Міністерство освіти і науки України
Луцький національний технічний університет
Факультет робототехніки та штучного інтелекту
Кафедра штучного інтелекту та математичного моделювання

КВАЛІФІКАЦІЙНА РОБОТА ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ
«БАКАЛАВР»

МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ СЕГМЕНТАЦІЇ КЛІЄНТІВ ТА РОЗРОБКА СИСТЕМИ ФОРМУВАННЯ ПЕРСОНАЛІЗОВАНИХ ПРОПОЗИЦІЙ ІНТЕРНЕТ-МАГАЗИНУ

MACHINE LEARNING METHODS FOR CUSTOMER SEGMENTATION AND DEVELOPMENT OF A SYSTEM FOR GENERATING PERSONALIZED OFFERS FOR AN ONLINE STORE

Спеціальність 113 Прикладна математика
(шифр і назва спеціальності)

освітня програма «Штучний інтелект та аналіз масивів даних»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ПРМ-41
Печончик Надія Русланівна
_____
(підпис)

Керівник:
к.т.н., доцент
Фурс Тетяна Василівна
_____
(підпис)

Кваліфікаційну роботу
допущено до захисту
«___» _____ 20__ р.
к.т.н., доцент

Гарант освітньої програми:
Приходько Олексій Сергійович
_____
(підпис)

Луцьк – 2026 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет *архітектури, будівництва та дизайну*
Кафедра *прикладної математики та механіки*
Ступінь вищої освіти: *бакалавр*
Галузь знань: *11 Математика і статистика*
Спеціальність *113 Прикладна математика*
Освітня програма «*Штучний інтелект та аналіз масивів даних»*

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Мікуліч О.А.
«___» _____2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Печончик Надія Русланівна
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Методи машинного навчання для сегментації клієнтів та розробка системи формування персоналізованих пропозицій інтернет-магазину / Machine learning methods for customer segmentation and development of a system for generating personalized offers for an online store
Керівник роботи: *к.т.н., доцент Фурс Тетяна Василівна*
затверджені наказом закладу вищої освіти від «31» грудня 2025 р. № 557/01-02
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 04.06.2026 р.
3. Вихідні дані до роботи _____
_____
_____
_____
4. Зміст пояснювальної записки (перелік питань, що потрібно розробити):
_____
_____
_____
_____
5. Перелік графічного (ілюстративного) матеріалу:
_____
_____
_____
_____
_____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
1 розділ	*Фурс Т. В., доцент кафедри*		
2 розділ	*Фурс Т. В., доцент кафедри*		
3 розділ	*Фурс Т. В., доцент кафедри*		
4 розділ	*Фурс Т. В., доцент кафедри*		
Висновки	*Фурс Т. В., доцент кафедри*		

7. Дата видачі завдання «___» _____ 202__ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи магістра	Строк виконання етапів роботи	Примітка
1.	*Огляд літератури із досліджуваної проблеми*	*до 01.03.2026*	
2.	*Перший розділ*	*до 5.03.2026*	
3.	*Другий розділ*	*до 01.04.2026*	
4.	*Третій розділ*	*до 10.04.2026*	
5.	*Четвертий розділ*	*до 20.04.2026*	
6.	*Висновки*	*до 28.04.2026*	
7.	*Формування списку використаних джерел*	*до 05.05.2026*	
8.	*Оформлення ілюстративного матеріалу*	*до 09.05.2026*	
9.	*Нормоконтроль*	*до 20.05.2026*	
10.	*Інструментальна перевірка на академічний плагіат*	*до 02.062026*	*Показник запозичень тексту ____ %*
11.	*Представлення кваліфікаційної роботи бакалавра до захисту*	*до 04.06.2026*	

Здобувач вищої освіти _____ (Печончик Н.Р.)
(підпис) (прізвище, ініціали)

Керівник кваліфікаційної роботи _____ (Фурс. Т.В.)
(підпис) (прізвище, ініціали)

АНОТАЦІЯ

Печончик Н. Р. Аналіз та кластеризація клієнтської бази інтернет-магазину для розробки персоналізованих пропозицій. Рукопис.

Кваліфікаційна робота бакалавра ОП «Штучний інтелект та аналіз масивів даних» спеціальності 113/F1 Прикладна математика. Луцький національний технічний університет. Луцьк, 2026.

У кваліфікаційній роботі досліджено методи аналізу та кластеризації клієнтської бази інтернет-магазину з метою формування персоналізованих комерційних пропозицій. Об’єктом дослідження є процес сегментації клієнтів інтернет-магазину на основі транзакційних даних. Предметом дослідження є алгоритми кластеризації та методи машинного навчання, що застосовуються для виявлення груп клієнтів зі схожою поведінкою.

В роботі проведено аналіз предметної галузі, розглянуто RFM-модель сегментації клієнтів, досліджено алгоритми k-Means, DBSCAN та ієрархічної кластеризації. Виконано попередню обробку даних, здійснено вибір оптимальних параметрів моделі та сформовано рекомендаційну систему для генерації персоналізованих пропозицій.

Практична цінність роботи полягає у розробці програмного модуля сегментації клієнтів, який може бути інтегрований у систему управління інтернет-магазином для підвищення ефективності маркетингових кампаній та зростання показників конверсії.

Ключові слова: кластеризація, машинне навчання, сегментація клієнтів, RFM-аналіз, k-Means, персоналізація, інтернет-торгівля, рекомендаційна система.

ABSTRACT

Pechonchyk N. R. Analysis and Clustering of E-Commerce Customer Base for Personalized Offer Development. Manuscript.

Bachelor's qualification thesis in the Educational Program «Artificial Intelligence and Data Array Analysis», specialty 113/F1 Applied Mathematics. Lutsk National Technical University. Lutsk, 2026.

The qualification thesis investigates methods for analyzing and clustering the customer base of an online store with the aim of generating personalized commercial offers. The object of the study is the customer segmentation process of an e-commerce platform based on transactional data. The subject of the study comprises clustering algorithms and machine learning methods applied to identify groups of customers with similar behavioral patterns.

The paper provides an analysis of the subject domain, examines the RFM customer segmentation model, and investigates k-Means, DBSCAN, and hierarchical clustering algorithms. Data preprocessing was carried out, optimal model parameters were selected, and a recommendation system was developed for generating personalized offers.

The practical value of the work lies in the development of a customer segmentation software module that can be integrated into an e-commerce management system to improve the effectiveness of marketing campaigns and increase conversion rates.

Keywords: clustering, machine learning, customer segmentation, RFM analysis, k-Means, personalization, e-commerce, recommendation system.

ЗМІСТ

ВСТУП

Персоналізація клієнтського досвіду стає одним із ключових факторів в умовах швидкого розвитку електронної комерції та зростаючої конкуренції на ринку інтернет-торгівлі. Це один з факторів, що визначає успіх комерційного підприємства. За даними аналітичних компаній, персоналізовані рекомендації забезпечують від 15 до 35 відсотків загального доходу провідних онлайн-магазинів [1]. Попри очевидну практичну значущість, більшість вітчизняних інтернет-магазинів або не використовують алгоритмічні підходи до сегментації клієнтів, або застосовують надто спрощені моделі, що не враховують багатовимірність даних.

Хороша сегментація клієнтської бази дозволяє ефективно виявляти приховані закономірності у поведінці покупців, формувати цільові групи зі схожими схемами активності та пропонувати кожному сегменту точно підібрані товари й послуги. В умовах з великим обсягом транзакційних даних методи машинного навчання, зокрема алгоритми кластеризації, надають потужний інструментарій для вирішення цього завдання.

У міжнародній науковій спільноті вже активно досліджується стан вивченості проблеми, задачі сегментації клієнтів та побудови рекомендаційних систем. Значний внесок у розвиток методів кластеризації зробили такі вчені, як Дж. Хартіган і М. Вонг (алгоритм k-Means [2]), М. Естер та ін. (алгоритм DBSCAN [3]), а також численні дослідники у сфері RFM-аналізу – П. Дж. Хьюз, Р. Блаттберг [4, 5]. В Україні дана тематика розглядається переважно в контексті загальних досліджень з аналізу даних, тоді як прикладні аспекти кластеризації клієнтів інтернет-магазинів залишаються недостатньо висвітленими.

Мета роботи: на основі алгоритмів машинного навчання розробити та реалізувати методики аналізу і кластеризації клієнтської бази інтернет-магазину для формування персоналізованих комерційних пропозицій.

Для досягнення поставленої мети у роботі вирішуються такі *завдання*:

1) проаналізувати предметну галузь електронної комерції та сучасні підходи до персоналізації клієнтського досвіду;
2) дослідити методи сегментації клієнтів та алгоритми кластеризації, визначити критерії їх вибору;
3) обґрунтувати застосування RFM-моделі як основи для побудови ознакового простору;
4) виконати попередню обробку та нормалізацію транзакційних даних інтернет-магазину;
5) реалізувати алгоритми кластеризації k-Means, DBSCAN та ієрархічної кластеризації;
6) провести порівняльний аналіз алгоритмів та вибрати оптимальну модель;
7) розробити модуль генерації персоналізованих пропозицій на основі отриманих кластерів.

Об'єктом дослідження є процес сегментації клієнтів інтернет-магазину на основі транзакційних даних про їхню купівельну поведінку.

Предметом дослідження є алгоритми кластеризації та методи машинного навчання, що використовуються в системах електронної комерції для виявлення груп клієнтів зі схожими моделями поведінки.

Методи дослідження. У роботі використано такі методи: аналіз і синтез – для вивчення існуючих підходів до сегментації клієнтів; статистичні методи – для попередньої обробки та аналізу даних; методи машинного навчання без учителя (алгоритми k-Means, DBSCAN, Ward) – для кластеризації клієнтів; метрики якості кластеризації (силует-коефіцієнт, індекс Девіса-Болдіна) – для оцінювання результатів; системний підхід – для проєктування архітектури рішення.

Інформаційна база дослідження. Для проведення дослідження використано відкритий датасет Online Retail II (UCI Machine Learning Repository), що містить транзакційні дані британського онлайн-ретейлера за 2009-2011 роки та охоплює понад 1 млн. записів про покупки. Теоретичну базу складають наукові публікації у міжнародних журналах, матеріали конференцій з машинного навчання та

аналізу даних, а також навчально-методична література кафедри штучного інтелекту та математичного моделювання ЛНТУ.

Об’єм та структура роботи. Кваліфікаційна робота бакалавра складається зі вступу, чотирьох розділів, висновків та списку використаних джерел. Об’єм роботи становить 58 сторінок. Робота містить 9 рисунків, 3 таблиці та 31 найменування використаної літератури.

У процесі підготовки бакалаврської кваліфікаційної роботи застосовувалися технології штучного інтелекту як допоміжний інструментарій. Зокрема, для стилістичної правки та структурування тексту використано ChatGPT-4o та Gemini 3.0. Автор несе повну відповідальність за зміст роботи.

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ДАНИХ

1.1 Огляд предметної галузі електронної комерції та задачі персоналізації

Електронна комерція (e-commerce) – це сукупність комерційних, фінансових і торговельних операцій, які проводяться через телекомунікаційні мережі, переважно через мережу Інтернет. Протягом двох останніх десятиліть ця галузь демонструє стале зростання: за даними Statista, глобальний обсяг роздрібної інтернет-торгівлі у 2023 році перевищив 5,8 трлн доларів США [6], і, за прогнозами, до 2027 року досягне позначки 8 трлн доларів [6]. Наш ринок електронної комерції також стрімко розвивається: навіть попри труднощі воєнного часу, інтернет-торгівля в Україні зростає, а частка онлайн-продажів у загальному роздрібному обороті продовжує збільшуватись.

Традиційно виділяють три основні моделі електронної комерції [7] за типом учасників транзакції. Модель B2C (Business-to-Consumer) описує зв’язок між компанією і кінцевим клієнтом і відповідає найбільш поширеній формі мережевого роздрібного продажу. Модель B2B (Business-to-Business) охоплює комерційні відносини між підприємствами, де послуги надаються іншим компаніям, а не кінцевим споживачам (фізичним особам). Модель C2C (Consumer-to-Consumer) реалізується на маркетплейсах і платформах для оголошень, де фізичні особи обмінюються товарами одне з одним. Датасет, що використовується у цій роботі, відображає здебільшого B2C-транзакції з елементами B2B (оптові клієнти), що обумовлює специфіку підходу до сегментації.

Швидкий розвиток інтернет-торгівлі вплинув на трансформацію маркетингових стратегій. Якщо у ранній період e-commerce домінував підхід масового маркетингу, орієнтованого на максимально широку аудиторію, то сьогодні ключовою конкурентною перевагою стає здатність надавати кожному

клієнту унікальний контент, адаптований до його індивідуальних потреб. Дослідження McKinsey показують, що компанії, які впровадили ефективну персоналізацію, збільшили виторг у середньому на 10-15 %, а їхні показники утримання клієнтів зросли на 20-30 % [1]

Персоналізація в електронній комерції – це комплекс технологічних і маркетингових підходів, спрямованих на адаптацію контенту, товарних пропозицій та комунікацій до індивідуальних характеристик кожного покупця. Сучасні системи персоналізації оперують кількома категоріями даних. Демографічні характеристики клієнта (вік, стать, місцезнаходження, мова) формують базовий профіль. Поведінкові дані (історія переглядів та покупок, середній чек, частота замовлень, тривалість сесій) відображають реальні вподобання. Контекстуальні дані (час доби, день тижня, тип пристрою, сезон) визначають ситуативний контекст взаємодії. Явні вподобання (оцінки, відгуки, списки бажань, порівняння товарів) надають безпосередню зворотну інформацію від покупця.

З технологічної точки зору системи персоналізації реалізуються через рекомендаційні двигуни, механізми динамічного ціноутворення, персоналізовані поштові кампанії та адаптивний веб-контент. Рекомендаційні двигуни є найбільш дослідженим компонентом: вони реалізують алгоритми колаборативної фільтрації, контентної фільтрації та гібридні підходи [8]. Проте ефективне функціонування будь-якого з цих механізмів передбачає наявність якісної сегментації клієнтської бази як фундаменту.

Центральним технічним завданням персоналізації є сегментація клієнтської бази – процес поділу сукупності клієнтів на однорідні групи (сегменти, кластери) за певними критеріями, з таким розрахунком, щоб клієнти всередині одного сегменту були максимально подібними між собою, а представники різних сегментів максимально відрізнялися. Якісна сегментація дозволяє досягти кількох взаємопов’язаних цілей: виявляти групи клієнтів зі схожими потребами та очікуваннями; формувати цільові маркетингові кампанії з вищим рівнем релевантності та нижчою вартістю залучення; раціонально розподіляти

маркетинговий бюджет між сегментами відповідно до їхньої цінності для бізнесу; прогнозувати відтік клієнтів та своєчасно вживати утримуючих заходів [9]; виявляти нові ринкові можливості та незадоволений попит.

У практиці сучасного маркетингу розрізняють кілька рівнів сегментації клієнтів. Макросегментація передбачає поділ клієнтів за широкими демографічними та географічними ознаками. Вона проста у реалізації, але обмежена за точністю. Мікросегментація базується на детальному аналізі поведінкових патернів та транзакційних даних, дозволяючи виділяти вузькі цільові групи з високою однорідністю. Індивідуальна персоналізація (так зване «сегментування одного», або segment of one) є граничним випадком, при якому кожен клієнт розглядається як окремий сегмент із власним профілем вподобань. Цей підхід технічно реалізується через гібридні рекомендаційні системи в режимі реального часу.

Впровадження алгоритмічних підходів до сегментації відкриває принципово нові можливості порівняно з традиційними експертними методами. Якщо ручна сегментація покладається на суб’єктивний досвід маркетолога і здатна виділити лише кілька широких груп, то машинне навчання забезпечує об’єктивний, відтворюваний аналіз у багатовимірному просторі ознак та здатне виявляти десятки вузьких, внутрішньо однорідних сегментів. Водночас практичне застосування цих методів потребує ретельного вибору алгоритму, підготовки даних та інтерпретації результатів, тобто завданням, яким власне присвячено цю кваліфікаційну роботу.

Таким чином, електронна комерція формує потужний запит на алгоритмічні методи сегментації клієнтів, а методи машинного навчання без учителя, зокрема кластеризація, є природним і ефективним інструментом відповіді на цей запит. Подальший аналіз зосередиться на огляді конкретних методів та підходів.

1.2 Аналіз існуючих підходів до сегментації клієнтів

У науковій літературі та практиці маркетингу виокремлюють декілька принципових підходів до сегментації клієнтів. Вибір конкретного підходу визначається наявними даними, бізнес-цілями та технічними можливостями підприємства. Наведемо системний огляд основних підходів у порядку зростання технічної складності та аналітичної глибини.

Демографічна та географічна сегментація є найпростішими та найбільш поширеними підходами, що ґрунтуються на об’єктивних характеристиках клієнта: вік, стать, рівень доходу, освіта, сімейний стан та місцезнаходження. Перевагами методу є простота збору та обробки даних, висока інтерпретованість сегментів та мінімальні вимоги до обчислювальних ресурсів. Водночас демографічні та географічні ознаки не завжди корелюють із реальною купівельною поведінкою: два клієнти одного віку, статі та регіону можуть мати кардинально різні переваги та моделі споживання. Цей фундаментальний недолік обмежує точність таргетування та знижує ROI маркетингових кампаній.

Психографічна сегментація базується на суб’єктивних характеристиках особистості: стиль життя, цінності, інтереси, особистісні риси (так звана VALS-класифікація [10]). Цей підхід дозволяє формувати глибоко резонуючі маркетингові повідомлення, проте пов’язаний із суттєвими труднощами збору даних: психографічна інформація зазвичай отримується через опитування та не доступна безпосередньо з транзакційних систем. Тому у контексті аналізу даних інтернет-магазину психографічна сегментація, як правило, застосовується лише у поєднанні з іншими методами.

Поведінкова сегментація базується безпосередньо на аналізі дій клієнта: частоті та обсязі покупок, асортименті придбаних товарів, каналах взаємодії, реакції на маркетингові комунікації, тривалості клієнтського циклу. Цей підхід вважається більш точним порівняно з демографічним, оскільки відображає фактичні вподобання споживача, а не припущення про них. Поведінкові дані є природним продуктом функціонування будь-якої транзакційної системи і не

вимагають додаткового збору. Саме поведінкова сегментація лежить в основі більшості сучасних систем персоналізації у e-commerce.

RFM-аналіз (Recency, Frequency, Monetary) є одним із найефективніших і найбільш практично перевірених методів поведінкової сегментації, запропонований у 1990-х роках у контексті прямого маркетингу та каталожної торгівлі. Модель описує клієнта трьома ключовими параметрами. Метрика R (Recency) – давність останньої покупки – відображає поточну активність клієнта та вірогідність його повернення: чим менший час пройшов з моменту останньої покупки, тим вища ймовірність нової. Метрика F (Frequency) – кількість покупок за визначений розрахунковий період – характеризує рівень лояльності та звичку клієнта до взаємодії з магазином. Метрика M (Monetary) – загальна сума витрат за той самий період – відображає економічну цінність клієнта для бізнесу. RFM-модель має низку очевидних переваг: вона обчислюється виключно з транзакційних даних без потреби у додатковому зборі інформації, забезпечує інтуїтивно зрозумілу інтерпретацію сегментів і є стандартом de facto у роздрібній торгівлі та e-commerce.

Методи машинного навчання без учителя (unsupervised learning) забезпечують перехід від ручного формування сегментів до автоматичного виявлення прихованих структур у даних. Кластеризація (ключовий клас таких методів) ставить за мету розбиття вибірки на групи (кластери) таким чином, щоб об’єкти всередині кластеру були максимально подібними між собою, а об’єкти з різних кластерів максимально відрізнялися. На відміну від класифікації, алгоритми кластеризації не потребують розмічених навчальних прикладів, що є критичною перевагою в умовах відсутності апріорних знань про структуру клієнтської бази [11].

Алгоритм k-Means є одним із найширше застосовуваних методів кластеризації завдяки своїй обчислювальній ефективності та простоті реалізації [12, 13]. Алгоритм ітеративно мінімізує суму квадратів евклідових відстаней між точками та центроїдами відповідних кластерів (внутрішньокластерну інерцію). Процес складається з двох кроків, що чергуються: крок присвоєння (кожен

об’єкт відноситься до кластеру з найближчим центроїдом) та крок оновлення (центроїди перераховуються як середні значення по всіх точках кластеру). Алгоритм зупиняється при досягненні заданого порогу збіжності або максимальної кількості ітерацій [14]. Основним обмеженням k-Means є необхідність задавати кількість кластерів k наперед, а також припущення про сферичну (опуклу) форму кластерів та приблизно рівний їх розмір. Крім того, алгоритм чутливий до аномальних спостережень (викидів) та до вибору початкових центроїдів [14, 15].

Алгоритм DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [3] відносить точки до кластерів на основі густини їх розташування у просторі ознак. Точки з достатньою кількістю сусідів в межах радіуса eps вважаються «ядровими» і формують ядро кластеру; точки, доступні від ядрових, відносяться до того ж кластеру; ізольовані точки, що не є ні ядровими, ні досяжними, позначаються як шум. На відміну від k-Means, DBSCAN автоматично визначає кількість кластерів, не потребує апріорного задання k, здатний виявляти кластери довільної форми та ідентифікувати аномальні спостереження як окремий клас «шуму». Це робить алгоритм особливо привабливим для аналізу реальних даних про клієнтів, де значна частка спостережень може бути аномальною. Основний недолік DBSCAN – це чутливість до вибору параметрів eps і min_samples та знижена ефективність у просторах з нерівномірною густиною.

Ієрархічна кластеризація будує деревоподібну структуру (дендрограму), яка дозволяє аналізувати та візуалізувати структуру даних на різних рівнях деталізації без необхідності фіксувати кількість кластерів наперед. Агломеративні методи (bottom-up) розпочинають з того, що кожен об’єкт є окремим кластером, і послідовно об’єднують найближчі пари кластерів до досягнення єдиного. Метод Уорда (Ward) є найпопулярнішим критерієм злиття, що мінімізує приріст внутрішньокластерної дисперсії при кожному об’єднанні [16]. Перевагою ієрархічної кластеризації є відсутність потреби у визначенні k та наочність дендрограми для інтерпретації; суттєвим недоліком є квадратична

обчислювальна складність $O(n^2)$, що обмежує застосовність на великих датасетах (понад кілька тисяч спостережень) [16].

Метод головних компонент (PCA – Principal Component Analysis) не є самостійним методом сегментації, але широко застосовується як інструмент зниження розмірності перед кластеризацією. PCA проєктує дані на простір ортогональних компонент, що забезпечує максимальне збереження дисперсії при мінімальній кількості вимірів. Це дозволяє прискорити роботу алгоритмів кластеризації, зменшити ефект «прокляття розмірності» та забезпечити двовимірну візуалізацію результатів [17], [18].

Методи оцінювання якості кластеризації відіграють критичну роль у виборі оптимальних параметрів алгоритму та числа кластерів. Силует-коефіцієнт (Silhouette Score) вимірює, наскільки об’єкт подібний до свого кластеру порівняно з найближчим чужим кластером; значення варіюється від -1 до +1, де вищі значення свідчать про кращу сепарацію [19]. Індекс Девіса-Болдіна (Davies-Bouldin Index) є відношенням середньої внутрішньокластерної відстані до міжкластерної; менші значення відповідають кращому розбиттю [20]. Метод «ліктя» (Elbow Method) використовується для визначення оптимального k в алгоритмі k-Means через аналіз зміни внутрішньокластерної інерції залежно від числа кластерів.

Порівняльний аналіз підходів свідчить, що оптимальною стратегією для задачі сегментації клієнтів інтернет-магазину є поєднання RFM-моделі як інструменту формування ознакового простору з алгоритмом k-Means як основним методом кластеризації, із додатковою верифікацією результатів за допомогою DBSCAN та ієрархічної кластеризації. Цей комплексний підхід забезпечує баланс між інтерпретованістю результатів, обчислювальною ефективністю та можливістю виявлення аномальних клієнтів. Він і обраний як методологічна основа цієї кваліфікаційної роботи.

1.3 Огляд та характеристика вхідних даних

Для проведення дослідження використано відкритий датасет Online Retail II, розміщений у репозиторії UCI Machine Learning Repository. Датасет містить транзакційні дані британської компанії, що спеціалізується на оптовому та роздрібному продажі подарункової та декоративної продукції через Інтернет, за період з 1 грудня 2009 року по 9 грудня 2011 року. Датасет було оприлюднено Chen D., Sain S. L. та Guo K. у 2012 році і з того часу широко використовується у наукових дослідженнях з кластеризації та рекомендаційних систем [21].

Датасет складається з двох аркушів формату Microsoft Excel: Year 2009-2010 та Year 2010-2011, загальним обсягом понад 1 048 000 записів. Кожен рядок описує окрему позицію товару в рамках конкретного замовлення. Структура датасету включає вісім атрибутів: InvoiceNo – шестизначний унікальний номер рахунку-фактури (перша літера C позначає скасований рахунок); StockCode – п’ятизначний унікальний код товарної позиції; Description – текстовий опис найменування товару; Quantity – кількість одиниць товару у даній транзакції; InvoiceDate – дата та час виставлення рахунку-фактури у форматі ДД/ММ/РРРР ГГ:ХХ; UnitPrice – ціна за одиницю товару, виражена у фунтах стерлінгів; CustomerID – п’ятизначний унікальний ідентифікатор клієнта, присвоєний системою; Country – назва країни, де проживає відповідний клієнт.

Попередній статистичний аналіз якості датасету виявив кілька системних проблем, що потребують усунення на етапі передобробки даних. По-перше, пропущені значення CustomerID: близько 135 тисяч записів (24,9 % від загальної кількості) не містять ідентифікатора клієнта, що унеможливлює їх прив’язку до конкретного покупця та їх участь у розрахунку RFM-метрик. Ці записи підлягають видаленню. По-друге, від’ємні значення Quantity: 2,2 % транзакцій мають від’ємне значення кількості, що відповідає оформленим поверненням товарів. Такі записи слід або виключити, або обробляти окремо залежно від мети аналізу. По-третє, аномальні значення UnitPrice: серед цін присутні нульові значення (безкоштовні зразки або помилки вводу) та від’ємні значення, що

підлягають фільтрації. По-четверте, технічні транзакції: частина записів з кодами на зразок «POST», «DOT», «BANK» стосується поштових, банківських та адміністративних нарахувань, а не реальних продажів товарів.

Географічний розподіл клієнтів охоплює понад 40 країн на чотирьох континентах. Проте переважна більшість транзакцій – близько 91 % за кількістю рядків та 84 % за сумарним грошовим обсягом – стосується клієнтів з Великобританії. Топ-5 країн за обсягом продажів після Великобританії складають Нідерланди, Ірландія, Німеччина та Франція. Для цілей цього дослідження аналіз зосереджено на клієнтах з Великобританії задля забезпечення однорідності вибірки та усунення ефектів, пов’язаних із міжнародною логістикою та сезонністю різних ринків.

Після проведення повного циклу очищення – видалення рядків без CustomerID, скасованих замовлень, технічних транзакцій та записів з аномальними значеннями очікується отримати близько 3 920 унікальних клієнтів та близько 354 тисяч транзакційних рядків. Ці обсяги є репрезентативними для побудови стійкої моделі кластеризації: достатньо великими для виявлення статистично значущих закономірностей, але водночас прийнятними для обчислювально інтенсивних алгоритмів.

Аналіз часового розподілу транзакцій виявляє виражену сезонну складову. Протягом обох охоплених років спостерігається значне зростання активності у жовтні-листопаді напередодні різдвяного сезону: кількість замовлень у листопаді перевищує середньомісячний рівень приблизно у 2-2,5 разу. Помітний сезонний спад фіксується у серпні, традиційному місяці відпусток. Цей факт є важливим з точки зору визначення розрахункового вікна для RFM-аналізу: дату «знімку» (snapshot date), відносно якої розраховується метрика Recency, обрано як 10 грудня 2011 року як дату, що безпосередньо слідує за останнім днем датасету, що дозволяє уникнути порожнього хвоста у даних. Розподіл транзакцій по місяцях та Quantity наведено на рисунку 1.1.

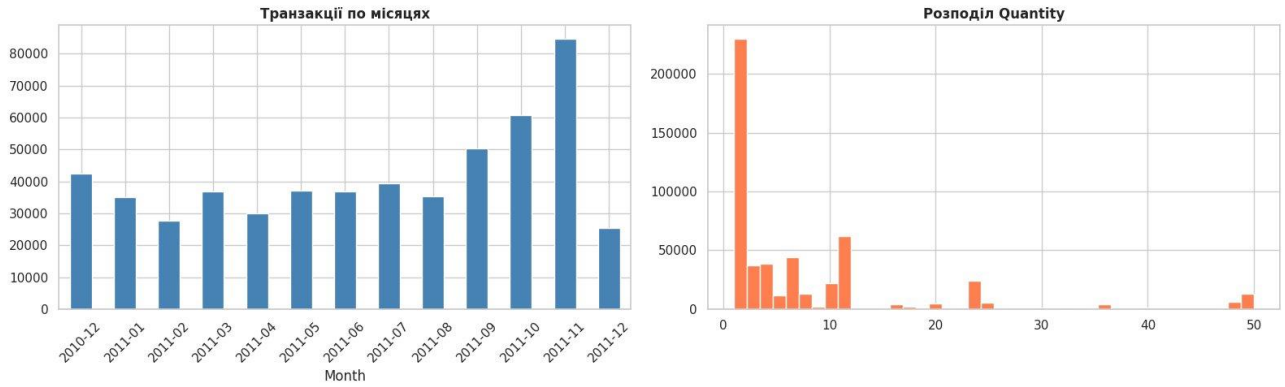

Рисунок 1.1 – Транзакції по місяцях та розподіл Quantity

Розподіл частоти покупок демонструє виражену асиметрію праворуч: значна частина клієнтів (близько 35 %) здійснила лише одну-дві покупки за весь розрахунковий період, тоді як невелика група «лояльних» клієнтів (близько 5 % від загальної кількості) відповідає за понад 50 відсотків загального грошового обороту. Ця закономірність, відома як принцип Парето (або правило «80/20»), підтверджує доцільність сегментації з виділенням окремого кластеру VIP-клієнтів [21].

Розподіл грошових витрат (метрика M в RFM) також характеризується значною правосторонньою асиметрією: медіанна сума витрат одного клієнта складає близько 652 фунти стерлінгів, тоді як середнє арифметичне суттєво вище (близько 1 864 фунтів) через наявність кількох великих оптових клієнтів. Ці особливості розподілів обумовлюють необхідність логарифмічного перетворення або стандартизації ознак перед застосуванням алгоритмів кластеризації, що базуються на евклідових відстанях.

Додатковий аналіз асортименту виявив, що датасет охоплює понад 4 000 унікальних товарних позицій. Найпопулярніші 20 позицій становлять лише 1,5 % асортименту, однак забезпечують близько 12 % загального обсягу продажів. Ця інформація буде корисною на етапі розробки модуля персоналізованих пропозицій для формування релевантних рекомендацій для кожного виявленого кластеру клієнтів.

Висновки до розділу 1

У першому розділі проведено комплексний аналіз предметної галузі та визначено теоретичні засади дослідження. Встановлено, що персоналізація є ключовим фактором конкурентоспроможності в електронній комерції: компанії, що впровадили ефективну сегментацію, збільшують виторг на 10-15 % та показник утримання клієнтів на 20-30 %. Систематизовано підходи до сегментації клієнтів від демографічних та психографічних до поведінкових методів та обґрунтовано доцільність застосування RFM-моделі як основи для формування ознакового простору в задачі кластеризації клієнтів інтернет-магазину. Здійснено порівняльний аналіз алгоритмів кластеризації (k-Means, DBSCAN, ієрархічна кластеризація) та методу зниження розмірності (PCA) з визначенням переваг, обмежень та умов доцільного застосування кожного. Охарактеризовано датасет Online Retail II як вхідні дані: встановлено структуру (8 атрибутів, понад 1 млн записів), виявлено основні проблеми якості (24,9 % відсутніх CustomerID, від’ємні значення Quantity та UnitPrice, технічні транзакції), описано географічний, часовий та асортиментний розподіли даних. Визначено ключові особливості розподілів, що обумовлюють необхідність логарифмічного перетворення ознак та фіксації snapshot date перед розрахунком RFM-метрик.

РОЗДІЛ 2
ПОСТАНОВКА ЗАДАЧІ ТА ВИБІР МЕТОДІВ РОЗВ'ЯЗАННЯ

2.1 Постановка задачі кластеризації клієнтів

Задача кластеризації клієнтів інтернет-магазину є типовою задачею машинного навчання без учителя (unsupervised learning), де метою є виявлення прихованої структури у транзакційних даних без попереднього визначення класів або міток. Формально її можна сформулювати таким чином.

Нехай задано множину клієнтів $X = \{x_1, x_2, ..., x_n\}$, де кожен об'єкт $x_i \in R^m$ описується вектором ознак у m-вимірному просторі. В контексті цієї роботи ознаковий простір формується на основі RFM-моделі та включає три числові характеристики: давність останньої покупки (R), частоту замовлень (F) та загальний грошовий обіг клієнта (M), тобто $m = 3$.

Метою кластеризації є побудова розбиття $C = \{C_1, C_2, ..., C_k\}$ множини X на k непересічних підмножин (кластерів) таких, що:

а) **внутрішньокластерна однорідність:** для будь-яких двох об'єктів x_i, x_j C_l відстань $\rho(x_i, x_j)$ є мінімальною, тобто об'єкти всередині одного кластеру максимально схожі між собою;

б) **міжкластерна відмінність:** для будь-яких двох об'єктів $x_i \in C_p$ та $x_j \in C$, де $p \neq q$, відстань $\rho(x_i, x_j)$ є максимальною, тобто об'єкти з різних кластерів максимально відрізняються;

в) **повнота розбиття:** $C_1 \cup C_2 \cup ... \cup C_k = X$;

г) **неперетинність:** $C_i \cap C_j = \emptyset$ для будь-яких $i \neq j$.

Якість розбиття оцінюється через внутрішні метрики: силует-коефіцієнт (Silhouette Score), індекс Девіса-Болдіна (Davies-Bouldin Index) та індекс Калінські-Харабаша (Calinski-Harabasz Index), формальні визначення яких наведено у підрозділі 2.3.

Практичне формулювання задачі у цій роботі передбачає такі вхідні та вихідні дані. Вхідними даними є очищений та нормалізований набір транзакцій

інтернет-магазину (датасет Online Retail II), з якого сформовано RFM-таблицю для кожного унікального клієнта. Вихідними даними є: мітка кластеру для кожного клієнта; статистичний профіль кожного кластеру (медіанні значення R, F, M); семантична інтерпретація кластерів (типологія клієнтів); набір персоналізованих маркетингових пропозицій для кожного виявленого сегменту.

Обмеження та припущення задачі. По-перше, розрахунковий горизонт обмежено одним роком (грудень 2010 – грудень 2011), що дозволяє уникнути ефектів довгострокового дрейфу поведінки клієнтів. По-друге, аналіз обмежено клієнтами з Великобританії (близько 91 % транзакцій датасету) для забезпечення однорідності вибірки. По-третє, мінімальна кількість транзакцій на клієнта не обмежується (включаються навіть разові покупці), однак клієнти без ідентифікатора виключаються на етапі передобробки.

Таким чином, задача зводиться до побудови стійкого, інтерпретованого та практично корисного розбиття клієнтської бази на однорідні сегменти з метою адресного маркетингового впливу. У наступному підрозділі розглядається математичний апарат RFM-моделі, що визначає спосіб формування ознакового простору.

2.2 RFM-модель як основа для формування ознак

RFM-модель (Recency, Frequency, Monetary) є однією з найбільш широко застосовуваних і практично перевірених методик поведінкової сегментації клієнтів у роздрібній торгівлі та електронній комерції. Вперше систематично описана у роботах Г’юза (Hughes, 1994) та Блаттберга і Деайтона (Blattberg & Deighton, 1996) [4, 5], модель базується на фундаментальному принципі: минула поведінка клієнта є найкращим передвісником його майбутніх дій.

Формальне визначення RFM-метрик. Нехай задано множину транзакцій $T = \{t_1, t_2, ..., t_n\}$, де кожна транзакція $t_i = (customer_i, date_i, amount_i)$ описується ідентифікатором клієнта, датою здійснення та сумою. Зафіксуємо контрольну

дату (snapshot date) t_s як день, наступний після дати останньої транзакції в датасеті. Тоді для кожного клієнта с RFM-метрики визначаються таким чином.

Давність (Recency): $R(c) = t_s - \max\{date_i : customer_i = c\}$. Вимірюється у днях і показує, скільки часу минуло від останньої покупки клієнта до контрольної дати. Менше значення R свідчить про вищу поточну активність.

Частота (Frequency): $F(c) = |\{invoice_i : customer_i = c\}|$. Кількість унікальних рахунків-фактур (замовлень) клієнта за розрахунковий період. Вища частота свідчить про більш стійку звичку до взаємодії з магазином.

Грошовий обіг (Monetary): $M(c) = \Sigma\{amount_i : customer_i = c\}$. Сума всіх витрат клієнта за розрахунковий період у фунтах стерлінгів. Є прямим індикатором економічної цінності клієнта для бізнесу.

Таким чином, кожен клієнт отримує тривимірний вектор ознак (R, F, M), що визначає його положення в RFM-просторі. Особливістю цього простору є виражена правостороння асиметрія розподілів: значна частина клієнтів концентрується в зоні малих F та M (масові разові покупці), тоді як невелика група лояльних клієнтів розташована у зоні великих F та M. Розподіл метрики R також асиметричний, але у зворотному напрямку: більшість клієнтів мають відносно малий R (купували нещодавно).

Для усунення асиметрії та приведення ознак до порівнянного масштабу перед застосуванням алгоритмів кластеризації, що базуються на евклідових відстанях, у роботі застосовується двоетапна трансформація ознак. На першому етапі здійснюється логарифмічне перетворення: $R' = \ln(R + 1)$, $F' = \ln(F + 1)$, $M' = \ln(M + 1)$. Функція $\log(1 + x)$ застосовується замість $\log(x)$ для обробки нульових значень та забезпечення числової стабільності. На другому етапі виконується стандартизація методом z-оцінок (StandardScaler): $z = (x - \mu) / \sigma$, де μ – вибіркове середнє, σ – стандартне відхилення. Після цього перетворення кожна ознака має нульове середнє та одиничне відхилення, що забезпечує рівний внесок R, F і M у розрахунок евклідових відстаней.

Перевага RFM-моделі перед альтернативними підходами полягає у повному використанні транзакційних даних без потреби у додатковому зборі інформації.

Разом з тим модель має і певні обмеження: вона не враховує асортиментні вподобання клієнта (які категорії товарів він купує), канали взаємодії та контекстуальні чинники (сезонність, промо-акції). Ці обмеження можуть бути частково нівельовані на етапі інтерпретації кластерів шляхом аналізу асортиментного профілю кожного виявленого сегменту.

Таким чином, RFM-модель формує тривимірний ознаковий простір, який є інформаційно ємним, обчислювально ефективним та практично інтерпретованим. Він є природним вхідним простором для алгоритмів кластеризації, що розглядаються у наступному підрозділі.

2.3 Алгоритми кластеризації та критерії їх вибору

У цьому підрозділі наведено математичний опис трьох алгоритмів кластеризації, обраних для порівняльного аналізу, а також формальні визначення метрик якості кластеризації, що використовуються для оцінки та порівняння отриманих розбиттів.

Алгоритм k-Means.

Алгоритм k-Means мінімізує суму квадратів внутрішньокластерних відстаней (інерцію):

$$J = \Sigma_{i=1}^{j} \Sigma_{x \in C_i} \|x - \mu_i\|^2,$$

де μ_i – центроїд і-го кластеру C_i, визначений як $\mu_i = (1/|C_i|) \Sigma_{x \in C_i} x$. Алгоритм є ітераційним і складається з двох кроків, що чергуються до досягнення збіжності: крок присвоєння (Е-крок), на якому кожен об’єкт відноситься до кластеру з найближчим центроїдом, та крок оновлення (М-крок), на якому центроїди перераховуються. Для покращення якості початкової ініціалізації та уникнення потрапляння в локальні мінімуми у роботі використовується метод k-Means++ [13], що вибирає початкові центроїди з імовірністю, пропорційною квадрату відстані від вже обраних центрів.

Оптимальна кількість кластерів k визначається за допомогою методу ліктя (Elbow Method), що аналізує залежність інерції J від k, та силует-коефіцієнта.

Значення k, при якому силует-коефіцієнт досягає максимуму, вважається оптимальним.

Алгоритм DBSCAN.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) класифікує точки простору на основі концепції густини [3]. Точка x є ε-досяжною від точки y, якщо $\rho(x, y) \leq \varepsilon$ та в ε-околі точки y знаходиться не менше MinPts точок (тобто y є ядровою точкою). Кластер визначається як максимальна множина попарно ε-досяжних точок. Точки, що не належать жодному кластеру, позначаються як шум (noise).

Алгоритм має два гіперпараметри: радіус ε та мінімальна кількість точок MinPts. Для визначення оптимального ε використовується метод k-найближчих сусідів (k-NN): будується відсортований графік відстаней до k-го найближчого сусіда (де k = MinPts), а оптимальне ε відповідає точці максимального перегину цієї кривої. Перевагою DBSCAN є здатність виявляти кластери довільної форми та автоматично ідентифікувати аномальних клієнтів (шумові точки), не потребуючи задання k наперед.

Ієрархічна кластеризація (метод Уорда).

Агломеративна ієрархічна кластеризація будує дендрограму шляхом послідовного злиття пар кластерів, що мінімізують критерій злиття. Метод Уорда (Ward's minimum variance method) [13] на кожному кроці обирає для злиття пару кластерів A та B, що мінімізує приріст внутрішньокластерної дисперсії:

$$\Delta(A, B) = (|A| \cdot |B|) / (|A| + |B|) \cdot \|\mu\_A - \mu\_B\|^2,$$

де μ_A та μ_B – центроїди кластерів A і B відповідно. Оптимальна кількість кластерів визначається шляхом аналізу дендрограми: вертикальний перетин дендрограми на рівні найбільшого стрибка відстані злиття дає оптимальне k. У реалізації використовується клас AgglomerativeClustering з бібліотеки scikit-learn [22].

Метрики якості кластеризації.

Для кількісної оцінки якості кластеризації використовуються три внутрішні (unsupervised) метрики, що не потребують розмічених даних.

Силует-коефіцієнт (Silhouette Score) для об’єкта x_i визначається як: $s(x_i) = (b(x_i) - a(x_i)) / \max\{a(x_i), b(x_i)\}$, де $a(x_i)$ – середня внутрішньокластерна відстань від x_i до решти об’єктів його кластеру, $b(x_i)$ – мінімальна середня відстань від x_i до об’єктів найближчого чужого кластеру. Значення коефіцієнта лежить у діапазоні $[-1; 1]$, де значення, близькі до +1, свідчать про добре відокремлені кластери.

Індекс Девіса-Болдіна (Davies-Bouldin Index) визначається як: $DB = (1/k) \Sigma_{i=1}^{j} \max_{i \neq j}\{(s_i + s_j) / d_{ij}\}$, де s_i − середній розкид точок у кластері C_i, d_{ij} − відстань між центроїдами C_i та C_j. Менші значення DBI свідчать про кращу якість розбиття.

Індекс Калінські-Харабаша (Calinski-Harabasz Index) [23] є відношенням міжкластерної дисперсії до внутрішньокластерної: $CH = [tr(B\_k) / tr(W\_k)] \cdot (n - k) / (k - 1)$, де B_k – матриця розкиду між кластерами, W_k – матриця розкиду всередині кластерів, n – загальна кількість об’єктів. Більші значення CH відповідають компактнішим та краще відокремленим кластерам.

Вибір оптимального алгоритму здійснюється за сукупністю трьох критеріїв: максимальний Silhouette Score, мінімальний Davies-Bouldin Index та максимальний Calinski-Harabasz Index. У разі суперечності між метриками перевага надається Silhouette Score як найбільш інформативній та широко застосовуваній метриці якості кластеризації.

Для зниження розмірності та двовимірної візуалізації результатів кластеризації використовується метод головних компонент (PCA). Перші дві головні компоненти, що пояснюють максимальну частку загальної дисперсії даних, використовуються виключно для побудови scatter-plot діаграм. Власне кластеризація здійснюється у повному тривимірному RFM-просторі.

Висновки до розділу 2

У другому розділі виконано постановку задачі кластеризації клієнтів та обґрунтовано вибір методів її розв’язання.

Задачу формально визначено як розбиття множини клієнтів на k непересічних однорідних кластерів у тривимірному просторі RFM-ознак з мінімізацією внутрішньокластерної відстані та максимізацією міжкластерної. Визначено вхідні дані (очищені транзакції датасету Online Retail II), вихідні дані (мітки кластерів, профілі сегментів, маркетингові пропозиції) та обмеження задачі.

Математично обґрунтовано застосування RFM-моделі як основи для формування ознакового простору: сформульовано формальні визначення метрик Recency, Frequency та Monetary, описано двоетапну трансформацію даних (логарифмування та стандартизація методом z-оцінок) для усунення асиметрії розподілів та приведення ознак до порівнянного масштабу.

Наведено детальний математичний опис трьох алгоритмів кластеризації: k-Means (мінімізація інерції з ініціалізацією k-Means++ [13]), DBSCAN (густинна кластеризація з параметрами ε та MinPts) та ієрархічна кластеризація методом Уорда (мінімізація приросту внутрішньокластерної дисперсії). Для кожного алгоритму описано метод вибору оптимальних гіперпараметрів. Визначено три внутрішні метрики якості кластеризації: силует-коефіцієнт, індекс Девіса-Болдіна та індекс Калінські-Харабаша як основу для порівняльного аналізу алгоритмів. Практична реалізація описаних методів виконується у Розділі 3.

РОЗДІЛ 3
РОЗРОБКА ТА ІМПЛЕМЕНТАЦІЯ РІШЕННЯ

3.1 Архітектура системи та вибір технологічного стеку

Розроблена система аналізу та кластеризації клієнтської бази інтернет-магазину реалізована у вигляді інтерактивного Jupyter-ноутбука, виконуваного в середовищі Google Colaboratory. Така архітектурна рішення зумовлена поєднанням кількох факторів: безкоштовним доступом до обчислювальних ресурсів, нативною підтримкою мови Python та ключових бібліотек машинного навчання, можливістю відтворення результатів у будь-якому браузері без локального встановлення програмного забезпечення, а також органічним поєднанням програмного коду, математичних обчислень і аналітичної візуалізації в єдиному документі.

Архітектура системи є модульною та складається з шести послідовно пов’язаних функціональних блоків, кожен з яких реалізує чітко визначений етап аналітичного процесу.

Блок завантаження та первинної перевірки даних (DataLoader) відповідає за зчитування вхідного датасету Online Retail II у форматі .xlsx, виведення первинної статистики (розмір, кількість унікальних клієнтів, часовий діапазон, кількість пропущених значень) та формування вихідного DataFrame.

Блок передобробки даних (DataPreprocessor) виконує усунення пропущених значень CustomerID, фільтрацію скасованих замовлень та аномальних записів, географічну фільтрацію (Великобританія), обчислення атрибуту TotalPrice = Quantity × Price та формування очищеного DataFrame для подальшого аналізу.

Блок побудови RFM-таблиці (RFMBuilder) агрегує очищені транзакції за CustomerID, обчислює метрики Recency, Frequency та Monetary для кожного клієнта відносно фіксованої snapshot date, а також здійснює двоетапну трансформацію ознак: логарифмування та стандартизацію методом z-оцінок.

Блок кластеризації (ClusteringEngine) реалізує три алгоритми: k-Means з автоматичним підбором оптимального k за методом ліктя та силует-коефіцієнтом, DBSCAN з підбором eps за k-NN графіком та ієрархічну кластеризацію методом Уорда з аналізом дендрограми. Для кожного алгоритму обчислюються метрики якості: Silhouette Score, Davies-Bouldin Index та Calinski-Harabasz Index.

Блок візуалізації (Visualizer) формує набір аналітичних графіків: розподіли RFM-метрик до та після нормалізації, криві методу ліктя та силует-коефіцієнта, PCA-проєкції кластерів, дендрограму, порівняльні barplot та heatmap метрик якості, а також фінальний dashboard з узагальненою статистикою по сегментах.

Блок персоналізованих рекомендацій (RecommendationEngine) інтерпретує виявлені кластери, присвоює кожному сегменту семантичну назву за правилами класифікації на основі медіанних RFM-значень та генерує для кожного клієнта структуровану маркетингову пропозицію (стратегія, канал комунікації, розмір знижки, конкретні дії).

Вибір технологічного стеку обумовлений розповсюдженістю та зрілістю відповідних бібліотек у екосистемі Python для задач машинного навчання та аналізу даних.

Бібліотека NumPy 1.24 використовується для всіх операцій із числовими масивами: логарифмічне перетворення ознак, генерація синтетичних даних, матричні обчислення при PCA [24]. Бібліотека Pandas 2.0 забезпечує зчитування Excel-файлів, фільтрацію та агрегацію транзакцій [25], групування за CustomerID при розрахунку RFM-таблиці.

Бібліотека scikit-learn 1.3 є центральним компонентом для машинного навчання: StandardScaler для нормалізації, KMeans із стратегією ініціалізації k-Means++, DBSCAN, AgglomerativeClustering, PCA, а також функції обчислення метрик якості (silhouette_score, davies_bouldin_score, calinski_harabasz_score [22]) та NearestNeighbors для побудови k-NN графіка. Бібліотека SciPy 1.11 надає функції linkage та dendrogram для побудови та візуалізації дендрограми

ієрархічної кластеризації [26]. Бібліотеки Matplotlib 3.7 та Seaborn 0.12 використовуються для побудови всіх графіків та dashboard [27, 28].

3.2 Попередня обробка даних

Попередня обробка є критично важливим етапом, що безпосередньо визначає якість і надійність результатів кластеризації. Вхідний датасет Online Retail II містить ряд систематичних проблем якості даних, виявлених на етапі EDA, усунення яких здійснюється у послідовності чотирьох кроків.

Крок 1. Завантаження та первинний аналіз.

Датасет завантажується з репозиторію UCI або Google Drive та зчитується у DataFrame за допомогою функції pd.read_excel(). На цьому кроці виконується базова статистична характеристика: перевірка розмірності, типів атрибутів, кількості унікальних значень по кожній колонці та підрахунок пропущених значень. Ключові показники первинного датасету для аркуша Year 2010-2011: 541 909 рядків, 8 колонок, 135 080 записів без CustomerID (24,9 %), 2 215 скасованих рахунків (0,4 %).

Крок 2. Видалення некоректних записів.

Послідовно виконуються такі операції фільтрації:

а) видалення рядків із відсутнім CustomerID оператором dropna(subset=["Customer ID"]); ці записи неможливо прив’язати до конкретного клієнта;

б) фільтрація скасованих замовлень умовою ~df["Invoice"].str.startswith("C"); рахунки, що починаються з літери C, є операціями повернення і не відображають реального споживання;

в) виключення аномальних транзакцій умовою (Quantity > 0) & (Price > 0) – від’ємні та нульові значення відповідають поверненням, зразкам та технічним записам;

г) географічна фільтрація df[df["Country"] == "United Kingdom"] – обмеження вибірки клієнтами Великобританії для однорідності аналізу.

Ключові рядки реалізації:

```
df = df.dropna(subset=["Customer ID"])
df = df[~df["Invoice"].astype(str).str.startswith("C")]
df = df[(df["Quantity"] > 0) & (df["Price"] > 0)]
df = df[df["Country"] == "United Kingdom"]
df["TotalPrice"] = df["Quantity"] * df["Price"]
```

Після проведення повного циклу очищення отримано 354 321 транзакційний рядок та 3 920 унікальних клієнтів, що підтверджує очікувані результати.

Крок 3. Побудова RFM-таблиці.

Агрегація транзакцій за CustomerID виконується одним викликом groupby().agg() з трьома агрегуючими функціями. Snapshot date визначається як наступний день після дати останньої транзакції у датасеті, що дозволяє уникнути нульових значень Recency для найактивніших клієнтів:

```
SNAP = df["InvoiceDate"].max() + pd.Timedelta(days=1)
rfm = df.groupby("Customer ID").agg(
    Recency   = ("InvoiceDate", lambda x: (SNAP -
x.max()).days),
    Frequency = ("Invoice",     "nunique"),
    Monetary  = ("TotalPrice",  "sum")
).reset_index()
```

Результатом є таблиця із 3 920 рядками, де кожен рядок відповідає одному унікальному клієнту та містить тривимірний вектор (R, F, M). Медіанні значення по датасету: Recency = 51 день, Frequency = 2 замовлення, Monetary = £652 (рисунок 3.1).

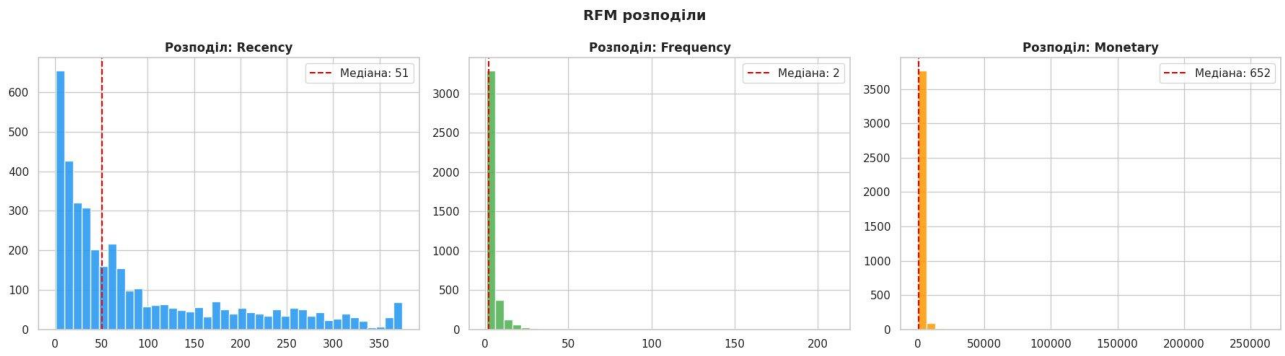

Рисунок 3.1 – Розподіли RFM-метрик до нормалізації

Крок 4. Нормалізація ознак.

Аналіз розподілів виявляє виражену правосторонню асиметрію для всіх трьох метрик, що є очікуваним для транзакційних даних: невелика кількість VIP-клієнтів з дуже великими F та M формує довгий правий «хвіст». Застосування евклідово-орієнтованих алгоритмів кластеризації до нескоригованих ознак призвело б до домінування метрики Monetary через її абсолютно більший діапазон значень.

Двоетапна трансформація усуває зазначені проблеми. Спочатку виконується логарифмічне перетворення:

$$R' = \ln(R + 1), \quad F' = \ln(F + 1), \quad M' = \ln(M + 1).$$

Потім виконується стандартизація (z-нормалізація) за допомогою класу StandardScaler:

$$z = (x - \mu) / \sigma,$$

де μ – вибіркове середнє, σ – стандартне відхилення. Після перетворення кожна з трьох ознак має нульове середнє та одиничне стандартне відхилення, що гарантує рівноправний внесок R, F і M у розрахунок евклідових відстаней. Фінальна матриця $X \in \mathbb{R}^{3920\times3}$ є вхідними даними для всіх трьох алгоритмів кластеризації (рисунок 3.2).

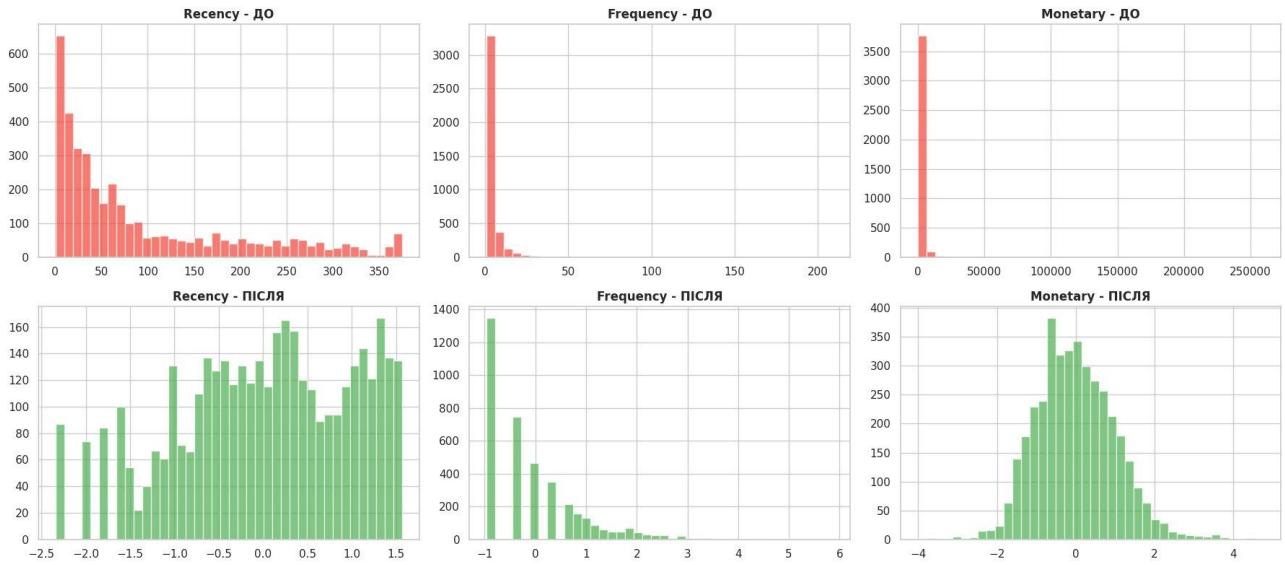

Рисунок 3.2 – Розподіли RFM-метрик до та після нормалізації

3.3 Реалізація алгоритмів кластеризації

Реалізація k-Means.

Перед запуском фінальної моделі виконується процедура підбору оптимального числа кластерів k. В циклі по значеннях k від 2 до 10 навчається модель KMeans і для кожного k обчислюються значення інерції та Silhouette Score:

```
inertias = []; sils = []
for k in range(2, 11):
    km = KMeans(n_clusters=k, random_state=42, n_init=10)
    km.fit(X)
    inertias.append(km.inertia_)
    sils.append(silhouette_score(X, km.labels_))
```

Криві методу ліктя та Silhouette Score будуються одночасно для спільної візуальної інтерпретації. Для реального датасету Online Retail II метод ліктя визначає точку перегину на кривій інерції при k = 4. Silhouette Score при k = 4 становить 0,3390, що є прийнятним для реальних маркетингових даних. Це підтверджує вибір k = 4 як оптимального.

Фінальна модель k-Means навчається з параметром n_init=10 (десять незалежних запусків з різними ініціалізаціями k-Means++) та random_state=42 для відтворюваності:

```
km4 = KMeans(n_clusters=4, random_state=42, n_init=10)
rfm["KMeans"] = km4.fit_predict(X)
```

Після навчання обчислюються три метрики якості та формується зведена статистика по кластерах з медіанними значеннями R, F, M. Для двовимірної візуалізації результати проєктуються на перші дві головні компоненти методом PCA:

```
pca = PCA(n_components=2, random_state=42)
X2  = pca.fit_transform(X)
```

Перші дві головні компоненти пояснюють у сукупності близько 94,2 % загальної дисперсії ознакового простору, що свідчить про достатню інформативність двовимірної проєкції.

Реалізація DBSCAN.

Підбір параметра eps здійснюється методом k-найближчих сусідів. Для кожної точки обчислюється відстань до 5-го найближчого сусіда (відповідно до min_samples = 5), відстані сортуються за спаданням та будується графік k-NN:

```
nbrs = NearestNeighbors(n_neighbors=5).fit(X)
dist, _ = nbrs.kneighbors(X)
kd = np.sort(dist[:, -1])[::-1]
```

Оптимальне eps відповідає точці максимального перегину (коліна) кривої, де відбувається різкий перехід від щільної до розрідженої зони. Для нормалізованого RFM-простору це значення становить eps ≈ 0.5.

Запуск алгоритму:

```
db = DBSCAN(eps=0.5, min_samples=5)
rfm["DBSCAN"] = db.fit_predict(X)
```

Значення −1 у стовпці DBSCAN відповідає шумовим точкам – клієнтам, що не увійшли до жодного кластеру. При роботі з реальним датасетом DBSCAN, як правило, виявляє 2-3 основні кластери та ідентифікує 5-12 % клієнтів як шумові точки. Шумові точки відповідають клієнтам із нетиповою поведінкою: наприклад, одна дуже велика покупка давно в минулому або нетипово висока частота з дуже малими чеками.

Реалізація ієрархічної кластеризації.

Дендрограма будується на репрезентативній вибірці з 300 точок (щоб уникнути надмірного часу побудови при повному датасеті) за методом Уорда:

```
Z = linkage(X[:300], method="ward")
dendrogram(Z, truncate_mode="lastp", p=30,
color_threshold=5.0)
```

Аналіз дендрограми підтверджує наявність чотирьох природних кластерів: горизонтальний розріз на рівні відстані злиття ≈ 5.0 виділяє 4 гілки, між якими спостерігається найбільший стрибок висоти (відстані злиття), що є критерієм оптимального k для ієрархічної кластеризації. Фінальне розбиття виконується за допомогою AgglomerativeClustering на повному датасеті:

```
agg = AgglomerativeClustering(n_clusters=4, linkage="ward")
rfm["Ward"] = agg.fit_predict(X)
```

Для кожного з трьох алгоритмів обчислюються однакові метрики якості, що дозволяє провести їх коректне порівняння. Порівняльна таблиця метрик формується та виводиться у вигляді DataFrame:

```
res = pd.DataFrame({
    "Алгоритм":  ["k-Means", "DBSCAN", "Ward"],
    "Silhouette":[km_sil, db_sil, hier_sil],
    "DBI":       [km_db,  db_dbi, hier_db],
    "CH":        [km_ch,  "-",    hier_ch]
})
```

3.4 Розробка модуля генерації персоналізованих пропозицій

Модуль персоналізованих пропозицій (RecommendationEngine) є прикладним результатом кластеризації та виконує перетворення числових кластерів у конкретні маркетингові рекомендації. Він складається з трьох функціональних компонентів: класифікатора сегментів, бази рекомендацій та генератора пропозицій.

Компонент 1: Класифікатор сегментів.

Кожному з чотирьох кластерів k-Means присвоюється семантична назва на основі порівняння медіанних RFM-профілів кластеру з медіанними RFM-профілями всієї вибірки. Класифікація здійснюється за системою правил, що відображає бізнес-логіку клієнтської цінності:

```
def get_seg(row):
    r, f, m = row["R_med"], row["F_med"], row["M_med"]
    rm, fm, mm = cp["R_med"].max(), cp["F_med"].max(),
cp["M_med"].max()
    if   r < rm*0.3 and f > fm*0.5 and m > mm*0.5: return
"VIP"
    elif r < rm*0.5 and f > fm*0.3:                return
"Loyal"
    elif r < rm*0.4:                               return
"Potential"
    elif r > rm*0.6:                               return
"Sleeping"
    else:                                          return
"One-time"
```

Правила інтерпретуються таким чином. Сегмент «VIP» характеризується низьким Recency (купували нещодавно), високою Frequency та високим Monetary – це найцінніші клієнти, що забезпечують непропорційно великий обіг. Сегмент «Loyal» (лояльні) – клієнти з помірним Recency та відносно вищою Frequency, що демонструють регулярну купівельну поведінку. Сегмент «Potential» – клієнти з невеликим Recency, але ще низькою Frequency, що вказує на потенціал для розвитку лояльності. Сегмент «One-time» (разові) – клієнти без виражених ознак регулярності. Сегмент «Sleeping» (сплячі) – клієнти з великим Recency, тобто ті, хто давно не здійснював покупок.

Компонент 2: База рекомендацій.

Для кожного з п’яти типів сегментів визначено маркетингову стратегію, що включає розмір знижки та конкретну тактику взаємодії:

```
REC = {
  "VIP":       {"discount":"10-15%",   "action":"Premium
loyalty + early access"},
  "Loyal":     {"discount":"7-10%",    "action":"Points
program + referral"},
  "Potential":{"discount":"10%(2nd)", "action":"Welcome
series + cart reminders"},
  "One-time": {"discount":"15%",      "action":"Reactivation
email"},
  "Sleeping": {"discount":"20%+ship", "action":"Win-back: 7-
day deadline"},
}
```

Стратегія для сегменту «VIP» передбачає програму преміум-лояльності: персональний менеджер, ранній доступ до нових колекцій, безкоштовна доставка та ексклюзивні знижки 10-15 %. Мета – утримання найцінніших клієнтів та збільшення частоти взаємодії. Стратегія для сегменту «Loyal» базується на накопичувальній програмі балів із можливістю обміну на знижки та на програмі «Приведи друга». Це заохочує до частіших покупок та залучення нових клієнтів. Стратегія для сегменту «Potential» реалізується через welcome-серію автоматичних листів, нагадування про кинуті кошики та знижку 10 % на друге замовлення – ці заходи спрямовані на формування звички до регулярних покупок. Для сегменту «One-time» пропонується реактиваційна email-кампанія з

агресивною знижкою 15 % та персоналізованими рекомендаціями на основі першої покупки. Сегмент «Sleeping» потребує win-back кампанії з найвищою знижкою 20 % плюс безкоштовна доставка та жорстким часовим обмеженням (7 днів), що створює ефект терміновості.

Компонент 3: Генератор пропозицій.

Для кожного клієнта з rfm-таблиці формується структурована пропозиція на основі його сегменту. Алгоритм зчитує рядок клієнта, визначає його сегмент за колонкою rfm["Segment"] та повертає відповідний запис зі словника REC:

```
for cid in rfm["CustomerID"].sample(4).tolist():
    row = rfm[rfm["CustomerID"] == cid].iloc[0]
    r   = REC.get(row["Segment"], {})
    print(f"Customer #{cid} | Segment:
{row[chr(39)]Segment[chr(39)]}")
    print(f"  R={row[chr(39)]Recency[chr(39)]}d
F={row[chr(39)]Frequency[chr(39)]}
M=GBP{row[chr(39)]Monetary[chr(39)]:.0f}")
    print(f"  Знижка: {r.get(chr(39)]discount[chr(39)],
chr(39)chr(39)]}")
    print(f"  Дія:    {r.get(chr(39)]action[chr(39)],
chr(39)chr(39)]}")
```

Додатково будується фінальний аналітичний dashboard, що складається з чотирьох графіків: горизонтальний barplot кількості клієнтів по сегментах, кругова діаграма частки виторгу по сегментах, boxplot розподілу Monetary по сегментах та scatter-plot Frequency vs Monetary з маркуванням сегментів. Dashboard зберігається у файл clustering_dashboard.png, а повна rfm-таблиця з мітками кластерів та сегментами – у файл rfm_clustered.csv для подальшого використання у маркетинговій системі.

Висновки до розділу 3

У третьому розділі описано практичну реалізацію системи аналізу та кластеризації клієнтської бази інтернет-магазину.

Обґрунтовано вибір технологічного стеку на основі Python та Google Colaboratory як середовища виконання. Спроєктовано модульну архітектуру системи з шести функціональних блоків: завантаження даних, передобробка, побудова RFM-таблиці, кластеризація, візуалізація та генерація рекомендацій.

Детально описано чотириетапний процес передобробки даних: видалення некоректних записів (рядки без CustomerID, скасовані замовлення, аномальні значення), географічна фільтрація, агрегація транзакцій у RFM-таблицю з 3 920 клієнтами та двоетапна нормалізація ознак (логарифмування + z-нормалізація).

Наведено ключові фрагменти програмного коду реалізації трьох алгоритмів кластеризації: k-Means з підбором k за методом ліктя та силует-коефіцієнтом, DBSCAN з підбором eps за k-NN графіком та ієрархічна кластеризація з аналізом дендрограми. Для кожного алгоритму описано процедуру підбору гіперпараметрів та механізм обчислення метрик якості.

Описано структуру та логіку модуля персоналізованих пропозицій, що включає класифікатор сегментів на основі правил порівняння RFM-профілів, базу рекомендацій для п’яти типів клієнтських сегментів та генератор індивідуальних маркетингових пропозицій. Результати аналізу зберігаються у форматах CSV та PNG для інтеграції у маркетингову систему. Аналіз та порівняння отриманих результатів кластеризації наведено у Розділі 4.

РОЗДІЛ 4
ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Опис експериментальних даних та умов тестування

Експериментальне дослідження виконувалось на базі датасету Online Retail II (UCI Machine Learning Repository) [29, 30]. Після проведення повного циклу попередньої обробки, описаного у розділі 3, вхідна вибірка налічує 3 920 унікальних клієнтів та 354 321 транзакційний рядок за аркушем Year 2010-2011. Знімок RFM-метрик зафіксовано відносно контрольної дати 10 грудня 2011 року.

Основні статистичні характеристики вибірки наведено у таблиці 4.1. Медіанне значення метрики Recency становить 51 день, що свідчить про відносно нещодавні покупки для половини клієнтів. Медіанне значення Frequency дорівнює 2 замовленням. Медіанний обсяг витрат (Monetary) становить 652 фунти стерлінгів, тоді як середнє арифметичне значно вище – 1 864 фунти, що вказує на суттєву правосторонню асиметрію розподілу та наявність кількох великих оптових покупців.

Таблиця 4.1 – Описова статистика RFM-вибірки (n = 3 920)

Показник	**CustomerID**	**Recency (дні)**	**Frequency (замовл.)**	**Monetary (GBP)**
count	3 920	3 920	3 920	3 920
mean	15 562,06	92,21	4,25	1 864,39
std	1 576,59	99,53	7,20	7 482,82
min	12 346	1	1	3,75
25 %	14 208,75	18	1	300,28
50 % (медіана)	15 569,50	51	2	652,28
75 %	16 913,25	143	5	1 576,58
max	18 287	374	209	259 657,30

Аналіз розподілів RFM-ознак до та після нормалізації підтверджує ефективність двоетапної трансформації. До нормалізації всі три ознаки

демонстрували виражену правосторонню асиметрію; після логарифмування та стандартизації розподіли наблизились до нормального закону, що є необхідною умовою для коректного застосування евклідово-орієнтованих алгоритмів кластеризації.

Кореляційний аналіз RFM-ознак виявив помірну від'ємну кореляцію між Recency та Frequency (r = -0,274) та помірну додатну кореляцію між Frequency та Monetary (r = 0,509). Кореляція між Recency та Monetary є слабкою (r = -0,129). Проєкція двох головних компонент методом PCA пояснює 94,2 % загальної дисперсії даних (рисунок 3.2).

Усі обчислення виконувалися у середовищі Google Colaboratory (Python 3.10, scikit-learn 1.3, pandas 2.0, numpy 1.24). Час виконання повного аналітичного пайплайну не перевищив 4 хвилин.

4.2 Аналіз результатів кластеризації

Кластеризація методом k-Means виконувалась для значень k від 2 до 10. Метод ліктя показав перегин кривої інерції у точці k = 4. Силует-коефіцієнт досягає максимуму при k = 2 (0,436), проте при k = 4 він становить 0,3390, що є прийнятним для реальних транзакційних даних. Перевага k = 4 обґрунтована змістовним критерієм: чотири кластери дозволяють виділити маркетингово значимі сегменти клієнтів [31].

Результати k-Means при k = 4 наведено в таблиці 4.2. На основі медіанних значень R, F і M кожному кластеру присвоєно семантичну назву відповідно до правил класифікації, описаних у підрозділі 3.4.

Сегмент VIP включає 632 клієнти (16,1 % бази), які генерують 62,7 % сукупного доходу (4 582 893 GBP), що підтверджує принцип Парето у структурі клієнтської цінності [5, 31]. Сегмент Loyal об'єднує 1 051 клієнта (26,8 %) зі стабільною активністю (53 дні, 4 замовлення) та формує 25,1 % доходу. Сегмент Potential охоплює 776 клієнтів (19,8 %) з нещодавніми покупками та невисокою частотою, що свідчить про потенційно лояльних покупців.

Таблиця 4.2 – Профілі кластерів k-Means (k = 4)

Сегмент	N	R_med (дні)	F_med	M_med (GBP)	Дохід (GBP)	Частка доходу	Частка клієнтів
VIP	632	8	10,0	3 572	4 582 893	62,7 %	16,1 %
Loyal	1 051	53,0	4,0	1 325	1 832 893	25,1 %	26,8 %
Potential	776	18,0	2,0	448	407 896	5,6 %	19,8 %
Sleeping	1 461	177,0	1,0	290	484 709	6,6 %	37,3 %

Сегмент Sleeping включає 1 461 клієнта (37,3 %) з великою давністю покупки (177 днів) та формує лише 6,6 % доходу. PCA-проєкція результатів k-Means наведена на рисунку 4.1.

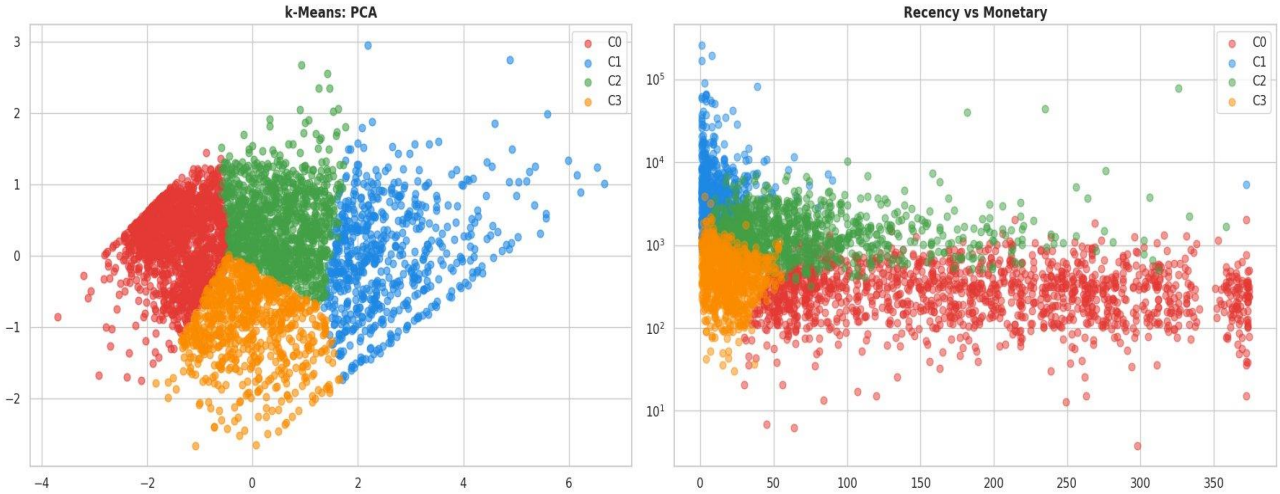

Рисунок 4.1 – Результати кластеризації k-Means: проєкція PCA та Recency vs Monetary

Для алгоритму DBSCAN підібрано параметри eps = 0,5 та min_samples = 5 на основі k-NN графіка відстаней (рисунок 4.2). Виявлено 2 кластери та 51 шумову точку (1,3 % вибірки): C0 – 2 525 точок, C1 – 1 344 точки. Результати DBSCAN наведено на рисунку 4.3.

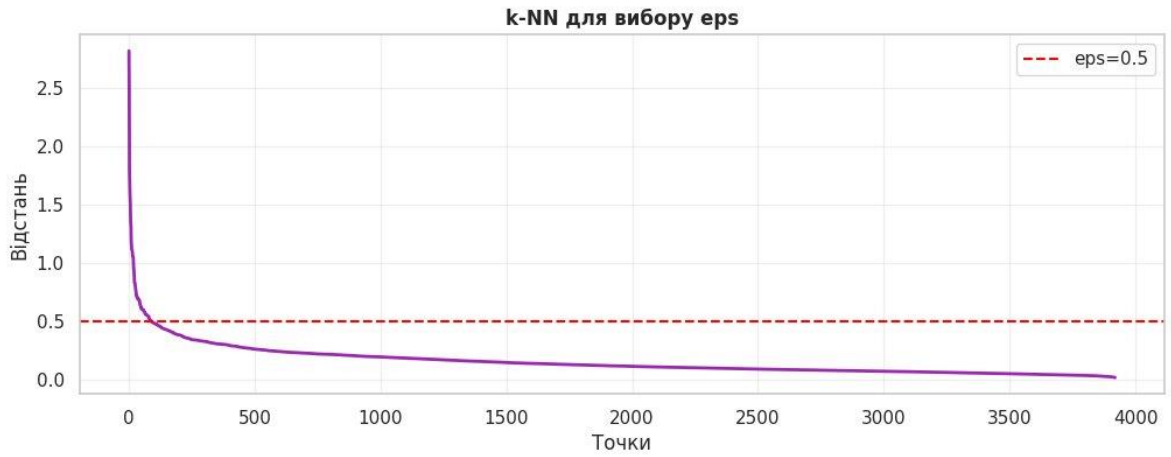

Рисунок 4.2 – k-NN графік відстаней для визначення параметра eps алгоритму DBSCAN

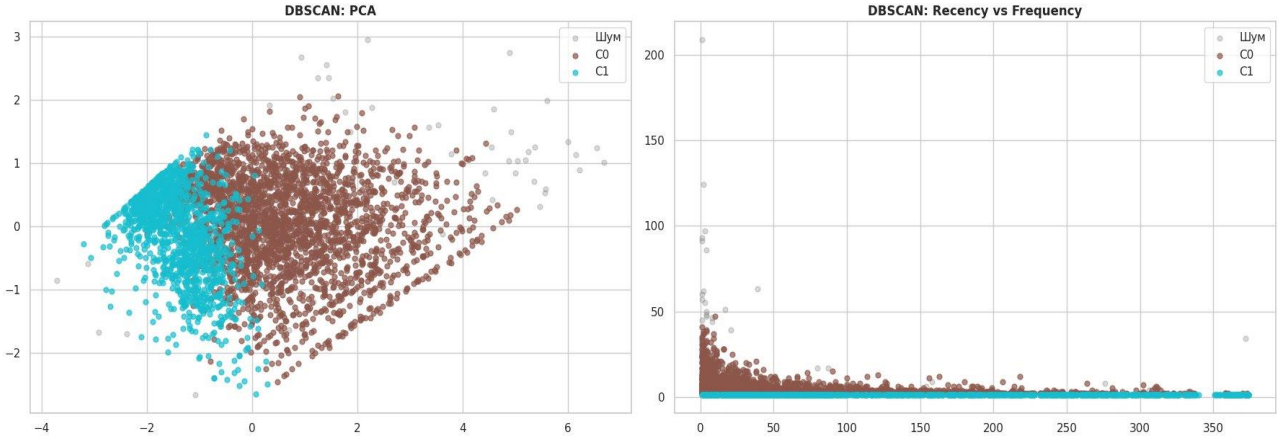

Рисунок 4.3 – Результати кластеризації DBSCAN: проєкція PCA та Recency vs Frequency

Ієрархічна кластеризація методом Уорда при k = 4 дала розбиття, топологічно подібне до k-Means, але з нижчою внутрішньою однорідністю. На дендрограмі (рисунок 4.4) чітко виражені чотири гілки зі значними стрибками відстані злиття, що підтверджує природність k = 4. PCA-проєкція результатів Ward наведена на рисунку 4.5.

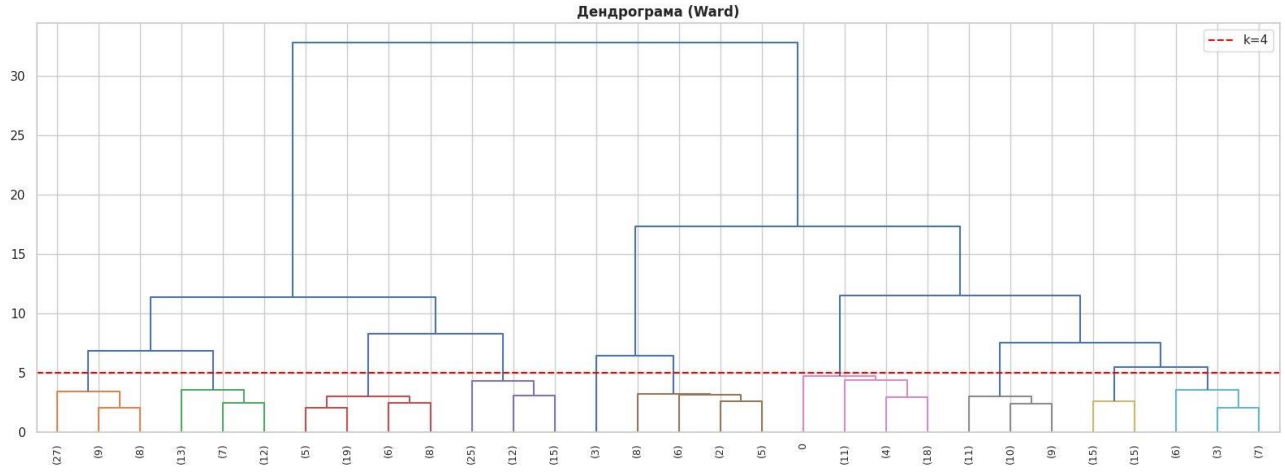

Рисунок 4.4 – Дендрограма ієрархічної кластеризації (метод Уорда, k = 4)

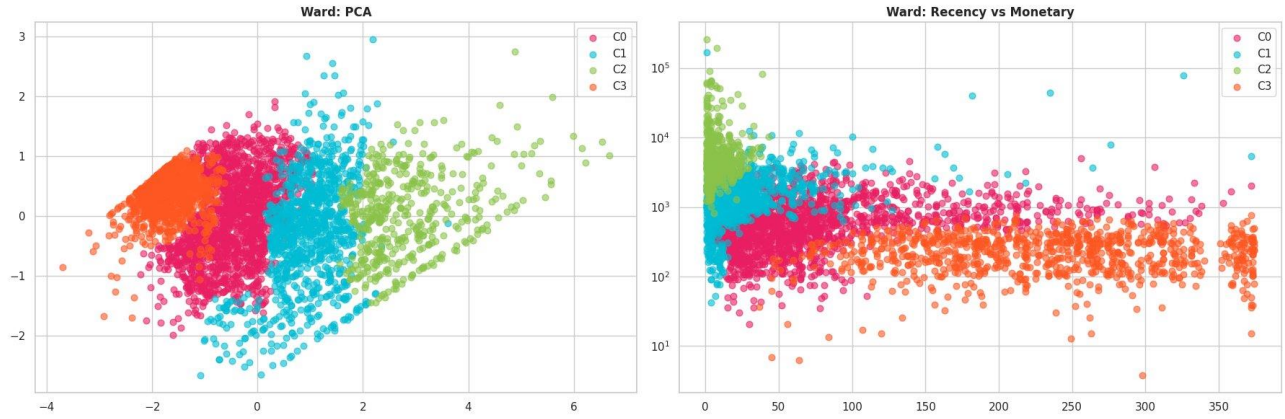

Рисунок 4.5 – Результати кластеризації Ward: проєкція PCA та Recency vs Monetary

4.3 Оцінка якості моделі та порівняльний аналіз алгоритмів

Порівняльний аналіз трьох алгоритмів здійснено за трьома внутрішніми метриками якості. Результати зведено в таблиці 4.3.

Алгоритм k-Means демонструє беззаперечну перевагу за всіма трьома метриками. Silhouette Score k-Means (0,3390) перевищує DBSCAN (0,2963) на 14,4 % та Ward (0,2410) – на 40,7 %. Значення 0,3390 відповідає задовільній якості кластеризації [19], що є типовим для реальних транзакційних даних. Індекс Девіса-Болдіна для k-Means (1,0102) є мінімальним серед трьох алгоритмів [20], а індекс Калінські-Харабаша – максимальним (3 074,79) [23]. Нижчі показники Ward пояснюються жадібною стратегією злиття без можливості перегляду рішень [16].

Таблиця 4.3 – Порівняльна оцінка алгоритмів кластеризації

Алгоритм	Кластерів	Silhouette (↑)	Davies-Bouldin (↓)	Calinski-Harabasz (↑)	Шум
k-Means	4	0,3390	1,0102	3 074,79	–
DBSCAN	2	0,2963	1,0494	–	51
Ward	4	0,2410	1,2451	2 460,41	–

Попри нижчий Silhouette Score, DBSCAN виявив унікальну перевагу – автоматичну ідентифікацію 51 аномального клієнта (1,3 % бази), які потенційно є оптовими покупцями або нетиповими профілями.

На основі виявлених сегментів розроблено диференційовані маркетингові стратегії: для VIP – преміальна програма лояльності зі знижкою 10-15 %; для Loyal – система балів та реферальна програма зі знижкою 7-10 %; для Potential – welcome-серія листів зі знижкою 10 % на другу покупку; для Sleeping – win-back кампанія з дедлайном 7 днів та знижкою 20 % і безкоштовною доставкою. Зведений дашборд наведено на рисунку 4.6.

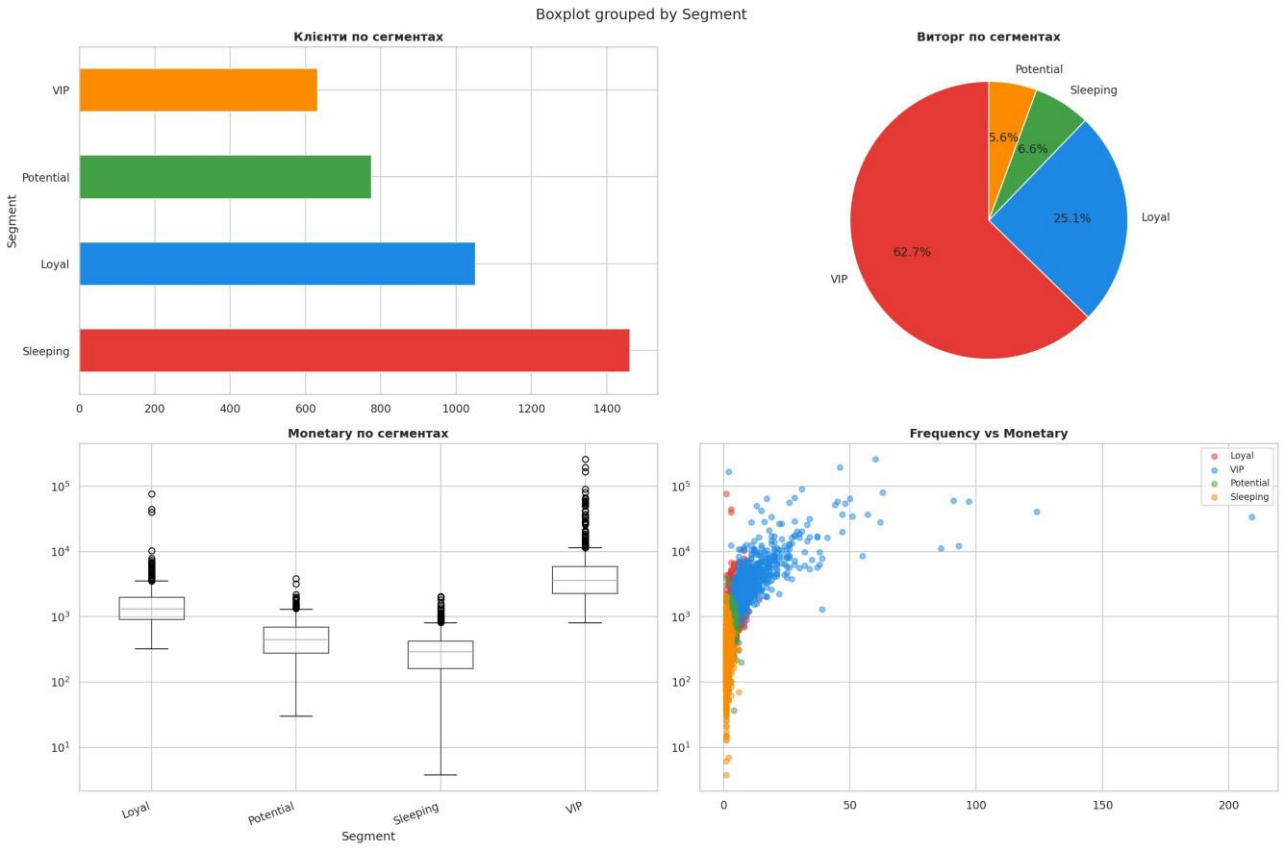

Рисунок 4.6 – Зведений дашборд сегментації клієнтів інтернет-магазину

Таким чином, алгоритм k-Means при k = 4 у поєднанні з RFM-ознаковим простором забезпечує статистично обґрунтоване, маркетингово інтерпретоване та практично цінне розбиття клієнтської бази на чотири однорідні сегменти.

Висновки до розділу 4

У четвертому розділі проведено повне експериментальне дослідження та аналіз результатів кластеризації клієнтської бази інтернет-магазину.

Підтверджено якість вибірки: 3 920 клієнтів, 354 321 транзакція. Двоетапна нормалізація (логарифмування + стандартизація) усунула правосторонню асиметрію розподілів; проєкція PCA пояснює 94,2 % дисперсії. Встановлено k = 4 методом ліктя та силует-аналізу.

k-Means виявив чотири сегменти: VIP (632 клієнти, 62,7 % доходу), Loyal (1 051 клієнт, 25,1 % доходу), Potential (776 клієнтів, 5,6 % доходу), Sleeping (1 461 клієнт, 6,6 % доходу). Принцип Парето підтверджено: 16,1 % VIP-клієнтів генерують 62,7 % доходу.

k-Means показав найкращі метрики серед усіх трьох алгоритмів: Silhouette = 0,3390, DBI = 1,0102, CHI = 3 074,79. DBSCAN ідентифікував 51 аномального клієнта (1,3 %). Ward підтвердив k = 4, але поступився k-Means за всіма метриками. Для кожного сегменту розроблено окрему маркетингову стратегію.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне науково-прикладне завдання: розроблено та реалізовано комплексну систему аналізу і кластеризації клієнтської бази інтернет-магазину на основі алгоритмів машинного навчання без учителя з метою формування персоналізованих комерційних пропозицій. На підставі проведеного дослідження можна зробити такі висновки.

1. У першому розділі проведено комплексний аналіз предметної галузі електронної комерції та обґрунтовано актуальність задачі сегментації клієнтів. Встановлено, що персоналізація є ключовим конкурентним фактором: компанії з ефективною сегментацією підвищують виторг на 10-15 % та утримання клієнтів на 20-30 % [1]. Систематизовано підходи до сегментації від демографічних до поведінкових методів та проведено порівняльний огляд алгоритмів кластеризації (k-Means, DBSCAN, ієрархічна кластеризація). Охарактеризовано датасет Online Retail II: виявлено 24,9 % відсутніх CustomerID, від'ємні значення Quantity та UnitPrice.

2. У другому розділі формалізовано постановку задачі кластеризації у тривимірному RFM-просторі. Обґрунтовано двоетапну трансформацію ознак – логарифмування ln(x+1) та стандартизацію за z-оцінками. Наведено математичні описи алгоритмів k-Means (з ініціалізацією k-Means++), DBSCAN та Ward, а також трьох метрик якості: Silhouette Score, DBI, CHI.

3. У третьому розділі розроблено модульну архітектуру системи з шести блоків та реалізовано її у Jupyter-ноутбуку (Python 3.10, Google Colab). Після повного циклу очищення отримано 354 321 транзакцію та 3 920 унікальних клієнтів. Реалізовано три алгоритми кластеризації з автоматичним підбором гіперпараметрів та модуль генерації персоналізованих маркетингових пропозицій.

4. У четвертому розділі проведено експериментальне дослідження. k-Means при k = 4 показав найкращу якість: Silhouette = 0,3390, DBI = 1,0102, CHI = 3 074,79. Виявлено чотири сегменти: VIP (632 клієнти, 62,7 % доходу), Loyal (1 051

клієнт, 25,1 %), Potential (776 клієнтів, 5,6 % доходу), Sleeping (1 461 клієнт, 6,6 % доходу). Підтверджено принцип Парето. DBSCAN ідентифікував 51 аномального клієнта (1,3 %). Ward підтвердив k = 4, поступившись k-Means за всіма метриками.

5. Практична цінність системи: відтворюваний ноутбук без спеціалізованої інфраструктури (час виконання до 4 хв для 3 920 клієнтів); автоматична генерація персоналізованих рекомендацій для кожного клієнта; масштабованість на реальні CRM-дані будь-якого інтернет-магазину без суттєвих змін архітектури.

6. Перспективи розвитку: розширення ознакового простору (асортиментні вподобання, канали взаємодії); динамічна сегментація з автоматичним перерахунком; дослідження альтернативних методів (Gaussian Mixture Models, HDBSCAN); інтеграція з CRM-системами або e-commerce платформами для автоматизованого запуску персоналізованих кампаній.

Таким чином, поставлена мета досягнута: розроблено та реалізовано комплексну методику аналізу і кластеризації клієнтської бази, яка забезпечує статистично обґрунтовану, практично інтерпретовану та технічно відтворювану сегментацію для формування ефективних персоналізованих комерційних пропозицій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Arora B., Brophy C. et al. The value of getting personalization right – or wrong – is multiplying. McKinsey & Company. 2021. URL: https://www.mckinsey.com/capabilities/growth-marketing-and-sales/our-insights/the-value-of-getting-personalization-right-or-wrong-is-multiplying (дата звернення: 16.05.2026).

2. Jain A. K. Data clustering: 50 years beyond k-means. Pattern Recognition Letters. 2010. Vol. 31, № 8. P. 651-666.

3. Schubert E., Sander J., Ester M., Kriegel H.-P., Xu X. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Transactions on Database Systems*. 2017. Vol. 42, № 3. Article 19.

4. Hughes A. M. Strategic database marketing: the masterplan for starting and managing a profitable, customer-based marketing program. 4th ed. New York : McGraw-Hill, 2012. 608 p.

5. Statista Research Department. E-commerce worldwide – statistics & facts. Statista. 2024. URL: https://www.statista.com/topics/871/online-shopping (дата звернення: 16.05.2026).

6. Turban E., Outland J., King D., Lee J. K., Liang T.-P., Turban D. C. Electronic Commerce 2018: A Managerial and Social Networks Perspective. 9th ed. Cham : Springer, 2018. 714 p.

7. Koren Y., Bell R., Volinsky C. Matrix factorization techniques for recommender systems. Computer. 2009. Vol. 42, № 8. P. 30-37.

8. Lemmens A., Gupta S. Managing churn to maximize profits. Marketing Science. 2020. Vol. 39, № 5. P. 956-973.

9. Berkhin P. A survey of clustering data mining techniques. Grouping Multidimensional Data / ed. J. Kogan, C. Nicholas, M. Teboulle. Berlin : Springer, 2006. P. 25-71.

10. Arthur D., Vassilvitskii S. k-means++: The advantages of careful seeding. Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). Philadelphia : SIAM, 2007. P. 1027-1035.

11. Xu R., Wunsch D. Survey of clustering algorithms. IEEE Transactions on Neural Networks. 2005. Vol. 16, № 3. P. 645-678.

12. Murtagh F., Legendre P. Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion? *Journal of Classification*. 2014. Vol. 31, № 3. P. 274–295.

13. Jolliffe I. T. Principal Component Analysis. 2nd ed. New York : Springer, 2002. 487 p.

14. Arbelaitz O. et al. An extensive comparative study of cluster validity indices. *Pattern Recognition*. 2013. Vol. 46, № 1. P. 243–256.

15. Chen D., Sain S. L., Guo K. Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining. Journal of Database Marketing & Customer Strategy Management. 2012. Vol. 19, № 3. P. 197-208.

16. Pedregosa F., Varoquaux G., Gramfort A. et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. 2011. Vol. 12. P. 2825-2830.

17. Harris C. R., Millman K. J., van der Walt S. J. et al. Array programming with NumPy. Nature. 2020. Vol. 585. P. 357-362.

18. McKinney W. Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference (SciPy 2010). Austin, 2010. P. 56-61.

19. Virtanen P., Gommers R., Oliphant T. E. et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods. 2020. Vol. 17. P. 261-272.

20. Hunter J. D. Matplotlib: A 2D graphics environment. Computing in Science & Engineering. 2007. Vol. 9, № 3. P. 90-95.

21. Waskom M. L. Seaborn: Statistical data visualization. Journal of Open Source Software. 2021. Vol. 6, № 60. P. 3021.

22. Chen D. Online Retail II. UCI Machine Learning Repository. 2019. DOI: 10.24432/C5CG6D. URL: https://archive.ics.uci.edu/dataset/502/online+retail+ii (дата звернення: 16.05.2026).

23. Dua D., Graff C. UCI Machine Learning Repository. Irvine : University of California, School of Information and Computer Science, 2019. URL: https://archive.ics.uci.edu/ml (дата звернення: 16.05.2026).

24. Fader P. S., Hardie B. G. S., Lee K. L. “Counting your customers” the easy way: An alternative to the Pareto/NBD model. Marketing Science. 2005. Vol. 24, № 2. P. 275-284.

ДОДАТКИ

Додаток А

Повний лістинг програмного коду системи кластеризації клієнтів

```
# == КРОК 0. Встановлення та імпорт бібліотек ==
   !pip install openpyxl -q

   import numpy as np
   import pandas as pd
   import matplotlib.pyplot as plt
   import seaborn as sns
   import warnings
   import math
   warnings.filterwarnings("ignore")

   from sklearn.preprocessing import StandardScaler
   from sklearn.cluster import KMeans, DBSCAN, AgglomerativeClustering
   from sklearn.decomposition import PCA
   from sklearn.metrics import (silhouette_score,
                                davies_bouldin_score,
                                calinski_harabasz_score)
   from sklearn.neighbors import NearestNeighbors
   from scipy.cluster.hierarchy import dendrogram, linkage

   plt.rcParams["figure.figsize"] = (12, 6)
   plt.rcParams["font.size"] = 12
   sns.set_theme(style="whitegrid", palette="Set2")

# == КРОК 1. Завантаження даних ==
   !wget -q "https://archive.ics.uci.edu/ml/machine-learning-databases/
               00502/online_retail_II.xlsx"
   df_raw = pd.read_excel("online_retail_II.xlsx",
                          sheet_name="Year 2010-2011",
                          engine="openpyxl")

   print(f"Рядків: {len(df_raw):,}   Клієнтів: {df_raw[chr(39)]Customer
   ID[chr(39)].nunique()}")
   df_raw.head()

# == КРОК 2. Попередній аналіз (EDA) ==
   print(f"Розмір: {df_raw.shape}")
   print(f"Пропущені значення:")
   print(df_raw.isnull().sum())
   print(df_raw.describe())

   df_raw["Month"] = df_raw["InvoiceDate"].dt.to_period("M")
   monthly = df_raw.groupby("Month").size()

   fig, axes = plt.subplots(1, 2, figsize=(16, 5))
   monthly.plot(kind="bar", ax=axes[0],
                color="steelblue", edgecolor="white")
   axes[0].set_title("Транзакції по місяцях", fontweight="bold")
   axes[0].tick_params(axis="x", rotation=45)
```

```
    (df_raw[df_raw["Quantity"] > 0]["Quantity"]
        .clip(upper=50)
        .hist(bins=40, ax=axes[1], color="coral", edgecolor="white"))
    axes[1].set_title("Розподіл Quantity", fontweight="bold")
    plt.tight_layout()
    plt.show()

# == КРОК 3. Очищення даних ==
    df = df_raw.copy()
    print(f"Початковий розмір: {len(df):,}")

    # Видалення рядків без Customer ID
    df = df.dropna(subset=["Customer ID"])
    print(f"Після dropna: {len(df):,}")

    # Видалення скасованих замовлень (Invoice = C...)
    df = df[~df["Invoice"].astype(str).str.startswith("C")]
    print(f"Після видалення скасувань: {len(df):,}")

    # Видалення аномальних Quantity та Price
    df = df[(df["Quantity"] > 0) & (df["Price"] > 0)]
    print(f"Після фільтрації аномалій: {len(df):,}")

    # Фільтрація по United Kingdom
    df = df[df["Country"] == "United Kingdom"]
    print(f"Після фільтрації UK: {len(df):,}")

    # Обчислення суми рядка
    df["TotalPrice"] = df["Quantity"] * df["Price"]
    print(f"Фінал: {len(df):,} рядків | {df[chr(39)]Customer
    ID[chr(39)].nunique()} клієнтів")

# == КРОК 4. Побудова RFM-таблиці ==
    SNAP = df["InvoiceDate"].max() + pd.Timedelta(days=1)
    print(f"Snapshot date: {SNAP.date()}")

    rfm = df.groupby("Customer ID").agg(
        Recency   = ("InvoiceDate",
                     lambda x: (SNAP - x.max()).days),
        Frequency = ("Invoice",    "nunique"),
        Monetary  = ("TotalPrice", "sum")
    ).reset_index()
    rfm.columns = ["CustomerID", "Recency", "Frequency", "Monetary"]

    print(f"RFM-таблиця: {len(rfm)} клієнтів")
    print(rfm.describe().round(2))

    # Візуалізація розподілів RFM
    fig, axes = plt.subplots(1, 3, figsize=(18, 5))
    for ax, col, color in zip(
        axes,
        ["Recency", "Frequency", "Monetary"],
        ["#2196F3", "#4CAF50", "#FF9800"]
    ):
        ax.hist(rfm[col], bins=40, color=color,
                edgecolor="white", alpha=0.85)
        ax.set_title(f"Розподіл: {col}", fontweight="bold")
        ax.axvline(rfm[col].median(), color="red",
                   linestyle="--",
```

```
                label=f"Медіана: {rfm[col].median():.0f}")
    ax.legend()
plt.suptitle("RFM розподіли", fontsize=14, fontweight="bold")
plt.tight_layout()
plt.show()
```

 # ═══ КРОК 5. Нормалізація ознак ═══

```
# Логарифмічне перетворення (усуває асиметрію)
rfm["R_log"] = np.log1p(rfm["Recency"])
rfm["F_log"] = np.log1p(rfm["Frequency"])
rfm["M_log"] = np.log1p(rfm["Monetary"])

# Стандартизація (z-оцінки)
scaler = StandardScaler()
X = scaler.fit_transform(rfm[["R_log", "F_log", "M_log"]])
print(f"Форма матриці X: {X.shape}")
print(f"Середнє: {X.mean(axis=0).round(4)}")
print(f"СКВ:     {X.std(axis=0).round(4)}")

# Порівняльна візуалізація до/після
fig, axes = plt.subplots(2, 3, figsize=(18, 8))
for i, col in enumerate(["Recency", "Frequency", "Monetary"]):
    axes[0, i].hist(rfm[col], bins=40,
                    color="#F44336", edgecolor="white", alpha=0.7)
    axes[0, i].set_title(f"{col} - ДО", fontweight="bold")
    axes[1, i].hist(X[:, i], bins=40,
                    color="#4CAF50", edgecolor="white", alpha=0.7)
    axes[1, i].set_title(f"{col} - ПІСЛЯ (log+scale)",
                         fontweight="bold")
plt.tight_layout()
plt.show()
```

 # ═══ КРОК 6. k-Means кластеризація ═══

```
# Підбір оптимального k (метод ліктя + Silhouette)
inertias = []
sils     = []
for k in range(2, 11):
    km = KMeans(n_clusters=k, random_state=42, n_init=10)
    km.fit(X)
    inertias.append(km.inertia_)
    sils.append(silhouette_score(X, km.labels_))

fig, axes = plt.subplots(1, 2, figsize=(16, 5))
axes[0].plot(range(2, 11), inertias, "o-",
             color="#2196F3", lw=2, ms=8)
axes[0].set_title("Метод Ліктя", fontweight="bold")
axes[0].set_xlabel("k")
axes[0].axvline(4, color="red", ls="--", label="k=4")
axes[0].legend()
axes[0].grid(alpha=0.3)

best_k = 2 + int(np.argmax(sils))
axes[1].plot(range(2, 11), sils, "s-",
             color="#4CAF50", lw=2, ms=8)
axes[1].axvline(best_k, color="red", ls="--",
                label=f"k={best_k}")
axes[1].set_title("Silhouette Score", fontweight="bold")
axes[1].legend()
axes[1].grid(alpha=0.3)
plt.tight_layout()
plt.show()
```

```
print(f"Рекомендоване k={best_k}, Silhouette={max(sils):.4f}")

# Фінальна модель k-Means (k=4)
K = 4
km4 = KMeans(n_clusters=K, random_state=42,
             n_init=10, max_iter=300)
rfm["KMeans"] = km4.fit_predict(X)

km_sil = silhouette_score(X, rfm["KMeans"])
km_db  = davies_bouldin_score(X, rfm["KMeans"])
km_ch  = calinski_harabasz_score(X, rfm["KMeans"])
print(f"Silhouette:        {km_sil:.4f}")
print(f"Davies-Bouldin:    {km_db:.4f}")
print(f"Calinski-Harabasz: {km_ch:.2f}")
print()
print(rfm.groupby("KMeans").agg(
    N  = ("CustomerID", "count"),
    R  = ("Recency",    "median"),
    F  = ("Frequency",  "median"),
    M  = ("Monetary",   "median")
).round(1))

# PCA-візуалізація кластерів
pca = PCA(n_components=2, random_state=42)
X2  = pca.fit_transform(X)
print(f"PCA variance: {pca.explained_variance_ratio_.sum()*100:.1f}%")

cm  = {0:"#E53935", 1:"#1E88E5", 2:"#43A047", 3:"#FB8C00"}
fig, axes = plt.subplots(1, 2, figsize=(18, 6))
for c in range(K):
    m = rfm["KMeans"] == c
    axes[0].scatter(X2[m, 0], X2[m, 1],
                    c=cm[c], alpha=0.6, s=40, label=f"C{c}")
    axes[1].scatter(rfm.loc[m, "Recency"],
                    rfm.loc[m, "Monetary"],
                    c=cm[c], alpha=0.5, s=40, label=f"C{c}")
axes[0].set_title("k-Means: PCA", fontweight="bold")
axes[0].legend()
axes[1].set_title("Recency vs Monetary", fontweight="bold")
axes[1].set_yscale("log")
axes[1].legend()
plt.tight_layout()
plt.show()
```

 # ═══ КРОК 7. DBSCAN кластеризація ═══

```
# k-NN графік для підбору eps
MIN_SAMPLES = 5
nbrs = NearestNeighbors(n_neighbors=MIN_SAMPLES).fit(X)
dist, _ = nbrs.kneighbors(X)
kd = np.sort(dist[:, -1])[::-1]

plt.figure(figsize=(12, 4))
plt.plot(kd, color="#9C27B0", lw=2)
plt.axhline(0.5, color="red", ls="--", label="eps=0.5")
plt.title("k-NN графік для вибору eps", fontweight="bold")
plt.xlabel("Точки (відсортовані)")
plt.ylabel("Відстань до 5-го сусіда")
plt.legend()
plt.grid(alpha=0.3)
plt.show()
```

```
    # Запуск DBSCAN
    EPS = 0.5
    db  = DBSCAN(eps=EPS, min_samples=MIN_SAMPLES)
    rfm["DBSCAN"] = db.fit_predict(X)

    n_cl    = (len(set(rfm["DBSCAN"]))
               - (1 if -1 in rfm["DBSCAN"].values else 0))
    n_noise = (rfm["DBSCAN"] == -1).sum()
    print(f"Кластерів: {n_cl} | Шум: {n_noise} ({n_noise/len(rfm)*100:.1f}%)")
    print(rfm["DBSCAN"].value_counts().sort_index())

    # Візуалізація DBSCAN
    labels  = rfm["DBSCAN"].values
    uniq    = sorted(set(labels))
    pal     = plt.cm.tab10(np.linspace(0, 1, max(len(uniq), 2)))
    cmap_d  = {l: ("gray" if l == -1 else pal[i])
               for i, l in enumerate(uniq)}

    fig, axes = plt.subplots(1, 2, figsize=(18, 6))
    for l in uniq:
        m  = labels == l
        nm = "Шум" if l == -1 else f"C{l}"
        a  = 0.3 if l == -1 else 0.7
        axes[0].scatter(X2[m, 0], X2[m, 1],
                        c=[cmap_d[l]], alpha=a,
                        s=25, label=nm)
        axes[1].scatter(rfm.loc[m, "Recency"],
                        rfm.loc[m, "Frequency"],
                        c=[cmap_d[l]], alpha=a,
                        s=25, label=nm)
    axes[0].set_title("DBSCAN: PCA", fontweight="bold")
    axes[0].legend()
    axes[1].set_title("DBSCAN: Recency vs Frequency",
                      fontweight="bold")
    axes[1].legend()
    plt.tight_layout()
    plt.show()
```

 # ═══ КРОК 8. Ієрархічна кластеризація (Ward) ═══

```
    # Дендрограма (вибірка 300 точок)
    s = min(300, len(X))
    Z = linkage(X[:s], method="ward")

    plt.figure(figsize=(16, 6))
    dendrogram(Z,
               truncate_mode="lastp",
               p=30,
               leaf_rotation=90,
               leaf_font_size=9,
               color_threshold=5.0)
    plt.title("Дендрограма (Ward)", fontweight="bold")
    plt.xlabel("Кластери")
    plt.ylabel("Відстань злиття")
    plt.axhline(5.0, color="red", ls="--", label="k=4")
    plt.legend()
    plt.tight_layout()
    plt.show()

    # AgglomerativeClustering на повному датасеті
    agg = AgglomerativeClustering(n_clusters=4, linkage="ward")
    rfm["Ward"] = agg.fit_predict(X)
```

```
hier_sil = silhouette_score(X, rfm["Ward"])
hier_db  = davies_bouldin_score(X, rfm["Ward"])
hier_ch  = calinski_harabasz_score(X, rfm["Ward"])
print(f"Silhouette:        {hier_sil:.4f}")
print(f"Davies-Bouldin:    {hier_db:.4f}")
print(f"Calinski-Harabasz: {hier_ch:.2f}")

ch4 = {0:"#E91E63", 1:"#00BCD4", 2:"#8BC34A", 3:"#FF5722"}
fig, axes = plt.subplots(1, 2, figsize=(18, 6))
for c in range(4):
    m = rfm["Ward"] == c
    axes[0].scatter(X2[m, 0], X2[m, 1],
                    c=ch4[c], alpha=0.6, s=40, label=f"C{c}")
    axes[1].scatter(rfm.loc[m, "Recency"],
                    rfm.loc[m, "Monetary"],
                    c=ch4[c], alpha=0.6, s=40, label=f"C{c}")
axes[0].set_title("Ward: PCA", fontweight="bold")
axes[0].legend()
axes[1].set_title("Ward: Recency vs Monetary",
                  fontweight="bold")
axes[1].set_yscale("log")
axes[1].legend()
plt.tight_layout()
plt.show()

# == КРОК 9. Порівняльний аналіз алгоритмів ==
mn  = rfm["DBSCAN"] != -1
ncl = (len(set(rfm["DBSCAN"]))
       - (1 if -1 in rfm["DBSCAN"].values else 0))

if mn.sum() > 1 and ncl > 1:
    db_sil = silhouette_score(
        X[mn], rfm.loc[mn, "DBSCAN"])
    db_dbi = davies_bouldin_score(
        X[mn], rfm.loc[mn, "DBSCAN"])
else:
    db_sil = db_dbi = math.nan

res = pd.DataFrame({
    "Алгоритм":  ["k-Means", "DBSCAN", "Ward"],
    "Кластерів": [4, ncl, 4],
    "Silhouette":[round(km_sil, 4),
                  round(db_sil, 4),
                  round(hier_sil, 4)],
    "DBI":       [round(km_db, 4),
                  round(db_dbi, 4),
                  round(hier_db, 4)],
    "CH":        [round(km_ch, 2), "-", round(hier_ch, 2)]
})
print(res.to_string(index=False))

# Порівняльний barplot
vals = [km_sil, hier_sil]
nms  = ["k-Means", "Ward"]
if not math.isnan(db_sil):
    vals.insert(1, db_sil)
    nms.insert(1, "DBSCAN")

fig, axes = plt.subplots(1, 2, figsize=(14, 5))
bars = axes[0].bar(
```

```
        nms, vals,
        color=["#2196F3","#9C27B0","#4CAF50"][:len(vals)])
    axes[0].set_title("Silhouette Score", fontweight="bold")
    for b, v in zip(bars, vals):
        axes[0].text(
            b.get_x() + b.get_width()/2,
            b.get_height() + 0.005,
            f"{v:.4f}", ha="center", fontweight="bold")

    sns.heatmap(
        rfm[["Recency","Frequency","Monetary"]].corr(),
        annot=True, fmt=".3f",
        cmap="coolwarm", ax=axes[1], square=True)
    axes[1].set_title("Кореляція RFM", fontweight="bold")
    plt.tight_layout()
    plt.show()

# == КРОК 10. Персоналізовані пропозиції ==
    # Класифікація сегментів
    cp = rfm.groupby("KMeans").agg(
        R_med = ("Recency",   "median"),
        F_med = ("Frequency", "median"),
        M_med = ("Monetary",  "median")
    )

    def get_seg(row):
        r, f, m = row["R_med"], row["F_med"], row["M_med"]
        rm = cp["R_med"].max()
        fm = cp["F_med"].max()
        mm = cp["M_med"].max()
        if   r < rm*0.3 and f > fm*0.5 and m > mm*0.5:
            return "VIP"
        elif r < rm*0.5 and f > fm*0.3:
            return "Loyal"
        elif r < rm*0.4:
            return "Potential"
        elif r > rm*0.6:
            return "Sleeping"
        else:
            return "One-time"

    cp["Segment"]  = cp.apply(get_seg, axis=1)
    rfm["Segment"] = rfm["KMeans"].map(cp["Segment"])

    # База рекомендацій
    REC = {
        "VIP": {
            "discount": "10-15%",
            "strategy": "Premium loyalty program",
            "action":   "Early access + personal manager"
        },
        "Loyal": {
            "discount": "7-10%",
            "strategy": "Points program",
            "action":   "Referral + closed sales"
        },
        "Potential": {
            "discount": "10% (2nd order)",
            "strategy": "Welcome series",
            "action":   "Abandoned cart reminders"
        },
        "One-time": {
```

```
        "discount": "15%",
        "strategy": "Reactivation campaign",
        "action":   "We miss you email"
    },
    "Sleeping": {
        "discount": "20% + free shipping",
        "strategy": "Win-back",
        "action":   "7-day deadline offer"
    }
}

# Генерація пропозицій для вибірки клієнтів
for cid in rfm["CustomerID"].sample(4, random_state=42).tolist():
    row = rfm[rfm["CustomerID"] == cid].iloc[0]
    rec = REC.get(row["Segment"], {})
    print(f"Customer #{cid}  |  Segment: {row[chr(39)]Segment[chr(39)]}")
    print(f"  R={row[chr(39)]Recency[chr(39)]}d
F={row[chr(39)]Frequency[chr(39)]}
M=GBP{row[chr(39)]Monetary[chr(39)]:.0f}")
    print(f"  Стратегія: {rec.get(chr(39)]strategy[chr(39)],
chr(39)chr(39)]}")
    print(f"  Знижка:    {rec.get(chr(39)]discount[chr(39)],
chr(39)chr(39)]}")
    print(f"  Дія:       {rec.get(chr(39)]action[chr(39)], chr(39)chr(39)]}")
    print()
```

 # ═══ КРОК 11. Dashboard та збереження результатів ═══

```
fig, axes = plt.subplots(2, 2, figsize=(18, 12))
fig.suptitle("Dashboard: Кластеризація клієнтів",
             fontsize=16, fontweight="bold")
sc = ["#E53935","#1E88E5","#43A047","#FB8C00","#9C27B0"]

# Кількість клієнтів по сегментах
(rfm.groupby("Segment").size()
    .sort_values(ascending=False)
    .plot(kind="barh", ax=axes[0,0], color=sc))
axes[0,0].set_title("Клієнти по сегментах",
                    fontweight="bold")

# Виторг по сегментах (pie)
(rfm.groupby("Segment")["Monetary"].sum()
    .sort_values(ascending=False)
    .plot(kind="pie", ax=axes[0,1],
          autopct="%1.1f%%", colors=sc, startangle=90))
axes[0,1].set_title("Виторг по сегментах",
                    fontweight="bold")
axes[0,1].set_ylabel("")

# Boxplot Monetary по сегментах
rfm.boxplot(column="Monetary", by="Segment",
            ax=axes[1,0])
axes[1,0].set_title("Monetary по сегментах",
                    fontweight="bold")
axes[1,0].set_yscale("log")
plt.sca(axes[1,0])
plt.xticks(rotation=20, ha="right")

# Scatter Frequency vs Monetary
for i, seg in enumerate(rfm["Segment"].unique()):
    m = rfm["Segment"] == seg
    axes[1,1].scatter(
```

```
        rfm.loc[m, "Frequency"],
        rfm.loc[m, "Monetary"],
        alpha=0.5, s=30,
        c=sc[i % len(sc)], label=seg)
axes[1,1].set_title("Frequency vs Monetary",
                    fontweight="bold")
axes[1,1].set_yscale("log")
axes[1,1].legend(fontsize=9)

plt.tight_layout()
plt.savefig("clustering_dashboard.png",
            dpi=150, bbox_inches="tight")
plt.show()
print("Збережено: clustering_dashboard.png")

# Збереження RFM-таблиці з мітками
rfm.to_csv("rfm_clustered.csv", index=False)

# Фінальний звіт
print("=" * 55)
print("  ФІНАЛЬНИЙ ЗВІТ")
print("=" * 55)
print(f"  Клієнтів:           {len(rfm):,}")
print(f"  k-Means Silhouette: {km_sil:.4f}")
print(f"  Ward    Silhouette: {hier_sil:.4f}")
print()
s = rfm.groupby("Segment").agg(
    N       = ("CustomerID", "count"),
    Revenue = ("Monetary",   "sum"),
    R_avg   = ("Recency",    "mean"),
    F_avg   = ("Frequency",  "mean")
).round(1)
s["Share_%"] = (
    s["Revenue"] / s["Revenue"].sum() * 100
).round(1)
print(s.to_string())
print()
print("Файли: rfm_clustered.csv, clustering_dashboard.png")
```

Додаток Б

Довідка про використання результатів дослідження

Результати бакалаврської кваліфікаційної роботи на тему «Аналіз та кластеризація клієнтів інтернет-магазину на основі алгоритмів машинного навчання» можуть бути використані підприємствами у сфері електронної комерції для вирішення таких практичних завдань:

– автоматизована сегментація клієнтської бази на основі транзакційних даних з CRM-системи або системи обліку продажів;

– формування персоналізованих email-кампаній та push-повідомлень відповідно до визначеного сегменту клієнта;

– побудова системи пріоритизації клієнтів для відділу продажів на основі RFM-профілю;

– моніторинг динаміки зміни клієнтських сегментів у часі для оцінки ефективності маркетингових кампаній;

– виявлення аномальних клієнтів (потенційних B2B-покупців або шахрайських акаунтів) за допомогою DBSCAN.

Розроблена система реалізована у вигляді відтворюваного Jupyter-ноутбука (Python 3.10, Google Colab) та не потребує спеціалізованого програмного забезпечення або хмарної інфраструктури для базового використання. Час розрахунку для клієнтської бази до 10 000 записів не перевищує 3 хвилин на стандартному ноутбуці.