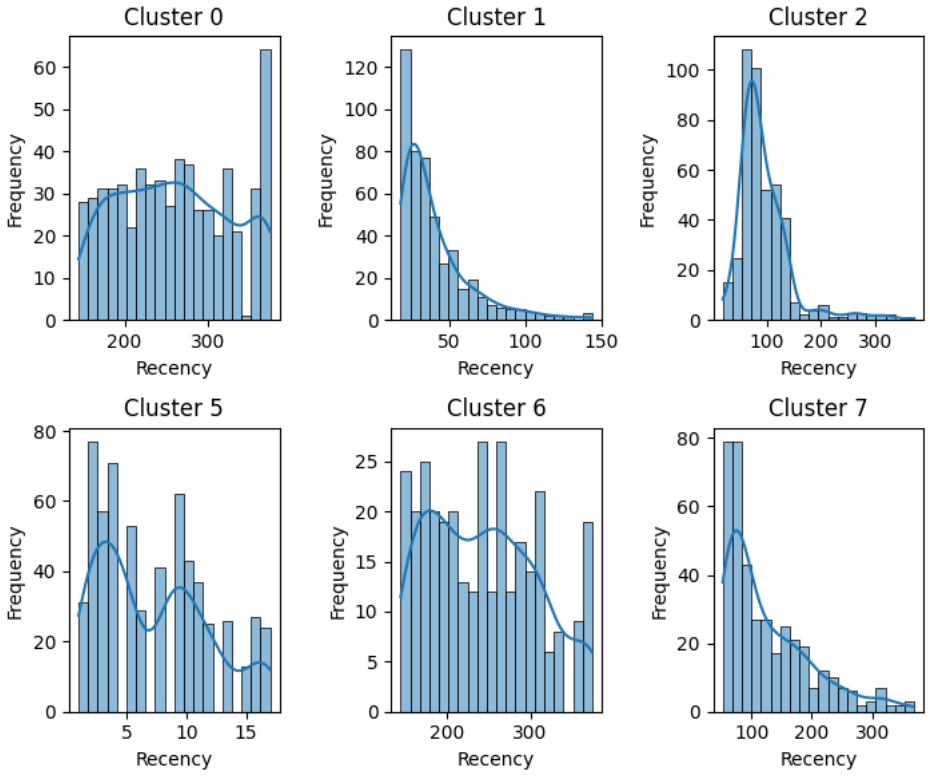

Рис. 2.2 Візулізація гістограм кластерів за покзником Recency від Fryquency

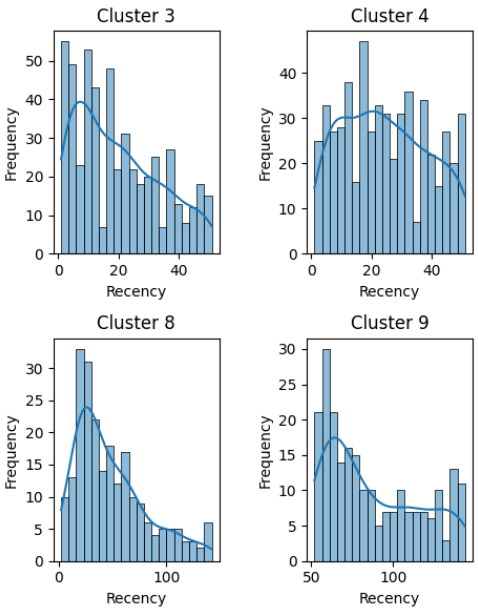

Рис. 2.3 Візулізація гістограм кластерів за покзником Recency від Fryquency

2.2.11 Рекомендація продуктів для кожного кластеру

```
num_clusters = 10
# Створення словника з рекомендаціями
cluster_recommendations = {}
# Цикл для проходження за кожним кластером
for cluster_id in range(num_clusters):
    # Пошук клієнтів за кожним кластером
    customers_in_cluster = rfm[rfm['kmeans_cluster'] == cluster_id].index

    # Пошук кращих продуктів в кожному кластері
    best_products =
data[data.CustomerID.isin(customers_in_cluster)].groupby(['StockCode'])['In
voiceNo'].count().sort_values(ascending=False).head(10)
```

```
    # Збереження кращих продуктів для кожного кластера
    cluster_recommendations[f'Кластер {cluster_id}'] = \
best_products.index.tolist()


# Відображення рекомендацій для кожного з кластерів
for cluster, recommended_products in cluster_recommendations.items():
    print(f"{cluster} :Кращі продукти: {recommended_products}")
```

Підібрані кращі продукти для кожного кластера:

Кластер 0 : Кращі продукти: ['85123A', '22423', '47566', '21034', '22457', 'POST', '22138', '22178', '22469', '84879']
Кластер 1 : Кращі продукти: ['22423', '85123A', '85099B', '20725', '84879', '47566', '22383', '23203', '23209', '22720']
Кластер 2 : Кращі продукти: ['85123A', 'POST', '22423', '47566', '85099B', '22960', '22138', '22720', '84879', '22086']
Кластер 3 : Кращі продукти: ['POST', '22423', '85123A', '23084', '22086', '84879', '47566', '22138', '23355', '22469']
Кластер 4 : Кращі продукти: ['84879', 'POST', '22086', '22138', '22423', '23084', '85123A', '84946', '21034', '23355']
Кластер5 : Кращі продукти: ['85123A', '85099B', '22423', '20725', '22197', '47566', '20727', '23203', '22383', '22720']
Кластер 6 : Кращі продукти: ['85123A', '22423', '47566', '84879', '21034', '22457', '22720', '22960', '21212', '22470']
Кластер 7 : Кращі продукти: ['85123A', '22423', '84879', '47566', '22720', '21212', '22960', '85099B', '22457', '20725']
Кластер 8 : Кращі продукти: ['21034', '85123A', '22086', '22197', '23321', '84879', '23103', '22557', '22139', '22469']
Кластер 9 : Кращі продукти: ['85123A', '22423', '84879', '23321', '84946', '23293', '85099B', '22487', '22961', 'POST']

2.2.12 Рекомендація продуктів для кожного клієнта

```
def customer_recommendations(num_clusters, num_customers, rfm, data):
    # Створення словника з рекомендаціями
    cluster_recommendations = {}

    # Цикл для проходження за кожним кластером
    for cluster in range(num_clusters):
        # Знаходить клієнтів у кластері
        customers_in_cluster = rfm[rfm.kmeans_cluster == cluster].index

        # Пошук кращих продуктів в кожному кластері
        best_products =
data[data.CustomerID.isin(customers_in_cluster)].groupby(['StockCode'])['In
voiceNo'].count().sort_values(ascending=False).head(10)

        # Клієнти, які не придбали кращі продукти
        not_buy_best_product = [customer for customer in customers_in_cluster
if not (data[(data.CustomerID == customer) &
(data.StockCode.isin(best_products.index.tolist()))]).empty]

        num_customers_to_display = min(num_customers,
len(not_buy_best_product))
```

```
        selected_customers = not_buy_best_product
[:num_customers_to_display]


        # Збереження кращих продуктів та обраних клієнтів для кожного
кластера
        cluster_recommendations[f'Кластер {cluster_id}'] = {
            'Обрані товари': best_products.index.tolist(),
            'Клієнти': selected_customers }

    return cluster_recommendations

num_clusters = 10
num_customers = 5

# Assuming you already have 'rfm' and 'df' dataframes
cluster_recommendations = customer_recommendations(num_clusters,
num_customers, rfm, data)

# Відображення рекомендацій клієнтів для кожного з кластерів
for cluster, recommendations_and_customers in
cluster_recommendations.items():
    print(f"{cluster} :")
    print("Рекомендовані продукти:")
    for customer_id in recommendations_and_customers['Клієнти']:
```

```
        print(f"Клієнт: {customer_id} : Рекомендовані продукти:
    {recommendations_and_customers['Обрані товари']}")

      print()
```

Кластер 0 :

Рекомендовані продукти:

Клієнт: 12350.0 : Рекомендовані продукти: ['85123A', '22423', '47566', '210 34', '22457', 'POST', '22138', '22178', '22469', '84879']

Клієнт: 12355.0 : Рекомендовані продукти: ['85123A', '22423', '47566', '210 34', '22457', 'POST', '22138', '22178', '22469', '84879']

Клієнт: 12361.0 : Рекомендовані продукти: ['85123A', '22423', '47566', '210 34', '22457', 'POST', '22138', '22178', '22469', '84879']

Клієнт: 12373.0 : Рекомендовані продукти: ['85123A', '22423', '47566', '210 34', '22457', 'POST', '22138', '22178', '22469', '84879']

Клієнт: 12401.0 : Рекомендовані продукти: ['85123A', '22423', '47566', '210 34', '22457', 'POST', '22138', '22178', '22469', '84879']

Кластер 1 :

Рекомендовані продукти:

Клієнт: 12349.0 : Рекомендовані продукти: ['22423', '85123A', '85099B', '2 0725', '84879', '47566', '22383', '23203', '23209', '22720']

Клієнт: 12356.0 : Рекомендовані продукти: ['22423', '85123A', '85099B', '2 0725', '84879', '47566', '22383', '23203', '23209', '22720']

Клієнт: 12357.0 : Рекомендовані продукти: ['22423', '85123A', '85099B', '2 0725', '84879', '47566', '22383', '23203', '23209', '22720']

Клієнт: 12370.0 : Рекомендовані продукти: ['22423', '85123A', '85099B', '2 0725', '84879', '47566', '22383', '23203', '23209', '22720']

Клієнт: 12371.0 : Рекомендовані продукти: ['22423', '85123A', '85099B', '2 0725', '84879', '47566', '22383', '23203', '23209', '22720']

Кластер 2 :

Рекомендовані продукти:

Клієнт: 12379.0 : Рекомендовані продукти: ['85123A', 'POST', '22423', '475 66', '85099B', '22960', '22138', '22720', '84879', '22086']

Клієнт: 12390.0 : Рекомендовані продукти: ['85123A', 'POST', '22423', '475 66', '85099B', '22960', '22138', '22720', '84879', '22086']

Клієнт: 12394.0 : Рекомендовані продукти: ['85123A', 'POST', '22423', '475 66', '85099B', '22960', '22138', '22720', '84879', '22086']

Клієнт: 12413.0 : Рекомендовані продукти: ['85123A', 'POST', '22423', '475 66', '85099B', '22960', '22138', '22720', '84879', '22086']

Клієнт: 12418.0 : Рекомендовані продукти: ['85123A', 'POST', '22423', '475 66', '85099B', '22960', '22138', '22720', '84879', '22086']

Кластер 3 :

Рекомендовані продукти:

Клієнт: 12352.0 : Рекомендовані продукти: ['POST', '22423', '85123A', '230 84', '22086', '84879', '47566', '22138', '23355', '22469']

Клієнт: 12358.0 : Рекомендовані продукти: ['POST', '22423', '85123A', '230 84', '22086', '84879', '47566', '22138', '23355', '22469']

Клієнт: 12364.0 : Рекомендовані продукти: ['POST', '22423', '85123A', '230 84', '22086', '84879', '47566', '22138', '23355', '22469']

Клієнт: 12374.0 : Рекомендовані продукти: ['POST', '22423', '85123A', '230 84', '22086', '84879', '47566', '22138', '23355', '22469']

Клієнт: 12421.0 : Рекомендовані продукти: ['POST', '22423', '85123A', '230 84', '22086', '84879', '47566', '22138', '23355', '22469']

Кластер 4 :

Рекомендовані продукти:

Клієнт: 12367.0 : Рекомендовані продукти: ['84879', 'POST', '22086', '2213 8', '22423', '23084', '85123A', '84946', '21034', '23355']

Клієнт: 12375.0 : Рекомендовані продукти: ['84879', 'POST', '22086', '22138', '22423', '23084', '85123A', '84946', '21034', '23355']

Клієнт: 12384.0 : Рекомендовані продукти: ['84879', 'POST', '22086', '22138', '22423', '23084', '85123A', '84946', '21034', '23355']

Клієнт: 12403.0 : Рекомендовані продукти: ['84879', 'POST', '22086', '22138', '22423', '23084', '85123A', '84946', '21034', '23355']

Клієнт: 12430.0 : Рекомендовані продукти: ['84879', 'POST', '22086', '22138', '22423', '23084', '85123A', '84946', '21034', '23355']

Кластер 5 :

Рекомендовані продукти:

Клієнт: 12347.0 : Рекомендовані продукти: ['85123A', '85099B', '22423', '20725', '22197', '47566', '20727', '23203', '22383', '22720']

Клієнт: 12359.0 : Рекомендовані продукти: ['85123A', '85099B', '22423', '20725', '22197', '47566', '20727', '23203', '22383', '22720']

Клієнт: 12362.0 : Рекомендовані продукти: ['85123A', '85099B', '22423', '20725', '22197', '47566', '20727', '23203', '22383', '22720']

Клієнт: 12381.0 : Рекомендовані продукти: ['85123A', '85099B', '22423', '20725', '22197', '47566', '20727', '23203', '22383', '22720']

Клієнт: 12388.0 : Рекомендовані продукти: ['85123A', '85099B', '22423', '20725', '22197', '47566', '20727', '23203', '22383', '22720']

Кластер 6 :

Рекомендовані продукти:

Клієнт: 12365.0 : Рекомендовані продукти: ['85123A', '22423', '47566', '84879', '21034', '22457', '22720', '22960', '21212', '22470']

Клієнт: 12410.0 : Рекомендовані продукти: ['85123A', '22423', '47566', '84879', '21034', '22457', '22720', '22960', '21212', '22470']

Клієнт: 12414.0 : Рекомендовані продукти: ['85123A', '22423', '47566', '84879', '21034', '22457', '22720', '22960', '21212', '22470']

Клієнт: 12426.0 : Рекомендовані продукти: ['85123A', '22423', '47566', '848
79', '21034', '22457', '22720', '22960', '21212', '22470']

Клієнт: 12559.0 : Рекомендовані продукти: ['85123A', '22423', '47566', '848
79', '21034', '22457', '22720', '22960', '21212', '22470']

Кластер 7 :

Рекомендовані продукти:

Клієнт: 12354.0 : Рекомендовані продукти: ['85123A', '22423', '84879', '475
66', '22720', '21212', '22960', '85099B', '22457', '20725']

Клієнт: 12383.0 : Рекомендовані продукти: ['85123A', '22423', '84879', '475
66', '22720', '21212', '22960', '85099B', '22457', '20725']

Клієнт: 12393.0 : Рекомендовані продукти: ['85123A', '22423', '84879', '475
66', '22720', '21212', '22960', '85099B', '22457', '20725']

Клієнт: 12399.0 : Рекомендовані продукти: ['85123A', '22423', '84879', '475
66', '22720', '21212', '22960', '85099B', '22457', '20725']

Клієнт: 12405.0 : Рекомендовані продукти: ['85123A', '22423', '84879', '475
66', '22720', '21212', '22960', '85099B', '22457', '20725']

Кластер 8 :

Рекомендовані продукти:

Клієнт: 12391.0 : Рекомендовані продукти: ['21034', '85123A', '22086', '221
97', '23321', '84879', '23103', '22557', '22139', '22469']

Клієнт: 12508.0 : Рекомендовані продукти: ['21034', '85123A', '22086', '221
97', '23321', '84879', '23103', '22557', '22139', '22469']

Клієнт: 12556.0 : Рекомендовані продукти: ['21034', '85123A', '22086', '221
97', '23321', '84879', '23103', '22557', '22139', '22469']

Клієнт: 12577.0 : Рекомендовані продукти: ['21034', '85123A', '22086', '221
97', '23321', '84879', '23103', '22557', '22139', '22469']

Клієнт: 12607.0 : Рекомендовані продукти: ['21034', '85123A', '22086', '221
97', '23321', '84879', '23103', '22557', '22139', '22469']

Кластер 9 :

Рекомендовані продукти:

Клієнт: 12454.0 : Рекомендовані продукти: ['85123A', '22423', '84879', '233
21', '84946', '23293', '85099B', '22487', '22961', 'POST']

Клієнт: 12492.0 : Рекомендовані продукти: ['85123A', '22423', '84879', '233
21', '84946', '23293', '85099B', '22487', '22961', 'POST']

Клієнт: 12512.0 : Рекомендовані продукти: ['85123A', '22423', '84879', '233
21', '84946', '23293', '85099B', '22487', '22961', 'POST']

Клієнт: 12604.0 : Рекомендовані продукти: ['85123A', '22423', '84879', '233
21', '84946', '23293', '85099B', '22487', '22961', 'POST']

Клієнт: 12641.0 : Рекомендовані продукти: ['85123A', '22423', '84879', '233
21', '84946', '23293', '85099B', '22487', '22961', 'POST']

2.3 Вирішення задачі методом колаборативної фільтрації

2.3.1 Створення матриці Клієнт-Продукти

```
customer_item_matrix = data.pivot_table(index='CustomerID',
columns='StockCode', values='Quantity',aggfunc='sum')


customer_item_matrix = \
customer_item_matrix.applymap(lambda x: 1 if x > 0 else 0)
customer_item_matrix.head()
```

Таблиця 2.6

Матриця Клієнт-Продукти

StockCode	10002	10080	10120	10123C	10124A	10124G	10125	10133
CustomerID								
-1-	-2-	-3-	-4-	-5-	-6-	-7-	-8-	-9-
12346.0	0	0	0	0	0	0	0	0
12347.0	0	0	0	0	0	0	0	0
12348.0	0	0	0	0	0	0	0	0
12349.0	0	0	0	0	0	0	0	0
12350.0	0	0	0	0	0	0	0	0

2.3.2 Рекомендація Клієнт-до-Клієнта

2.3.2.1 Заповнення нулів за допомогою міри косинусної подібності

```
user_to_user = pd.DataFrame(cosine_similarity(customer_item_matrix))
user_to_user
```

Таблиця 2.7

Заповнена матриця Клієнт-до-Клієнта

	1	2	3	4	5	6	7	8	9
-1-	**-2-**	**-3-**	**-4-**	**-5-**	**-6-**	**-7-**	**-8-**	**-9-**	**-10-**
0	1.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000
1	0.0	1.000000	0.063022	0.046130	0.047795	0.038484	0.0	0.025876	0.136641
2	0.0	0.063022	1.000000	0.024953	0.051709	0.027756	0.0	0.027995	0.118262
3	0.0	0.046130	0.024953	1.000000	0.056773	0.137137	0.0	0.030737	0.032461
4	0.0	0.047795	0.051709	0.056773	1.000000	0.031575	0.0	0.000000	0.000000

2.3.2.2 Заміна індексів на унікальні номери клієнтів

```
user_to_user.columns = customer_item_matrix.index

user_to_user['CustomerID'] = customer_item_matrix.index

user_to_user = user_to_user.set_index('CustomerID')
```

2.3.2.3 Рекомендація продуктів для клієнта

user_to_user.loc[12358].sort_values(ascending=False)

Таблиця 2.8

Значення подібності клієнтів до клієнта

CustomerID	Значення
-1-	**-2-**
12358.0	1.000000
14145.0	0.452911
14155.0	0.452911
18240.0	0.452911
13551.0	0.416025

items_bought_12358 = \
set(customer_item_matrix.loc[12358].iloc[customer_item_matrix.loc[12358].to_numpy().nonzero()].index)

items_bought_by_12358

{'15056BL', '15056N', '15056P', '15060B',
'20679', '21232', '22059', '22063',
'22646', '37447', '37449', '48185', 'POST'}

Продукти, які купляв ближчий до нього клієнт:

items_bought_by_14145 =\
set(customer_item_matrix.loc[14145.0].iloc[customer_item_matrix.loc[14145.0].to_numpy().nonzero()].index)

items_bought_by_14145

{'15056BL', '15056N', '15056P', '20679', '85014A', '85014B'}

Відбираємо, продукти, які не купляв клієнт:

items_to_recommend_to_14145 = items_bought_by_12358 - items_bought_by_14145

items_to_recommend_to_14145

{'15060B', '21232', '22059', '22063',

'22646', '37447', '37449', '48185', 'POST'}

data.loc[data.StockCode.isin(items_to_recommend_to_14145), ['StockCode', 'Description']].drop_duplicates().set_index('StockCode')

Таблиця 2.9

Рекомендації для клієнта

StockCode	Description
-1-	**-2-**
POST	POSTAGE
22646	CERAMIC STRAWBERRY CAKE MONEY BANK
48185	DOORMAT FAIRY CAKE
21232	STRAWBERRY CERAMIC TRINKET BOX
22059	CERAMIC STRAWBERRY DESIGN MUG
37449	CERAMIC CAKE STAND + HANGING CAKES
15060B	FAIRY CAKE DESIGN UMBRELLA

Функція для повернення рекомендацій клієнтам:

```
def recommendet_item_to_customer(customer_id):
  most_similar_user =
user_to_user.loc[cust_a].sort_values(ascending=False).reset_index().iloc[1, 0]
  items_bought_by_customer_a =
set(customer_item_matrix.loc[cust_a].iloc[customer_item_matrix.loc[cust_a].
to_numpy().nonzero()].index)
  items_bought_by_customer_b =
set(customer_item_matrix.loc[most_similar_user].iloc[customer_item_matrix
.loc[most_similar_user].to_numpy().nonzero()].index)
  items_to_recommend_to_customer = items_bought_by_customer_b -
items_bought_by_customer_a
  items_description =
data.loc[data.StockCode.isin(items_to_recommend_to_customer),
['StockCode', 'Description']].drop_duplicates().set_index('StockCode')
  return items_description

get_items_to_recommend_cust(12348.0)
```

Таблиця 2.10

Рекомендації для клієнта

StockCode	Description
-1-	**-2-**
21986	PACK OF 12 PINK POLKADOT TISSUES
M	Manual

2.3.3 Рекомендація Продукт-до-Продукту

2.3.3.1 Заповнення нулів за допомогою міри косинусної подібності транспонованої матриці

```
item_to_item = pd.DataFrame(cosine_similarity(customer_item_matrix.T))

item_to_item.columns = customer_item_matrix.T.index

item_to_item['StockCode'] = customer_item_matrix.T.index
item_to_item = item_to_item.set_index('StockCode')
item_to_item.head()
```

Таблиця 2.11

Заповнена матриця Продукт-до-Продукту

Stock Code	**10002**	**10080**	**10120**	**10123C**	**10124A**	**10124G**	**10125**	**10133**
-1-	**-2-**	**-3-**	**-4-**	**-5-**	**-6-**	**-7-**	**-8-**	**-9-**
10002	1.000000	0.0	0.094868	0.091287	0.0	0.000000	0.090351	0.062932
10080	0.000000	1.0	0.000000	0.000000	0.0	0.000000	0.032774	0.045655
10120	0.094868	0.0	1.000000	0.115470	0.0	0.000000	0.057143	0.059702
10123C	0.091287	0.0	0.115470	1.000000	0.0	0.000000	0.164957	0.000000
10124A	0.000000	0.0	0.000000	0.000000	1.0	0.447214	0.063888	0.044499

2.3.3.2 Рекомендація продуктів для продуктів

```
item_to_item.loc['10080'].sort_values(ascending=False)
```

Таблиця 2.12

Значення подібності Продукт-до-Продукт

StockCode	Значення
-1-	**-2-**
10080	1.000000
23694	0.191346
22039	0.187317
47504H	0.166924
21650	0.165567

Визначення п’яти кращих продуктів:

```
top_5_items = \
list(item_to_item.loc['10080'].sort_values(ascending=False).iloc[:5].index)

top_5_items

['10080', '23694', '22039', '47504H','21650']

data.loc[data.StockCode.isin(top_5_items), ['StockCode',
'Description']].drop_duplicates().set_index('StockCode').loc[top_5_items]
```

Таблиця 2.13

Кращі продукти

StockCode	Description
-1-	**-2-**
10080	GROOVY CACTUS INFLATABLE
23694	PAISLEY PARK CARD
22039	BOTANICAL LILY GIFT WRAP
47504H	ENGLISH ROSE SPIRIT LEVEL
21650	ASSORTED TUTTI FRUTTI BRACELET

Функція для повернення рекомендацій для продуктів:

```
def get_top_items(item):
  top_5_items =
list(item_to_item.loc[item].sort_values(ascending=False).iloc[:5].index)
  top_5_display = data.loc[data.StockCode.isin(top_5_items), ['StockCode',
'Description']].drop_duplicates().set_index('StockCode').loc[top_5_items]
  return top_5_display

get_top_items('84029E')
```

Таблиця 2.14

Рекомендація Продукт-до-Продукт

StockCode	Description
-1-	**-2-**
84029E	RED WOOLLY HOTTIE WHITE HEART.
84029G	KNITTED UNION FLAG HOT WATER BOTTLE
21479	WHITE SKULL HOT WATER BOTTLE
21485	RETROSPOT HEART HOT WATER BOTTLE
22111	SCOTTIE DOG HOT WATER BOTTLE

2.4 Висновок

У технічному розділі було розглянуто вирішення задачі рекомендації. В задачі було проаналізовано область роздрібного рітейлу. Для задачі було обрано два методи вирішення. В першому методі, клієнти були кластеризовані на 10 кластерів, та визначені рекомендації, в залежності від кластера. В другому методі, була розглянута колаборативна фільтрація, та визначені рекомендації Клієнт-до-Клієнта і Продукт-до-Продукту.

ВИСНОВКИ

У кваліфікаційній роботі була розглянута задача рокомендації для роздрібного рітейлу. Виявлено ключові кроки для попередньої обробки даних. Розроблені та налаштовані методи для вирішення проблем визначення рекомендацій. Інструментом рішення були обрані: RFM – сегментація, з подальшим розрахунком рекомендація в залежності від кластера; Колаборативна фільтрація, з вирішенням двома підходами: рекомендація Клієнт-до-Клієнта та Продукт-до-Продукту.

Для набору даних попередньо були розглянуті данні, та виявлені критичні точки для попередньої обробки даних.

Для методу RFM – сегментації, були розраховані необхідні показники, присвоєні відповідні оцінки для кожного клієнта. Надалі була розроблена модель машинного кавчання K-means, для кластеризації клієнтів за попередніми оцінками. В кінці, біли проаналізовані кластери, та виявлені, рекомендації кращих продуктів для кожного клієнта.

Для методу колаборативної фільтрації, була розрахована матриця характеризуюча кожного клієнта за його вподобаннями з історії покупок.

Для методу колаботативної фільтрації, розглянуто два підходи: рекомендації Клієнт-до-Клієнта, та Продукт-до-Продукту. Розроблені методи видачі рекомендацій для кожного клієнта, та продукту.

В залежності від маркетингової задачі, є змога розраховувати рекомендації, та підвищувати клієнтський досвід, контролювати запаси та підвищувати лояльність клієнта до рітейлора.

СПИСИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. [Penguin, 2011] Eli Pariser. The Filter Bubble: How the New Personalized Web Is Changing What We Read and How We Think / Eli Pariser – 304 c.

2. [Springer Science & Business Media, 2010] Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor. Recommender Systems Handbook, Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor – 842 c.

3. [Deng-Wickham-2011] Deng, Henry, and Hadley Wickham. “Density Estimation in R.” September 2011.

4. [Donoho-2015] Donoho, David. “50 Years of Data Science.” September 18, 2015. https://oreil.ly/kqFb0. [Duong-2001] Duong, Tarn. “An Introduction to Kernel Density Estimation.” 2001.

5. [Few-2007] Few, Stephen. “Save the Pies for Dessert.” Visual Business Intelligence Newsletter. Perceptual Edge. August 2007.

6. [Manning, 2019] Practical Recommender Systems, Kim Falk: 1–432

7. [ggplot2] Wickham, Hadley. ggplot2: Elegant Graphics for Data Analysis. New York: Springer-Verlag New York, 2009.

8. [Hyndman-Fan-1996] Hyndman, Rob J., and Yanan Fan. “Sample Quantiles in Statis- tical Packages.” American Statistician 50, no. 4 (1996): 361– 65. [lattice] Sarkar, Deepayan. Lattice: Multivariate Data Visualization with R. New York: Springer, 2008.

9. [Legendre] Legendre, Adrien-Marie. Nouvelle méthodes pour la détermination des orbites des comètes. Paris: F. Didot, 1805. “Measures of Skewness and Kurtosis.” In NIST/SEMATECH e-Handbook of Statistical Methods. 2012.

10. [Salsburg-2001] Salsburg, David. The Lady Tasting Tea: How Statistics Revolutionized Science in the Twentieth Century. New York: W. H. Freeman, 2001. [seaborn] Waskom, Michael. “Seaborn: Statistical Data Visualization.” 2015

11. [Trellis-Graphics] Becker, Richard A., William S.Cleveland, MingJen Shyu, and Ste- phen P. Kaluzny. “A Tour of Trellis Graphics.” April 15, 1996

12. [Tukey-1962] Tukey, John W. “The Future of Data Analysis.” The Annals of Mathemat- ical Statistics 33, no. 1 (1962): 1–67.

13. Методичні рекомендації до виконання кваліфікаційної роботи магістра студентами галузі знань 12 Інформаційні технології спеціальності 124 Системний аналіз / Т. А. Желдак, Т.В. Хом’як; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». Дніпро: НТУ «ДП», 2021. – 32 с.

Додаток А. Відомість матеріалів кваліфікаційної роботи

№ з/п	Позначення	Найменування	Кількість аркушів	Примітки
1				
2		Документація		
3				
4	САУ.КР.23.16.ПЗ	Пояснювальна записка	74	Формат А4
5				
6		Демонстраційний матеріал	10	Презентація на CD диску
7				
8		Копія роботи	1	CD диск
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

					САУ.КР.23.16.ДА.ПЗ			
Змін.	Аркуш	№ докум.	Підпис	Дата				
Розроб.		Сербін			**Матеріали кваліфікаційної роботи**	Літ.	Аркуш	Аркушів
К. розд.								
Керівн.		Коряшкіна				НТУ «ДП», 12; 124м-22-1		
Н.контр.		Хом’як						
Зав. каф.		Желдак						